Astra Automation

26.02.2025 г.

Оглавление

1 Быстрый старт **5**
1.1 Начало работы 5
1.2 Разработка контента Ansible 8

2 Возможности платформы **11**
2.1 Управление облачными ресурсами 11
2.2 Установка программного обеспечения 12
2.3 Управление безопасностью 12
2.4 Управление сетевым оборудованием 12
2.5 Рутинные операции по управлению инфраструктурой 12
2.6 Разработка собственного инфраструктурного кода 13

3 Обзор архитектуры **15**
3.1 Структура платформы 15
3.2 Структура кластера 16
3.3 Сеть 17
3.4 СУБД 18

4 Развертывание платформы **19**
4.1 Системные требования 19
4.2 Подготовка узлов 22
4.3 Описание инвентаря 25
4.4 Рекомендуемая топология 38
4.5 Развертывание без доступа к интернету 44
4.6 Установка системы 46
4.7 Обновление версии 47
4.8 Миграция 50
4.9 Расчет ресурсов 52
4.10 Справочные данные 52

5 Automation Hub **65**
5.1 Коллекции Ansible 65
5.2 Версионирование коллекций 68
5.3 Сертифицированные и проверенные коллекции 69
5.4 Настройка доступа к Automation Hub 72
5.5 Справочные данные 73

6 Private Automation Hub **77**
6.1 Особенности архитектуры 77
6.2 Структура управления 78
6.3 Применение Private Automation Hub 86
6.4 Графический интерфейс 95
6.5 Программный интерфейс (API) 130
6.6 Справочные данные 130

7 Контроллер **135**
7.1 Особенности архитектуры . . . 135
7.2 Структура управления . . . 136
7.3 Графический интерфейс . . . 206
7.4 Программный интерфейс (API) . . . 319
7.5 Интерфейс командной строки (CLI) . . . 393
7.6 Утилита awx-manage . . . 534
7.7 Настройка контроллера через Ansible . . . 535
7.8 Среда исполнения . . . 536

8 Разработка контента и среды исполнения **539**
8.1 Инструменты . . . 539
8.2 Среда исполнения . . . 573
8.3 Разработка коллекций . . . 575
8.4 Проверка стиля и синтаксиса . . . 575
8.5 Тестирование . . . 576
8.6 Устранение неисправностей . . . 577
8.7 Контроль целостности . . . 577
8.8 Справочные данные . . . 578

9 Управление событиями **637**

10 Средства аналитики **639**

11 Примеры решений **641**
11.1 Типовой проект . . . 641
11.2 Развертывание веб-сервера NGINX . . . 643
11.3 Защита информации . . . 649
11.4 Пример организации процесса CI/CD . . . 659
11.5 Пример настройки GitFlic webhook . . . 667
11.6 Отправка уведомлений в Mattermost . . . 669
11.7 Настройка SSO в Automation Controller с использованием ALD Pro . . . 672

12 Обеспечение безопасности **677**
12.1 Сетевые настройки . . . 677
12.2 Сертификаты TLS/SSL . . . 679

13 Производительность **681**
13.1 Концепции . . . 681
13.2 Мониторинг . . . 681
13.3 Управление . . . 681

14 Техническое обслуживание **683**
14.1 Использование журналов контроллера . . . 683
14.2 Поиск причин неисправностей . . . 688
14.3 Примеры сценариев . . . 688
14.4 Использование базы знаний . . . 689
14.5 Техническая поддержка . . . 689
14.6 Резервное копирование и восстановление . . . 690
14.7 Дополнительные настройки контроллера . . . 691

15 Дополнительные материалы **693**
15.1 Приложение 1. Общие настройки . . . 693
15.2 Приложение 2. Устанавливаемые продукты . . . 694
15.3 Приложение 3. Облачные сервисы . . . 704
15.4 Приложение 4. Инструментарий . . . 716

16 Термины и определения **741**

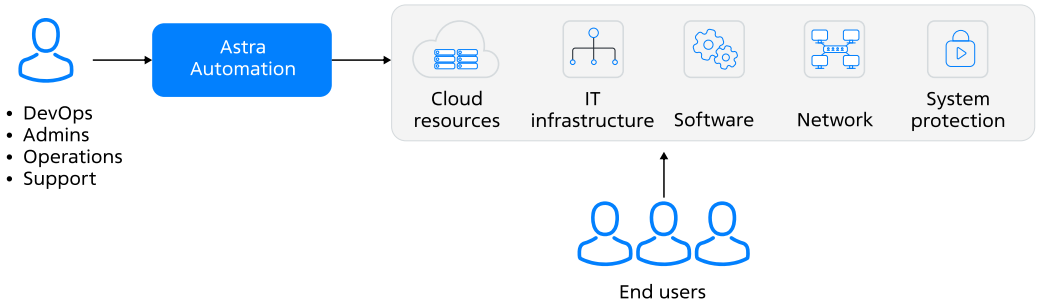
17 Обновления **745**
17.1 Жизненный цикл продукта . . . 745
17.2 Версионирование . . . 746

17.3 Что означает техническая предварительная версия (technical preview) . . . 746
17.4 Цикл разработки документации 746
17.5 Состав платформы 747
17.6 История обновлений 748
17.7 Стабильные версии 748

Алфавитный указатель **769**

Платформа для автоматизации множества рутинных операций по управлению информационной инфраструктурой заказчика, позволяющая существенно сократить затраты на ее поддержку и администрирование.

Решения базируются на использовании известного открытого проекта Ansible со свободно распространяемым исходным кодом.



Реализуя принцип инфраструктурного кода *IaC*, платформа позволяет автоматизировать решение следующих задач:

- Развертывание IT-инфраструктуры на базе различных операционных систем, используя как физическое оборудование, так и виртуальное окружение в необходимых сочетаниях.
- Управление объектами облачной инфраструктуры, включая их создание, настройку и обновление.
- Установка и настройка программного обеспечения, включая одиночные приложения и комплексы.
- Управление сетевым оборудованием от различных производителей.
- Настройка систем безопасности. Приведение систем в соответствие с требованиями стандартов безопасности.

Преимущества

Преимущества использования Astra Automation по отношению к последовательным ручным способам развертывания сложных информационных систем:

- существенное упрощение и ускорение процесса развертывания, обеспечивающие кратное снижение трудозатрат и времени на весь процесс;
- обеспечение безопасности в соответствии с существующими требованиями;
- обеспечение безошибочной непротиворечивой конфигурации устанавливаемых продуктов, значительно снижающее риски возникновения инцидентов и связанных с ними простоев систем;
- развертывание информационной инфрастуктуры любой сложности.

Пользователи

Данная платформа ориентирована на крупные компании и интеграторов сложных систем и обеспечивает необходимые потребительские качества.

С применением данной платформы *крупные компании* решают следующие важные проблемы:

- снижение стоимости владения информационной инфрастуктурой;
- высвобождение высококвалифицированных инженеров для решения внутренних сложных задач;
- ускорение развертывания инфраструктуры и снижение рисков.

Платформа удовлетворяет следующие насущные потребности *интеграторов*:

- обеспечение безопасного развертывания продуктов и отсутствия ошибок в их настройках;
- уменьшение затрат на весь процесс развертывания;
- сокращение времени развертывания комплексной инфрастуктуры.

Знакомство с платформой

Узнайте как это устроено и как работает.

Здесь мы приводим начальные сведения о структуре платформы, включая ее основные компоненты, их взаимодействие между собой и с компонентами развертываемой инфраструктуры. Вы познакомитесь с основами установки и управления платформой.

Быстрый старт Быстрое освоение базовых приемов работы с платформой.

Обзор архитектуры Структура платформы и общие сведения о ее составных частях.

Automation Hub Структура и концепции облачного реестра инфраструктурного кода.

Контроллер Назначение, структура и настройка контроллера для решения задач автоматизации.

Инструменты для разработки Назначение, состав и применение средств разработки собственного инфраструктурного кода.

Обновления Astra Automation Жизненный цикл продукта и документации, объявления о выпусках новых версий.

Сценарии

Получите практику развертывания информационных систем, следуя пошаговым инструкциям.

Здесь вы освоите на практике основные сценарии установки Astra Automation и развертывание с ее помощью сложных систем в различных средах, используя известные облачные среды или собственное оборудование с применением средств виртуализации.

Развертывание платформы Развертывание платформы с различной топологией, от одиночного узла до распределенной кластерной структуры.

Примеры создания комплексной инфраструктуры Примеры применения библиотеки структурного кода для решения различных типовых задач.

Защита информации Применение коллекций Ansible для настройки систем защиты информации в соответствии с требованиями регуляторов.

Справочники

То, что всегда пригодится профессионалу.

Документация содержит справочную информацию, необходимую для профессиональной работы с платформой.

Состав репозитория Содержимое репозитория Automation Hub.

Утилиты управления Утилиты развертывания и управления структурой платформы.

Графический интерфейс Полное описание графического интерфейса контроллера.

REST API Спецификация API контроллера.

Типографский стиль

Для улучшения читаемости в документации используются различные стили для выделения частей, которые по смыслу отличаются от основного текста.

Жирный (Bold) – слова, к которым требуется привлечь внимание, а также названия элементов пользовательского интерфейса (за исключением ссылок и кнопок).

Наклонный (Italic) – термины при первом употреблении и значения, которые необходимо указать в полях формы.

`Моноширинный (Monospace)` – названия параметров, названия ядер и их версии, названия пакетов, названия и значения переменных, названия утилит, имена файлов и каталогов.

`Enter` – названия клавиш и их сочетаний.

File ▸ *Exit* – названия пунктов меню и последовательность их выбора.

Запустить – названия кнопок и ссылок.

Совет: Полезная информация, которая может упростить работу.

Примечание: Описание какой-либо особенности работы с платформой.

Важно: Важная информация, которую необходимо учитывать при использовании платформы.

Предупреждение: Важная информация, игнорирование которой может привести к проблемам в работе платформы, утечке или потере данных и другим неприятным последствиям.

Последнее обновление: 26.02.2025

ГЛАВА 1

Быстрый старт

1.1 Начало работы

Для быстрого знакомства с возможностями платформы Astra Automation разверните Automation Controller и с его помощью запустите несложный playbook.

Для развертывания Automation Controller понадобится узел, обладающий характеристиками не ниже указанных:

Характеристика	Значение
Количество ядер CPU	4
Объем RAM, ГБ	8
Объем хранилища, ГБ	30
Операционная система	Astra Linux Special Edition 1.7.5uu1
Режим защищенности	Базовый («Орел») или усиленный («Воронеж»)

Совет: Вы можете использовать для развертывания контроллера виртуальную машину VirtualBox, созданную с помощью *Vagrant*. Соответствующий `Vagrantfile` доступен по ссылке.

Для настройки узла необходим доступ к нему по протоколу SSH, а для работы с самим контроллером – веб-браузер.

1.1.1 Установка контроллера

Для установки контроллера выполните следующие действия:

1. Подключитесь к целевому узлу по SSH.
2. В каталоге `/etc/apt/sources.list.d/` создайте файл `astra-automation.list` со ссылкой на репозиторий Astra Automation:

   ```
   deb https://dl.astralinux.ru/aa/aa-debs-for-alse-1.7 <version> main
   ```

   Вместо <version> необходимо подставить версию устанавливаемой платформы, например, `1.1-upd1`.

Доступные версии продукта опубликованы в таблице *История обновлений*.

3. Обновите список доступных пакетов:

```
sudo apt update
```

4. Установите пакет `astra-automation-setup`:

```
sudo apt install astra-automation-setup --yes
```

5. Перейдите в каталог `/opt/rbta/aa/astra-automation-setup/`:

```
cd /opt/rbta/aa/astra-automation-setup/
```

6. Внесите изменения в файл `inventory`:

   - Раздел `[automationcontroller]` приведите к следующему виду:

     ```
     [automationcontroller]
     controller.example.com  ansible_host=<IP>  ansible_connection=local
     ```

     Здесь:

     - `controller.example.com` – название узла;
     - `<IP>` – IP-адрес настраиваемого узла. Если вы используете ВМ, созданную на основе упомянутого ранее `Vagrantfile`, укажите в этом поле значение `192.168.56.11`.

   - В разделе `[all:vars]` укажите реквизиты администратора контроллера:

     - `admin_username` – название учетной записи;
     - `admin_password` – пароль.

7. Запустите установку:

```
sudo ./aa-setup
```

Дождитесь завершения выполнения команды, это может занять некоторое время. По окончании установки в терминал выводится набор строк следующего вида:

```
PLAY RECAP *********************************************************************
controller.example.com     : ok=298   changed=141   unreachable=0    failed=0    ␣
↪skipped=114   rescued=0    ignored=5
localhost                  : ok=0     changed=0     unreachable=0    failed=0    ␣
↪skipped=1     rescued=0    ignored=0

___________________________________________
Astra Automation Setup has been finished
___________________________________________
Astra Automation Controller host:
https://192.168.56.11/

Login: admin
Password: awx
___________________________________________
```

Установка считается успешной, если значение параметра `failed` равно `0`.

8. Для отключения от настраиваемого узла выполните команду:

```
exit
```

1.1.2 Настройка контроллера

1. Запустите веб-браузер и перейдите по адресу `http://<IP>/`, где <IP> – IP-адрес узла, на котором развернут Automation Controller.

   При первом подключении браузер выводит предупреждение о том, что подключение не защищено. Это нормальное поведение, поскольку для защиты подключения контроллер использует самоподписанный сертификат. При использовании браузера на основе Chromium выполните следующие действия:

   a. Нажмите кнопку *Дополнительные*.

   b. Нажмите ссылку *Перейти на сайт <IP> (небезопасно)*.

2. Авторизуйтесь в веб-интерфейсе контроллера с учетными данными администратора контроллера. Значения по умолчанию:

   - имя пользователя – `admin`;
   - пароль – `awx`.

3. Создайте и заполните инвентарь:

   a. На панели навигации выберите *Ресурсы* ▸ *Инвентарь* (*Resources* ▸ *Inventories*).

   b. Нажмите кнопку *Добавить* (Add) и выберите **Добавить инвентарь** (Add inventory).

   c. В форме **Создать новый инвентарь** (Create new inventory) в поле **Название** (Name) введите строку `Quick Start Inventory`.

   d. Нажмите кнопку *Сохранить* (Save). Откроется окно просмотра свойств созданного инвентаря.

   e. Выберите вкладку **Управляемые узлы** (Managed Nodes).

   f. Нажмите кнопку *Добавить* (Add).

   g. Заполните форму **Создать новый узел** (Create New Node):

      - **Название** (Name): `localhost`.
      - **Переменные** (Variables):

      ```
      ---
      ansible_host: localhost
      ansible_connection: local
      ```

   h. Нажмите кнопку *Сохранить* (Save).

4. Создайте проект:

   a. На панели навигации выберите *Ресурсы* ▸ *Проекты* (*Resources* ▸ *Projects*).

   b. Нажмите кнопку *Добавить* (Add).

   c. Заполните форму **Создать новый проект** (Create new project):

      - **Название** (Name): `Quick Start Project`.
      - **Тип системы управления исходными данными** (Source Control Type): **Git**.
      - **URL системы управления исходными данными** (Source Control URL):

      ```
      https://gitflic.ru/project/astra_team/astra-automation-quick-start.git
      ```

   d. Нажмите кнопку *Сохранить* (Save).

   Дождитесь завершения синхронизации проекта: значение в поле **Статус последнего задания** (Last Job Status) должно измениться на Успех (Success).

5. Создайте шаблон задания:

   a. На панели навигации выберите *Ресурсы* ▸ *Шаблоны* (*Resources* ▸ *Templates*).

b. Нажмите кнопку *Добавить* (Add) и в открывшемся меню выберите *Добавить шаблон задания* (Add job template).

c. Заполните форму **Создать новый шаблон задания** (Create New Job Template):

1. **Название** (Name): `Quick Start Template`.
2. **Тип задания** (Job Type): **Выполнение** (Run).
3. **Инвентарь** (Inventory): `Quick Start Inventory`.
4. **Проект** (Project): `Quick Start Project`.
5. **Playbook**: `playbooks/quick-start.yml`.

d. Нажмите кнопку *Сохранить* (Save). Откроется окно просмотра свойств шаблона.

6. Для запуска задания нажмите кнопку *Запустить* (Run). Откроется окно просмотра журнала выполнения задания.
7. Дождитесь перехода задания в статус Успех (Success), после чего обновите страницу браузера. В окне вывода будет отображен журнал выполнения задания, например:

```
PLAY [Quick Start Playbook] ****************************************************

TASK [Print Quick Start] *******************************************************
ok: [localhost] => {
    "msg": "Quick Start with Astra Automation on localhost"
}

PLAY RECAP *********************************************************************
localhost                  : ok=1    changed=0    unreachable=0    failed=0    ␣
↪skipped=0    rescued=0    ignored=0
```

1.1.3 Заключение

В этом сценарии вы познакомились с основными шагами по развертыванию Automation Controller и его использованию для настройки управляемых узлов. Из всей последовательности шагов важно выделить следующие действия:

- Указание учетных данных администратора контроллера в файле `inventory`.
- Получение кода проекта из системы контроля версий.
- Создание инвентаря и записей об управляемых узлах.
- Создание шаблона и запуск задания на его основе.

1.2 Разработка контента Ansible

Автоматизация управления информационной инфраструктурой основана на использовании инфраструктурного кода со следующими компонентами:

- коллекции, ориентированные на решение определенной группы задач, например, на установку определенного программного обеспечения;
- сборники сценариев (playbooks), определяющие задания по применению коллекций в конкретном окружении.

Приведенное здесь руководство ориентировано на разработку простейшей коллекции и применение ее с помощью playbook.

Важно: Эта часть документации находится в стадии разработки.

Приведенные здесь инструкции предназначены для быстрого освоения базовых приемов работы с платформой:

- установка основных компонентов платформы;
- использование платформы для установки программ;
- разработка простейшего контента Ansible для управления инфраструктурой с помощью платформы.

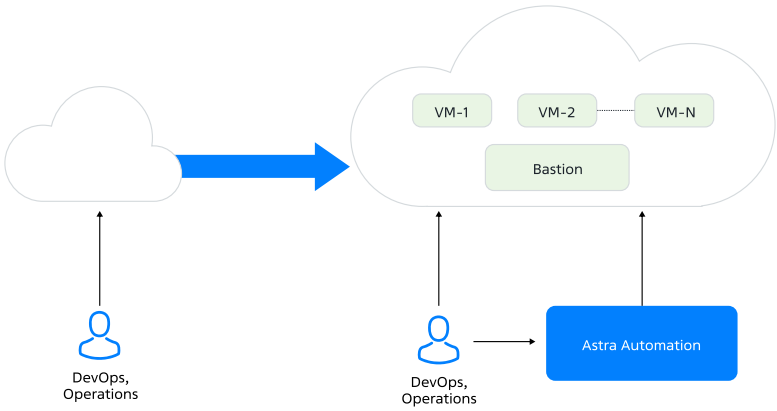
ГЛАВА 2

Возможности платформы

Платформа Astra Automation решает множество задач по управлению IT-инфраструктурой, решение которых описано в следующих секциях. Многие задачи можно решать в произвольной последовательности и в различных сочетаниях. Жизненный цикл инфраструктуры содержит множество повторяющихся операций развертывания, установки, замены и обновления различных ее компонентов.

2.1 Управление облачными ресурсами

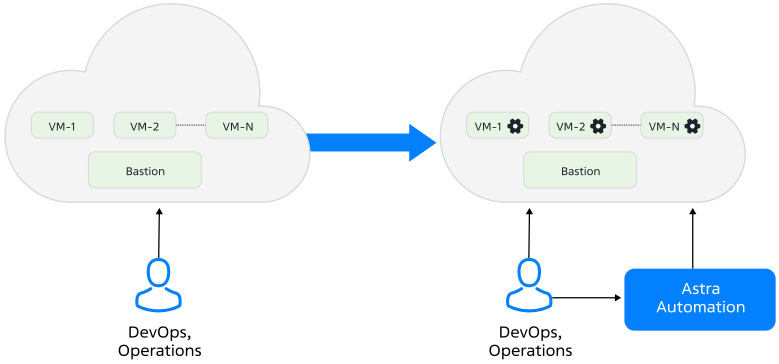
Обладая доступом к облачному пространству, публичному или частному, организация может использовать Astra Automation для создания базовой инфраструктуры, используя технологии виртуализации.



Для создания облачных ресурсов и управления ими предлагается использовать модули Ansible.

2.2 Установка программного обеспечения

Astra Automation предоставляет развитый реестр инфраструктурного кода, позволяющего устанавливать разнообразные одиночные системы или комплексы систем на оборудовании организации или в облачном пространстве.



Сертифицированный инфраструктурный код ориентирован на использование операционной системы Astra Linux Special Edition.

2.3 Управление безопасностью

Astra Automation предоставляет специальную коллекцию Ansible для автоматизации применения требуемых мер безопасности в соответствии с рекомендациями CIS[1], ФСТЭК России[2] и других сообществ и организаций. Коллекция ориентирована на использование операционной системы Astra Linux Special Edition.

2.4 Управление сетевым оборудованием

Сетевое оборудование (маршрутизаторы и коммутаторы), составляющие основу любой IT-инфраструктуры, в зависимости от сложности сети подлежат достаточно частой настройке под нужды этой инфраструктуры. К таким настройкам, например, относятся следующие составляющие:

- таблицы маршрутизации;
- состав виртуальных сетей;
- входные и выходные фильтры.

Astra Automation предоставляет необходимые модули и коллекции для настройки отечественного и зарубежного оборудования.

2.5 Рутинные операции по управлению инфраструктурой

IT-инфраструктура, подобно живому организму, подлежит частым изменениям, требующим каждодневной настройки, связанной с организационными изменениями, изменениями в ролевых моделях доступа к различным сервисам и другими. Подобные операции требуют значительных трудозатрат, которые можно значительно снизить, используя возможности Astra Automation по автоматизации рутинных операций. Для этого необходимо подготовить шаблоны для запуска типовых заданий или потоков заданий, которые может использовать персонал с минимальными знаниями платформы.

[1] https://www.cisecurity.org/
[2] https://fstec.ru/

2.6 Разработка собственного инфраструктурного кода

Несмотря на значительный объем инфраструктурного кода, предоставляемого платформой, практически в каждой организации найдутся специфичные задачи, для выполнения которых недостаточно создания сценариев использования готовых решений. При необходимости разработчики организации могут воспользоваться набором инструментов для разработки контента Ansible (CDK, content development kit) от Astra Automation для эффективного решения специфичных задач. CDK позволяет разрабатывать необходимые компоненты:

- контейнерные образы для расширения возможностей среды исполнения, поставляемой в составе Astra Automation;
- коллекции Ansible, включающие роли, модули, расширения и другие элементы инфраструктурного кода.

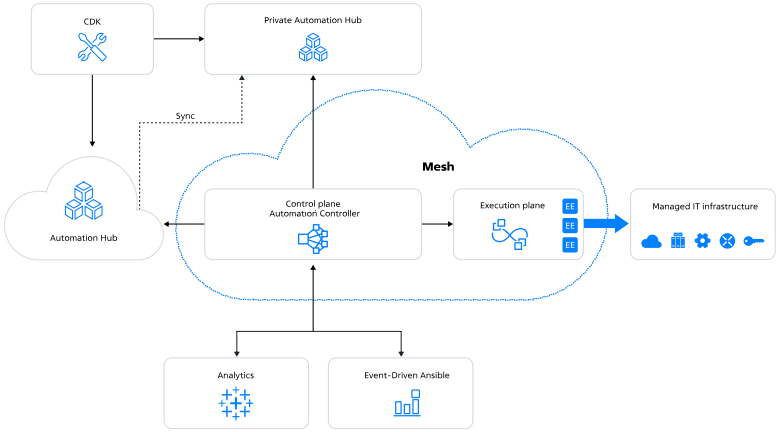
ГЛАВА 3

Обзор архитектуры

Astra Automation является гибридной платформой, объединяющей облачный сервис Automation Hub, доступный всем пользователем платформы, и компоненты управления, устанавливаемые в пространстве организации.

3.1 Структура платформы

Платформа включает ряд компонентов для управления *инфраструктурой заказчика* (Managed IT infrastructure).



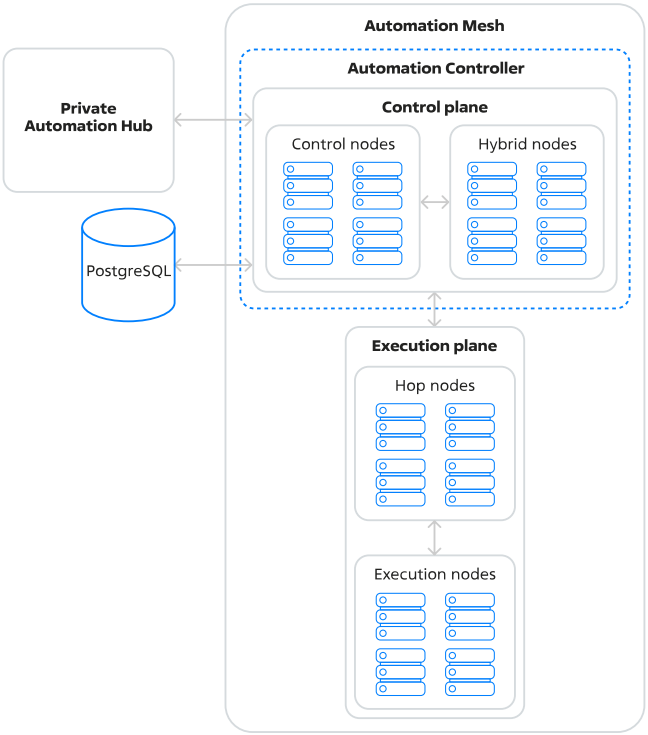
К платформе относятся следующие компоненты:

- *Контроллер* (Automation Controller) представляет собой плоскость управления (control plane) на основе кластера. Он предоставляет пользователю удобные средства управления через графический web-интерфейс, вызовы API, командный интерфейс (CLI, command-line interface) и специальную коллекцию Ansible.
- *Плоскость исполнения* (Execution plane) является частью общего кластера, объединяющей узлы, которые выполняют задания контроллера. Особенности плоскости исполнения:
  - Она обеспечивает эффективное управление *распределенной* инфраструктурой путем расположения узлов в одном сетевом окружении с узлами управляемой инфраструктуры.

  - Непосредственным исполнителем заданий в виде playbook является среда исполнения (EE, Execution Environment), выполненная в виде контейнера Podman.
  - Контроллер позволяет назначать различные задания на исполнение определенным группам узлов плоскости исполнения, а также выполнять их с помощью указанных образов EE.
- *Сеть прикладного уровня* (Mesh) объединяет плоскость управления и плоскость исполнения в единый кластер. Настройки платформы позволяют гибко определять связи (топологию) между компонентами этих плоскостей.
- *Automation Hub* является облачным реестром инфраструктурного кода для реализации требуемых задач автоматизации. В состав реестра входят различные типы инфраструктурного кода:
  - сертифицированные коллекции Ansible для выполнения различных задач автоматизации;
  - образы среды исполнения Ansible.
- *Собственный реестр* инфраструктурного кода (Private Automation Hub) позволяет организации использовать контент, предоставляемый через Automation Hub или другие источники, а также хранить и использовать собственный контент и образы среды исполнения. При создании собственного реестра последний получает контент из Automation Hub.
- *Средства разработки* (CDK, content development kit) содержат необходимые инструменты для организации полноценного процесса разработки собственного инфраструктурного кода и образов среды исполнения. Созданные артефакты следует сохранять в собственном реестре инфраструктурного кода.
- *Служба управления событиями* (Event-Driven Ansible) позволяет отслеживать события, возникающие в различных компонентах инфраструктуры пользователя. Применение этой службы приводит к существенному снижению объема ручной работы а также к быстрому своевременному реагированию на изменившуюся ситуацию в инфраструктуре.
- Средства аналитики (Analytics) позволяют собирать данные о работе контроллера и формировать отчеты для анализа эффективности организации процессов автоматизации.

3.2 Структура кластера

Центральное место в структуре Astra Automation занимает кластер управления, состоящий из набора узлов, объединенных сетью Mesh.



В Astra Automation используются узлы четырех типов:

- *Управляющие узлы* (control nodes) управляют работой контроллера и формируют задания автоматизации.
- *Исполняющие узлы* (execution nodes) используются для непосредственного запуска заданий автоматизации, сформированных управляющими узлами.
- *Промежуточные* (переходные) узлы (hop nodes) используются в качестве посредников между управляющими и исполняющими узлами. Их назначение – связывать узлы, когда прямой доступ управляющих узлов к исполняющим по какой-либо причине невозможен, например, из-за технических ограничений или соображений безопасности.

  Примечание: Промежуточные узлы нельзя использовать для запуска заданий.

- *Гибридные узлы* (hybrid nodes) выполняют функции управляющих и исполняющих узлов.

Automation Hub – это облачный сервис ПАО Группа Астра, предоставляющий зарегистрированным пользователям доступ к хранилищу инфраструктурного кода.

3.3 Сеть

С помощью полносвязной сети прикладного уровня с автоматическим обнаружением узлов (automation mesh) кластер Astra Automation объединяет узлы разных типов в две плоскости – плоскость управления (control plane) и плоскость исполнения (execution plane). При объединении узлов в сеть происходит настройка связей между узлами плоскостей.

Плоскость управления представляет собой Automation Controller и состоит из управляющих и гибридных узлов, а плоскость исполнения – из промежуточных и исполняющих узлов.

Узлы кластера объединяются в сеть с помощью службы рецепторов, установленной на каждом узле.

3.4 СУБД

СУБД на базе PostgreSQL используется для хранения данных контроллера. Она должна быть доступна для всех узлов из плоскости управления.

По умолчанию СУБД развертывается вместе с контроллером, однако, можно использовать и уже существующую СУБД, развернутую на собственном оборудовании или в одном из облачных сервисов управляемых баз данных.

При развертывании СУБД средствами контроллера необходимо выделить для нее отдельный узел, который не должен использоваться для других целей.

Automation Controller не имеет поддержки кластеризации СУБД. Это значит, что высокую доступность СУБД следует обеспечить самостоятельно.

Использование внешней СУБД накладывает следующие ограничения:

- Контроллер не поддерживает автоматическое переключение на новый мастер в случае переключения мастера в кластере СУБД.
- Не рекомендуется использовать менеджеры соединений типа pgBouncer[3].
- Нельзя использовать менеджер соединений в режиме «Transaction pooling».

[3] https://www.pgbouncer.org/

ГЛАВА 4

Развертывание платформы

Развертывание Astra Automation предусматривает использование следующих видов окружения:

- виртуальные машины;
- физическое оборудование (bare metal).

Для тестового окружения можно создать необходимые ВМ (виртуальная машина) с помощью *Vagrant*.

4.1 Системные требования

Для развертывания платформы необходимо обеспечить выполнение системных требований, предъявляемых к рабочим местам администратора и пользователей, узлам платформы и используемой ей СУБД.

4.1.1 Рабочие места администратора и пользователей

Для работы с платформой через веб-интерфейс необходимо наличие на рабочих станциях администратора и пользователей одного из следующих веб-браузеров:

Браузер	Версия, не ниже
Google Chrome	109
Mozilla Firefox	102
Яндекс Браузер	23.7.2.767

4.1.2 Общие требования

Фактический список узлов для развертывания платформы и их технические характеристики определяются требованиями к отказоустойчивости, обеспечению высокой доступности и быстродействию.

Дополнительные требования:

- Все узлы должны иметь постоянные IP-адреса или доменные имена (в зависимости от того, как заполняется файл инвентаря, используемый установщиком).
- Рекомендуется обеспечить передачу данных между всеми узлами плоскости управления Automation Controller с минимальными задержками.
- Узлы, используемые для развертывания Private Automation Hub, должны быть доступны с других узлов, входящих в состав развертываемой конфигурации. В противном случае некоторые узлы не смогут использовать содержимое, хранящееся в Private Automation Hub.
- При развертывании без доступа к интернету (Offline Bundle) зарезервируйте на каждом узле не менее 10 ГБ свободного места для хранения установочных файлов платформы.

4.1.3 Automation Controller

Типовые системные требования к узлам Automation Controller зависят от их типа.

Управляющие узлы и гибридные узлы

Типовые системные требования к управляющим и гибридным узлам:

Параметр	Минимальное значение
Количество ядер CPU	4
Количество RAM, ГБ	16
Дисковое пространство, ГБ	50
Тип дискового накопителя	SSD
Производительность, IOPS	1500
Тип ОС	Astra Linux Special Edition 1.7.4
Версия ядра ОС	`5.15.0-generic`
Режим защищенности ОС	Базовый («Орел») или усиленный («Воронеж»)

Исполняющие узлы

Типовые системные требования к исполняющим узлам:

Параметр	Минимальное значение
Количество ядер CPU	4
Количество RAM, ГБ	16
Дисковое пространство, ГБ	50
Тип ОС	Astra Linux Special Edition 1.7.4
Версия ядра ОС	`5.15.0-generic`
Режим защищенности ОС	Базовый («Орел») или усиленный («Воронеж»)

На производительность исполняющих узлов наибольшее влияние оказывают количество ядер CPU и RAM.

Промежуточные узлы

Типовые системные требования к промежуточным узлам:

Параметр	Минимальное значение
Количество ядер CPU	4
Количество RAM, ГБ	16
Дисковое пространство, ГБ	50
Тип ОС	Astra Linux Special Edition 1.7.4
Версия ядра ОС	`5.15.0-generic`
Режим защищенности ОС	Базовый («Орел») или усиленный («Воронеж»)

На производительность промежуточных узлов наибольшее влияние оказывают пропускная способность сети и низкие задержки при передаче пакетов.

Рекомендации

На управляющих, гибридных и исполняющих узлах зарезервируйте не менее 20 ГБ дискового пространства каталога `/var/lib/awx/`, используемого для хранения проектов и образов среды исполнения. При большом количестве проектов или их значительном размере увеличьте размер пространства, зарезервированного для хранения содержимого каталога `/var/lib/awx/projects/`, как минимум вдвое. Это позволит избежать ошибок, связанных с недостатком дискового пространства.

Требования к количеству RAM зависят от количества одновременно настраиваемых узлов. Оно, в свою очередь, зависит от количества *ответвлений (forks)*, которое задается в настройках шаблона задания или в конфигурационном файле `ansible.cfg`. Чтобы избежать проблемы нехватки ресурсов, рекомендуется выделить не менее 1 ГБ RAM на каждые 10 ответвлений, а также зарезервировать еще 2 ГБ RAM для собственных нужд контроллера. Например, если в настройках количества ответвлений указано значение `200`, для успешного выполнения заданий рекомендуется иметь на узлах не менее 22 ГБ RAM.

4.1.4 Private Automation Hub

Типовые системные требования к узлам Private Automation Hub:

Параметр	Минимальное значение	Рекомендуемое значение
Количество ядер CPU	4	≥ 8
Количество RAM, ГБ	16	≥ 16
Дисковое пространство, ГБ	60	≥ 120
Тип дискового накопителя	SSD	SSD
Тип ОС	Astra Linux Special Edition 1.7.4	Astra Linux Special Edition 1.7.5
Версия ядра ОС	`5.15.0-generic`	`6.1.50-generic`
Режим защищенности ОС	Базовый («Орел») или усиленный («Воронеж»)	

4.1.5 СУБД

Типовые системные требования к узлам СУБД PostgreSQL, в том числе развернутой в стороннем сервисе:

Параметр	Минимальное значение	Рекомендуемое значение
Версия PostgreSQL	13	13
Количество ядер CPU	4	≥ 8
Количество RAM, ГБ	16	≥ 16
Дисковое пространство, ГБ	20	≥ 150
Производительность, IOPS	1500	≥ 1500
Тип дискового накопителя	HDD	SSD

Astra Automation не имеет поддержки кластеризации по отношению к СУБД. Это значит, что для обеспечения ее высокой доступности необходимо использовать сторонние средства.

4.1.6 Балансировщик нагрузки

Для доступа к компонентам платформы рекомендуется использовать балансировщики нагрузки. Убедитесь, что настройки используемых балансировщиков не мешают прохождению трафика между узлами платформы через порты 80 и 443.

4.2 Подготовка узлов

Для успешного развертывания платформы необходимо настроить узел, на котором будет запущен установщик, и все узлы платформы. Узел, на котором запускается установщик, называется *установочным узлом*. При необходимости его можно включить в состав платформы.

4.2.1 Подготовка установочного узла

Для настройки установочного узла выполните следующие действия:

1. Если в файле `/etc/apt/sources.list` не содержатся ссылки на базовый (base) и расширенный (extended) репозитории Astra Linux Special Edition, добавьте их.

   Важно: Версии используемых репозиториев должны соответствовать версии ОС, установленной на узлах платформы.

   Например, для Astra Linux Special Edition 1.7.5 ссылки на репозитории имеют следующий вид:

   ```
   deb http://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.5/repository-base     1.7_
   ↪x86-64 main non-free contrib
   deb http://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.5/repository-extended 1.7_
   ↪x86-64 main non-free contrib
   ```

   Подробности о сетевых репозиториях Astra Linux Special Edition 1.7 см. в статье справочного центра Интернет-репозитории Astra Linux Special Edition x.7[4].

2. В каталоге `/etc/apt/sources.list.d/` создайте файл `astra-automation.list` со ссылкой на репозиторий Astra Automation:

   ```
   deb https://dl.astralinux.ru/aa/aa-debs-for-alse-1.7 <version> main
   ```

[4] https://wiki.astralinux.ru/pages/viewpage.action?pageId=158598882

Вместо <version> необходимо подставить версию устанавливаемой платформы, например, `1.1-upd1`.

Доступные версии продукта опубликованы в таблице *История обновлений*.

3. Обновите список доступных пакетов:

```
sudo apt update
```

4. Установите пакет `astra-automation-setup`:

```
sudo apt install astra-automation-setup --yes
```

По окончании установки выводится сообщение следующего вида:

```
-----------------------------------
Welcome to Astra Automation Setup!
-----------------------------------

Follow next steps to install Astra Automation:
1. Go to /opt/rbta/aa/astra-automation-setup
2. Edit inventory file
3. Run aa-setup
4. Enjoy!

For guidance on installing Astra Automation, consult https://docs.astra-automation.
↪ru/
```

4.2.2 Подготовка узлов

Для подготовки рабочего окружения к развертыванию платформы выполните следующие действия на всех узлах:

1. Обеспечьте сетевую связность всех узлов.
2. Настройте корректное разрешение имен узлов.
3. Обеспечьте доступ к узлам по SSH с использованием ключей согласно *инструкции*.
4. Обеспечьте выполнение команд с использованием `sudo` без ввода пароля согласно *инструкции*.

4.2.3 Настройка SSH для доступа к узлам

Рекомендуемым способом доступа к узлам является подключение по протоколу SSH с использованием ключей.

Генерация ключей SSH

Для генерации пар ключей SSH на установочном узле выполните следующие действия:

1. В каталоге `/opt/rbta/aa/astra-automation-setup/` создайте подкаталог `ssh-keys/`.

```
sudo mkdir -p /opt/rbta/aa/astra-automation-setup/ssh-keys/
```

2. Перейдите в созданный каталог:

```
cd /opt/rbta/aa/astra-automation-setup/ssh-keys/
```

3. Для каждого узла создайте пару ключей SSH:

```
sudo ssh-keygen -C "<comment>" -f ./<filename> -N "<password>"
```

где:

- <comment> – комментарий к ключу, например:

  ```
  administrator@node1.example.com
  ```

  Из этого комментария следует, что ключ создан для пользователя `administrator` узла `node1.example.com`.

- <filename> – шаблон имени файлов, в которые следует сохранить пару ключей. Приватный ключ сохраняется в файл `<filename>`, публичный – `<filename>.pub`.
- <password> – пароль для защиты приватного ключа. Если пароль пустой, приватный ключ создается без защиты.

Настройка целевых узлов

Чтобы разрешить подключение к узлу по SSH с использованием ключей, выполните следующие действия:

1. В каталоге `/etc/ssh/sshd_config.d/` создайте файл `ssh_auth_keys` со следующим содержимым:

   ```
   PubkeyAuthentication  yes
   Authorized KeysFile   .ssh/authorized_keys  .ssh/authorized_keys2
   ```

2. Перезапустите сервер SSH:

   ```
   sudo systemctl restart ssh
   ```

3. В домашнем каталоге пользователя-администратора создайте подкаталог `.ssh/`, а в нем – файл `authorized_keys`.
4. Добавьте в файл `authorized_keys` содержимое соответствующего публичного ключа, например (часть содержимого ключа опущена с целью сокращения):

   ```
   ssh-rsa AAAAB3NzaC1yc2EAd...4bwy3tY2/ administrator@node1.example.com
   ```

   Совет: Если узел разрешает подключение по SSH с использованием пароля, публичную часть ключей SSH можно разместить с помощью утилиты `ssh-copy-id`:

   ```
   ssh-copy-id -i <filename> <user>@<host>
   ```

   где:

   - <filename> – путь к файлу публичного ключа SSH;
   - <user> – название учетной записи пользователя узла;
   - <host> – IP-адрес или FQDN узла.

4.2.4 Использование sudo без ввода пароля

Для настройки узлов с помощью Ansible необходимо разрешить на них использование команды sudo без ввода пароля.

Чтобы выключить запрос пароля при использовании sudo, выполните команду:

```
sudo astra-sudo-control disable
```

Примечание: В универсальных базовых образах Astra Linux[5] использование sudo без ввода пароля включено по умолчанию.

[5] https://registry.astralinux.ru/

4.3 Описание инвентаря

Для утилиты развертывания необходимо описание инвентаря для всех компонентов платформы. К этим компонентам относятся:

- Automation Controller;
- Private Automation Hub.

Описание инвентаря включает:

- параметры каждого используемого узла;
- разделение узлов по компонентам платформы;
- параметры баз данных;
- специальные параметры (TLS, SSO и другие).

4.3.1 Инвентарь платформы

Параметры компонентов платформы задаются в файле *инвентаря* `inventory`, находящемся в каталоге `/opt/rbta/aa/astra-automation-setup/`. Для этого используются следующие секции:

- `[automationcontroller]` – узлы плоскости управления;
- `[execution_nodes]` – узлы плоскости исполнения;
- `[automationhub]` – узел Private Automation Hub;
- `[database]` – узел СУБД Automation Controller;
- `[all:vars]` – параметры, общие для всех узлов, а также различные настройки развертывания платформы, в том числе учетные данные администраторов Automation Controller и Private Automation Hub, пути к сертификатам и ключам TLS и так далее.

Важно: Файл инвентаря установщика Astra Automation должен иметь формат INI.

Выберите подходящую конфигурацию платформы и укажите параметры узлов в инвентаре.

4.3.2 Реквизиты доступа к узлам

С целью сокращения в примерах опущены параметры, в которых задаются реквизиты для подключения к узлам:

- `ansible_user` – название учетной записи пользователя, используемой для подключения к узлу;
- `ansible_ssh_private_key_file` – путь к файлу приватного ключа SSH.

Если для подключения ко всем узлам используются одни и те же реквизиты, укажите их в секции `[all:vars]`, например:

```
[automationcontroller]
node1.example.com  ansible_host=192.0.2.11
node2.example.com  ansible_host=192.0.2.12

[all:vars]
ansible_user=administrator
ansible_ssh_private_key_file=./ssh-keys/ssh_key
```

Если реквизиты для доступа к узлам различаются, укажите их в параметрах соответствующих узлов, например:

```
[automationcontroller]
node1.example.com  ansible_host=192.0.2.11  ansible_user=alex  ansible_ssh_private_key_
↪file=./ssh-keys/node1_key
node2.example.com  ansible_host=192.0.2.12  ansible_user=john  ansible_ssh_private_key_
↪file=./ssh-keys/node2_key

[execution_nodes]
node3.example.com  ansible_host=192.0.2.21  ansible_user=jack  ansible_ssh_private_key_
↪file=./ssh-keys/node3_key
```

4.3.3 Главный узел

При развертывании платформы данные о ее узлах указываются в инвентаре установщика. При первом запуске установщика узел, указанный первым в группе `automationcontroller`, становится главным узлом (primary node).

К главном узлу предъявляются следующие требования:

- При развертывании или обновлении контроллера главный узел всегда должен быть доступен.
- Запись о главном узле в инвентаре установщика должна размещаться на первой позиции в группе `automationcontroller`.

Параметры Automation Controller

В этом разделе приводятся примеры заполнения инвентаря для нескольких типовых конфигураций Automation Controller.

Учетные данные системного администратора Automation Controller укажите в секции `[all:vars]`, например:

```
[all:vars]
admin_username=admin
admin_email=admin@example.com
admin_password=p@ssw0rd!
```

Назначение переменных:

- `admin_username` – имя учетной записи;
- `admin_email` – адрес электронной почты;
- `admin_password` – пароль.

Контроллер на одном узле

В самом простом случае контроллер и СУБД могут быть развернуты на одном узле. Такая конфигурация не подходит для промышленной эксплуатации, но проста в развертывании и нетребовательна к ресурсам.

Если используется внутренняя СУБД, указывать ее параметры не требуется – она будет развернута на том же узле, что и контроллер.

Укажите сведения об узле в секции `[automationcontroller]`, например:

```
[automationcontroller]
aac.example.com  ansible_host=192.0.2.11
```

Если развертывание кластера производится на установочном узле для тестовых целей, для подключения не требуется использование SSH. В этом случае используйте параметр `ansible_connection=local`.

Кластер из гибридных узлов

По умолчанию все узлы, входящие в группу `automationcontroller`, считаются гибридными. Допускается создание кластера, состоящего только из гибридных узлов.

Пример заполнения секции `[automationcontroller]`:

```
[automationcontroller]
hybrid-node-1.example.com ansible_host=192.0.2.11
hybrid-node-2.example.com ansible_host=192.0.2.12
hybrid-node-3.example.com ansible_host=192.0.2.13
```

Если планируется в будущем добавить в кластер узлы других типов, для управляющих узлов рекомендуется создать дополнительную группу, в переменных которой явно задать тип узлов:

```
[automationcontroller]
node-1.example.com ansible_host=192.0.2.11
node-2.example.com ansible_host=192.0.2.12
node-3.example.com ansible_host=192.0.2.13
node-4.example.com ansible_host=192.0.2.14

# Отдельная группа для управляющих узлов
[control_group]
node-4.example.com

# Переменные группы control_group
[control_group:vars]
node_type=control
```

Кластер из одного управляющего и одного исполняющего узла

Для создания кластера Automation Controller из одного управляющего и одного исполняющего узла заполните инвентарь следующим образом:

- В секцию `[automationcontroller]` добавьте запись об управляющем узле.
- Создайте секцию `[execution_nodes]` и добавьте в нее запись об исполняющем узле.
- Создайте для группы `automationcontroller` переменные `node_type` и `peers` со значениями `control` и `execution_nodes` соответственно.

Пример описания кластера из одного управляющего и одного исполняющего узла:

```
# Плоскость управления
[automationcontroller]
control-node-1.example.com  ansible_host=192.0.2.11

# Переменные группы automationcontroller
[automationcontroller:vars]
node_type=control
peers=execution_nodes

# Плоскость исполнения
[execution_nodes]
execution-node-1.example.ru  ansible_host=192.0.2.21
```

Примечание: Переменная `peers` со значением `execution_nodes` указывает, что управляющие узлы должны связываться с исполняющими узлами напрямую, без промежуточных узлов.

Кластер из управляющих и исполняющих узлов

Отказоустойчивая конфигурация Automation Controller в минимальной комплектации требует наличия в кластере хотя бы двух управляющих и двух исполняющих узлов. Для получения такой конфигурации заполните инвентарь следующим образом:

1. Добавьте в секцию `[automationcontroller]` записи об управляющих узлах.
2. Создайте секцию `[execution_nodes]` и добавьте в нее записи об исполняющих узлах.
3. Создайте для группы `automationcontroller` переменные `node_type` и `peers` со значениями `control` и `execution_nodes` соответственно.

Пример описания кластера из двух управляющих и двух исполняющих узлов:

```
# Плоскость управления
[automationcontroller]
control-node-1.example.com  ansible_host=192.0.2.11
control-node-2.example.com  ansible_host=192.0.2.12

# Переменные группы automationcontroller
[automationcontroller:vars]
node_type=control
peers=execution_nodes

# Плоскость исполнения
[execution_nodes]
execution-node-1.example.com  ansible_host=192.0.2.21
execution-node-2.example.com  ansible_host=192.0.2.22
```

Все управляющие узлы здесь имеют прямой доступ ко всем исполняющим узлам.

Кластер с удаленным исполнением

Конфигурация с промежуточными узлами позволяет подключаться через них к удаленным исполняющим узлам.

Пример отказоустойчивого кластера Automation Controller с промежуточным узлом `hop-node-1`, через который выполняется подключение к исполняющему узлу `execution-node-2`:

```
# Плоскость управления
[automationcontroller]
controller-node-1.example.com  ansible_host=192.0.2.11
controller-node-2.example.com  ansible_host=192.0.2.12

[automationcontroller:vars]
node_type=control
peers=instance_group_local

# Плоскость исполнения
[execution_nodes]
execution-node-1.example.com  ansible_host=192.0.2.21
execution-node-2.example.com  ansible_host=192.0.2.22
execution-node-3.example.com  ansible_host=192.0.2.23
hop-node-1.example.com  ansible_host=192.0.2.31

# Исполняющие узлы с прямым подключением к управляющим узлам
[instance_group_local]
execution-node-1.example.com
execution-node-2.example.com

# Промежуточный узел
[hop]
hop-node-1.example.com
```

(continues on next page)

(продолжение с предыдущей страницы)

```
[hop:vars]
# Прямое подключение к управляющим узлам
node_type=hop
peers=automationcontroller

# Удаленный исполняющий узел
[instance_group_remote]
execution-node-3.example.com

# Прямое подключение к узлам из группы hop
[instance_group_remote:vars]
peers=hop
```

Примечание: Переменная `peers` со значением `instance_group_local` указывает, что управляющие узлы должны связываться с исполняющими узлами из группы `instance_group_local` напрямую, без промежуточных узлов.

Параметры Private Automation Hub

Added in version 1.1-upd1.

Private Automation Hub можно развернуть на одном или нескольких узлах.

Внимание: Private Automation Hub и Automation Controller нельзя развертывать на одном узле.

Развертывание Private Automation Hub на одном узле

Для развертывания Private Automation Hub на одном узле выполните следующие действия:

1. Создайте секцию `[automationhub]` и добавьте в нее сведения об узле:

   ```
   [automationhub]
   hub.example.com
   ```

2. В секции `[all:vars]` укажите пароль администратора:

   ```
   [all:vars]
   automationhub_admin_password = "<admin_password>"
   ```

Фрагмент файла инвентаря для развертывания Private Automation Hub на одном узле:

```
[automationhub]
hub.example.com

[all:vars]
ansible_python_interpreter = /usr/bin/python3
ansible_user = 'admin'
ansible_ssh_private_key_file = '/path/to/private/ssh/key'
# Доменное имя, по которому будет доступен Private Automation Hub.
automationhub_main_url = https://hub.example.com/
automationhub_admin_password = 'automationhub'
```

Отказоустойчивая конфигурация Private Automation Hub

Для развертывания отказоустойчивой конфигурации Private Automation Hub необходимо выполнение следующих условий:

- Использование балансировщика нагрузки.
- Использование внешнего высокодоступного сервиса PostgreSQL. Предполагается, что такой сервис уже настроен и доступен по доменному имени `rdbs.example.com` через порт 5432.
- Развертывание Private Automation Hub не менее чем на двух узлах.

С учетом описанных требований, для развертывания отказоустойчивой конфигурации Private Automation Hub выполните следующие действия:

1. В секции `[automationhub]` перечислите все узлы, на которых необходимо развернуть Private Automation Hub:

```
[automationhub]
hub1.example.com
hub2.example.com
```

2. В секции `[automationhub:vars]` укажите параметры подключения к S3-совместимому бакету, который будет использоваться для хранения *содержимого*:

```
[automationhub:vars]
s3_endpoint_url = https://s3-storage.example.com
s3_region_name = ru-central1
s3_access_key = '<access_key>'
s3_secret_key = '<secret_key>'
s3_bucket_name = 'private-hub-example-com'
```

3. В секции `[all:vars]` укажите следующие данные:
   - параметры подключения к СУБД;
   - пароль администратора;
   - URL, который пользователи будут использовать для доступа к Private Automation Hub.

```
[all:vars]
automationhub_pg_host = rdbs.example.com
automationhub_pg_port = '5432'
automationhub_pg_database = 'automationhub'
automationhub_pg_username = 'admin'
automationhub_pg_password = 'automationhub'

automationhub_admin_password = 'automationhub'

automationhub_main_url = https://hub.example.com/
```

Фрагмент файла инвентаря с примером отказоустойчивой конфигурации Private Automation Hub:

```
[automationhub]
hub1.example.com
hub2.example.com

[database]

[all:vars]
ansible_python_interpreter = /usr/bin/python3
ansible_user = 'admin'
ansible_ssh_private_key_file = '/path/to/private/ssh/key'
# Адрес узла, к которому будут подключаться клиенты. Это должен быть адрес␣
```

(continues on next page)

(продолжение с предыдущей страницы)

```
↪балансировщика нагрузки, например, https://<load_balancer_host>.
automationhub_main_url = https://hub.example.com/
automationhub_admin_password = 'automationhub'
automationhub_pg_host = rdbs.example.com
automationhub_pg_port = '5432'

[automationhub:vars]
s3_endpoint_url = https://s3-storage.example.com
s3_region_name = ru-central1
s3_access_key = '<access_key>'
s3_secret_key = '<secret_key>'
s3_bucket_name = 'private-hub-example-com'
```

Подробное описание настроек и их возможные значения см. в *справочнике*.

Управление сертификатами TLS/SSL

В этом документе описываются следующие операции с сертификатами TLS, используемыми для защиты подключения к различным компонентам платформы Astra Automation:

- настройка платформы на использование сертификатов, выданных удостоверяющим центром;
- перевыпуск самоподписанного сертификата средствами платформы;
- самостоятельная генерация самоподписанного сертификата.

Все приведенные ниже инструкции необходимо выполнять на *установочном узле*.

Использование сертификатов TLS, выданных удостоверяющим центром

Чтобы использовать для защиты подключения к компонентам платформы сертификаты, выданные удостоверяющим центром, выполните следующие действия:

1. Убедитесь, что используемые сертификаты удовлетворяют требованиям, указанным в документе *Сертификаты TLS/SSL*.
2. Скопируйте сертификаты и соответствующие им ключи в каталог `/opt/rbta/aa/astra-automation-setup/` или один из его подкаталогов.
3. Перейдите в каталог `/opt/rbta/aa/astra-automation-setup/`:

   ```
   cd /opt/rbta/aa/astra-automation-setup/
   ```

4. В файле `inventory` укажите пути к файлам сертификатов и соответствующих им ключей:

   - `web_server_ssl_cert` и `web_server_ssl_key` – сертификат и ключ для веб-сервера Automation Controller.

     Если для каждого узла из плоскости управления Automation Controller выпущен отдельный сертификат, укажите пути к файлам в свойствах узлов, например:

     ```
     [automationcontroller]
     ctrl1.example.com  ansible_host=192.0.2.11  web_server_ssl_cert=/path/to/ctrl-
     ↪cert1.pem    web_server_ssl_key=/path/to/ctrl-key1.key
     ctrl2.example.com  ansible_host=192.0.2.12  web_server_ssl_cert=/path/to/ctrl-
     ↪cert2.pem    web_server_ssl_key=/path/to/ctrl-key2.key
     ctrl3.example.com  ansible_host=192.0.2.13  web_server_ssl_cert=/path/to/ctrl-
     ↪cert3.pem    web_server_ssl_key=/path/to/ctrl-key3.key
     ```

     Если для всех узлов плоскости управления Automation Controller используется один и тот же Wildcard-сертификат, пути к файлам можно указать в переменных группы `automationcontroller`:

```
[automationcontroller]
ctrl1.example.com  ansible_host=192.0.2.11
ctrl2.example.com  ansible_host=192.0.2.12
ctrl3.example.com  ansible_host=192.0.2.13

[automationcontroller:vars]
web_server_ssl_cert=/path/to/ctrl-cert.pem
web_server_ssl_key=/path/to/ctrl-key.key
```

- `automationhub_ssl_cert` и `automationhub_ssl_key` – сертификат и ключ для *Private Automation Hub*.

  Если для каждого узла Private Automation Hub выпущен отдельный сертификат, укажите пути к файлам в свойствах узлов, например:

```
[automationhub]
hub1.example.com   automationhub_ssl_cert=/path/to/hub-cert1.pem  automationhub_
↪ssl_key=/path/to/hub-key1.key
hub2.example.com   automationhub_ssl_cert=/path/to/hub-cert2.pem  automationhub_
↪ssl_key=/path/to/hub-key2.key
hub3.example.com   automationhub_ssl_cert=/path/to/hub-cert3.pem  automationhub_
↪ssl_key=/path/to/hub-key3.key
```

  Если для всех узлов Private Automation Hub используется один и тот же Wildcard-сертификат, пути к файлам можно указать в переменных группы `automationhub`:

```
[automationhub]
hub1.example.com
hub2.example.com
hub3.example.com

[automationhub:vars]
automationhub_ssl_cert=/path/to/hub-cert.pem
automationhub_ssl_key=/path/to/hub-key.key
```

- `automationedacontroller_ssl_cert` и `automationedacontroller_ssl_key` – сертификат и ключ для *Event-Driven Ansible Controller*.

  Значения этих параметров необходимо указывать в секции `[all:vars]`:

```
[all:vars]
automationedacontroller_ssl_cert=/path/to/cert.pem
automationedacontroller_ssl_key=/path/to/key.key
```

- `postgres_ssl_cert` и `postgres_ssl_key` – сертификат и ключ сервера СУБД PostgreSQL, развернутого средствами платформы Astra Automation.

  Значения этих параметров необходимо указывать в блоке `[all:vars]`, например:

```
[all:vars]
postgres_ssl_cert=/path/to/pg-cert.pem
postgres_ssl_key=/path/to/pg-key.key
```

  Для сервера PostgreSQL, развернутого иным способом, настройка этих параметров не требуется.

- `mesh_ca_certfile` и `mesh_ca_keyfile` – сертификат и ключ для сети Mesh, используемой Automation Controller.

  Значения этих параметров необходимо указывать в блоке `[all:vars]`, например:

```
[all:vars]
mesh_ca_certfile=/path/to/mesh-cert.pem
mesh_ca_keyfile=/path/to/mesh-key.key
```

Совет: Заполнение инвентаря можно упростить с помощью использования переменных. Пример см. в разделе *Использование переменных при заполнении инвентаря*.

5. Если в секции `[all:vars]` присутствует параметр `aap_service_regen_cert`, закомментируйте или удалите строку с ним.
6. Запустите утилиту `aa-setup`:

```
sudo ./aa-setup
```

Автоматизированный перевыпуск самоподписанного сертификата

Чтобы перевыпустить и переустановить самоподписанный сертификат, используя средства платформы, выполните следующие действия:

1. Перейдите в каталог `/opt/rbta/aa/astra-automation-setup/`:

```
cd /opt/rbta/aa/astra-automation-setup/
```

2. В файле `inventory` добавьте в секцию `[all:vars]` переменную `aap_service_regen_cert` со значением `true`:

```
[all:vars]
aap_service_regen_cert=true
```

3. Запустите утилиту `aa-setup`:

```
sudo ./aa-setup
```

4. Удалите или закомментируйте добавленную ранее строку с параметром `aap_service_regen_cert`.

Использование самоподписанных сертификатов

Чтобы выпустить самоподписанный сертификат вручную, выполните следующие действия:

1. Для генерации ключа выполните команду:

```
openssl genrsa -out /path/to/key.key 2048
```

   Здесь `/path/to/key.key` – путь к файлу для сохранения ключа.

2. Для выпуска сертификата на основе созданного ранее ключа выполните команду:

```
openssl req \
    -x509 \
    -sha256 \
    -new \
    -nodes \
    -key /path/to/key.key \
    -days 3650 \
    -out /path/to/cert.pem \
    -subj "/C=RU/ST=Moscow/L=Moscow/O=Company/OU=IT/CN=CertificateAuthority"
```

   где:

   - `-x509` – команда выпуска сертификата;
   - `-sha256` – алгоритм хэширования;
   - `-new` – создание нового сертификата;

- `-key` – полный путь к файлу созданного ранее ключа;
- `-days` – срок действия сертификата, в днях;
- `-out` – путь к файлу для сохранения сертификата;
- `-subj` – параметры сертификата:
  - `C` – двухсимвольный код страны в соответствии с ISO-3166-1;
  - `ST` – название региона, в котором зарегистрирована организация;
  - `L` – город, в котором зарегистрирована организация;
  - `O` – наименование организации в соответствии с уставом организации;
  - `OU` – наименование подразделения;
  - `CN` – FQDN веб-сервера.

Подробности об использовании команды `openssl` см. в документации OpenSSL[6].

3. Установите сертификат согласно инструкций, приведенных в секции *Использование сертификатов TLS, выданных удостоверяющим центром*.

Параметры СУБД

СУБД PostgreSQL необходима для хранения данных компонентов платформы. Возможно использование уже существующего кластера PostgreSQL, например, развернутого в одном из облачных сервисов управляемых баз данных. Если отдельного кластера PostgreSQL нет, экземпляр СУБД можно средствами платформы развернуть на одном из узлов.

Примечание: СУБД PostgreSQL используется платформой для хранения данных, но не является ее частью. Настройка отказоустойчивой конфигурации PostgreSQL в этом руководстве не рассматривается. Для получения соответствующих инструкций обращайтесь к документации PostgreSQL[7].

Развертывание средствами платформы

Чтобы средствами платформы развернуть экземпляр СУБД на отдельном узле, выполните следующие действия:

1. Создайте в инвентаре секцию `[database]` и добавьте в нее сведения об узле, например:

```
[database]
rdbs.example.com   ansible_host=192.0.2.101
```

Важно: Узел, используемый для развертывания PostgreSQL, не должен использоваться для других целей, например, запуска заданий или управления контроллером.

---

2. В секции `[all:vars]` укажите значения параметров подключения к СУБД.

[6] https://docs.openssl.org/master/man1/openssl-req/
[7] https://www.postgresql.org/docs/current/high-availability.html

Внешняя СУБД

Если у вас уже есть кластер PostgreSQL, выполните следующие действия:

1. Создайте в кластере PostgreSQL пользователей и принадлежащие им базы данных для используемых компонентов платформы.
2. Убедитесь, что в инвентаре установщика секция `[database]` существует, но не содержит узлов.
3. В секции `[all:vars]` укажите значения параметров подключения к СУБД.

Параметры подключения к PostgreSQL

Параметры подключения Automation Controller и Private Automation Hub к СУБД задаются в секции `[all:vars]`:

```
[all:vars]
# СУБД для Automation Controller
pg_host = ''
pg_port = 5432
pg_database = 'aac'
pg_username = 'aac'
pg_password = ''
pg_sslmode = 'prefer'
postgres_use_ssl = True
postgres_ssl_cert = <path_to_SSL_cert>
postgres_ssl_key = <path_to_SSL_key>

# СУБД для Automation Hub
automationhub_pg_host = '192.0.2.101'
automationhub_pg_port = '5432'
automationhub_pg_database = 'automationhub'
automationhub_pg_username = 'automationhub'
automationhub_pg_password = 'hubpgpassword'
automationhub_pg_sslmode = 'prefer'
```

Здесь:

- `pg_host` и `automationhub_pg_host` – IP-адреса или FQDN серверов СУБД для Automation Controller и Private Automation Hub соответственно.

  Значение по умолчанию – пустая строка (используется локальное подключение).

- `pg_port` и `automationhub_pg_port` – порты для подключения к серверам СУБД для Automation Controller и Private Automation Hub соответственно.

  Значение по умолчанию – 5432.

- `pg_database` и `automationhub_pg_database` – названия БД для Automation Controller и Private Automation Hub соответственно.
- `pg_username` и `automationhub_pg_username` – названия учетных записей пользователей БД Automation Controller и Private Automation Hub соответственно.
- `pg_password` и `automationhub_pg_password` – пароли пользователей БД Automation Controller и Private Automation Hub соответственно.

  При использовании локального подключения пароль указывать не требуется.

- `pg_sslmode` и `automationhub_pg_sslmode` – режим использования SSL.

  Может принимать следующие значения:

  - `prefer` – подключение к БД будет защищено с помощью SSL, если использование шифрования поддерживается настройками сервера (по умолчанию);
  - `verify-full` – защита подключения с помощью SSL необходима, производится строгая верификация сервера.

- `postgres_use_ssl` – использование SSL при подключении к БД Automation Controller:
  - `True` – включено;
  - `False` – выключено.
- `postgresql_ssl_cert` – полный путь к файлу сертификата SSL, используемого для защиты подключения к СУБД Automation Controller.
- `postgresql_ssl_key` – полный путь к файлу ключа для сертификата SSL, используемого для защиты подключения к СУБД Automation Controller.

Примечание: Файлы сертификата и его ключа должны размещаться в каталоге `/opt/rbta/aa/astra-automation-setup/` или одном из его подкаталогов.

Подробности о поддерживаемых параметрах шифрования см. в документации PostgreSQL[8].

Использование SSO

Установщик платформы позволяет настроить компоненты платформы на использование единой системы идентификации пользователей (*SSO*) с использованием метода SAML.

Примечание: Automation Controller может использовать *различные методы аутентификации*, но в инвентаре установщика можно задать настройки только для SAML.

Подробное описание параметров SSO и их возможные значения см. в *справочнике*.

Подготовка к работе

Для настройки SSO необходимы следующие сведения о поставщике услуг SAML:

- URL;
- идентификатор профиля платформы на стороне поставщика услуг SAML (Entity ID);
- открытый сертификат;
- закрытый ключ;
- URL поставщика идентификационных данных (Identity Provider, IdP);
- идентификатор поставщика идентификационных данных.

Для получения указанных сведений обратитесь к поставщику услуг SAML.

Примечание: Сервис SAML на момент установки должен быть в рабочем состоянии.

---

[8] https://www.postgresql.org/docs/current/libpq-ssl.html

Параметры SSO для Automation Controller

В секции `[all:vars]` укажите параметры подключения к поставщику услуг SAML. Например, чтобы использовать сервер Keycloak[9], укажите параметры подключения к нему следующим образом:

```
[all:vars]
sso_setup=true
sso_url='https://<saml_host>:<saml_port>'
entity_id='<ac_entity>'
pub_cert_path='./saml.crt'
private_key_path='./saml.key'
provider_url='https://<saml_host>:<saml_port>/auth/realms/master/protocol/saml'
provider_id='https://<saml_host>:<saml_port>/auth/realms/master'
sso_displayname='Single Sign-On with Keycloak'
attr_email='email'
attr_groups='groups'
attr_username='email'
attr_last_name='last_name'
attr_first_name='first_name'
attr_user_permanent_id='email'
tech_contact_name='admin'
tech_contact_email='tech-support@example.com'
support_contact_name='admin'
support_contact_email='user-support@example.com'
```

Здесь:

- <saml_host> – адрес сервера поставщика услуг SAML;
- <saml_port> – порт для подключения к серверу поставщика услуг SAML;
- <ac_entity> – идентификатор объекта поставщика услуг SAML. Как правило, здесь указывается базовый URL Automation Controller.

Параметры SSO для Private Automation Hub

Чтобы включить авторизацию в Private Automation Hub через SSO, в секции `[all:vars]` укажите следующие параметры:

```
[all:vars]
sso_host='<host>'
sso_port='443'
sso_automation_platform_realm='<platform_realm>'
sso_protocol='https'
sso_automation_hub_client_id='<client_id>'
sso_client_secret="<client_secret>"
sso_realm_public_key="<realm_public_key>"
sso_automation_hub_authorization_url="<auth_url>"
sso_automation_hub_token_url="<token_url>"
```

[9] https://www.keycloak.org/

Известные проблемы

Развертывание платформы будет прервано сообщением об ошибке, если в инвентаре существует пустая секция `[automationcontroller]`, а значения переменных `admin_username`, `admin_email` и `admin_password` в секции `[all:vars]` не заданы.

4.4 Рекомендуемая топология

Согласно *архитектуре*, платформа содержит определенный набор компонентов. Конкретная топология платформы должна учитывать множество требований, предъявляемых к процессам и средствам автоматизации. Рекомендуется использовать один из рассматриваемых здесь вариантов, прошедших необходимое тестирование. Все описанные топологии содержат полный набор компонентов платформы, но различаются отказоустойчивостью, сложностью настройки и количеством узлов.

- *Минимальная топология* – проста в настройке, нетребовательна к количеству узлов и их характеристикам, хорошо подходит для изучения возможностей платформы, но не обеспечивает отказоустойчивости.

  Важно: Эта топология не рекомендуется для использования в производственных целях.

- *Базовая топология* – расширяет возможности минимальной топологии, снижая нагрузку на Automation Controller за счет использования отдельного узла для запуска заданий Ansible. Не обеспечивает отказоустойчивости, однако, может быть удобна для начальной стадии развития процессов автоматизации информационных технологий в компании.
- *Топология уровня предприятия* – обеспечивает высокое быстродействие и отказоустойчивость, однако требует больших затрат на ресурсы. Рекомендуется для промышленного применения на предприятиях с высокими требованиями к надежности информационной инфраструктуры.

4.4.1 Характеристики узлов

Характеристики узлов, на которых тестировались приведенные топологии:

- Узлы платформы и развернутой ее средствами СУБД

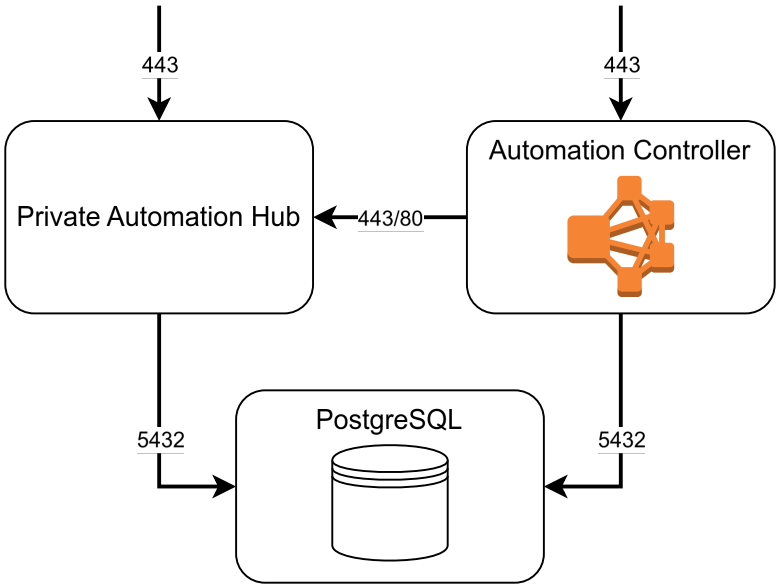
Параметр	Значение
Количество ядер CPU	8
Количество RAM, ГБ	16
Дисковое пространство, ГБ	50
Тип дискового накопителя	Сетевой SSD
IOPS	2000
Тип ОС	Astra Linux Special Edition 1.7.5uu1
Версия ядра ОС	`6.1.50-generic`
Режим защищенности ОС	Базовый («Орел»)

- Узлы СУБД во внешнем сервисе

Параметр	Значение
Количество ядер CPU	8
Количество RAM, ГБ	32
Дисковое пространство, ГБ	100
Лимит на количество одновременно открытых подключений	300

4.4.2 Минимальная топология

Топология платформы, развернутой на минимальном количестве узлов, показана на схеме:



Подключение пользователей к Automation Controller и Private Automation Hub выполняется напрямую через порт 443.

Для СУБД, Private Automation Hub и Automation Controller требуется всего по одному узлу.

Automation Controller взаимодействует с Private Automation Hub через порты 80 и 443.

СУБД развертывается на отдельном узле средствами платформы. Automation Controller и Private Automation Hub подключаются к своим базам данных через порт 5432.

Параметры сетевого взаимодействия компонентов платформы:

Порт	Протокол	Служба	Источник	Цель
80/443	TCP	HTTP/HTTPS	Automation Controller	Private Automation Hub
443	TCP	HTTPS	Пользователи	Automation Controller
443	TCP	HTTPS	Пользователи	Private Automation Hub
5432	TCP	PostgreSQL	Private Automation Hub	База данных
5432	TCP	PostgreSQL	Automation Controller	База данных

Пример файла инвентаря для минимальной топологии:

```
[automationcontroller]
ctrl.example.com  ansible_host=192.168.56.11

[automationhub]
hub.example.com   ansible_host=192.168.56.12

[database]
db.example.com    ansible_host=192.168.56.101

[all:vars]
ansible_ssh_private_key_file=/path/to/private/ssh/key
ansible_user=admin

admin_email=admin@example.com
admin_password="ctRlp@ssW0rd"
admin_username=admin

pg_host="192.168.56.101"
pg_port=5432
```

(continues on next page)

(продолжение с предыдущей страницы)

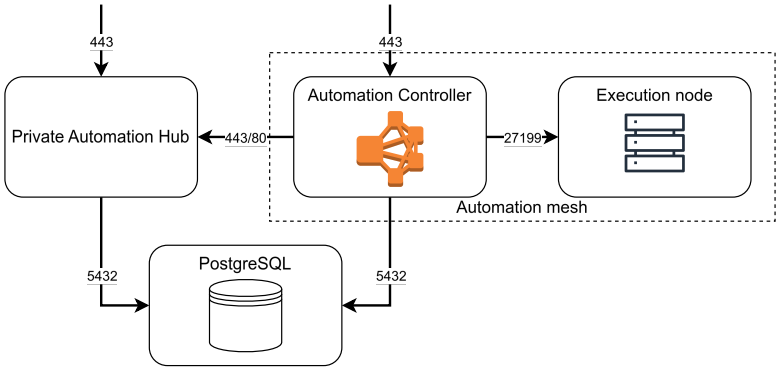
```
pg_database='automationcontroller'
pg_username='automationcontroller'
pg_password='ctrlDBpaS$123456'
pg_ssl_mode='prefer'

# Automation Hub
automationhub_admin_password="hU8p4ssWd1234"
automationhub_main_url="https://hub.example.com"
automationhub_pg_host="192.168.56.101"
automationhub_pg_port=5432
automationhub_pg_database='automationhub'
automationhub_pg_username='automationhub'
automationhub_pg_password='hUbPa55I2345b'
automationhub_pg_sslmode='prefer'
```

Здесь:

IP-адрес	Компонент	Примечание
192.168.56.11	Automation Controller	Гибридный узел
192.168.56.12	Private Automation Hub	–
192.168.56.101	СУБД PostgreSQL	Развертывается средствами платформы

4.4.3 Базовая топология

Топология платформы, использующей для запуска заданий отдельный узел, показана на схеме:



Подключение пользователей к Automation Controller и Private Automation Hub выполняется напрямую через порт 443.

Для СУБД, Private Automation Hub, управляющей и исполняющей плоскостей Automation Controller требуется всего по одному узлу.

Automation Controller взаимодействует с Private Automation Hub через порты 80 и 443.

СУБД развертывается на отдельном узле средствами платформы. Automation Controller и Private Automation Hub подключаются к своим базам данных через порт 5432.

Связь между управляющим и исполняющим узлами Automation Controller обеспечивается через порт 27199 сети Automation mesh.

Параметры сетевого взаимодействия компонентов платформы:

Порт	Протокол	Служба	Источник	Цель
80/443	TCP	HTTP/HTTPS	Automation Controller	Private Automation Hub
443	TCP	HTTPS	Пользователи	Automation Controller
443	TCP	HTTPS	Пользователи	Private Automation Hub
5432	TCP	PostgreSQL	Private Automation Hub	База данных
5432	TCP	PostgreSQL	Automation Controller	База данных
27199	TCP	Receptor	Automation Controller	Исполняющий узел

Пример файла инвентаря для базовой топологии:

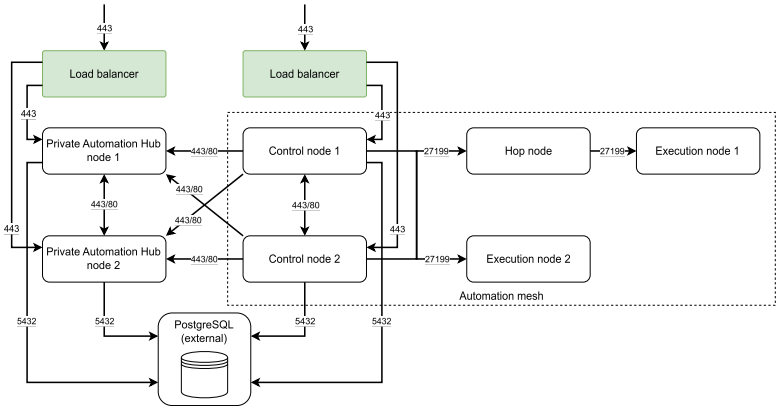
```
[automationcontroller]
ctrl.example.com  ansible_host=192.168.56.11  node_type=control

[automationcontroller:vars]
peers=execution_nodes

[execution_nodes]
exec.example.com  ansible_host=192.168.56.12

[automationhub]
hub.example.com   ansible_host=192.168.56.13

[database]
db.example.com    ansible_host=192.168.56.101

[all:vars]
ansible_ssh_private_key_file=/path/to/private/ssh/key
ansible_user=admin

admin_email=admin@example.com
admin_password="ctRlp@ssW0rd"
admin_username=admin

pg_host="192.168.56.101"
pg_port=5432
pg_database='automationcontroller'
pg_username='automationcontroller'
pg_password='ctrlDBpaS$123456'
pg_ssl_mode='prefer'

# Automation Hub
automationhub_admin_password="hU8p4ssWd1234"
automationhub_main_url="https://hub.example.com"
automationhub_pg_host="192.168.56.101"
automationhub_pg_port=5432
automationhub_pg_database='automationhub'
automationhub_pg_username='automationhub'
automationhub_pg_password='hUbPa55I2345b'
automationhub_pg_sslmode='prefer'
```

Здесь:

IP-адрес	Компонент	Примечание
192.168.56.11	Automation Controller	Управляющий узел
192.168.56.12	Automation Controller	Исполняющий узел, подключенный к плоскости управления напрямую
192.168.56.13	Private Automation Hub	–
192.168.56.101	СУБД PostgreSQL	Развертывается средствами платформы

4.4.4 Топология уровня предприятия

Одним из важнейших требований, предъявляемых к платформе со стороны предприятий, является отказоустойчивость. Топология платформы, развернутой в отказоустойчивой конфигурации, показана на схеме:



Такая топология обеспечивает отказоустойчивость и повышенное быстродействие платформы путем горизонтального масштабирования кластеров.

Подключение к узлам Private Automation Hub и Automation Controller выполняется через балансировщики нагрузки или отказоустойчивые прокси-серверы (не входят в состав платформы) через порт 443.

Узлы Automation Controller взаимодействуют с узлами Private Automation Hub через порты `80` и 443.

Для хранения баз данных используется внешний сервис PostgreSQL. Automation Controller и Private Automation Hub подключаются к своим базам данных через порт 5432.

Для обеспечения отказоустойчивости и Private Automation Hub, и Automation Controller развертываются на нескольких узлах каждый.

Связь между управляющими, промежуточными и исполняющими узлами Automation Controller выполняется через порт 27199 сети Automation mesh. Подключение управляющих узлов к исполняющим может выполняться как напрямую, так и через промежуточные узлы. Если все узлы платформы находятся в одной сети, использование промежуточных узлов не требуется.

Параметры сетевого взаимодействия компонентов платформы:

Порт	Протокол	Служба	Источник	Цель
`80`/443	TCP	HTTP/HTTPS	Automation Controller	Private Automation Hub
443	TCP	HTTPS	Балансировщик нагрузки	Пользователи
443	TCP	HTTPS	Балансировщик нагрузки	Automation Controller
443	TCP	HTTPS	Балансировщик нагрузки	Private Automation Hub
5432	TCP	PostgreSQL	Private Automation Hub	База данных
5432	TCP	PostgreSQL	Automation Controller	База данных
27199	TCP	Receptor	Automation Controller	Промежуточный и исполняющий узлы
27199	TCP	Receptor	Промежуточный узел	Исполняющий узел

Пример файла инвентаря для топологии уровня предприятия:

```
[automationcontroller]
ctrl1.example.com  ansible_host=192.168.56.11
ctrl2.example.com  ansible_host=192.168.56.12

[automationcontroller:vars]
node_type=control
peers=instance_group_local


[execution_nodes]
hop.example.com    ansible_host=192.168.56.13
exec1.example.com  ansible_host=192.168.56.14
exec2.example.com  ansible_host=192.168.56.15

[instance_group_local]
exec1.example.com

[hop]
hop.example.com

[hop:vars]
node_type=hop
peers=automationcontroller

[instance_group_remote]
exec1.example.com

[instance_group_remote:vars]
peers=hop

[automationhub]
hub1.example.com   ansible_host=192.168.56.16
hub2.example.com   ansible_host=192.168.56.17

[database]
# Используется внешний кластер PostgreSQL

[all:vars]
ansible_ssh_private_key_file=/path/to/private/ssh/key
ansible_user=admin

admin_email=admin@example.com
admin_password="ctRlp@ssW0rd"
admin_username=admin

pg_host="pg-ha.example.com"
pg_port=5432
pg_database='automationcontroller'
pg_username='automationcontroller'
pg_password='ctrlDBpaS$123456'
pg_ssl_mode='prefer'

# Automation Hub
automationhub_admin_password="hU8pAssWd1234"
automationhub_main_url="https://hub.example.com"
automationhub_pg_host="pg-ha.example.com"
automationhub_pg_port=5432
automationhub_pg_database='automationhub'
automationhub_pg_username='automationhub'
automationhub_pg_password='hUbPa55I2345b'
automationhub_pg_sslmode='prefer'
```

Здесь:

IP-адрес	Компонент	Примечание
192.168.56.11	Automation Controller	Узел плоскости управления
192.168.56.12	Automation Controller	Узел плоскости управления
192.168.56.13	Automation Controller	Промежуточный узел
192.168.56.14	Automation Controller	Исполняющий узел, подключенный к плоскости управления через промежуточный узел
192.168.56.15	Automation Controller	Исполняющий узел, подключенный к плоскости управления напрямую
192.168.56.16	Private Automation Hub	–
192.168.56.17	Private Automation Hub	–

Секция `[database]` оставлена пустой, поскольку развертывание сервера PostgreSQL средствами платформы в этом случае не используется. Предполагается, что отказоустойчивый сторонний сервис PostgreSQL доступен по доменному имени `pg-ha.example.com`.

4.5 Развертывание без доступа к интернету

Развертывание платформы в окружении без доступа к интернету имеет ряд особенностей, которые рассматриваются ниже.

4.5.1 Развертывание зеркал репозиториев

Поскольку при установке в окружении без доступа к интернету невозможно использовать репозитории ПАО Группа Астра, следует самостоятельно развернуть их копии таким образом, чтобы они были доступны для узлов, используемых для развертывания платформы.

Пошаговые инструкции по созданию зеркал приведены в статье справочного центра Создание локальных и сетевых репозиториев[10].

При использовании ОС Astra Linux Special Edition 1.7 локально должны быть развернуты копии базового (base) и расширенного (extended) репозиториев.

Важно: Версии используемых репозиториев должны соответствовать версии ОС, установленной на узлах платформы.

4.5.2 Подготовка узлов

Общие настройки узлов в окружении без доступа к интернету не отличаются от настроек в окружении с доступом к интернету:

1. Обеспечьте сетевую связность всех узлов.
2. Настройте корректное разрешение имен узлов.
3. Обеспечьте доступ к узлам по SSH с использованием ключей согласно *инструкции*.
4. Обеспечьте выполнение команд с использованием `sudo` без ввода пароля согласно *инструкции*.

[10] https://wiki.astralinux.ru/pages/viewpage.action?pageId=61575159

4.5.3 Настройка установочного узла

Для настройки установочного узла выполните следующие действия:

1. Добавьте в файл `/etc/apt/sources.list` ссылки на развернутые зеркала репозиториев, например:

```
deb http://192.168.0.1/astra/frozen/1.7_x86-64/1.7.5/repository-base     1.7_x86-64␣
↪main non-free contrib
deb http://192.168.0.1/astra/frozen/1.7_x86-64/1.7.5/repository-extended 1.7_x86-64␣
↪main non-free contrib
```

2. Загрузите архив с установщиком на установочный узел.

   Примечание: Архив доступен для загрузки в Личном кабинете[11] при наличии действующей лицензии на продукт.

3. Создайте каталог для файлов установщика, например:

```
sudo mkdir -p /var/aa-setup/
```

4. Распакуйте содержимое архива в созданный каталог, например:

```
sudo tar -xvzf <archive>.tgz -C /var/aa-setup/
```

4.5.4 Настройка инвентаря

При настройке инвентаря выполните следующие действия:

1. Опишите параметры платформы согласно инструкций из раздела *Описание инвентаря*.
2. Добавьте в секцию `[all:vars]` следующие строки:

```
bundle_install=true
registry_ip='<registry_ip>'
registry_port='5000'
bundle_dir=/var/aa-setup/
```

   Здесь:

   - `registry_ip` – IP-адрес установочного узла.

     Важно: Нельзя использовать адрес локальной петли `127.0.0.1`.

     Изменено в версии 1.1-upd1: До версии 1.1-upd1 этот параметр назывался `docker_registry_ip`.

   - `registry_port` – порт, который слушает служба образов.

     Изменено в версии 1.1-upd1: До версии 1.1-upd1 этот параметр назывался `docker_registry_port`.

   - `bundle_dir` – путь к каталогу с файлами установщика платформы.

     Примечание: Значение этого параметра необходимо задавать только в том случае, когда путь к каталогу с файлами установщика платформы отличается от `/opt/rbta/aa/astra-automation-setup/`.

[11] https://lk-new.astralinux.ru

Примечание: Подробное описание параметров приводится в документе *Параметры инвентаря*.

4.6 Установка системы

После указания в инвентаре учетных данных системного администратора Automation Controller, пароля администратора Private Automation Hub, СУБД и параметров узлов запустите развертывание платформы. Ее работоспособность проверьте простейшим тестированием.

4.6.1 Процедура установки

В каталоге `/opt/rbta/aa/astra-automation-setup/` выполните команду:

```
./aa-setup --log-path=<log_path>
```

В значении параметра `--log-path` укажите путь к каталогу для сохранения журнала установки. Этот каталог должен быть доступен для записи от имени текущего пользователя.

Чтобы использовать путь к журналу установки по умолчанию, выполните команду с привилегиями суперпользователя:

```
sudo ./aa-setup
```

По умолчанию необходимые пакеты устанавливаются из репозитория `dl.astralinux.ru`.

Дождитесь завершения выполнения команды, это может занять некоторое время. По окончании установки в терминал выводится набор строк вида (названия узлов и их количество должны совпадать со значениями, указанными в файле инвентаря):

```
INFO: PLAY RECAP *********************************************************************
aac01                      : ok=233  changed=118  unreachable=0    failed=0   ␣
↪skipped=74   rescued=0    ignored=4
aac02                      : ok=191  changed=107  unreachable=0    failed=0   ␣
↪skipped=72   rescued=0    ignored=2
db                         : ok=41   changed=20   unreachable=0    failed=0    skipped=9␣
↪   rescued=0    ignored=0
ex01                       : ok=82   changed=46   unreachable=0    failed=0   ␣
↪skipped=47   rescued=0    ignored=1
ex02                       : ok=82   changed=46   unreachable=0    failed=0   ␣
↪skipped=47   rescued=0    ignored=1
INFO: Astra Automation Setup has been finished

You can now login to Astra Automation Controller using following credentials:

Login: admin
Password: p@ssw0rd!
```

Развертывание считается успешным, если для всех узлов в поле `failed` указано значение `0`.

Примечание: Синтаксис и полный список параметров утилиты `aa-setup` приведены в *справочных данных*.

4.6.2 Проверка работоспособности компонентов платформы

Здесь приводятся методики проверки работоспособности компонентов платформы Astra Automation после развертывания и обновления.

Automation Controller

Чтобы проверить корректность развертывания контроллера, выполните следующие действия:

1. Запустите браузер и введите в адресной строке IP-адрес или FQDN управляющего узла Automation Controller.

   Если управляющих узлов несколько – выберите любой из них и подключитесь к нему.

   Примечание: Если используется самоподписанный сертификат, в окне браузера может выводиться предупреждение о вероятной угрозе безопасности. В данном случае подтвердите исключение безопасности.

2. В форму для ввода учетных данных введите имя и пароль администратора контроллера.
3. На панели навигации выберите раздел *Администрирование* ▸ *Узлы контроллера* (*Administration* ▸ *Instances*).
4. Убедитесь, что все узлы, указанные ранее в файле `inventory`, находятся в статусе **Готов** (Ready).

Private Automation Hub

Чтобы проверить корректность развертывания Private Automation Hub, выполните следующие действия:

1. Запустите браузер и введите в адресной строке IP-адрес или FQDN Private Automation Hub.

   Примечание: Если используется самоподписанный сертификат, в окне браузера может выводиться предупреждение о вероятной угрозе безопасности. В данном случае подтвердите исключение безопасности.

2. В форме для ввода учетных данных введите имя пользователя `admin` и пароль администратора, указанный в инвентаре установщика в значении параметра `automationhub_admin_password`.

4.7 Обновление версии

Обновление версии Astra Automation осуществляется с помощью утилиты `aa-setup`. Она может обновить Astra Automation до определенной или последней версии.

4.7.1 Подготовка к обновлению

Необходимо обеспечить выполнение следующих условий:

- Для обновления используется тот же установочный узел, с помощью которого происходило первоначальное развертывание платформы. Вместо такого узла можно воспользоваться другим, на котором установлен пакет для развертывания Astra Automation и восстановлены необходимые настройки установочного узла.
- На этом узле остался в неизменном виде каталог `/opt/rbta/aa/astra-automation-setup/`, в частности файл инвентаря `/opt/rbta/aa/astra-automation-setup/inventory`, описывающий состав платформы.

4.7.2 Обновление до последней версии

Запустите процесс обновления одним из следующих способов:

- Запуск с использованием привилегий суперпользователя:

  ```
  sudo ./aa-setup --upgrade
  ```

- Запуск с привилегиями обычного пользователя:

  ```
  ./aa-setup --upgrade --log-path=<log_path>
  ```

  В значении параметра `--log-path` укажите путь к каталогу для сохранения журнала установки. Этот каталог должен быть доступен активному пользователю для записи.

При запуске с параметром `--upgrade` утилита `aa-setup` выполняет следующие действия:

1. Соединяется с репозиторием, из которого был установлен Astra Automation, и считывает из него список доступных версий.
2. Определяет версию пакета, использованного для установки или обновления Astra Automation в предыдущий раз. Текущая версия хранится в файле `/etc/astra_automation.version`.
3. Выполняет обновление при необходимости:
   - если последняя версия в репозитории совпадает с версией, установленной на узле, то обновление не производится, `aa-setup` завершает свою работу;
   - если версия, установленная на узле, ниже самой последней в репозитории, то автоматически производится обновление утилиты `aa-setup` на самую новую версию, утилита перезапускается и обновляет Astra Automation.

Необходимо дождаться завершения выполнения команды, это может занять некоторое время. По окончании обновления в терминал выводится строка вида:

```
aac      : ok=233  changed=130  unreachable=0    failed=0    skipped=55    rescued=0   ␣
↪ignored=4
```

Обновление считается успешным, если в поле `failed` указано значение `0`.

4.7.3 Обновление до определенной версии

Для обновления Astra Automation до определенной версии используйте параметр `--upgrade` в сочетании со следующими параметрами:

- `--repo-url` – когда необходимо указать репозиторий, отличный от используемого по умолчанию (`dl.astralinux.ru`).

  Например, для использования репозитория `https://example.com/artifactory/automation-debs-for-alse` аналогично тому, как тот же репозиторий использовался бы при начальной установке Astra Automation, воспользуйтесь командой:

  ```
  sudo ./aa-setup --upgrade --repo-url https://example.com/artifactory/automation-
  ↪debs-for-alse
  ```

- `--product-version` – когда необходимо обновить платформу до определенной версии.

  Если необходимо обновить платформу до определенной версии <version> (например, `1.1-upd1`), воспользуйтесь командой:

  ```
  sudo ./aa-setup --upgrade --product-version <version>
  ```

Если необходимо использовать другой репозиторий и указать определенную версию воспользуйтесь командой:

```
sudo ./aa-setup --upgrade --repo-url https://some.company.ru/artifactory/automation-debs-
↪for-alse --product-version <version>
```

4.7.4 Универсальный способ обновления

Примечание: Данный способ обновления не является рекомендуемым. Его следует использовать в крайнем случае. Например, когда невозможно удовлетворить требованиям по подготовке.

---

1. Убедитесь, что в списке репозиториев APT в каталоге `/etc/apt/` указана ссылка на репозиторий, содержащий требуемую версию установочного пакета Astra Automation:

   ```
   deb https://dl.astralinux.ru/aa/aa-debs-for-alse-1.7 <version> main
   ```

   Вместо <version> необходимо подставить версию, до которой обновляется платформа, например, `1.1-upd1`.

   Доступные версии продукта опубликованы в таблице *История обновлений*.

2. Обновите индекс пакетов:

   ```
   sudo apt update
   ```

3. Обновите пакет `astra-automation-setup`:

   ```
   sudo apt install --only-upgrade astra-automation-setup
   ```

4. Запустите процесс обновления таким же образом, как вы изначально развертывали платформу:

   ```
   sudo ./aa-setup
   ```

4.8 Миграция

В организациях может возникнуть необходимость мигрировать платформу Astra Automation с одних узлов на другие. При этом часто изменяются доменные имена и IP-адреса узлов платформы.

Чтобы выполнить такую миграцию, следует воспользоваться функциональностью утилиты `aa-setup`. Здесь рассматривается пример миграции платформы Astra Automation с одной конфигурации на другую.

Важно: Допускается миграция платформы между конфигурациями с разным количеством узлов.

Пусть исходная конфигурация платформы имеет следующие параметры:

- Развернут кластер Automation Controller из трех узлов:

Узел	Назначение	FQDN	IP-адрес
`node1`	Управляющий узел	`node1.example.com`	`192.168.56.11`
`node2`	Управляющий узел	`node2.example.com`	`192.168.56.12`
`node3`	Узел СУБД	`node3.example.com`	`192.168.56.13`

- Для администрирования узлов используется учетная запись `administrator`.
- Сервер СУБД PostgreSQL развернут средствами платформы.

Далее эта платформа будет называться исходной платформой, а соответствующий ей установочный узел – исходным установочным узлом.

Пусть целевая конфигурация платформы имеет следующие параметры:

- Для развертывания платформы подготовлены три узла:

Узел	Назначение	FQDN	IP-адрес
`aac1`	Управляющий узел	`aac1.example.com`	`10.120.0.11`
`aac2`	Управляющий узел	`aac2.example.com`	`10.120.0.12`
`rdbs`	Узел СУБД	`rdbs.example.com`	`10.120.0.101`

- Установочный узел и узлы платформы настроены согласно инструкций из документа *Подготовка узлов*.
- Для администрирования узлов платформы используется учетная запись `devops`.
- Сервер СУБД PostgreSQL будет развернут средствами платформы.

Далее эта платформа будет называться целевой платформой, а соответствующий ей установочный узел – целевым установочным узлом.

Совет: В качестве целевого установочного узла можно использовать исходный установочный узел.

4.8.1 Создание резервной копии

Для создания резервной копии подключитесь к исходному *установочному узлу* и выполните следующие действия:

1. Перейдите в каталог `/opt/rbta/aa/astra-automation-setup/`:

```
cd /opt/rbta/aa/astra-automation-setup/
```

2. Запустите утилиту `aa-setup` с аргументом `--backup` (`-b`):

```
sudo ./aa-setup --backup
```

3. Перейдите в каталог `/tmp/backups/astra-automation/`:

```
cd /tmp/backups/astra-automation/
```

4. Получите список файлов, упорядоченных по дате создания:

```
ls -l --sort=time
```

5. Если необходимо использовать другой установочный узел, любым удобным способом скопируйте на него самый новый архив с резервной копией. Например, чтобы скопировать архив по сети, выполните команду:

```
scp <backup> <user>@<host>:<dir>
```

   где:

   - <backup> – имя файла резервной копии, например, `astra-automation-backup-28-05-2024-114821.tar.gz`;
   - <user> – название учетной записи пользователя целевого установочного узла;
   - <host> – FQDN или IP-адрес установочного узла целевого кластера;
   - <dir> – полный путь к каталогу на установочном узле целевого кластера.

4.8.2 Восстановление из резервной копии

Для восстановления платформы из резервной копии подключитесь к целевому установочному узлу и выполните следующие действия:

1. Создайте каталог `/tmp/backups/astra-automation/`:

```
mkdir -p /tmp/backups/astra-automation/
```

2. Скопируйте файл резервной копии в каталог `/tmp/backups/astra-automation/`:

```
cp <backup> /tmp/backups/astra-automation/
```

3. Перейдите в каталог `/opt/rbta/aa/astra-automation-setup/`:

```
cd /opt/rbta/aa/astra-automation-setup/
```

4. В файле `inventory` внесите изменения, соответствующие новой конфигурации платформы. Для описанной выше конфигурации файл `inventory` необходимо привести к следующему виду:

```
[automationcontroller]
aac1.example.com   ansible_host=10.120.0.11
aac2.example.com   ansible_host=10.120.0.12

[database]
rdbs.example.com   ansible_host=10.120.0.101
```

(continues on next page)

(продолжение с предыдущей страницы)

```
[vars:all]
ansible_user=devops
ansible_ssh_private_key_file=/path/to/private/ssh/key
admin_username=admin
admin_email=admin@example.com
admin_password=p@ssW0rd!
pg_host=10.120.0.101
pg_port=5432
```

5. Запустите развертывание платформы:

```
sudo ./aa-setup
```

6. В каталоге `/tmp/` создайте структуру подкаталогов `backups/astra-automation/`:

```
mkdir -p /tmp/backups/astra-automation/
```

7. Скопируйте файл резервной копии в каталог `/tmp/backups/astra-automation/` под именем `astra-automation-backup-latest.tar.gz`, например:

```
cp astra-automation-backup-28-05-2024-114821.tar.gz \
   /tmp/backups/astra-automation/astra-automation-backup-latest.tar.gz
```

8. Запустите восстановление платформы из резервной копии:

```
sudo ./aa-setup --restore
```

9. Для проверки корректности восстановления выполните вход в веб-интерфейс контроллера от имени администратора и проверьте содержимое разделов.

4.9 Расчет ресурсов

Важно: Эта часть документации находится в стадии разработки.

4.10 Справочные данные

Для развертывания платформы и управления ею в процессе эксплуатации используются специальные утилиты. Здесь приведены справочные данные по их назначению и использованию.

4.10.1 aa-setup

В этом документе описаны назначение и синтаксис утилиты Astra Automation Setup (`aa-setup`).

Назначение

Утилита `aa-setup` выполняет следующие действия:

- развертывает платформу в виде нескольких связанных компонентов;
- обновляет версии и структуру компонентов платформы;
- выполняет резервное копирование и восстановление базы данных;
- генерирует и распространяет секретный ключ;
- проверяет версии Astra Automation, доступные в репозитории;
- удаляет Astra Automation.

Установка и запуск

Для установки утилиты `aa-setup` выполните следующие действия:

1. В каталоге `/etc/apt/sources.list.d/` создайте файл `astra-automation.list` со ссылкой на репозиторий Astra Automation:

   ```
   deb https://dl.astralinux.ru/aa/aa-debs-for-alse-1.7 <version> main
   ```

   Вместо <version> необходимо подставить версию устанавливаемой платформы, например, `1.1-upd1`.

   Доступные версии продукта опубликованы в таблице *История обновлений*.

2. Обновите список доступных пакетов:

   ```
   sudo apt update
   ```

3. Выполните команду установки:

   ```
   sudo apt install astra-automation-setup --yes
   ```

Описание аргументов

Запустить утилиту можно в каталоге `/opt/rbta/aa/astra-automation-setup/` со следующими аргументами:

- **`-h`, `--help`**

  Выводит информацию о доступных опциях и соответствующих аргументах.

- **`-i`** `<inventory_path>`, **`--inventory`** `<inventory_path>`

  Путь к файлу инвентаризации. Значение по умолчанию – `./inventory`.

- **`-p`** `<log_path>`, **`--log-path`** `<log_path>`

  Путь к альтернативному каталогу для сохранения журнала установки. Текущий пользователь должен иметь привилегию на создание файлов и запись в этом каталоге. Значение по умолчанию – `./`.

- **`-k`, `--generate-key`**

  Генерирует и распространяет новый секретный ключ, используемый для выполнения следующих задач:

  - шифрование данных пользовательских сессий;
  - генерация токенов для паролей;
  - защита данных контроллера.

- **-b, --backup**

  Сохраняет в виде *резервной копии* данные, необходимые для полного восстановления платформы на узлах, описанных в инвентаре установщика.

- **-r, --restore**

  Использует резервную копию данных для восстановления платформы на узлах, описанных в инвентаре установщика.

- **-d, --debug**

  Включает режим отладки для более подробного вывода информации о процессе.

- **-u, --uninstall**

  Запускает процесс удаления компонентов Astra Automation, перечисленных в файле инвентаризации.

  Added in version 1.1-upd1.

- **-y, --force-yes**

  Ключ отключает запрос на подтверждение выполнения операции, подразумевая, что ответ всегда Да (Yes).

- **--force-postgres-removal**

  Запускает процесс удаления PostgreSQL.

- **--plain**

  Отключает расширенный вариант текстового пользовательского интерфейса.

- **--repo-url** `<repo_url>`

  Указывает URL альтернативного репозитория для установки из него Astra Automation. Например, для использования репозитория `https://example.com/artifactory/automation-debs-for-alse` воспользуйтесь командой:

  ```
  sudo ./aa-setup --repo-url https://example.com/artifactory/automation-debs-for-
  ↪alse
  ```

- **--default-job-ee** `<ee_url>`

  Указывает на образ контейнейнера, который будет использован в качестве *среды исполнения* по умолчанию (Default execution environment). Например, для использования образа `registry.astralinux.ru/aa/aa-base-ee:0.5.1` воспользуйтесь командой:

  ```
  sudo ./aa-setup --default-job-ee registry.astralinux.ru/aa/aa-base-ee:0.5.1
  ```

- **--component** `<component_name>`

  Указывает альтернативный компонент репозитория. Например, если необходимо использовать пакеты, размещенные в компоненте `testing`, воспользуйтесь командой:

  ```
  sudo ./aa-setup --component testing
  ```

- **--product-version** `<product_version>`

  Указывает определенную версию Astra Automation. По умолчанию устанавливается новейшая версия. Для установки версии <version> (например, `1.1-upd1`) воспользуйтесь командой:

  ```
  sudo ./aa-setup --product-version <version>
  ```

- **-v, --version**

  Показывает версию утилиты `aa-setup`.

- **--upgrade**

  Обновляет компоненты Astra Automation до новейшей версии. Может быть использован совместно с аргументом `--product-version` для обновления до определенной версии.

Для обновления до версии <version> (например, 1.1-upd1) используйте команду:

```
sudo ./aa-setup --upgrade --product-version <version>
```

- **--check-releases**

  Выводит на экран терминала версии Astra Automation, доступные в репозитории.

Также вы можете передать дополнительные аргументы для команды ansible-playbook после --, например:

```
sudo ./aa-setup -b -- -vvv --ask-pass
```

4.10.2 Параметры инвентаря

Настройки развертывания платформы Astra Automation задаются в конфигурационном файле inventory, доступного в следующих каталогах:

- /opt/rbta/aa/astra-automation-setup/ – при использовании интернет-репозиториев ПАО Группа Астра;
- в корневом каталоге распакованного архива – при использовании оффлайн-установщика.

Файл inventory это обычный инвентарь в формате INI, а значит, при описании настроек узлов вы можете использовать все возможности, предоставляемые Ansible при работе с файлами этого типа.

При указании имен файлов рекомендуется использовать полные пути. Относительные пути рассчитываются по отношению к каталогу, в котором находится файл inventory.

Общие принципы заполнения инвентаря

Настройки компонентов платформы можно задавать на следующих уровнях:

- Для всех узлов.

  Настройки из секции [all:vars] называются глобальными, потому что они действуют на все узлы, указанные в инвентаре.

- Для группы узлов.

  Чтобы задать настройки для всех узлов определенной группы, создайте соответствующую секцию. Название секции формируется по следующему принципу:

  ```
  [<group>:vars]
  ```

  Здесь <group> – название группы узлов.

  Например, чтобы задать настройки для узлов из группы main создайте секцию [main:vars]:

  ```
  [main]
  node1.example.com   ansible_host=192.168.56.1
  node2.example.com   ansible_host=192.168.56.2
  node3.example.com   ansible_host=192.168.56.3

  [main:vars]
  ansible_user=administrator
  ansible_ssh_private_key=/path/to/ssh/private/key
  ```

  Эта запись эквивалентна следующей:

```
[main]
node1.example.com   ansible_host=192.168.56.1   ansible_user=administrator
↪ansible_ssh_private_key=/path/to/ssh/private/key
node2.example.com   ansible_host=192.168.56.2   ansible_user=administrator
↪ansible_ssh_private_key=/path/to/ssh/private/key
node3.example.com   ansible_host=192.168.56.3   ansible_user=administrator
↪ansible_ssh_private_key=/path/to/ssh/private/key
```

- Для отдельных узлов в соответствующих строках.

При определении значений настроек используется следующий порядок:

1. глобальные настройки;
2. настройки группы;
3. настройки отдельных узлов.

Более поздние значения перезаписывают более ранние.

Настройки подключения

Тип подключения к узлам платформы задается в настройке `ansible_connection`:

- `local` – локальное подключение к узлу. Используется только в том случае, когда все компоненты платформы развертываются на одном узле.
- `ssh` – подключение с использованием протокола SSH (по умолчанию).
- `paramiko` – подключение с использованием протокола Paramiko.

Основные настройки подключения

Для всех типов подключения возможно использование следующих дополнительных настроек:

- `ansible_host` – IP-адрес управляемого узла.
- `ansible_port` – номер порта, через который следует выполнять подключение к управляемому узлу. При использовании протокола SSH значение по умолчанию – 22.
- `ansible_user` – название учетной записи, от имени которой следует выполнять подключение к управляемому узлу.
- `ansible_password` – пароль учетной записи.

Настройки протокола SSH

Для протокола SSH доступны следующие настройки:

- `ansible_scp_extra_args` – дополнительные аргументы вызова утилиты `scp`.
- `ansible_sftp_extra_args` – дополнительные аргументы вызова утилиты `sftp`.
- `ansible_ssh_common_args` – дополнительные аргументы вызова утилит `ssh`, `scp` и `sftp`.
- `ansible_ssh_executable` – путь к исполняемому файлу утилиты `ssh`, который следует использовать вместо системного.
- `ansible_ssh_extra_args` – дополнительные аргументы вызова утилиты `ssh`.
- `ansible_ssh_pipelining` – сокращение количества операций подключения к управляемым узлам за счет выполнения некоторых модулей без копирования файлов:
  - `true` – включено;
  - `false` – выключено (по умолчанию).

- `ansible_ssh_private_key_file` – путь к файлу приватного ключа SSH.

Повышение привилегий

Для выполнения некоторых операций требуются повышенные привилегии. Способы их получения задаются в следующих настройках:

- `ansible_become` – принудительное использование повышения привилегий;
- `ansible_become_exe` – путь к исполняемому файлу, используемому для повышения привилегий;
- `ansible_become_flags` – дополнительные флаги, используемые для выбранного метода повышения привилегий;
- `ansible_become_method` – метод повышения привилегий;
- `ansible_become_user` – название учетной записи, используемой для повышения привилегий;
- `ansible_become_password` – пароль учетной записи, используемой для повышения привилегий.

Настройки окружения удаленных узлов

Параметры выполнения сценариев Ansible на управляемых узлах задаются в следующих настройках:

- `ansible_shell_type` – тип оболочки на управляемом узле:
  - `csh`;
  - `fish`;
  - `sh` (по умолчанию).

  Важно: Значение этой настройки нужно изменять только если значением настройки `ansible_shell_executable` выбрана оболочка, не совместимая с `sh`.

- `ansible_shell_executable` – путь к командному интерпретатору.

  Важно: Значение этой настройки нужно изменять только в том случае, когда использование `/bin/sh` по какой-либо причине невозможно.

  Значение по умолчанию – `/bin/sh`.

Группы узлов

Настройки узлов для отдельных компонентов платформы задаются в соответствующих секциях:

- Automation Controller:
  - `[automationcontroller]` – узлы плоскости управления;
  - `[execution_nodes]` – исполняющие узлы;
  - `[instance_group_<name>]` – группа исполняющих узлов;
  - дополнительные группы для промежуточных и исполняющих узлов.
- Private Automation Hub – `[automationhub]`.

- СУБД – `[database]`.

  На указанном узле средствами платформы будет развернута СУБД PostgreSQL, а в ней создана база данных для Automation Controller.

```
[database]
rdbs.example.com   ansible_host=192.168.56.101
```

Важно: Узел, используемый для развертывания PostgreSQL, не должен использоваться для других целей, например, запуска заданий или управления контроллером.

Если для хранения баз данных используется сторонний сервис, секцию `[database]` нужно создать, но оставить пустой.

Настройки Automation Controller

Настройки узлов Automation Controller задаются в соответствующих секциях, а учетные данные администратора, настройки подключения к СУБД и настройки SSO – в секции `[all:vars]`.

Настройки узлов

Для узлов Automation Controller доступны дополнительные настройки – тип узла (`node_type`) и связанная группа узлов (`peers`).

- `node_type` – *тип узла*.
  - `control` – управляющий;
  - `execution` – исполняющий;
  - `hop` – промежуточный;
  - `hybrid` – гибридный.

  Для узлов из секции `[automationcontroller]` можно использовать только типы `control` и `hybrid` (по умолчанию).

  Для узлов из секции `[execution_nodes]` можно использовать только типы `hop` и `execution` (по умолчанию).
- `peers` – название связанной группы узлов.

  С помощью `peers` настраивается связь между плоскостью управления и группами исполняющих узлов.

  Пример использования настройки `peers` см. в описании развертывания *кластера Automation Controller с удаленным исполнением*.

Учетные данные системного администратора

Учетные данные системного администратора Automation Controller задаются в следующих глобальных настройках:

- `admin_email` – адрес электронной почты;
- `admin_password` – пароль;
- `admin_username` – название учетной записи.

Сеть

Параметры сети Automation Controller задаются в следующих глобальных настройках:

- `mesh_ca_certfile` – путь к файлу сертификата сети Mesh.
- `mesh_ca_keyfile` – путь к файлу ключа сертификата сети Mesh.
- `receptor_listener_port` – номер порта, используемый рецептором.

  Значение по умолчанию – 27199.
- `web_server_ssl_cert` – путь к файлу сертификата веб-сервера.
- `web_server_ssl_key` – путь к файлу ключа сертификата веб-сервера.

СУБД

Параметры подключения Automation Controller к СУБД задаются в следующих глобальных настройках:

- `pg_database` – название базы данных.
- `pg_host` – IP-адрес или FQDN узла с СУБД.

  Если указана пустая строка (по умолчанию), используется локальное подключение к СУБД, развернутой на том же узле, что и Automation Controller.
- `pg_password` – пароль пользователя СУБД.
- `pg_port` – порт СУБД.

  Значение по умолчанию – 5432.
- `pg_sslmode` – режим использования SSL при подключении к СУБД:
  - `prefer` – подключение к СУБД будет защищено с помощью SSL, если использование шифрования поддерживается настройками сервера (по умолчанию);
  - `verify-full` – защита подключения с помощью SSL необходима, производится строгая верификация сервера.
- `pg_username` – название учетной записи пользователя СУБД.
- `postgres_ssl_cert` – путь к файлу сертификата, который используется для защиты подключения к серверу СУБД PostgreSQL, развернутому средствами платформы.
- `postgres_ssl_key` – путь к файлу ключа сертификата, который используется для защиты подключения к серверу СУБД PostgreSQL, развернутому средствами платформы.
- `postgres_use_ssl` – использование SSL для защиты подключения к СУБД:
  - `true` – включено;
  - `false` – выключено (по умолчанию).

SSO

Параметры SSO для Automation Controller задаются в следующих глобальных настройках:

- `attr_email` – название атрибута, в котором хранится адрес электронной почты пользователя.
- `attr_first_name` – название атрибута, в котором хранится фамилия пользователя.
- `attr_groups` – название атрибута, в котором хранится список групп пользователя.
- `attr_last_name` – название атрибута, в котором хранится имя пользователя.
- `attr_user_permanent_id` – название атрибута, в котором хранится постоянный идентификатор пользователя.

- attr_username – название атрибута, в котором хранится название учетной записи пользователя.
- entity_id
- private_key_path – путь к файлу приватного ключа сертификата, используемого для защиты подключения к поставщику услуг SSO.
- provider_id – идентификатор поставщика услуг SSO.
- provider_url – URL поставщика услуг SSO.
- pub_cert_path – путь к файлу сертификата, используемого для защиты подключения к поставщику услуг SSO.
- sso_displayname – название поставщика услуг SSO, отображаемое в Automation Controller.
- sso_setup – использование SSO:
  - true – включено;
  - false – выключено (по умолчанию).
- sso_url – URL поставщика услуг SSO.
- support_contact_email – адрес электронной почты службы технической поддержки поставщика услуг SSO.
- support_contact_name – название службы поддержки поставщика услуг SSO.
- tech_contact_email – адрес электронной почты службы поддержки пользователей поставщика услуг SSO.
- tech_contact_name – название службы поддержки пользователей поставщика услуг SSO.

Настройки Private Automation Hub

Настройки Private Automation Hub задаются в секциях [automationhub], [automationhub:vars] и [all:vars].

Настройки узлов

В секции [automationhub] задаются настройки подключения к узлам Private Automation Hub, например:

```
[automationhub]
hub1.example.com   ansible_host=192.168.56.101
hub2.example.com   ansible_host=192.168.56.102
```

Учетные данные администратора

Учетные данные администратора Private Automation Hub задаются в следующих глобальных настройках:

- automationhub_admin_password – пароль;
- automationhub_admin_user – название учетной записи.

Сеть

- `automationhub_main_url` – URL Private Automation Hub.

  Указанное здесь значение используется при формировании шаблонов настроек, позволяющих утилитам Ansible загружать необходимое содержимое из Private Automation Hub.
- `automationhub_ssl_cert` – путь к файлу сертификата TLS, используемого веб-сервером.
- `automationhub_ssl_key` – путь к файлу ключа сертификата TLS, используемого веб-сервером.

СУБД

Параметры подключения Private Automation Hub к СУБД задаются в следующих глобальных настройках:

- `automationhub_pg_database` – название базы данных.
- `automationhub_pg_host` – IP-адрес или FQDN узла с СУБД.
- `automationhub_pg_password` – пароль пользователя СУБД.
- `automationhub_pg_port` – порт СУБД.

  Значение по умолчанию – 5432.
- `automationhub_pg_username` – название учетной записи пользователя СУБД.

SSO

Параметры SSO для Private Automation Hub задаются в следующих глобальных настройках:

- `sso_authomation_hub_authorization_url` – URL страницы входа, на которую перенаправляется пользователь при авторизации через SSO.
- `sso_automation_hub_client_id` – идентификатор клиента, который будет использоваться для авторизации приложения при взаимодействии с сервером SSO.
- `sso_authomation_hub_token_url` – URL страницы, используемой Private Automation Hub для получения токена от поставщика услуг SSO.
- `sso_automation_platform_realm` – область в системе SSO.
- `sso_client_secret` – секретный ключ клиента, который используется для подтверждения подлинности при запросах к серверу SSO.
- `sso_host` – адрес сервера поставщика услуг SSO.
- `sso_port` – порт сервера поставщика услуг SSO.
- `sso_protocol` – протокол, который будет использоваться для соединения с сервером SSO.
- `sso_realm_public_key` – публичный ключ области, используемый для проверки цифровых подписей токенов и других данных авторизации.

Настройки хранилища S3

Параметры подключения к S3-совместимому хранилищу задаются в следующих глобальных настройках:

- `s3_access_key` – ключ доступа к бакету.
- `s3_addressing_style` – стиль адресации объектов в бакете:
  - `auto` – если возможно, используется стиль `virtual`, в противном случае – `path`;
  - `path` – название бакета включено в путь URI (по умолчанию);
  - `virtual` – название бакета включено в FQDN узла.
- `s3_bucket_name` – название бакета.
- `s3_default_acl` – список контроля доступа (access control list, ACL), используемый при работе с хранилищем по умолчанию.

  Список возможных значений зависит от типа используемого хранилища. Рекомендуется не задавать значение этого параметра в инвентаре, а выполнить необходимые настройки на стороне хранилища.

  Значение по умолчанию – `omit`.
- `s3_endpoint_url` – точка доступа к хранилищу.
- `s3_region_name` – название региона, в котором размещен бакет.

  Список возможных значений зависит от типа используемого хранилища.
- `s3_secret_key` – секретный ключ для доступа к хранилищу.
- `s3_signature_version` – версия подписи, используемая для работы с хранилищем:
  - s3v4 (по умолчанию);
  - s3.

  Примечание: Это значение считается устаревшим и не рекомендуется к использованию.
- `s3_use_sigv4` – подписывание запросов к хранилищу с помощью AWS Signature V4[12]:
  - `true` – включено (по умолчанию);
  - `false` – выключено.

Настройки развертывания без доступа к интернету

Параметры развертывания платформы без доступа к интернету задаются в следующих глобальных настройках:

- `bundle_dir` – путь к каталогу с файлами установщика платформы.

  Значение по умолчанию – `/opt/rbta/aa/astra-automation-setup/`.
- `bundle_repo_folder` – путь к каталогу, в который будут скопированы DEB-пакеты, необходимые для развертывания платформы. Указанный каталог будет использоваться при настройке репозиториев APT.

  Значение по умолчанию – `/var/lib/astra-automation-setup/bundle/`.

  Added in version 1.1-upd1.
- `bundle_install` – развертывание платформы в окружении без доступа к интернету:
  - `true` – включено;

[12] https://docs.aws.amazon.com/AmazonS3/latest/API/sig-v4-authenticating-requests.html

  - `false` – выключено, используются интернет-репозитории ПАО Группа Астра (по умолчанию).

- `registry_ip` – IP-адрес узла, на котором должен быть развернут реестр образов среды исполнения.

  Изменено в версии 1.1-upd1: Ранее этот параметр назывался `docker_registry_ip`.

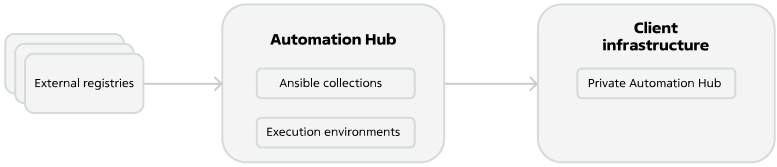
- `registry_port` – порт, который должна использовать служба реестра образов среды исполнения.

  Изменено в версии 1.1-upd1: Ранее этот параметр назывался `docker_registry_port`.

ГЛАВА 5

Automation Hub

Реестр Automation Hub предоставляет коллекции Ansible и образы среды исполнения, разработанные ПАО Группа Астра и ее партнерами. Содержимое этого реестра рекомендуется загрузить на *Private Automation Hub* и использовать последний в своих проектах.



Хранящиеся в Automation Hub образы среды исполнения созданы на базе ОС Astra Linux Special Edition.

Примечание: Примеры кода с использованием контента Automation Hub доступны на сайте https://source.astragroup.ru/.

5.1 Коллекции Ansible

Использование коллекций позволяет значительно упростить создание playbook для решения самых разных задач по развертыванию и настройке ПО с помощью Ansible.

5.1.1 Назначение

Коллекции Ansible, доступные в реестре Automation Hub[13], предназначены для развертывания и настройки продуктов ПАО Группа Астра и вспомогательного стороннего ПО.

Совет: Рекомендуется загрузить необходимые коллекции из Automation Hub в Private Automation Hub.

От коллекций, имеющихся в свободном доступе, они отличаются следующим:

- Все роли распространяются в виде коллекций Ansible, в которые включаются также модули, специфичные для продуктов ПАО Группа Астра.

[13] https://hub.astra-automation.ru/

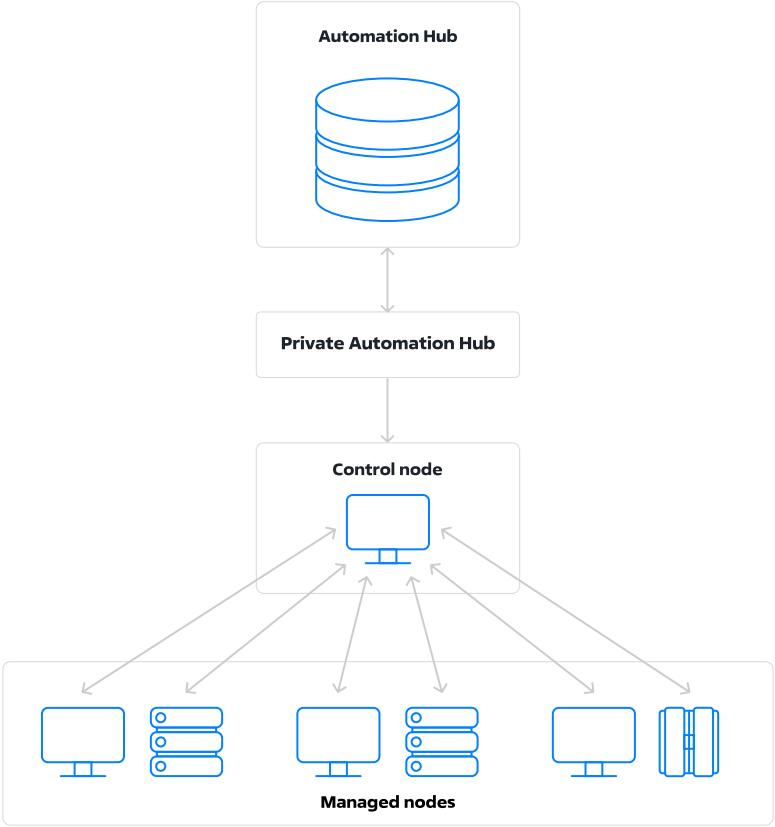
- При разработке коллекций учитываются особенности продуктов ПАО Группа Астра.
- Длительный срок поддержки.
- Доработка функциональности исходя из потребностей клиентов ПАО Группа Астра.
- Оперативное исправление всех найденных ошибок.
- Многократная проверка кода, в том числе в реальных условиях.
- Единая логика и стандартизированные подходы при написании кода, повышающие качество продукта.
- Названия всех коллекций используют пространство имен `astra`.
- Версии коллекций формируются с использованием семантического версионирования[14] (semantic versioning).

Роли, размещенные в Automation Hub, обладают следующими свойствами:

- Распространение только в составе коллекций.
- Каждая роль используется для установки и настройки только определенной функциональности. Например, роль `astra.ald_pro.client` используется только для настройки клиентов домена ALD Pro. Для настройки контроллера или его реплики следует использовать роли `astra.ald_pro.controller` и `astra.ald_pro.replica` соответственно.

5.1.2 Структура рабочего окружения

При использовании коллекций из реестра Automation Hub схема работы Ansible имеет вид:



Перед выполнением playbook следует установить на управляющий узел все необходимые коллекции. Инструкции по установке коллекций приведены ниже в секции *Применение*.

[14] https://semver.org/

5.1.3 Настройки Ansible

Во время работы Ansible использует настройки, переданные в аргументах командной строки или указанные в конфигурационном файле `ansible.cfg`. Поиск файла настроек `ansible.cfg` выполняется в следующем порядке:

1. Каталог, указанный в значении переменной окружения `ANSIBLE_CONFIG`.
2. Текущий каталог.
3. Домашний каталог активного пользователя.
4. Каталог `/etc/ansible/`.

Поиск прекращается как только файл `ansible.cfg` будет найден в любом из указанных расположений.

> **Предупреждение:** При использовании для работы с Ansible *среды исполнения* файл `ansible.cfg` следует размещать в каталоге проекта.

Подробности о файле настроек см. в документации Ansible[15].

5.1.4 Применение

Использование коллекций Ansible из реестра Automation Hub имеет следующие особенности:

- Необходимо настроить доступ к реестру Automation Hub согласно *инструкции*.
- Для описания зависимостей создайте в каталоге проекта файл `requirements.yml` с записями следующего вида:

  ```
  ---
  collections:
    - name: astra.<collection_name>
    # ...
  ```

  где <collection_name> – название коллекции. Например, для коллекции `astra.ceph` указанная запись должна иметь вид:

  ```
  ---
  collections:
    - name: astra.ceph
  ```

- При использовании роли в playbook указывайте *FQCN*, например:

  ```
  ---
  - name: Set up ALD Pro domain controller
    hosts: dc01
    # ...
    roles:
      - role: astra.ald_pro.controller
        vars:
          aldpro_domain: aldpro.example.com
          aldpro_pdc_ip: 192.168.56.11
          aldpro_pdc_name: dc01
          aldpro_admin_password: p@ssW0rD!
  ```

- Для установки необходимых ресурсов согласно зависимостям, определенным в файле `requirements.yml`, используйте команду:

[15] https://docs.ansible.com/ansible/latest/reference_appendices/config.html

```
ansible-navigator exec \
   --eei registry.astralinux.ru/aa/aa-base-ee \
   --eev ~/.ssh/hub.astra-automation.ru:/root/.ssh/id_rsa \
   -- ansible-galaxy install -r requirements.yml
```

- Для запуска playbook используйте команду:

```
ansible-navigator run playbook.yml --eei registry.astralinux.ru/aa/aa-base-ee -m␣
↪stdout
```

Для выполнения команд `ansible-navigator` использует *среду исполнения*, заданную аргументом `--eei`.

5.2 Версионирование коллекций

При описании зависимостей Ansible рекомендуется всегда явно указывать номер версии используемой коллекции. Это позволит избежать незапланированных изменений в существующем окружении и несовместимости новой версии коллекции с ним.

Коллекции версионируются согласно правилу семантического версионирования[16]. Версия коллекции выглядит как:

```
<major_version>.<minor_version>.<patch>[-rc.<rc_version>]
```

где:

- `<major_version>` – номер мажорной версии. Версия `0` используется для начальной разработки. Номер мажорной версии увеличивается, если в код коллекции добавляются изменения API.
- `<minor_version>` – номер минорной версии. Увеличивается при добавлении новой функциональности или существенных изменениях в уже существующей функциональности без нарушения совместимости с более ранним API. Минорная версия обнуляется при увеличении мажорной версии.
- `<patch>` – номер срочного обновления при исправлении ошибок. Увеличивается после исправления ошибок в имеющейся функциональность без потери обратной совместимости. Новая функциональность не вносится, существенные изменения в уже существующей не производятся. Это поле обнуляется при увеличении мажорной и минорной версий.
- (опционально) `-rc.<rc_version>` – версия Release Candidate (RC) указывает на то, что это еще не финальная версия, но функциональность почти готова к выпуску. Номер версии RC увеличивается по мере подготовки коллекции к выпуску.

5.2.1 Получение списка версий коллекции

Для получения списка доступных версий коллекции выполните следующие действия:

1. Перейдите на страницу коллекции в соответствующем разделе Automation Hub[17].
2. Нажмите на выпадающий список **Версия** (Version).

[16] https://semver.org/
[17] https://hub.astra-automation.ru/

5.2.2 Указание версии коллекции в файле зависимостей

Чтобы зафиксировать номер версии используемой коллекции, добавьте к соответствующей записи в *файле зависимостей* строку с параметром `version`, например:

```
---
collections:
  - name: astra.ald_pro
    version: "<version>"
```

Если это поле отсутствует, будет загружена самая свежая версия коллекции из доступных. В значении поля `version` можно задать условия, определяющие допустимый диапазон номеров устанавливаемой версии коллекции:

- `*` – самая свежая (latest) версия. Это значение используется по умолчанию.
- `!=` – использовать любую версию, кроме указанной.
- `==` – использовать строго указанную версию.
- `>=` – использовать указанную версию или более новую.
- `>` – использовать более новую версию, чем указанная.
- `<=` – использовать указанную версию или более старую.
- `<` – использовать версию более старую, чем указанная.

Смотрите более подробную информацию о версионировании коллекций в документации Ansible[18].

5.2.3 Примеры

- Строгое указание версии

  ```
  version: "==0.1.1"
  ```

- Указанная или более новая версия

  ```
  version: ">=0.1"
  ```

5.3 Сертифицированные и проверенные коллекции

5.3.1 Форматы распространения кода Ansible

Код для автоматизации с помощью Ansible может включать следующие элементы:

- сценарии;
- определения переменных;
- модули и другие.

Код может иметь произвольную структуру каталогов в проекте. Для обеспечения модульности и переиспользования Astra Automation предлагает контент в виде коллекций Ansible, которые объединяют элементы кода в стандартную структуру. Смотрите более подробную информацию в *описании коллекций Ansible*.

Все коллекции Astra Automation доступны в Automation Hub[19] и являются *проверенными*.

[18] https://docs.ansible.com/ansible/latest/collections_guide/collections_installing.html#installing-an-older-version-of-a-collection
[19] https://hub.astra-automation.ru/

5.3.2 API

Коллекции Ansible используются посредством подключения их к другому коду Ansible (например, к playbook или другим коллекциям) в качестве зависимостей и «библиотек» – подобно тому, как аналогичный подход применяется в высокоуровневых языках программирования. У пользователя коллекции нет необходимости редактировать ее код, чтобы использовать необходимую функциональность в своем окружении.

Пользователь взаимодействует с коллекцией с помощью следующих элементов интерфейса:

- переменные ролей и playbook, объявленные в коллекции;
- инвентарь в формате, определенном коллекцией;
- параметры модулей, содержащихся в коллекции;
- названия ролей, модулей и playbook.

Совокупность перечисленных интерфейсов можно воспринимать как API коллекции, который для каждой коллекции описан в ее документации. API определяется функциональными возможностями коллекции, но также на него может влиять состав *Execution Environment*.

5.3.3 Проверенный контент

В условиях динамического развития продуктов ПАО Группа Астра, а также продуктов других компаний, потенциальные пользователи Astra Automation хотели бы получать код автоматизации на тестирование как можно раньше, даже если он не полностью охватывает все запланированные возможности и не стабилизирован. К такому коду со стороны Astra Automation предъявляется ограниченный набор требований:

- Код организован в соответствии со структурой коллекции Ansible.
- Коллекция версионируется в соответствии с SemVer (Semantic Versioning).
- Код не требует интерактивного взаимодействия с пользователем.
- Коллекция содержит документацию на все компоненты (роли, модули, общий `README.md`).

После прохождения проверки инженерами Astra Automation коллекция, отвечающая данным требованиям, получает статус проверенной.

Проверенный контент имеет следующие ограничения:

- Отсутствует фиксированный жизненный цикл и выпуски с длительным сроком поддержки.
- Код может быть неидемпотентным[20].

5.3.4 Сертифицированный контент

Коллекции Ansible могут быть сертифицированы в рамках определенной версии Astra Automation. К сертифицированным коллекциям предъявляются следующие требования (помимо требований, предъявляемых к проверенным коллекциям):

- Код не должен дублироваться. Если коллекция collection1 разработана для настройки сервиса service1, которому для функционирования необходим сервис service2, код настройки которого уже есть в составе отдельной сертифицированной коллекции collection2, то запрещается дублировать его в коллекцию collection1.

[20] https://en.wikipedia.org/wiki/Idempotence

- Коллекция настраивает только тот объект (сервис, устройство), для которого разрабатывается. Коллекция не должна содержать реализации функций настройки каких-либо других объектов – такие функции должны быть добавлены в соответствующие сертифицированные коллекции.
- Код коллекции должен быть идемпотентным.
- Возможности коллекции должны быть покрыты тестами.

5.3.5 Жизненный цикл

Для сертифицированных коллекций поддерживаются следующие типы выпусков:

- **FR (feature release)** – выпуски с новыми возможностями коллекции. Периодичность выпуска составляет 4 недели, если есть готовые к публикации изменения.
- **LTS (long-term support)** – выпуски с длительным сроком поддержки в течение 24 месяцев. В таких выпусках зафиксированы функциональность коллекции и версии поддерживаемых продуктов. Выпуски получают только обновления безопасности и устранение ошибок, а также совместимость с новой версией Automation Controller, если подходит к концу срок поддержки текущей версии контроллера.

Порядок выпуска FR

В рамках выпуска FR в сертифицированных коллекциях могут происходить следующие изменения, отражающиеся в соответствующем изменении версии коллекции согласно семантическому версионированию:

- **Исправления** – исправления обнаруженных ошибок, в том числе исправление и дополнение документации.
- **Минорные обновления** – добавление новых функций или новой версии коллекции, не затрагивающие обратную совместимость с текущей мажорной версией API. Это означает, что playbook и другие коллекции, использующие данную коллекцию, не потребуют адаптации для уже задействованных в них возможностей коллекции.
- **Мажорные обновления** – значительные изменения в коллекции, которые меняют формат входных/выходных параметров ролей, playbook и других компонентов коллекции (API). Для таких версий теряется обратная совместимость с предыдущими мажорными версиями коллекции. Изменения могут быть связаны с добавлением поддержки новой версии продукта, исправлениями критических ошибок или удалением компонентов коллекции.

При внесении изменений в служебные файлы (`taskfile.yml`, конфигурации linter и другие) и тесты версия коллекции не меняется.

Версия FR выпускается каждые 4 недели или чаще в случае обнаружения значительных ошибок. Если изменений в коллекции за цикл выпуска не произошло, новая версия не выпускается.

В рамках выпусков FR может добавляться поддержка новых версий продуктов. Для сертифицированных коллекций это должно происходить при очередном выпуске FR после даты релиза продукта или ранее.

Порядок выпуска LTS

Использование долговременно-поддерживаемых версий (LTS, long-term support) позволяет сопровождать инфраструктуру с помощью коллекции с зафиксированным API в течение длительного времени. Новый выпуск LTS формируется каждые 18 месяцев на основе текущего выпуска FR, что позволяет плавно мигрировать с предыдущего LTS-выпуска.

Выпуски LTS поддерживаются в течение 24 месяцев со дня выпуска. В течение этого периода коллекция получает следующие обновления:

- исправления обнаруженных ошибок;
- исправления безопасности;
- новые функции, если это требуется для исправлений.

Мажорная и минорные значения версии коллекции всегда зафиксированы для выпуска LTS. В версии коллекции изменяется только поле patch.

5.3.6 Поддержка

Для сертифицированных и проверенных коллекций клиент может направлять запросы о нововведениях или исправлениях ошибок в поддержку Astra Automation. Запросы будут отрабатываться инженерами Astra Automation и партнерами, сопровождающими сертифицированную коллекцию.

5.3.7 Удаление коллекций

Коллекции могут терять свой статус сертифицированной или проверенной, а также быть удалены с Automation Hub[21].

Коллекция теряет статус сертифицированной в следующих случаях:

- Если сопровождающий коллекцию не обеспечивает устранение ошибок для выпусков LTS.
- Если сопровождающий коллекцию не обеспечил совместимость с новой версией Astra Automation.
- Если закончилась поддержка версии Astra Automation, для которой сертифицировалась коллекция.

Коллекция теряет статус проверенной, когда версия Automation Controller, для которой она предназначена, перестает поддерживаться.

5.4 Настройка доступа к Automation Hub

Для получения содержимого из Automation Hub необходим токен. Чтобы создать и использовать его, выполните следующие действия:

1. Авторизуйтесь на портале Automation Hub[22].
2. На панели навигации выберите *Коллекции* ▸ *Токен API* (*Collections* ▸ *API Token*).
3. Нажмите кнопку *Загрузить токен* (Download token).
4. Используйте показанный на странице токен для выполнения следующих операций:
   - *синхронизация коллекций* в Private Automation Hub с Automation Hub;
   - *создание образа среды исполнения*, содержащего необходимые коллекции.

[21] https://hub.astra-automation.ru/
[22] https://hub.astra-automation.ru/

Важно: Период действия токена – 24 часа. По истечении указанного времени его нужно будет создать заново.

При создании нового токена предыдущий перестает действовать немедленно.

5.5 Справочные данные

В реестре Automation Hub[23] доступны следующие коллекции Ansible:

Коллекция	Название ПО	Комментарии
`astra.aa-controller`	Astra Automation	Развертывание и настройка компонентов платформы Astra Automation
`astra.ald_pro`	ALD Pro[24]	Домен, управляемый заданным количеством контроллеров, на базе ALD Pro
`astra.astralinux`	Astra Linux Special Edition[25]	Автоматизация базовых операций по управлению ОС Astra Linux Special Edition
`astra.brest`	ПК СВ «Брест»[26]	Программный комплекс средств виртуализации
`astra.ceph`	Ceph[27]	Система хранения данных Ceph с вариациями: • распределенная; • на одном сервере.
`astra.chrony`	Chrony[28]	Сервер времени
`astra.cups`	CUPS[29]	Служба печати
`astra.dcimanager`	DCImanager[30]	Платформа для управления физической и информационной инфраструктурой
`astra.dhcp`	ISC DHCP[31]	Служба динамического назначения IP-адресов и сетевых имен
`astra.docker`	Docker[32]	Система контейнеризации
`astra.etcd`	etcd[33]	Распределенное хранилище данных в формате «ключ-значение»
`astra.freeipa`	FreeIPA[34]	Система идентификации пользователей на базе FreeIPA в составе: • сервер; • реплика; • клиент.
`astra.grafana`	Grafana[35]	Система мониторинга
`astra.haproxy`	HAProxy[36]	Балансировщик нагрузки
`astra.hardening`	Astra Linux Special Edition	Автоматизация настройки систем безопасности Astra Linux Special Edition

continues on next page

[23] https://hub.astra-automation.ru/

Таблица 1 – продолжение с предыдущей страницы

Коллекция	Название ПО	Комментарии
`astra.iscsi`	Open-iSCSI[37]	Open-iSCSI в составе: • сервер-инициатор; • сервер-цель.
`astra.keepalived`	Keepalived[38]	Балансировщик нагрузки
`astra.keycloak`	Keycloak[39]	Поставщик услуг аутентификации (SSO)
`astra.memcached`	Memcached[40]	Система кеширования
`astra.nfs`	NFS[41]	Настройка клиента и сервера NFS
`astra.nginx`	NGINX[42]	Web-сервер
`astra.ocfs2`	OCFS2[43]	Кластерная файловая система *OCFS2* на заданном количестве узлов. Требует предварительной установки с помощью коллекций `astra.ceph` или `astra.iscsi`.
`astra.patroni`	Patroni[44]	Сервис кластеризации СУБД PostgreSQL
`astra.pgbouncer`	PgBouncer[45]	Менеджер подключений к кластеру СУБД PostgreSQL
`astra.postgresql`	PostgreSQL[46]	Кластер СУБД PostgreSQL
`astra.prometheus`	Prometheus[47]	Система мониторинга
`astra.rabbitmq`	RabbitMQ[48]	Брокер сообщений
`astra.repo_mirror`	Astra Linux Special Edition	Создание зеркал DEB-репозиториев и управление ими
`astra.rubackup`	RuBackup[49]	Система резервного копирования
`astra.rupost`	RuPost[50]	Почтовая система на базе RuPost. Коллекция зависит от коллекции PostgreSQL. Для полноценной работы требует настроенного домена LDAP.
`astra.termidesk`	Termidesk[51]	Диспетчер подключений виртуальных рабочих мест (VDI)
`yandex.cloud`	–	Управление ресурсами в Yandex Cloud

[24] https://www.aldpro.ru/
[25] https://astralinux.ru/
[26] https://astra.ru/software-services/application-software-astra-group/brest/
[27] https://ceph.io/
[28] https://chrony-project.org/
[29] https://www.cups.org/
[30] https://www.ispsystem.ru/dcimanager
[31] https://www.isc.org/dhcp/
[32] https://www.docker.com/
[33] https://etcd.io/
[34] https://www.freeipa.org/
[35] https://grafana.com/
[36] https://www.haproxy.org/
[37] https://www.open-iscsi.com/
[38] https://keepalived.org/
[39] https://www.keycloak.org/
[40] https://memcached.org/
[41] https://ru.wikipedia.org/wiki/Network_File_System
[42] https://nginx.org/
[43] https://archive.kernel.org/oldwiki/ocfs2.wiki.kernel.org/
[44] https://patroni.readthedocs.io/en/latest/
[45] https://www.pgbouncer.org/
[46] https://www.postgresql.org/
[47] https://prometheus.io/
[48] https://www.rabbitmq.com/
[49] https://www.rubackup.ru/
[50] https://www.rupost.ru/
[51] https://termidesk.ru/

ГЛАВА 6

Private Automation Hub

Added in version 1.1-upd1.

Private Automation Hub является средством распространения образов среды исполнения и коллекций Ansible. Содержит контент, выбранный владельцем платформы:

- разработанный собственными силами;
- загруженный из Automation Hub[52], в том числе сертифицированные и проверенные коллекции Ansible и образы среды исполнения;
- загруженный из Ansible Galaxy[53] и других сторонних источников.

Примечание: В основе Private Automation Hub лежит Pulp Project[54] – программное обеспечение с открытым исходным кодом, предназначенное для получения, выгрузки и распространения пакетов ПО в собственной или облачной инфраструктуре.

6.1 Особенности архитектуры

Важно: Эта часть документации находится в стадии разработки.

6.1.1 Хранилище

По умолчанию Private Automation Hub использует для хранения содержимого локальную файловую систему. При развертывании на нескольких узлах следует использовать для хранения содержимого NFS или бакет S3-совместимого облачного хранилища.

При использовании NFS нужно будет смонтировать том вручную на каждом узле Private Automation Hub.

Чтобы использовать S3-совместимое облачное хранилище, необходимо выполнить следующие действия:

1. задать настройки доступа к бакету в *инвентаре установщика платформы*;

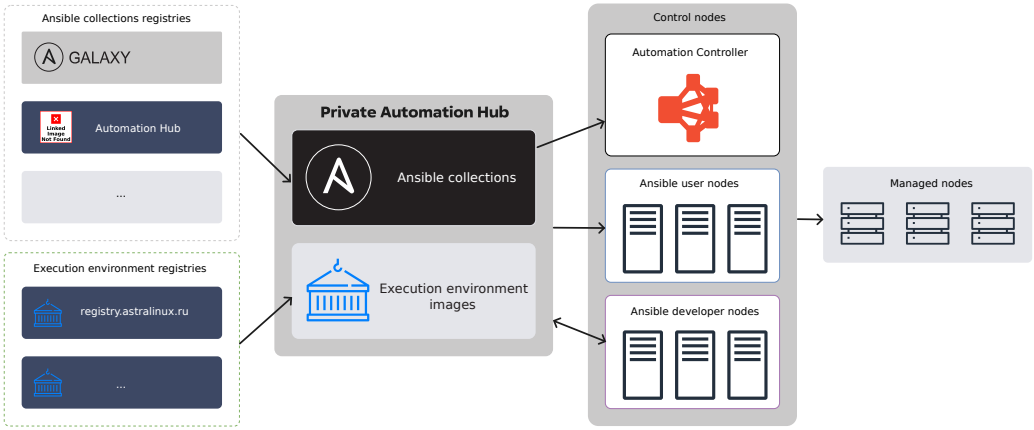
[52] https://hub.astra-automation.ru/
[53] https://galaxy.ansible.com/
[54] https://pulpproject.org/

2. запустить установщик платформы.

При изменении типа хранилища обновление ссылок в базе данных Private Automation Hub не производится. Это значит, что при условии сохранения путей к файлам смена локального хранилища на NFS или в обратную сторону не требует дополнительных действий. В то же время переход на использование бакета S3-совместимого хранилища или в обратную сторону требует удаления и повторной загрузки всего содержимого.

6.2 Структура управления

Private Automation Hub используется для распространения коллекций Ansible и образов среды исполнения. Они необходимы для выполнения сценариев автоматизации с помощью различных средств Ansible, например, *Automation Controller* или *Ansible Navigator*.



Коллекции могут быть загружены из внешних источников (remotes) и с рабочих станций разработчиков. Образы среды исполнения могут быть загружены из внешних реестров (image registries) и с рабочих станций разработчиков.

6.2.1 Внешние источники коллекций

Внешний источник коллекций – это сторонний сервер, реализующий определенный API для работы с коллекциями, например:

- Automation Hub[55];
- Ansible Galaxy[56];
- другие Private Automation Hub.

6.2.2 Репозитории

Репозиторий ограничивает доступность хранящихся в нем версий коллекций для пользователей. В зависимости от настроек репозитория, размещенные в нем версии коллекций могут быть доступны следующим категориям пользователей:

- только администраторы;
- только авторизованные пользователи;
- все пользователи.

Репозиторий может быть защищенным. *Защищенные репозитории* нельзя удалить, однако, можно изменить некоторые из их настроек.

Список защищенных репозиториев (в алфавитном порядке):

[55] https://hub.astra-automation.ru/
[56] https://galaxy.ansible.com/

- `aa-certified` – *сертифицированные* коллекции, разработанные ПАО Группа Астра и ее партнерами;
- `community` – коллекции, разработанные сообществом пользователей Ansible;
- `published` – коллекции, размещенные владельцем реестра для определенных целей;
- `rejected` – коллекции, не прошедшие успешно процедуру проверки (validation);
- `staging` – коллекции, ожидающие согласования в процессе их проверки (validation);
- `validated` – *проверенные* коллекции.

Остальные репозитории создаются пользователями Private Automation Hub и могут быть удалены или изменены по их усмотрению.

Репозиторий может быть настроен на получение коллекций из внешнего источника. В этом случае для него становится доступна *синхронизация*.

Синхронизация

Синхронизация – это процесс, в результате которого содержимое репозитория Private Automation Hub приводится в соответствие с содержимым связанного внешнего источника коллекций.

Синхронизация может выполняться в режиме **Зеркало** (Mirror). В этом случае Private Automation Hub выполняет с версиями коллекций следующие действия:

- удаляет версии, отсутствующие в источнике;
- загружает из внешнего источника версии, которых нет в репозитории;
- отмечает как устаревшие версии, отмеченные как устаревшие в источнике;
- отмечает как актуальные версии, отмеченные как актуальные в источнике.

Приватность

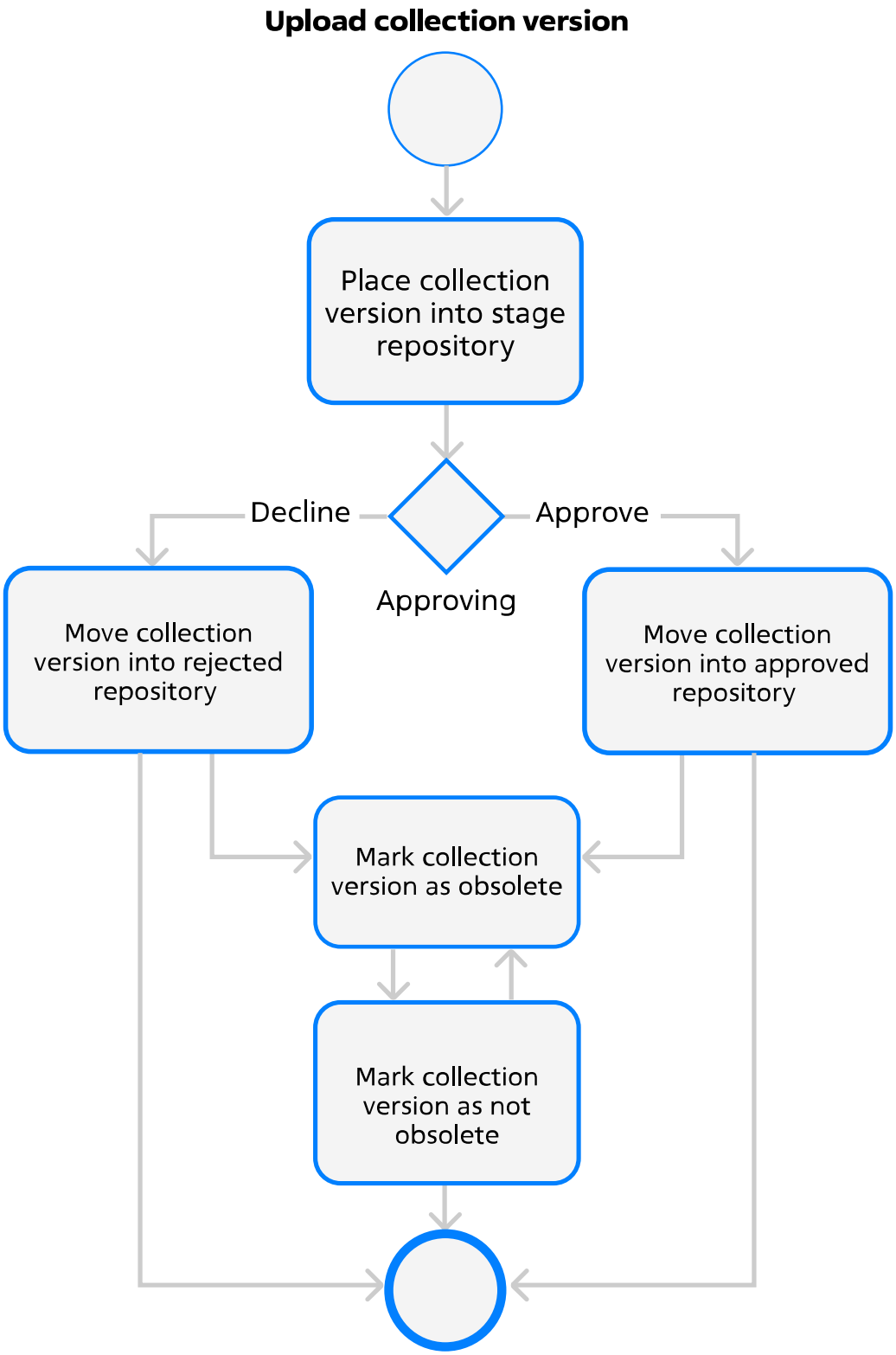
По умолчанию содержимое репозиториев доступно для использования всем пользователям Private Automation Hub. Если отметить репозиторий как приватный, доступ к хранящимся в нем коллекциям будет только у пользователей, явно указанных в настройках репозитория.

Жизненный цикл коллекции

Private Automation Hub позволяет гибко настроить жизненный цикл коллекций и их версий:

- версии коллекций могут быть доступны пользователям сразу или после *согласования*;
- версия коллекции может быть отмечена как устаревшая.

Жизненный цикл коллекции с учетом этапа согласования показан на схеме:



1. Версия коллекции загружается вручную разработчиком или автоматически самим Private Automation Hub при синхронизации с внешним источником коллекций. В обоих случаях версия коллекции помещается в репозиторий с меткой `pipeline: staging`.
2. Версия коллекции проходит согласование.

   Если версия коллекция согласована, она перемещается в соответствующий репозиторий с меткой `pipeline: approved`.

   Если версия коллекции отклонена, она перемещается в защищенный репозиторий `rejected`.

Этап согласования пропускается, если у соответствующего репозитория установлена метка `pipeline: approved`.

3. Независимо от того, в каком репозитории находится версия коллекции, она может быть отмечена как устаревшая. С устаревшей коллекции может быть снята отметка об устаревании.
4. Любая версия любой коллекции может быть удалена из Private Automation Hub.

Конвейер задач

В настройках репозитория может быть задан *конвейер задач* – порядок публикации версии коллекции для пользователей:

- **Согласовано** (Approved) – версия коллекции доступна пользователям сразу же после загрузки.
- **Промежуточный вариант** (Staging) – версия коллекции не будет доступна пользователям до тех пор, пока не будет согласована.
- **Ничего** (None) – версия коллекции размещается в защищенном репозитории `staging`.

Метки

Метки управляют доступностью коллекций для пользователей.

- `pipeline: approved` – версии коллекций доступны пользователям.

  Эта метка назначается автоматически при выборе конвейера задач **Согласовано** (Approved).
- `pipeline: rejected` – версии коллекций недоступны пользователям, так как были отклонены при согласовании.

  Эта метка назначена защищенному репозиторию `rejected` и не может быть присвоена другим репозиториям.
- `pipeline: staging` – версии коллекций недоступны пользователям и ожидают согласования.

  Эта метка присваивается автоматически при выборе конвейера задач **Промежуточный вариант** (Staging).
- `hide_from_search` – версии коллекции не выводятся в результатах поиска.

  Эта метка может быть присвоена при выборе конвейера задач **Согласовано** (Approved) или **Ничего** (None). При выборе конвейера задач **Промежуточный вариант** (Staging) эта метка присваивается автоматически.

6.2.3 Пространства имен

Для удобства управления коллекциями и предотвращения коллизий названий в Private Automation Hub используются *пространства имен*.

Например, коллекции `community.postgresql` и `astra.postgresql` относятся к пространствам имен `community` и `astra` соответственно.

Требования к названию пространств имен:

- длина не менее двух знаков;
- может содержать буквы латинского алфавита, цифры и символ _;
- не может начинаться с символа _.

6.2.4 Образы среды исполнения

Образы среды исполнения могут быть загружены в Private Automation Hub из других реестров образов или напрямую с рабочих станций разработчиков контента Ansible.

Коллекции

Private Automation Hub предъявляет к коллекциям следующие требования:

- Название:
  - длина не менее двух знаков;
  - может содержать буквы латинского алфавита, цифры и символ _;
  - не может начинаться с символа _.
- Коллекция должна иметь определенную структуру файлов и каталогов (см. ниже).
- Семантическое версионирование[57].

К коду коллекций, хранящихся в репозиториях Git, предъявляются следующие требования:

- Номера версий должны указываться в тегах[58].
- Тег **должен** содержать номер версии, например, `1.0.1`.
- Тег **может** содержать префикс `v`, например, `v1.0.1`.

Структура коллекции

В составе коллекции могут быть следующие файлы и каталоги:

- `meta/`

  Каталог с файлами, содержащими служебную информацию о коллекции, используемую утилитой `ansible-galaxy`.
- `plugins/`

  Опциональный каталог, содержащий расширения, необходимые для работы коллекции.
- `roles/`

  Каталог с подкаталогами ролей. Подробности см. в секции *Структура роли*.
- `tests/`

  Каталог с файлами автоматических тестов для коллекции. Файлы этого каталога могут быть полезны при изучении особенностей использования коллекции и подготовке окружения для работы с ней.
- `CHANGELOG.md`

  Файл с историей изменений между версиями.
- `LICENSE`

  Файл лицензии, под которой распространяется код коллекции.
- `README.md`

  Описание коллекции, включающее в себя:
  - Предназначение коллекции.
  - Требования к окружению.
  - Список поддерживаемых операционных систем.

[57] https://semver.org/
[58] https://git-scm.com/docs/git-tag

  - Порядок описания коллекции в файле зависимостей Ansible и способ установки.
  - Список ролей.
  - Порядок запуска автоматических тестов.
  - Тип лицензии, под которой распространяется код коллекции.
  - Информация об авторских правах.

- `galaxy.yml`

  Сведения о коллекции, используемые утилитой `ansible-galaxy`, в том числе номер актуальной версии коллекции.

Список 1: Образец galaxy.yml

```yaml
---
namespace: astra
name: ald_pro
version: 1.0.0
description: Collection for ALD Pro deployment
readme: README.md
authors:
  - LLC "RusBITech-Astra"
dependencies:
  "freeipa.ansible_freeipa": "1.10.0"
  "ansible.utils": "3.0.0"
tags:
  - astra
  - ald_pro
repository: https://hub.astra-automation.ru/aa-gca/ARFA/ald_pro
documentation: https://hub.astra-automation.ru/aa-gca/ARFA/ald_pro/-/blob/master/
↪README.md
```

  Здесь:

  - `astra` – название пространства имен.
  - `ald_pro` – название коллекции.
  - `version` – номер версии коллекции.
  - `description` – краткое описание коллекции.
  - `readme` – путь к файлу `README`, содержащему подробное описание коллекции.
  - `authors` – список авторов коллекции.
  - `dependencies` – зависимости, необходимые для использования коллекции.
  - `tags` – список тегов, по которым можно найти коллекцию среди множества других.
  - `repository` – ссылка на репозиторий с исходным кодом коллекции.
  - `documentation` – ссылка на документацию коллекции.

- `requirements_ansible.yml`

  Зависимости Ansible, необходимые для использования коллекции.

Структура роли

Для использования в Private Automation Hub роли Ansible должны быть включены в состав коллекции.

В составе роли могут быть следующие файлы и каталоги:

- `defaults/`

  Каталог с файлами, содержащими значения по умолчанию для переменных роли.

- `files/`

  Каталог с вспомогательными файлами, используемыми ролью, например, шаблоны HTML-страниц, изображения и т. д. При выполнении роли файлы из этого каталога копируются на управляемые узлы.

- `handlers/`

  Каталог с файлами обработчиков, выполняемых при использовании роли.

- `meta/`

  Каталог с файлами, содержащими служебную информацию о роли, используемую утилитой `ansible-galaxy`.

- `tasks/`

  Каталог с файлами, выполняющимися при использовании роли.

- `templates/`

  Каталог с шаблонами формата Jinja 2.

- `vars/`

  Каталог с файлами, содержащими список переменных роли. Каждая роль содержит список переменных, позволяющих управлять настройками соответствующего ПО.

- `LICENSE`

  Файл лицензии, под которой распространяется код роли.

- `README.md`

  Описание роли, включающее в себя:

  - Предназначение роли.
  - Требования к окружению.
  - Список переменных роли.

    Переменные роли делятся на обязательные и опциональные. Для опциональных переменных, значения которых не заданы, будут использованы значения по умолчанию.

    > **Предупреждение:** Значения обязательных переменных должны быть явно заданы в playbook или используемом им файле с переменными.

  - Примеры использования.
  - Порядок запуска автоматических тестов.
  - Тип лицензии, под которой распространяется код роли.
  - Информация об авторских правах.
  - Список поддерживаемых операционных систем.

Разграничение доступа

В Private Automation Hub используется разграничение доступа на основе ролей (*RBAC*).

Привилегии и роли

Привилегия это разрешение на доступ к определенному типу ресурсов или выполнение операции с ним. Например, привилегия `View collection remote` предоставляет разрешение на просмотр внешних репозиториев коллекций.

Изменить существующие привилегии или создать новые нельзя.

Роль – это набор привилегий.

Роли могут быть *встроенными* и *пользовательскими*.

- Встроенные роли нельзя изменить или удалить.

  Полный список встроенных ролей и их описание доступны в *справочнике*.

- Пользовательские роли могут быть созданы, изменены и удалены администраторами Private Automation Hub.

  Требования к названию пользовательской роли:

  - начинается с префикса `galaxy.`;
  - может содержать только символы латинского алфавита, цифры, точки и символы подчеркивания;
  - уникальность на уровне Private Automation Hub.

Группы

Группа может быть связана с одной или несколькими ролями.

Важно: Группы могут быть связаны только с ролями, названия которых начинаются с префикса `galaxy.`.

Участникам группы предоставляются привилегии, связанные со всеми ролями группы.

Пользователи

Необходимые привилегии можно предоставить пользователю следующими способами:

- Установить флаг суперпользователя.

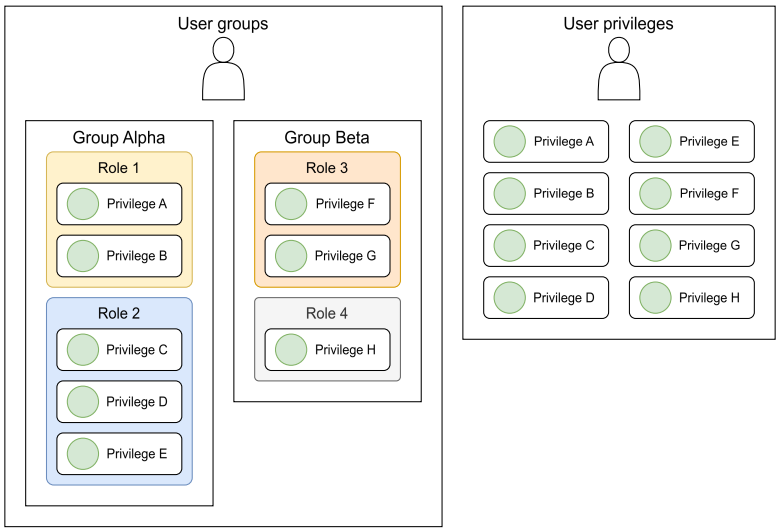
  Флаг суперпользователя дает полный доступ ко всем ресурсам Private Automation Hub. Снять флаг суперпользователя с самого себя нельзя.

- Сделать пользователя участником группы с нужной ролью.

Пример

Рассмотрим связь пользователей, групп, ролей и привилегий на примере. Пусть имеются следующие ресурсы:

- один пользователь, не являющийся суперпользователем;
- две группы, `Alpha` и `Beta`;
- четыре роли, `Role 1` – `Role 4`;
- восемь привилегий, A – H.



Роли предоставляют следующие привилегии:

- `Role 1` – A, B.
- `Role 2` – C, D, E.
- `Role 3` – F, G;
- `Role 4` – H.

Группа `Alpha` связана с ролями `Role 1` и `Role 2`, а группа `Beta` – с ролями `Role 3` и `Role 4`.

Пользователь является участником обеих групп, а значит, он получает привилегии всех связанных с ними ролей, от A до H включительно.

6.3 Применение Private Automation Hub

Применение Private Automation Hub включает следующие типовые операции:

- Настройка Private Automation Hub на получение содержимого из сторонних реестров, в том числе Automation Hub[59].
- Настройка проектов Ansible на работу с Private Automation Hub независимо от способа управления задачами автоматизации.

[59] https://hub.astra-automation.ru/

6.3.1 Синхронизация коллекций

Синхронизация Private Automation Hub с другими реестрами коллекций позволяет решить следующие задачи:

- сокращение времени загрузки коллекций в Automation Controller и на рабочие места пользователей Ansible;
- поддержание актуальности версий коллекций в Private Automation Hub;
- тонкая настройка доступности версий коллекций.

В результате синхронизации в Private Automation Hub будут доступны коллекции, распространяемые через сторонние реестры. Например, вы можете настроить синхронизацию Private Automation Hub с реестром Automation Hub[60], чтобы всегда иметь под рукой необходимые версии коллекций, созданных ПАО Группа Астра и ее партнерами.

Для настройки синхронизации коллекций с внешним реестром необходимо выполнить описанные далее действия.

Получение полномочий

Некоторые реестры коллекций, например, Ansible Galaxy[61], не требуют полномочий для доступа к содержимому. Другие предоставляют доступ только авторизованным пользователям. В этом случае для синхронизации коллекций необходимо получить соответствующие полномочия.

- Авторизация в Automation Hub осуществляется по токену. Для его создания воспользуйтесь *инструкцией*.
- Прочие реестры коллекций могут использовать для авторизации имя пользователя и пароль, токен, ключи SSH и тому подобное. Для получения необходимых полномочий обратитесь к владельцу соответствующего реестра.

Private Automation Hub может использоваться как источник коллекций. Для авторизации в нем можно использовать следующие способы:

- токен;
- имя пользователя и пароль.

Настройка синхронизации

Чтобы Private Automation Hub мог получать коллекции из стороннего реестра коллекций, выполните следующие действия:

1. Согласно *инструкции* добавьте внешний репозиторий со следующими свойствами:
   - **Название** (Name): произвольное название внешнего репозитория, например, `Automation Hub`.
   - **URL**: URL репозитория во внешнем реестре, например:

     ```
     https://hub.example.com/api/galaxy/content/<repository>/
     ```

     Важно: URL репозитория должен завершаться символом `/`.

     Здесь <repository> – название репозитория во внешнем реестре.

     По умолчанию коллекции загружаются из репозитория `published`. В этом случае допускается указать сокращенный URL:

[60] https://hub.astra-automation.ru/
[61] https://galaxy.ansible.com/

```
https://hub.example.com/api/galaxy/
```

- **Требования YAML** (YAML requirements) – если вы хотите синхронизировать все доступные коллекции, оставьте это поле пустым. Чтобы синхронизировать только определенные коллекции, укажите их названия, например:

```yaml
---
collections:
  - name: astra.ald_pro
  - name: astra.ceph
  - name: astra.nginx
```

Также укажите полномочия, используемые для авторизации во внешнем реестре. При настройке синхронизации Private Automation Hub с реестром Automation Hub заполните поле **Токен**, а поля **Имя пользователя** и **Пароль** оставьте пустыми.

2. Согласно *инструкции* создайте репозиторий, в котором будут размещаться коллекции, полученные из стороннего реестра. В поле **Внешний репозиторий** (Remote) выберите внешний репозиторий, созданный на предыдущем шаге.

Синхронизация

Следуя *инструкции* запустите синхронизацию репозитория с внешним источником. Дождитесь перехода задания синхронизации в статус **Успешно** (Success).

Совет: Для отслеживания процесса синхронизации в реальном времени выполните следующие действия:

1. На панели навигации выберите пункт **Управление задачами** (Task management).
2. Нажмите на ссылку с названием задачи синхронизации.

   Сообщения о ходе синхронизации выводятся на панели **Сообщения о ходе исполнения** (Progress messages).

Публикация

Чтобы загруженные коллекции были доступны пользователям, разместите их в репозитории со следующими свойствами:

- Присвоена *метка* `pipeline: approved`.

  Если версии коллекций загружены в репозиторий с меткой `pipeline: staging`, проведите согласование.

  Если версии коллекций загружены в репозиторий без меток, измените его конвейер задач на **Согласовано** (Approved). Метка `pipeline: approved` будет присвоена автоматически.

- Включена настройка **Создать дистрибутив** (Create distribution).

  Совет: В свойствах защищенных репозиториев `aa-certified`, `validated`, `community` и `published` эта настройка включена по умолчанию.

Проверка результатов синхронизации

Для проверки успешности синхронизации выполните следующие действия в Private Automation Hub и внешнем реестре:

1. На панели навигации выберите пункт *Коллекции* ▸ *Пространства имен* (*Collections* ▸ *Namespaces*).
2. Нажмите на ссылку с названием синхронизированного пространства имен.
3. Убедитесь, что выбранное пространство имен в Private Automation Hub и внешнем реестре содержит один и тот же набор коллекций.

Рекомендуемые настройки синхронизации с Automation Hub

В Private Automation Hub по умолчанию доступны репозитории `validated` и `aa-certified`. Чтобы настроить их на получение *проверенных* и *сертифицированных* коллекций из Automation Hub, выполните следующие действия:

1. Создайте токен для доступа к Automation Hub согласно *инструкции*.
2. В свойствах внешнего репозитория `aa-certified` укажите значение токена для доступа к Automation Hub.
3. Добавьте внешний репозиторий согласно *инструкции*:
   - **Название**: `aa-validated`.
   - **URL**:

     ```
     https://hub.astra-automation.ru/api/galaxy/content/validated/
     ```

   - **Токен**: укажите значение токена для доступа к Automation Hub.
4. Измените настройки репозитория `validated` согласно *инструкции*. В поле **Внешний репозиторий** (Remote) выберите внешний репозиторий `aa-validated`.

   Совет: Изменять настройки репозитория `aa-certified` не нужно – он по умолчанию настроен на получение содержимого из внешнего репозитория `aa-certified`.

5. Запустите синхронизацию обоих репозиториев Private Automation Hub с репозиториями Automation Hub согласно *инструкции*.

6.3.2 Настройка проекта на использование Private Automation Hub

Управлять задачами автоматизации можно различными способами, например, с применением *Automation Controller* и *Ansible Navigator*. В обоих случаях для получения коллекций из внешних источников используется утилита `ansible-galaxy`. С настройками по умолчанию она может загружать коллекции только из реестра Ansible Galaxy[62].

Для загрузки коллекций из Private Automation Hub необходимо выполнить ряд дополнительных настроек.

Совет: Использование образов среды исполнения, содержащих все необходимые коллекции, является более эффективным способом управления зависимостями Ansible, чем их загрузка из реестров коллекций. Для создания собственного образа среды исполнения, содержащего коллекции из Private Automation Hub, следуйте *инструкции*.

[62] https://galaxy.ansible.com/

Предварительные настройки

Предварительно в Private Automation Hub необходимо *создать токен доступа* для проектов.

Важно: Период действия токена по умолчанию – 24 часа. По истечении указанного времени его нужно будет создать заново.

При создании нового токена предыдущий перестает действовать немедленно.

Указание реквизитов

При управлении задачами автоматизации с помощью *Ansible Navigator* укажите в файле `ansible.cfg` реквизиты доступа к Private Automation Hub:

1. В секцию `[galaxy]` добавьте параметр `server_list`. В его значении одной строкой через запятую укажите список условных названий репозиториев, из которых будет загружаться содержимое, например:

   ```
   [galaxy]
   server_list = my_private_hub, galaxy
   ```

   Важно: Особенности заполнения параметра `server_list`:

   - Допускается использование пробела после запятой.
   - Указанный список заменяет собой список реестров по умолчанию. Это значит, что для использования реестра Ansible Galaxy[63] нужно будет выполнить следующие дополнительные действия:
     1. Указать в значении параметра `server_list` условное название реестра Ansible Galaxy.
     2. Создать дополнительную секцию с описанием настроек подключения к Ansible Galaxy.

   ---

2. По шаблону создайте и заполните секции для всех упомянутых серверов:

   ```
   [galaxy_server.<name>]
   url = <url>

   # При авторизации с использованием токена:
   token = <token>

   # При авторизации по названию учетной записи пользователя и паролю:
   username = <username>
   password = <password>
   ```

   Здесь:

   - <name> – условное название репозитория, указанное в `server_list`.
   - `url` – URL репозитория.

     Важно: URL репозитория должен завершаться символом `/`.

     По умолчанию коллекции загружаются из репозитория `published`. В этом случае допускается указать сокращенный URL:

     ```
     https://hub.example.com/api/galaxy/
     ```

[63] https://galaxy.ansible.com/

Чтобы получить доступ к репозиторию, отличному от published, приведите URL к следующему виду:

```
https://hub.example.com/api/galaxy/content/<repository>/
```

Здесь <repository> – название репозитория в Private Automation Hub.

- <token> – значение токена, используемого для доступа к реестру.
- <username> – название учетной записи пользователя реестра.
- <password> – пароль пользователя реестра.

Совет: Для доступа к реестру Ansible Galaxy[64] указание учетных данных не требуется.

Создание полномочия

При управлении задачами автоматизации с помощью Automation Controller достаточно создать полномочие типа *API-токен Ansible Galaxy/Automation Hub* и связать его с организацией.

Хотя вы все еще можете добавить в проект файл `ansible.cfg`, созданный согласно приведенной выше инструкции, использование полномочия имеет ряд преимуществ:

- не требуется добавление в файл `ansible.cfg` учетных данных в открытом виде или настройка Automation Controller на работу с *внешней системой управления секретами*;
- при устаревании или отзыве токена не нужно вносить изменения в файлы проекта;
- действие полномочия распространяется на всех пользователей организации.

Для получения и установки зависимостей из Private Automation Hub выполните следующие действия:

1. В корневом каталоге проекта создайте файл зависимостей `requirements.yml`, например:

```yaml
---
collections:
  - name: astra.rupost
    version: 0.14.4
  - name: astra.nginx
    version: 1.8.2
  - name: astra.postgresql
    version: 3.0.0
```

2. Согласно *инструкции* создайте полномочие типа *API-токен Ansible Galaxy/Automation Hub*. Поля формы в секции **Сведения о типе** (Type Details) заполните следующим образом:

   - **URL сервера Galaxy** (Galaxy Server URL) – URL репозитория в Private Automation Hub.

     Важно: Указанный URL должен завершаться символом /.

     По умолчанию коллекции загружаются из репозитория published. В этом случае допускается указать сокращенный URL:

     ```
     https://hub.example.com/api/galaxy/
     ```

     Чтобы получить доступ к репозиторию, отличному от published, приведите URL к следующему виду:

[64] https://galaxy.ansible.com/

```
https://hub.example.com/api/galaxy/content/<repository>/
```

Здесь <repository> – название репозитория в Private Automation Hub.

- **Токен API** (API Token): содержимое токена.

3. Если для защиты подключения к Private Automation Hub используется самоподписанный сертификат, в настройках контроллера *включите игнорирование результатов проверки сертификатов SSL* при работе с реестрами коллекций.

   Если эта настройка выключена, проверка сертификатов при загрузке и установке зависимостей проекта будет завершаться ошибкой:

```
[WARNING]: Skipping Galaxy server https://hub.example.com/api/galaxy/content/aa-
↪validated/.
Got an unexpected error when getting available versions of
collection astra.chrony: Unknown error when attempting to call Galaxy at
'https://hub.example.com/api/galaxy/content/aa-validated/api': <urlopen error
[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: self signed
certificate (_ssl.c:1129)>. <urlopen error [SSL: CERTIFICATE_VERIFY_FAILED]
certificate verify failed: self signed certificate (_ssl.c:1129)>
ERROR! Failed to resolve the requested dependencies map. Could not satisfy the␣
↪following requirements:
* astra.chrony:* (direct request)
```

4. Согласно *инструкции* измените свойства организации, добавив в список **Учетные данные Galaxy** (Galaxy Credentials) созданное полномочие.
5. Синхронизируйте проект с источником согласно *инструкции*. Automation Controller использует связанное с организацией полномочие для загрузки необходимых коллекций из Private Automation Hub.

Пример

Пусть проект использует коллекции из следующих реестров:

- Ansible Galaxy[65];
- Private Automation Hub, доступный по адресу `https://private-hub.example.com`.

  При этом необходимо настроить доступ к следующим репозиториям:

  - `published` – репозиторий по умолчанию;
  - `testing` – коллекции, не прошедшие полное тестирование;
  - `automation-hub` – коллекции, полученные из Automation Hub при *синхронизации*.

Чтобы настроить проект на работу со всеми указанными реестрами и репозиториями, содержимое `ansible.cfg` необходимо привести к следующему виду:

```
[galaxy]
server_list = galaxy, hub_published, hub_testing, automation_hub

[galaxy_server.galaxy]
url = https://galaxy.ansible.com/

[galaxy_server.hub_published]
url = https://private-hub.example.com/api/galaxy/
token = b23...41e

[galaxy_server.hub_testing]
url = https://private-hub.example.com/api/galaxy/content/testing/
token = b23...41e
```

(continues on next page)

[65] https://galaxy.ansible.com/

(продолжение с предыдущей страницы)

```
[galaxy_server.automation_hub]
url = https://private-hub.example.com/api/galaxy/content/automation-hub/
token = b23...41e
```

При использовании Automation Controller достаточно описать зависимости в файле `requirements.yml` и создать три полномочия типа *API-токен Ansible Galaxy/Automation Hub*, по одному на каждый репозиторий.

6.3.3 Загрузка образов среды исполнения

Загрузка образов среды исполнения в Private Automation Hub позволяет решить следующие задачи:

- сокращение времени загрузки образов среды исполнения на узлы Automation Controller и рабочие места пользователей Ansible;
- предоставление пользователям образов среды исполнения, разработанных собственными силами.

В результате загрузки образов в Private Automation Hub они будут доступны пользователям точно так же, как если бы они были загружены в один из публичных реестров, например, Docker Hub[66] или Quay.io[67].

Загрузка образов может быть выполнена из внешних реестров образов или напрямую с рабочих станций разработчиков контента Ansible.

Загрузка из внешнего реестра

Для загрузки образов из внешнего реестра выполните следующие действия:

1. Согласно *инструкции* создайте запись о внешнем реестре (remote registry).
2. Согласно *инструкции* создайте запись о среде исполнения, содержащую следующие сведения:
   - внешний реестр (выберите реестр, созданный на предыдущем шаге);
   - название пространства имен контейнеров;
   - название образа;
   - список меток, по которым выполняется отбор версий образа.

     В названиях меток допускается использование подстановочного символа *.
3. Согласно *инструкции* запустите синхронизацию среды исполнения с внешним реестром.

Пример

Чтобы загрузить из реестра Automation Hub[68] все образы `aa-base-ee` ветки `0.6.*`, кроме `0.6.3`, выполните следующие действия:

1. Создайте запись о внешнем реестре со следующим URI:

   ```
   https://hub.astra-automation.ru/
   ```

2. Создайте запись о среде исполнения со следующими свойствами:
   - **Внешний реестр** (Registry): выберите реестр, созданный на предыдущем шаге.

[66] https://hub.docker.com
[67] https://quay.io/
[68] https://hub.astra-automation.ru/

- **Название вышестоящего объекта** (Upstream name): `aa/aa-base-ee`.
- **Текущие включенные метки** (Currently included tags): `0.6.*`.
- **Текущие исключенные метки** (Currently excluded tags): `0.6.3`.

Загрузка с рабочей станции

Для загрузки образа среды исполнения с рабочей станции выполните на ней следующие действия:

1. Предоставьте Podman учетные данные для доступа к Private Automation Hub:

   ```
   podman login -u=<username> -p=<password> <hub>
   ```

   Здесь:

   - <username> – название учетной записи пользователя Private Automation Hub.

     Указанный пользователь должен обладать привилегиями на загрузку образов среды исполнения.
   - <password> – пароль.
   - <hub> – доменное имя Private Automation Hub.

   В случае успешной авторизации в терминал выводится следующая строка:

   ```
   Login Succeeded!
   ```

   Подробности о команде `login` см. в документации Podman[69].

2. Присвойте образу название и метку, с которыми он должен быть опубликован в Private Automation Hub:

   ```
   podman tag <image_local>:<label_local> <hub>/<image_hub>:<label_hub>
   ```

   Здесь:

   - <image_local> – название образа, собранного на рабочей станции;
   - <label_local> – метка образа, собранного на рабочей станции;
   - <hub> – FQDN Private Automation Hub;
   - <image_hub> – название, под которым образ должен быть опубликован в Private Automation Hub;
   - <label_hub> – метка, с которой образ должен быть опубликован в Private Automation Hub.

3. Загрузите образ в Private Automation Hub:

   ```
   podman push <hub>/<image_hub>:<tag_hub>
   ```

   Здесь:

   - <hub> – FQDN Private Automation Hub.
   - <image_hub> – название, под которым образ должен быть опубликован в Private Automation Hub.
   - <tag_hub> – метка, с которой образ должен быть опубликован в Private Automation Hub.

   Подробности о команде `push` см. в документации Podman[70].

[69] https://docs.podman.io/en/latest/markdown/podman-login.1.html
[70] https://docs.podman.io/en/latest/markdown/podman-push.1.html

Пример

Private Automation Hub доступен по FQDN `hub.example.com`. На рабочей станции разработчика собран образ среды исполнения с названием `ee-automation-image:latest`. В Private Automation Hub этот образ нужно опубликовать с названием и меткой `custom-ee-image:1.0.0`.

1. Авторизуйтесь в Private Automation Hub:

```
podman login -u=<user> -p=<password> hub.example.com
```

2. Присвойте образу название и метку, с которыми он должен быть опубликован в Private Automation Hub:

```
podman tag ee-automation-image:latest hub.example.com/custom-ee-image:1.0.0
```

3. Опубликуйте образ в Private Automation Hub:

```
podman push hub.example.com/custom-ee-image:1.0.0
```

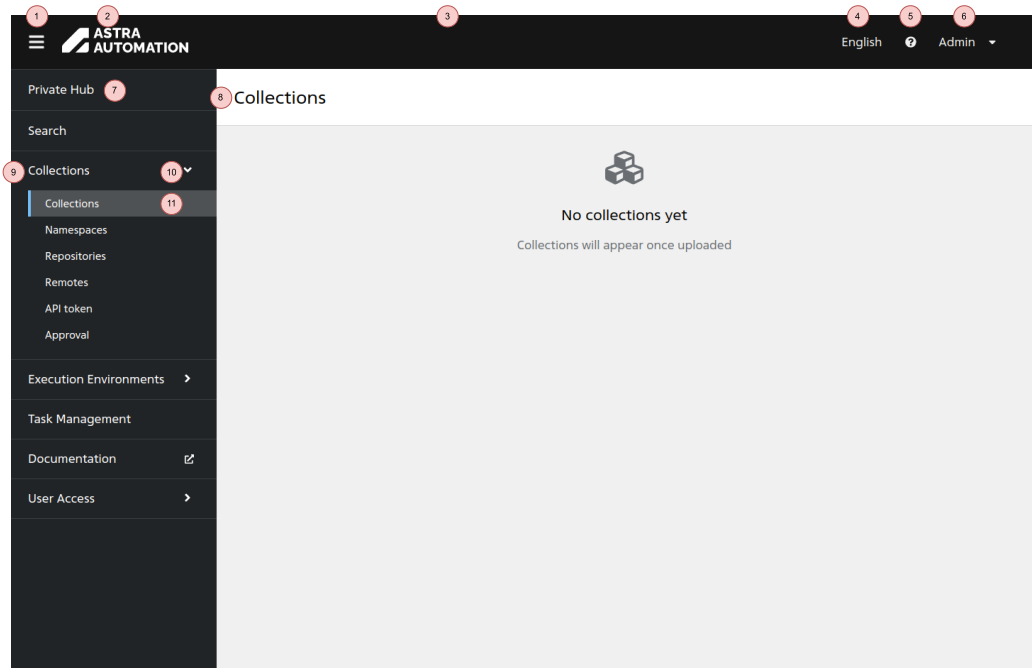
4. Убедитесь, что образ доступен в Private Automation Hub:
   1. После авторизации выберите на панели навигации *Среды исполнения* ▸ *Среды исполнения* (*Execution Environments* ▸ *Execution Environments*).
   2. Убедитесь, что в таблице сред исполнения появилась запись `custom-ee-image`.
   3. Нажмите на ссылку `custom-ee-image`.
   4. Выберите вкладку **Образы** (Images) и убедитесь в наличии образа с меткой `1.0.0`.

6.4 Графический интерфейс

Private Automation Hub предоставляет удобный веб-интерфейс для управления его работой. Расположенная в левой части панель навигации предоставляет доступ к управлению всеми ресурсами.

6.4.1 Основные компоненты

На схеме представлены основные компоненты пользовательского интерфейса Private Automation Hub:



Здесь:

1. Кнопка переключения видимости панели навигации.
2. Логотип.
3. Панель инструментов.
4. Кнопка смены языка.
5. Кнопка вызова справки.
6. Пользовательское меню.
7. Панель навигации.
8. Заголовок окна.
9. Раздел навигации.
10. Кнопка переключения видимости раздела навигации.
11. Активный раздел навигации.

6.4.2 Поиск

Раздел **Поиск** (Search) содержит поле ввода для поиска коллекций, пространств имен и сред исполнения, а также ссылки для быстрого перехода в окна **Коллекции** (Collections), **Пространства имен** (Namespaces) и **Среды исполнения** (Execution Environments).

При выполнении поиска его результаты отображаются на панелях **Коллекции** (Collections), **Пространства имен** (Namespaces) и **Среды исполнения** (Execution Environments).

6.4.3 Коллекции

Раздел *Коллекции* (Collections) содержит ссылки на разделы, используемые для управления следующими ресурсами:

- **Коллекции** (Collections) – коллекции Ansible;
- **Пространства имен** (Namespaces) – пространства имен;
- **Репозитории** (Repositories) – репозитории, используемые в качестве источников кода коллекций;
- **Внешние репозитории** (Remotes) – внешние ресурсы, используемые для получения коллекций;
- **Токен API** (API token) – управление токенами, используемыми для доступа к Private Automation Hub;
- **Согласование** (Approval) – принятие и отклонение коллекций и их версий.

6.4.4 Среды исполнения

Раздел *Среды исполнения* (Execution Environments) содержит ссылки на разделы, используемые для управления следующими ресурсами:

- **Среды исполнения** (Execution Environments) – образы среды исполнения, загруженные в Private Automation Hub.
- **Внешние реестры** (Remote Registries) – внешние реестры, используемые для получения образов среды исполнения.

6.4.5 Управление задачами

Раздел *Управление задачами* (Task Management) используется для управления задачами, обеспечивающими работу Private Automation Hub.

6.4.6 Документация

Раздел *Документация* (Documentation) – это ссылка на сайт с документацией Private Automation Hub.

6.4.7 Доступ пользователя

Раздел *Доступ пользователя* (User Access) содержит ссылки на разделы, используемые для управления параметрами доступа к Private Automation Hub:

- **Пользователи** (Users) – управление пользователями;
- **Группы** (Groups) – управление группами пользователей;
- **Роли** (Roles) – просмотр списка встроенных пользовательских ролей и управление собственными пользовательскими ролями.

Поиск

Окно **Поиск** (Search) позволяет быстро найти коллекции, пространства имен и среды исполнения.

Поиск выполняется среди опубликованных коллекций по частичному совпадению с названием или полным совпадением с ключевыми словами, указанными в описании коллекции. В результаты поиска не попадают отклоненные коллекции и коллекции, ожидающие *согласования*.

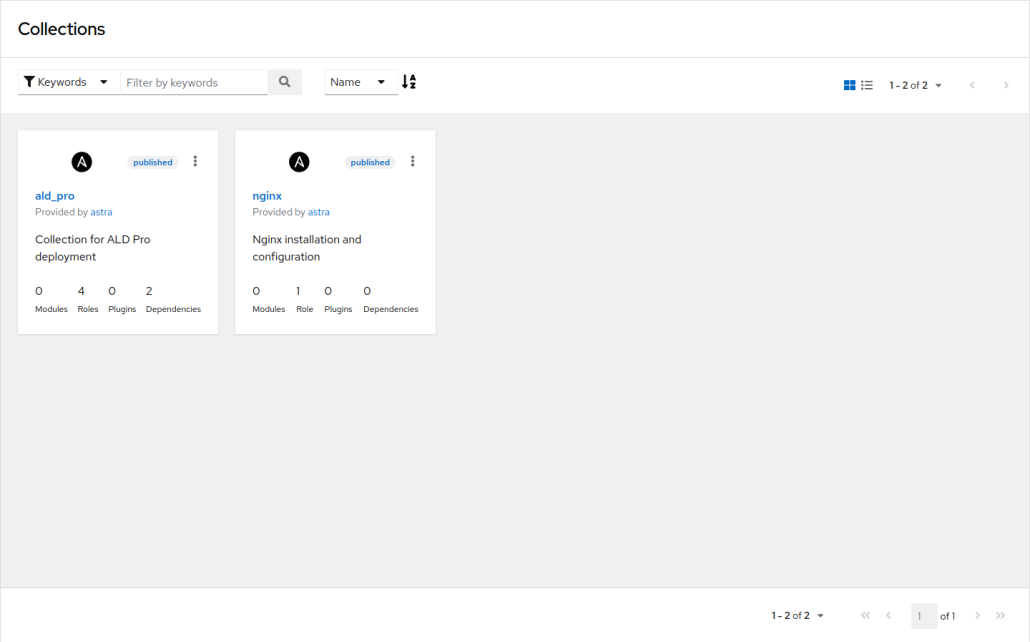
Поиск пространств имен и образов среды исполнения выполняется по частичному совпадению с названием.

Коллекции

Раздел **Коллекции** (Collections) содержит подразделы, используемые для управления пространствами имен, коллекциями, репозиториями, внешними источниками коллекций и токенами API. Здесь же находится раздел **Согласование**, используемый для проверки коллекций и их версий перед публикацией в Private Automation Hub.

Коллекции

Окно **Коллекции** (Collections) используется для управления *коллекциями Ansible*. Для перехода к нему выберите на панели навигации *Коллекции* ▸ *Коллекции* (*Collections* ▸ *Collections*).



Окно состоит из панели инструментов и списка коллекций.

На панели инструментов размещаются следующие поля:

- поле, позволяющее сократить список отображаемых коллекций путем фильтрации по ключевым словам, репозиторию, пространству имен или тегам;
- выпадающий список для выбора поля, по которому сортируются коллекции: - название; - пространство имен; - дата последнего изменения; - версия;
- кнопка изменения направления фильтрации;
- переключатели способа отображения списка коллекций.

Просмотр информации о коллекции

Для просмотра подробной информации о коллекции нажмите на ссылку с ее названием на соответствующей панели.

Окно просмотра сведений о коллекции состоит из выпадающего списка, позволяющего выбрать версию, и следующих вкладок:

- **Установить** (Install) – инструкция по установке коллекции с использованием утилиты `ansible-galaxy`.
- **Документация** (Documentation) – документация коллекции, входящих в нее ролей и модулей.
- **Содержимое** (Contents) – краткая информация о модулях и ролях, предоставляемых коллекцией.
- **Журнал импорта** (Import log) – журнал импорта коллекции в Private Automation Hub.
- **Зависимости** (Dependencies) – список зависимостей коллекции.
- **Источники контента** (Distributions) – список репозиториев, в которых размещена коллекция, а также краткая инструкция по настройке `ansible-galaxy` для их использования.

Загрузка коллекции

Для загрузки коллекции выполните следующие действия:

1. На панели навигации выберите *Коллекции* ‣ *Пространства имен* (*Collections* ‣ *Namespaces*).
2. На панели с названием нужного пространства имен нажмите на ссылку *Просмотреть коллекции* (View collections).
3. Нажмите кнопку *Загрузить коллекцию* (Upload collection).
4. В открывшемся окне заполните форму:
   - **Выбрать файл** (Select file) – файл в формате `.tar.gz`, сформированный с помощью команды `ansible-galaxy build`.
   - **Репозиторий** (Repository):
     - **Промежуточные репозитории** (Staging Repos) – коллекция будет загружена в репозиторий на Private Automation Hub.
     - **Все репозитории** (All Repos) – выберите один из *репозиториев* для загрузки кода коллекции в него.

     Важно: Для успешной загрузки коллекции необходимо наличие привилегий, позволяющих публиковать код в указанном репозитории от имени активного пользователя.
5. Нажмите кнопку *Загрузить* (Upload).

Удаление коллекции или ее версии

Для удаления версии коллекции выполните следующие действия:

1. В списке коллекций нажмите на ссылку с названием нужной коллекции.
2. В окне просмотра сведений о коллекции выберите в выпадающем списке версию, которую хотите удалить.
3. Нажмите кнопку ⋮ и в открывшемся меню выберите соответствующий пункт:
   - **Удалить коллекцию из системы** (Delete collection from system) – удаление всех версий коллекции из Private Automation Hub.
   - **Удалить коллекцию из репозитория** (Remove collection from repository) – удаление записи о коллекции из репозитория.
   - **Удалить версию <version> из системы** (Delete version <version> from system) – удаление из Private Automation Hub только выбранной версии коллекции.
   - **Удалить версию <version> из репозитория** (Remove version <version> from repository) – удаление из репозитория записи только о выбранной версии коллекции.
4. В открывшемся окне включите флаг **Я понимаю, что это действие не может быть отменено** (I understand that this action cannot be undone) и нажмите кнопку *Удалить* (Delete).

Изменение статуса устаревания

Чтобы отметить версию коллекции как устаревшую или актуальную, выполните следующие действия:

1. В списке коллекций нажмите на ссылку с названием нужной коллекции.
2. В окне просмотра сведений о коллекции выберите в выпадающем списке нужную версию.
3. Нажмите кнопку ⋮ и в открывшемся меню выберите пункт **Перевести в статус устаревшее** (Deprecated) / **Вывести из состояния устаревшей** (Undeprecate).

Копирование в другие репозитории

Чтобы скопировать версию коллекции в другие репозитории, выполните следующие действия:

1. В списке коллекций нажмите на ссылку с названием нужной коллекции.
2. В окне просмотра сведений о коллекции выберите в выпадающем списке нужную версию.
3. Нажмите кнопку ⋮ и в открывшемся меню выберите пункт **Скопировать версию <version> в репозитории** (Copy version <version> to repositories).
4. В открывшемся окне включите флаги в строках с нужными репозиториями и нажмите кнопку *Выбрать* (Select).

Загрузка новой версии

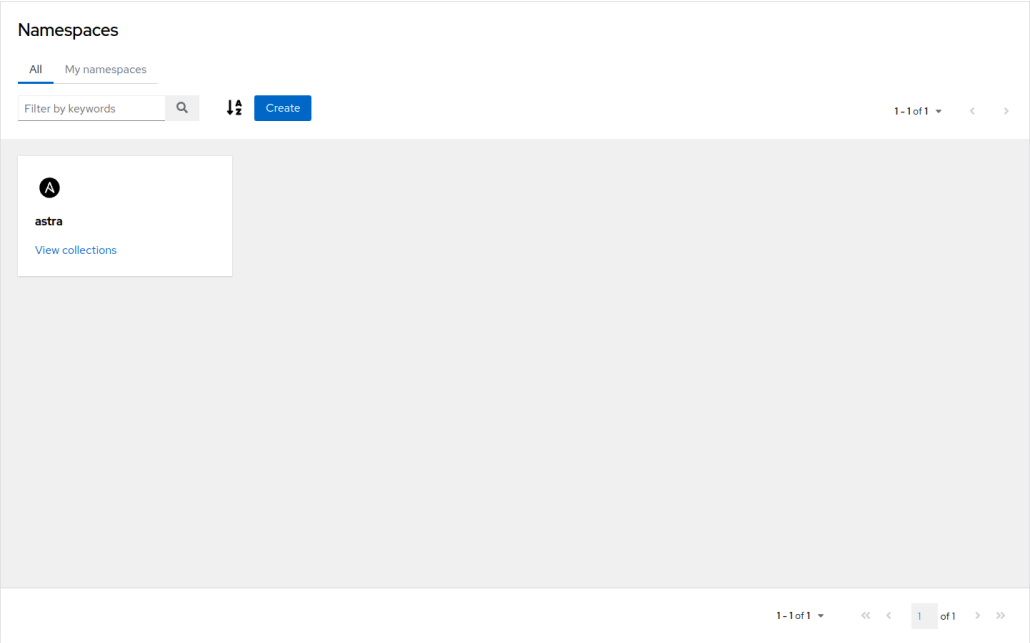
Чтобы загрузить новую версию коллекции, выполните следующие действия:

1. В списке коллекций нажмите на ссылку с названием нужной коллекции.
2. Нажмите кнопку ⋮ и в открывшемся меню выберите пункт **Загрузить новую версию** (Upload new version).
3. В открывшемся окне заполните форму:
   - **Выбрать файл** (Select file) – файл в формате `.tar.gz`, сформированный с помощью команды `ansible-galaxy build`.
   - **Репозиторий** (Repository):
     - **Промежуточные репозитории** (Staging Repos) – коллекция будет загружена в репозиторий Private Automation Hub.
     - **Все репозитории** (All Repos) – выберите один из *репозиториев* для загрузки кода коллекции в него.

     Важно: Для успешной загрузки коллекции необходимо наличие привилегий, позволяющих публиковать код в указанном репозитории от имени активного пользователя.
4. Нажмите кнопку *Загрузить* (Upload).

Пространства имен

Окно **Пространства имен** (Namespaces) используется для управления *пространствами имен коллекций Ansible*. Для перехода к нему выберите на панели навигации *Коллекции* ▸ *Пространства имен* (*Collections* ▸ *Namespaces*).



Окно состоит из двух вкладок:

- **Все** (All) – все пространства имен, существующие в Private Automation Hub.
- **Мои пространства имен** (My namespaces) – пространства имен, доступ к которым предоставлен активному пользователю непосредственно или через группы, в которых он состоит.

В обеих вкладках выводятся панели, содержащие логотип пространства имен, его название и ссылку *Просмотреть коллекции* (View collections).

Просмотр информации о пространстве имен

Для просмотра подробной информации о пространстве имен нажмите на ссылку *Просмотреть коллекции* (View collections) на соответствующей панели.

Окно просмотра сведений о пространстве имен состоит из следующих вкладок:

- **Коллекции** (Collections) – таблица коллекций, имеющихся в пространстве имен.

  Примечание: В списке отображаются только коллекции, прошедшие согласование.

- **Конфигурация CLI** (CLI configuration) – URI со ссылкой на Private Automation Hub. Эта ссылка может быть использована для настройки `ansible-galaxy` и Automation Controller на использование Private Automation Hub в качестве источника коллекций.
- **Ресурсы** (Resource) – дополнительные сведения о пространстве имен.

  Примечание: Эта вкладка не отображается, если соответствующий раздел не заполнен.

- **Доступ** (Access) – управление доступом к пространству имен как отдельных пользователей, так и групп.

Создание пространства имен

Для создания пространства имен выполните следующие действия:

1. В окне **Пространства имен** (Namespaces) нажмите кнопку *Создать* (Create).
2. Заполните форму **Создать новое пространство имен** (Create a new namespace):

   В поле **Название** (Name) укажите название пространства имен.

   Требования к названию пространств имен:

   - длина не менее двух знаков;
   - может содержать буквы латинского алфавита, цифры и символ _;
   - не может начинаться с символа _.

   Важно: Будьте внимательны при выборе названия пространства имен – его нельзя будет изменить.

3. Нажмите кнопку *Создать* (Create).
4. *Загрузите коллекции*.
5. *Настройте параметры CLI*.
6. Укажите дополнительную информацию о пространстве имен.
7. *Настройте параметры доступа*.

Изменение свойств пространства имен

Для изменения основных сведений о пространстве имен выполните следующие действия:

1. На панели с названием нужного пространства имен нажмите на ссылку *Просмотреть коллекции* (View collections).
2. В окне с информацией о пространстве имен нажмите кнопку ⋮ и в открывшемся меню выберите **Редактировать пространство имен** (Edit namespace).
3. Во вкладке **Редактировать подробности** (Edit details) укажите следующие сведения о пространстве имен:
   - **Название компании** (Company name);
   - **URL логотипа** (Logo URI) – логотип по указанной ссылке будет загружен в Private Automation Hub при сохранении изменений;
   - **Описание** (Description) – общее описание пространства имен;
   - **Полезные ссылки** – список ссылок, которые могут быть полезны при поиске информации о пространстве имен и хранящихся в нем коллекциях, например, сайт компании-разработчика.
4. Во вкладке **Редактировать ресурсы** (Edit resources) укажите дополнительную информацию о пространстве имен, которая будет выводиться при его просмотре. Для оформления содержимого используйте правила разметки Markdown.

Загрузка коллекций

Для загрузки коллекции в пространство имен выполните следующие действия:

1. На панели с названием нужного пространства имен нажмите на ссылку *Просмотреть коллекции* (View collections).
2. Нажмите кнопку *Загрузить коллекцию* (Upload collection).
3. В открывшемся окне заполните форму:
   - **Выбрать файл** (Select file) – файл в формате `.tar.gz`, сформированный с помощью команды `ansible-galaxy build`.
   - **Репозиторий** (Repository):
     - **Промежуточные репозитории** – коллекция будет загружена в репозиторий на Private Automation Hub.
     - **Все репозитории** – выберите один из *репозиториев* для загрузки кода коллекции в него.

     Важно: Для успешной загрузки коллекции необходимо наличие привилегий, позволяющих публиковать код в указанном репозитории от имени активного пользователя.
4. Нажмите кнопку *Загрузить* (Upload).

Конфигурация CLI

Для настройки утилит командной строки на работу с Private Automation Hub выполните следующие действия:

1. На панели с названием нужного пространства имен нажмите на ссылку *Просмотреть коллекции* (View collections).
2. Выберите вкладку **Конфигурация CLI** (CLI configuration).
3. Скопируйте ссылку, отображаемую во вкладке.
4. Укажите параметры подключения к Private Automation Hub в конфигурационном файле `ansible.cfg` согласно *инструкции*.

При использовании Automation Controller *создайте полномочие* типа «API-токен Ansible Galaxy/Automation Hub».

Настройка доступа

Доступ к пространству имен может быть предоставлен отдельным пользователям и группам. Порядок настройки одинаков:

1. На панели с названием нужного пространства имен нажмите на ссылку *Просмотреть коллекци* (View collections).
2. Выберите вкладку **Доступ** (Access).
3. Чтобы предоставить доступ к пространству имен пользователю или группе нажмите кнопку *Выбрать пользователя* (Select a user) или *Выбрать группу* (Select a group) соответственно.
4. В открывшемся окне выберите нужного пользователя или группу и нажмите кнопку *Далее* (Next).
5. Во вкладке **Выбрать роль (роли)** выберите роли, предоставляемые пользователю или группе:

Роль	Предоставляемые привилегии
`galaxy.collection_namespace_owner`	• Изменение свойств пространства имен • Загрузка коллекций
`galaxy.collection_publisher`	• Создание пространств имен • Изменение свойств пространства имен • Загрузка коллекций

6. Нажмите кнопку *Далее* (Next).
7. Во вкладке **Предварительный просмотр** (Preview) проверьте корректность настроек и нажмите кнопку *Добавить* (Add).

Удаление пространства имен

> **Предупреждение:**
> - При удалении пространства имен также удаляются все содержащиеся в нем коллекции.
> - Это действие нельзя отменить.

Чтобы удалить пространство имен, выполните следующие действия:

1. На панели с названием удаляемого пространства имен нажмите на ссылку *Просмотреть коллекции* (View collections).
2. В окне с информацией о пространстве имен нажмите кнопку ⋮ и в открывшемся меню выберите **Удалить пространство имен** (Delete namespace).
3. В открывшемся окне включите флаг **Я понимаю, что это действие не может быть отменено** (I understand that this action cannot be undone) и нажмите кнопку *Удалить*.

Репозитории

Окно **Репозитории** (Repositories) позволяет выполнять следующие действия:

- *просматривать сведения о существующих репозиториях*;
- *создавать новые репозитории*;
- *изменять настройки существующих репозиториев*;
- *управлять доступом к репозиториям*;
- *получать настройки для использования репозитория утилитами Ansible*;
- *синхронизировать содержимое репозиториев Private Automation Hub с внешними репозиториями*;
- *управлять версиями репозиториев*;
- *удалять репозитории*.

Для перехода в окно **Репозитории** (Repositories) выберите на панели навигации *Коллекции ‣ Репозитории* (*Collections ‣ Repositories*).

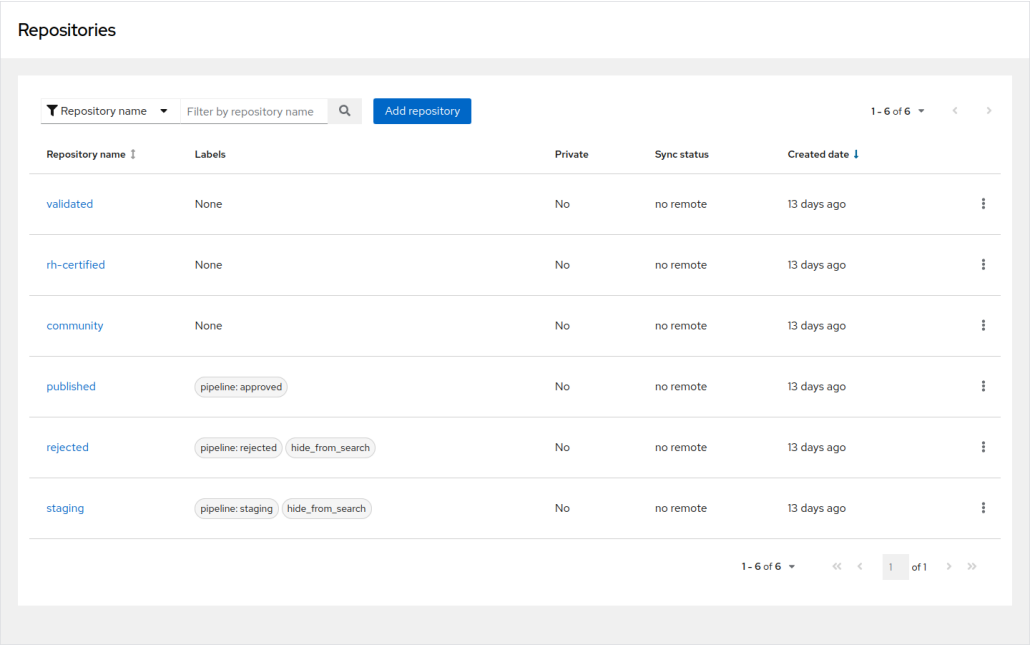


Таблица репозиториев состоит из следующих колонок:

- **Название репозитория** (Repository name) – название репозитория. При нажатии на ссылку происходит переход в окно просмотра подробной информации о репозитории.
- **Метки** (Labels) – метки, которыми отмечен репозиторий.
- **Приватный** (Private) – если репозиторий приватный, доступ к нему имеют только аутентифицированные пользователи.
- **Состояние синхронизации** (Sync status) – количество времени с момента последней синхронизации репозитория с внешним репозиторием.
- **Дата создания** (Created date) – количество времени с момента создания репозитория.
- Кнопка ⋮ для вызова меню дополнительных действий.

Просмотр

Для получения подробных сведений о репозитории нажмите на ссылку с его названием в таблице репозиториев. Окно просмотра сведений о репозитории состоит из следующих вкладок:

- **Подробности** (Details) – общие сведения о репозитории: название, описание, количество версий, сохраненное в Private Automation Hub, URL репозитория и так далее.
- **Доступ** – пользователи-владельцы и группы-владельцы репозитория.
- **Версии коллекции** (Collections versions) – список коллекций с указанием версий, загруженных в Private Automation Hub.
- **Версии** – версии репозитория, хранящегося в Private Automation Hub.

  Таблица версий состоит из следующих колонок:

  - **Номер версии** (Version number) – номер версии *репозитория*.
  - **Дата создания** (Created date) – количество времени с момента создания версии.
  - Кнопка ⋮, позволяющая вернуть состояние репозитория к выбранной версии.

Создание репозитория

Для создания репозитория выполните следующие действия:

1. В окне **Репозитории** (Repositories) нажмите кнопку *Добавить репозиторий* (Add repository).
2. Заполните форму **Добавить новый репозиторий** (Add new repository):
   - **Название** (Name) – название репозитория в Private Automation Hub.

     Важно: Будьте внимательны при выборе названия – его нельзя будет изменить.

   - **Описание** (Description) – краткое описание репозитория, например, для каких целей он создается.
   - **Сохраненное количество версий** (Retained number of versions) – количество версий репозитория, которое будет храниться в Private Automation Hub. При создании новой версии самая старая будет удаляться автоматически.
   - **Создать дистрибутив** (Create a distribution) – если этот флаг включен, пользователи Private Automation Hub смогут синхронизировать, загружать и искать содержимое, хранящееся в репозитории. В противном случае содержимое репозитория будет недоступно для пользователей.
   - **Конвейер задач** (Pipeline) – порядок попадания содержимого в репозиторий:

  - **Ничего** (None) – для загрузки коллекций пользователям требуются привилегии на изменение содержимого.
  - **Согласовано** (Approved) – коллекции попадают в репозиторий после согласования. Публикация коллекций напрямую в этот репозиторий отключена.
  - **Промежуточный** (Staging) – хранящиеся в репозитории коллекции требуют согласования, чтобы попасть на страницу поиска. Загружать коллекции в этот репозиторий могут любые пользователи, имеющие привилегии на загрузку содержимого в пространства имен.
- **Скрыть от поиска** (Hide from search) – если этот флаг включен, хранящиеся в репозитории коллекции не будут выводиться в результатах поиска.
- **Сделать приватным** (Make private) – если этот флаг включен, содержимое репозитория смогут использовать только его пользователи-владецы.
- **Внешний** (Remote) – внешний репозиторий, из которого при синхронизации загружается содержимое.

3. Нажмите кнопку *Сохранить* (Save).

Изменение репозитория

Для изменения свойств существующего репозитория выполните следующие действия:

1. В окне **Репозитории** (Repositories) нажмите на ссылку с названием нужного репозитория.
2. В окне свойств выбранного репозитория нажмите кнопку *Редактировать* (Edit).
3. Измените свойства репозитория.

Важно: Название существующего репозитория нельзя изменить. У защищенных репозиториев также недоступны для изменения некоторые другие свойства.

- **Описание** (Description) – краткое описание репозитория, например, для каких целей он создается.
- **Сохраненное количество версий** (Retained number of versions) – количество версий репозитория, которые будет храниться в Private Automation Hub. При создании новой версии самая старая будет удаляться автоматически.
- **Создать дистрибутив** (Create a distribution) – если этот флаг включен, пользователи Private Automation Hub смогут синхронизировать, загружать и искать содержимое, хранящееся в репозитории. В противном случае содержимое репозитория будет недоступно для пользователей.

  Примечание: В свойствах защищенных репозиториев `aa-certified`, `validated`, `community` и `published` эта настройка по умолчанию **включена**, и выключить ее нельзя.

  В свойствах защищенных репозиториев `rejected` и `staging` эта настройка по умолчанию **выключена**, и включить ее нельзя.

- **Конвейер задач** (Pipeline) – порядок попадания содержимого в репозиторий:
  - **Ничего** (None) – для загрузки коллекций пользователям требуются привилегии на изменение содержимого.
  - **Согласовано** (Approved) – коллекции попадают в репозиторий после согласования. Публикация коллекций напрямую в этот репозиторий отключена.
  - **Промежуточный** (Staging) – хранящиеся в репозитории коллекции требуют согласования, чтобы попасть на страницу поиска. Загружать коллекции в этот

репозиторий могут любые пользователи, имеющие привилегии на загрузку содержимого в пространства имен.

- **Скрыть от поиска** (Hide from search) – если этот флаг включен, хранящиеся в репозитории коллекции не будут выводиться в результатах поиска.
- **Сделать приватным** (Make private) – если этот флаг включен, содержимое репозитория смогут использовать только его пользователи-владецы.
- **Внешний** (Remote) – внешний репозиторий, из которого при синхронизации загружается содержимое.

4. Нажмите кнопку *Сохранить* (Save).

Настройка владения

Владельцами репозитория могут быть пользователи и группы пользователей. Порядок настройки одинаков.

1. В таблице со списком репозиториев нажмите на ссылку с названием нужной записи.
2. Выберите вкладку **Доступ** (Access).
3. Чтобы сделать владельцем репозитория пользователя или группу, нажмите кнопку *Выбрать пользователя* (Select a user) или *Выбрать группу* (Select a group) соответственно.
4. В открывшемся окне выберите нужного пользователя или группу и нажмите кнопку *Далее* (Next).
5. Во вкладке **Выбрать роль (роли)** (Select role(s)) включите флаг роли `galaxy.ansible_repository_owner` и нажмите кнопку **Далее** (Next).
6. Во вкладке **Предварительный просмотр** (Preview) проверьте корректность настроек и нажмите кнопку *Добавить* (Add).

Настройка CLI на работу с репозиторием

Чтобы настроить инструменты командной строки Ansible на работу с репозиторием, выполните следующие действия:

1. В таблице со списком репозиториев нажмите на ссылку с названием нужной записи.
2. Нажмите кнопку *Скопировать конфигурацию CLI* (Copy CLI configuration). Необходимые настройки будут автоматически скопированы в буфер обмена.
3. Добавьте строки из буфера обмена в конфигурационный файл `ansible.cfg`.

Синхронизация с внешним репозиторием

Чтобы синхронизировать содержимое репозитория Private Automation Hub с внешним репозиторием, выполните следующие действия:

1. В таблице со списком репозиториев нажмите на ссылку с названием нужной записи.
2. Нажмите кнопку *Синхронизация* (Sync).
3. В открывшемся окне укажите настройки синхронизации:
   - **Зеркало** (Mirror) – если этот флаг включен, Private Automation Hub выполняет с версиями коллекций следующие действия:
     - удаляет версии, отсутствующие в источнике;
     - загружает из внешнего источника версии, которых нет в репозитории;
     - отмечает как устаревшие версии, отмеченные как устаревшие в источнике;

  - отмечает как актуальные версии, отмеченные как актуальные в источнике.
- **Оптимизировать** (Optimize) – если этот флаг включен, загрузка содержимого из внешнего репозитория будет выполняться только в том случае, если внешний репозиторий подтвердит наличие изменений.

4. Нажмите кнопку *Синхронизация*.

Управление версиями репозитория

Если в настройках репозитория количество сохраненных версий больше единицы, при необходимости можно вернуть репозиторий в одно из прежних состояний:

1. В таблице со списком репозиториев нажмите на ссылку с названием нужной записи.
2. Выберите вкладку **Версии** (Version).
3. В таблице версий нажмите кнопку ⋮ в строке с нужной записью и в открывшемся меню выберите **Вернуть к этой версии** (Revert to this version).

Удаление репозитория

Предупреждение: При удалении репозитория также удаляются все связанные с ним ресурсы.

Для удаления репозитория выполните следующие действия:

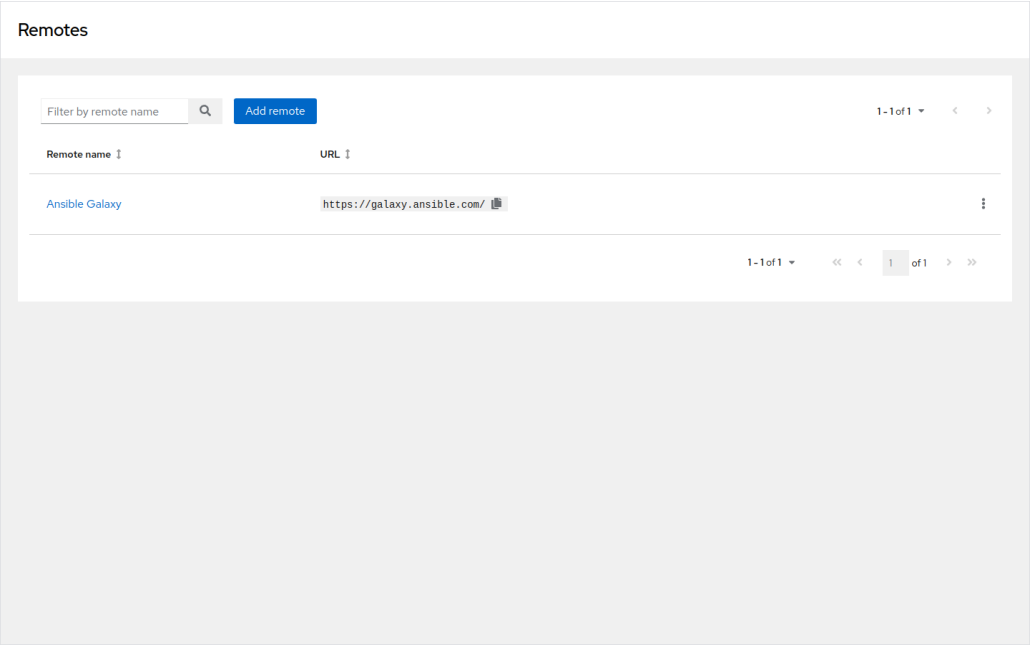
1. В таблице со списком репозиториев нажмите на ссылку с названием нужной записи.
2. Нажмите кнопку *Удалить*.
3. Подтвердите удаление.

Внешние репозитории

Окно **Внешние репозитории** (Remotes) позволяет выполнять следующие действия:

- *просматривать сведения о существующих внешних репозиториях*;
- *добавлять записи о внешних репозиториях*;
- *изменять записи о внешних репозиториях*;
- *управлять владельцами записей о внешних репозиториях*;
- *удалять записи о внешних репозиториях*.

Для перехода к нему выберите на панели навигации *Коллекции* ‣ *Внешние репозитории* (*Collections* ‣ *Remotes*).



Окно состоит из панели инструментов и списка внешних репозиториев.

На панели инструментов размещаются поле поиска и кнопка *Добавить внешний репозиторий*.

Таблица внешних репозиториев состоит из следующих колонок:

- **Название внешнего репозитория** (Remote name) – название внешнего репозитория в Private Automation Hub.

  При нажатии на ссылку происходит переход в окно просмотра подробной информации о внешнем репозитории.
- **URL** – ссылка на внешний репозиторий.
- Кнопка ⋮ для вызова меню дополнительных действий.

Просмотр

Для получения подробных сведений о внешнем репозитории нажмите на ссылку с его названием в таблице внешних репозиториев. Окно просмотра сведений о внешнем репозитории состоит из следующих вкладок:

- **Подробности** (Details) – общие сведения о репозитории: название, URL, настройки защиты подключения, список загружаемых коллекций и так далее.
- **Доступ** (Access) – пользователи-владельцы и группы-владельцы репозитория.

Добавление внешнего репозитория

Для добавления записи о внешнем репозитории выполните следующие действия:

1. В окне **Внешние репозитории** (Remotes) нажмите кнопку *Добавить внешний репозиторий* (Add remote).
2. Заполните форму **Добавить новый внешний репозиторий** (Add new remote):
   - **Название** (Name) – название внешнего репозитория в Private Automation Hub.

     Важно: Будьте внимательны при выборе названия – его нельзя будет изменить.

   - **URL** – ссылка на внешний репозиторий.

- **Включить все зависимости при синхронизации коллекции.** (Include all dependencies when syncing a collection.) – если этот переключатель включен, при синхронизации коллекций с Private Automation Hub будут загружены также все необходимые зависимости. Если переключатель выключен, коллекции загружаются без зависимостей.
- **Токен** (Token) – токен для доступа к внешнему репозиторию.
- **URL единого входа** (SSO URL) – URL страницы единого входа, используемой для доступа к внешнему репозиторию.
- **Требования YAML** (YAML requirements) – список коллекций, которые Private Automation Hub должен загрузить из внешнего репозитория, например:

```
---
collections:
  - name: astra.ald_pro
  - name: astra.brest
  - name: astra.ceph
  - name: astra.nginx
```

- **Название учетной записи пользователя** (Username) – название учетной записи, используемой для доступа к внешнему репозиторию.
- **Пароль** (Password) – пароль учетной записи, используемой для доступа к внешнему репозиторию.

При нажатии на ссылку *Показать дополнительные параметры* (Show advanced options) в форме появляются дополнительные поля.

- **URL прокси-сервера** (Proxy URL) – IP-адрес или FQDN используемого прокси-сервера.
- **Название учетной записи в прокси-сервере** (Proxy username) и **Пароль прокси-сервера** (Proxy password) – учетные данные пользователя, которые следует использовать для авторизации на прокси-сервере.
- **Проверка TLS** – если флаг включен, выполняется одноранговая проверка сертификата TLS, используемого для защиты подключения к внешнему репозиторию.
- **Ключ клиента** (Client key) – файл с приватным ключом клиента.
- **Сертификат клиента** (Client certificate) – файл сертификата клиента.
- **Сертификат CA** (CA certificate) – файл сертификата CA.
- **Параллельность загрузок** (Download concurrency) – максимальное количество параллельных загрузок, которые Private Automation Hub может запустить при работе с внешним репозиторием.
- **Ограничение скорости** (Rate Limit) – ограничение на количество запросов в секунду при работе Private Automation Hub с внешним репозиторием.

3. Нажмите кнопку *Сохранить* (Save).

Редактирование основных сведений о внешнем репозитории

Для изменения основной информации о внешнем репозитории выполните следующие действия:

1. В таблице со списком внешних репозиториев нажмите на ссылку с названием нужного внешнего репозитория.
2. Нажмите кнопку *Редактировать* (Edit).
3. Заполните форму сведений о внешнем репозитории.
4. Нажмите кнопку *Сохранить*.

Управление владельцами

Владельцами внешнего репозитория могут быть пользователи и группы пользователей. Порядок настройки одинаков.

1. В таблице со списком внешних репозиториев нажмите на ссылку с названием нужной записи.
2. Выберите вкладку **Доступ** (Access).
3. Чтобы назначить владельцем репозитория пользователя или группу, нажмите кнопку *Выбрать пользователя* (Select a user) или *Выбрать группу* (Select a group) соответственно.
4. В открывшемся окне выберите нужного пользователя или группу и нажмите кнопку *Далее* (Next).
5. Во вкладке **Выбрать роль (роли)** (Select role(s)) включите флаг роли `galaxy.collection_remote_owner` и нажмите кнопку **Далее** (Next).
6. Во вкладке **Предварительный просмотр** (Preview) проверьте корректность настроек и нажмите кнопку *Добавить* (Add).

Удаление внешнего репозитория

Примечание: При удалении записи о внешнем репозитории загруженные из него коллекции остаются в Private Automation Hub. Для удаления коллекции следуйте *соответствующей инструкции*.

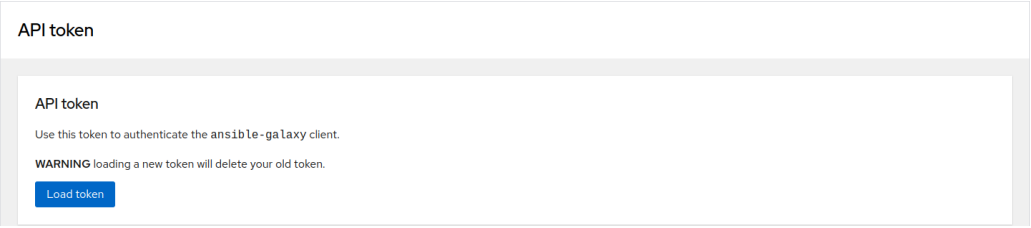
Для удаления записи о внешнем репозитории выполните следующие действия:

1. В таблице записей о внешних репозиториях нажмите на ссылку с названием удаляемого внешнего репозитория.
2. Нажмите кнопку *Удалить*.
3. Подтвердите удаление.

Токен API

Окно **Токен API** (API token) используется для создания токена, позволяющего авторизоваться в Private Automation Hub.

Для перехода в окно **Токен API** (API token) выберите на панели навигации *Коллекции* ▸ *Токены API* (*Collections* ▸ *API token*).



Важно: Период действия токена по умолчанию – 24 часа. По истечении указанного времени его нужно будет создать заново.

При создании нового токена предыдущий перестает действовать немедленно.

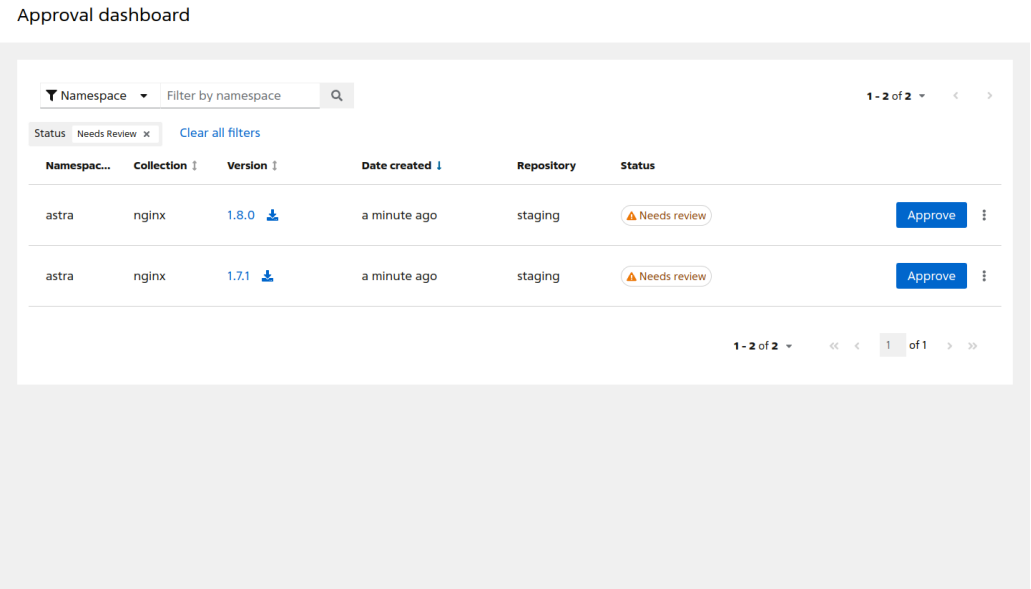
Для создания токена выполните следующие действия:

1. Нажмите кнопку *Загрузить токен* (Load token).

2. Скопируйте токен в буфер обмена и сохраните в надежном месте – это единственная возможность его увидеть. В случае утери токена его придется создать заново.
3. Настройте окружение на использование созданного токена согласно *инструкции*.

Согласование

Чтобы загруженная в Private Automation Hub версия коллекции стала доступна пользователям, она должна быть *согласована*. Для просмотра версий коллекций, ожидающих согласования, их согласования или отклонения выберите на панели навигации пункт *Коллекции* ▸ *Согласование* (*Collections* ▸ *Approval*).



Окно состоит из панели фильтрации и таблицы версий коллекций, ожидающих согласования. Таблица версий коллекций состоит из следующих колонок:

- **Пространство имен** (Namespace) – название *пространства имен*, к которому относится коллекция.
- **Коллекция** (Collection) – название коллекции.
- **Версия** (Version) – ссылка с номером версии коллекции. В этой же колонке находится ссылка на загрузку архива с версией коллекции.
- **Дата создания** (Date created) – количество времени с момента создания репозитория.
- **Репозиторий** (Repository) – название *репозитория*, в котором сейчас находится версия коллекции.
- **Статус** (Status) – текущий статус версии коллекции.
- Кнопка *Согласовать* (Approve) для согласования версии коллекции и кнопка ⋮ для вызова меню дополнительных действий.

Просмотр

Перед согласованием коллекции рекомендуется просмотреть сведения о ней. Эту задачу можно выполнить несколькими способами.

Загрузка кода коллекции на локальный компьютер

Чтобы изучить исходный код коллекции, загрузите его на локальный компьютер.

1. Нажмите на ссылку в колонке **Версия** (Version).
2. Сохраните архив на локальный компьютер.
3. Создайте каталог для распаковки архива, например:

```
mkdir astra-nginx-1.8.0/
```

4. Распакуйте содержимое архива в созданный каталог:

```
tar xf astra-nginx-1.8.0.tar.gz -C ./astra-nginx-1.8.0/
```

Просмотр основных сведений о версии коллекции

Чтобы просмотреть общую информацию о версии коллекции до ее согласования, нажмите на ссылку в колонке **Версия** (Version).

Окно просмотра сведений о коллекции состоит из выпадающего списка, позволяющего выбрать версию, и следующих вкладок:

- **Установить** (Install) – инструкция по установке коллекции с использованием утилиты `ansible-galaxy`.
- **Документация** (Documentation) – документация коллекции, входящих в нее ролей и модулей.
- **Содержимое** (Contents) – краткая информация о модулях и ролях, предоставляемых коллекцией.
- **Журнал импорта** (Import log) – журнал импорта коллекции в Private Automation Hub.
- **Зависимости** (Dependencies) – список зависимостей коллекции.
- **Источники контента** (Distributions) – список репозиториев, в которых размещена коллекция, а также краткая инструкция по настройке `ansible-galaxy` для их использования.

Согласование и отклонение версий коллекции

Чтобы согласовать версию коллекции, нажмите кнопку *Согласовать* (Approve) в соответствующей строке. Версия коллекции будет перемещена в соответствующий репозиторий. Если в настройках репозитория включено создание дистрибутива, коллекция будет доступна для поиска и загрузки пользователями.

Чтобы отклонить версию коллекции, в соответствующей строке нажмите кнопку ⋮ и в открывшемся меню выберите пункт **Отклонить** (Reject).

Важно: При отклонении версия коллекции не удаляется из Private Automation Hub, а перемещается в защищенный репозиторий `rejected`. Для окончательного удаления версии коллекции *удалите* ее из этого репозитория.

Среды исполнения

Раздел **Среды исполнения** (Execution environments) содержит подразделы, используемые для управления образами среды исполнения и записями о реестрах таких образов.

Среды исполнения

Окно **Среды исполнения** (Execution Environments) позволяет выполнять следующие действия:

- *просматривать сведения о средах исполнения*;
- *создавать записи о средах исполнения*;
- *изменять основную информацию о средах исполнения*;
- *изменять описания сред исполнения*;
- *запускать синхронизацию сред исполнения с внешними реестрами*;
- *удалять загруженные образы сред исполнения*;
- *настраивать доступ пользователей и групп к средам исполнения*.

Для перехода в окно **Среды исполнения** (Execution Environments) выберите на панели навигации *Среды исполнения* ▸ *Среды исполнения* (*Execution Environments* ▸ *Execution Environments*).

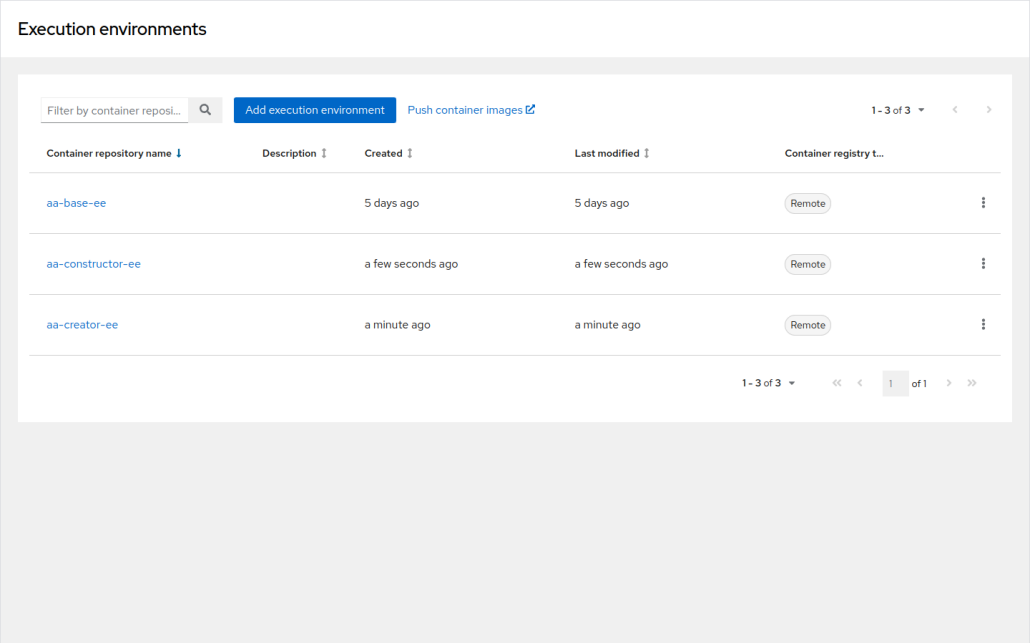


Таблица сред исполнения состоит из следующих колонок:

- **Название репозитория контейнера** (Container repository name) – общее название среды исполнения в Private Automation Hub. Например, `aa-base-ee` – общее название для всех образов среды исполнения, созданных ПАО Группа Астра для использования в Automation Controller.

  При нажатии на ссылку происходит переход в окно просмотра подробной информации о среде исполнения.
- **Описание** (Description) – краткое описание среды исполнения.
- **Создан** (Created) – время, прошедшее с момента создания записи о среде исполнения.
- **Последнее изменение** (Last modified) – время, прошедшее с момента последнего изменения записи о среде исполнения.

- **Тип реестра контейнеров** (Container registry type) – тип источника образов среды исполнения:
  - **Удаленный** (Remote) – образы загружаются в Private Automation Hub из стороннего реестра;
  - **Локальный** (Local) – образы загружаются в Private Automation Hub напрямую, например, с рабочих мест разработчиков образов.
- Кнопка ⋮ — вызов меню дополнительных действий.

Просмотр

Для получения подробных сведений о среде исполнения нажмите на ссылку с ее названием в таблице сред исполнения. Окно просмотра сведений о среде исполнения состоит из следующих вкладок:

- **Подробность** (Detail) – ссылка на загрузку образа из Private Automation Hub и инструкция по его использованию (при наличии).
- **Действие** (Activity) – история действий со средой исполнения. Здесь выводятся краткие сведения о добавленных и удаленных тегах, загрузке образов и так далее.
- **Образы** (Images) – таблица версий образов среды исполнения, состоит из следующих колонок:
  - **Метка** (Tag) – метка, указывающая версию образа, например, `0.6.4` или `latest`.
  - **Опубликован** (Published) – время, прошедшее с момента публикации образа в реестре Private Automation Hub.
  - **Слои** (Layers) – количество слоев в образе.
  - **Размер** (Size) – размер образа.
  - **Обработанные данные** (Digest) – тип и значение хеша образа.
  - Ссылка на загрузку образа из реестра Private Automation Hub.
  - Кнопка ⋮ – вызов меню дополнительных действий.
- **Доступ** (Access) – списки пользователей и групп, имеющих привилегии на управление записью о среде исполнения.

Создание среды исполнения

Для создания записи о среде исполнения выполните следующие действия:

1. В окне **Среды исполнения** (Execution Environments) нажмите кнопку *Добавить среду исполнения* (Add execution environment).
2. Заполните форму **Добавить среду исполнения** (Add execution environment):
   - **Название** (Name) – название среды исполнения, например, `aa-base-ee`.

     Важно: Будьте внимательны при выборе названия – его нельзя будет изменить. Название среды исполнения используется при формировании ссылок на загрузку образов из Private Automation Hub.

     Требования к названию среды исполнения:
     - Допускается использование строчных букв латинского алфавита, цифр, дефиса, подчеркивания и точки.
     - Первый и последний символы – не дефис.
   - **Название вышестоящего (upstream) объекта** (Upstream name) – название раздела во внешнем реестре образов, например, `aa/aa-base-ee`.

- **Реестр** (Registry) – внешний реестр образов.

  Значение в этом поле выбирается из справочника *внешних реестров*.

- **Добавить тег(и) для включения** (Add tag(s) to include) – список тегов, образы с которыми необходимо загрузить из внешнего реестра, например, `0.6.4`, `latest`. Возможна загрузка версий образов с использованием шаблона, например, `0.4.*`.
- **Добавить тег(и) для исключения** (Add tag(s) to exclude) – список тегов, образы с которыми не нужно загружать из внешнего реестра. Возможно исключение версий с использованием шаблона, например, `dev-*`.
- **Описание** – это поле доступно для изменения только при редактировании существующей записи о среде исполнения.

3. Нажмите кнопку *Сохранить* (Save).
4. Если источником образов служит внешний реестр, *запустите синхронизацию образов*.

Изменение основной информации о среде исполнения

Для изменения основной информации о среде исполнения выполните следующие действия:

1. В таблице со списком сред исполнения нажмите на ссылку с названием нужной среды исполнения.
2. Нажмите кнопку *Редактировать* (Edit) в строке с названием среды исполнения.
3. Заполните форму **Редактировать среду исполнения** (Edit execution environment).
   - **Название вышестоящего (upstream) объекта** (Upstream name) – название раздела во внешнем реестре образов, например, `aa/aa-base-ee`.
   - **Реестр** (Registry) – внешний реестр образов.

     Значение в этом поле выбирается из справочника *внешних реестров*.

   - **Добавить тег(и) для включения** (Add tag(s) to include) – список тегов, образы с которыми необходимо загрузить из внешнего реестра, например, `0.6.4`, `latest` и тому подобное. Возможна загрузка версий образов с использованием шаблона, например, `0.4.*`.
   - **Добавить тег(и) для исключения** (Add tag(s) to exclude) – список тегов, образы с которыми не нужно загружать из внешнего реестра. Возможно исключение версий с использованием шаблона, например, `dev-*`.
   - **Описание** – описание среды исполнения.
4. Нажмите кнопку *Сохранить* (Save).
5. *Запустите синхронизацию образов с внешним реестром*.

Описание среды исполнения

Если у вас есть описание среды исполнения, например, руководство по ее использованию, для его размещения в Private Automation Hub выполните следующие действия:

1. В таблице со списком сред исполнения нажмите на ссылку с названием нужной среды исполнения.
2. Во вкладке **Подробность** (Detail) нажмите кнопку *Добавить* (Add).
3. Добавьте в поле **Необработанный Markdown** (Raw Markdown) инструкцию по использованию образа среды исполнения в формате Markdown.
4. Нажмите кнопку *Сохранить*.

Синхронизация с внешним реестром

Чтобы запустить синхронизацию образов среды исполнения с внешним реестром, выполните следующие действия:

1. В таблице со списком сред исполнения нажмите кнопку ⋮ в строке с нужной записью.
2. В открывшемся меню выберите **Синхронизация из реестра** (Sync from registry).

Удаление образов среды исполнения

Примечание: После удаления образов *измените список тегов* в основных данных о среде исполнения. В противном случае удаленные образы будут загружены заново при первой же синхронизации.

Для удаления ненужных образов среды исполнения выполните следующие действия:

1. В таблице со списком сред исполнения нажмите на ссылку с названием нужной записи.
2. Выберите вкладку **Образы** (Images).
3. В строке с удаляемым образом нажмите кнопку ⋮ и в открывшемся меню выберите **Удалить** (Delete).
4. Включите флаг **Я понимаю, что это действие не может быть отменено.** (I understand that this action cannot be undone.) и нажмите кнопку *Удалить* (Delete).

Настройка доступа

Доступ к среде исполнения может быть предоставлен отдельным пользователям и группам. Порядок настройки одинаков.

1. В таблице со списком сред исполнения нажмите на ссылку с названием нужной записи.
2. Выберите вкладку **Доступ** (Access).
3. Чтобы предоставить доступ к среде исполнения пользователю или группе нажмите кнопку *Выбрать пользователя* (Select a user) или *Выбрать группу* (Select a group) соответственно.
4. В открывшемся окне выберите нужного пользователя или группу и нажмите кнопку *Далее* (Next).
5. Во вкладке **Выбрать роль (роли)** (Select role(s)) выберите роли, назначаемые пользователю или группе:

Роль	Предоставляемые привилегии
`galaxy.execution_environment_collabor`	• Изменение основных данных среды исполнения • Изменение списка тегов среды среды исполнения • Публикация загруженных образов в реестре Private Automation Hub
`galaxy.execution_environment_namespac`	• Изменение разрешений на доступ к пространству имен среды исполнения • Изменение основных данных среды исполнения • Изменение списка тегов среды среды исполнения • Публикация загруженных образов в реестре Private Automation Hub • Загрузка и публикация в Private Automation Hub новых образов среды исполнения • Управление ролями в пространстве имен образа среды исполнения
`galaxy.execution_environment_publishe`	• Изменение разрешений на доступ к пространству имен среды исполнения • Изменение основных данных среды исполнения • Изменение списка тегов среды среды исполнения • Публикация загруженных образов в реестре Private Automation Hub • Загрузка и публикация в Private Automation Hub новых образов среды исполнения • Управление ролями в пространстве имен образа среды исполнения • Создание записей о средах исполнения

6. Нажмите кнопку *Далее* (Next).
7. Во вкладке **Предварительный просмотр** (Preview) проверьте корректность настроек и нажмите кнопку *Добавить* (Add).

Внешние реестры

Окно **Внешние реестры** (Remote registries) позволяет выполнять следующие действия:

- *добавлять записи о внешних реестрах контейнеров*;
- *синхронизировать списки образов с внешними реестрами контейнеров*;
- *удалять записи о внешних реестрах контейнеров*.

Для перехода в окно **Внешние реестры** (Remote registries) выберите на панели навигации *Среды исполнения* ▸ *Внешние реестры* (*Execution Environments* ▸ *Remote Registries*).

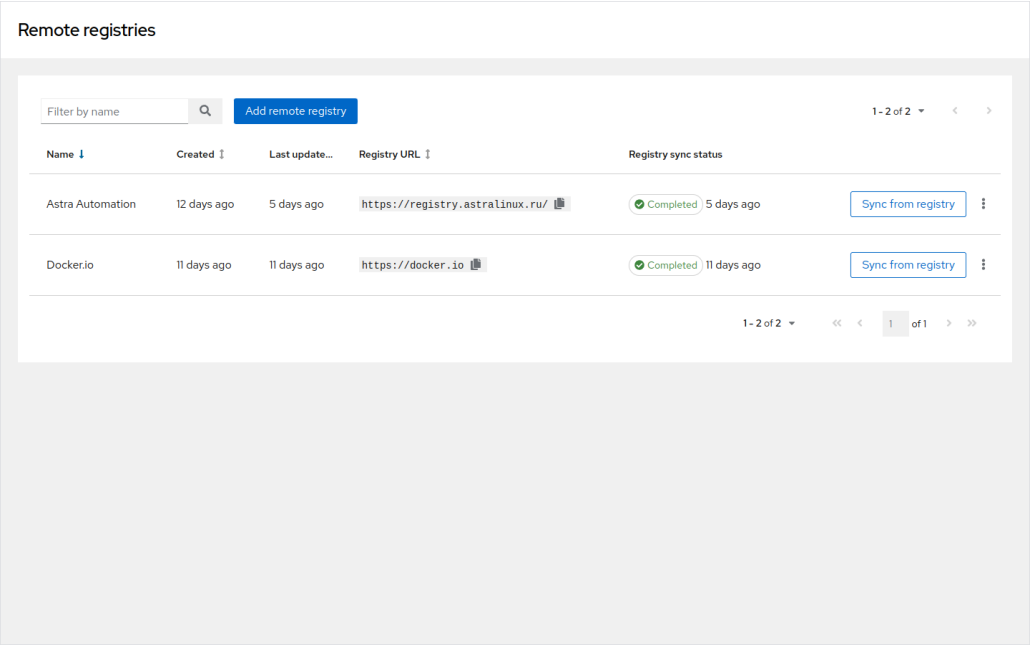


Таблица записей о внешних реестрах состоит из следующих колонок:

- **Название** (Name) – название внешнего реестра.
- **Создан** (Created) – время, прошедшее с момента создания записи о внешнем реестре.
- **Последнее обновление** (Last update) – время, прошедшее с момента последней синхронизации Private Automation Hub с внешним реестром.
- **Статус синхронизации реестра** (Registry sync status) – краткие сведения об успешности последней синхронизации Private Automation Hub с внешним реестром.
- Кнопка *Синхронизация реестра* (Sync from registry) – запускает синхронизацию Private Automation Hub с внешним реестром.
- Кнопка ⋮ – вызов меню дополнительных действий.

Добавление внешнего реестра

Для добавления записи о внешнем реестре выполните следующие действия:

1. В окне **Внешние реестры** (Remote registries) нажмите кнопку *Добавить внешний реестр* (Add remote registry).
2. В открывшемся окне **Добавить внешний реестр** (Add remote registry) заполните форму:
   - **Название** (Name) – произвольное название внешнего реестра. Должно быть уникальным на уровне Private Automation Hub.

Важно: Будьте внимательны при выборе названия внешнего реестра – его нельзя будет изменить.

- **URL** – ссылка на внешний реестр. Должна начинаться с указания используемого протокола – `http://` или `https://`, например:

```
https://registry.astralinux.ru/
```

- **Название учетной записи пользователя** (Username) и **Пароль** (Password) – учетные данные пользователя, которые следует использовать для авторизации при доступе к внешнему реестру.

При нажатии на ссылку *Показать дополнительные параметры* (Show advanced options) в форме появляются дополнительные поля.

- **URL прокси-сервера** (Proxy URL) – IP-адрес или FQDN используемого прокси-сервера.
- **Название учетной записи в прокси-сервере** (Proxy username) и **Пароль прокси-сервера** (Proxy password) – учетные данные пользователя, которые следует использовать для авторизации на прокси-сервере.
- **Проверка TLS** – если флаг включен, выполняется одноранговая проверка сертификата TLS, используемого для защиты подключения к внешнему реестру.
- **Ключ клиента** (Client key) – файл с приватным ключом клиента.
- **Сертификат клиента** (Client certificate) – файл сертификата клиента.
- **Сертификат CA** (CA certificate) – файл сертификата CA.
- **Параллельность загрузок** (Download concurrency) – максимальное количество параллельных загрузок, которые Private Automation Hub может запустить при работе с внешним реестром.
- **Лимит ставки** (Rate Limit) – ограничение на количество запросов в секунду при работе Private Automation Hub с внешним реестром.

3. Нажмите кнопку *Сохранить* (Save).

Синхронизация с внешним реестром образов

Чтобы синхронизировать хранящиеся в Private Automation Hub образы среды исполнения, связанные с *внешним реестром образов*, выполните следующие действия:

1. В таблице записей о внешних реестрах нажмите кнопку *Синхронизация из реестра*.
2. Дождитесь перехода записи о внешнем реестре в статус **Завершена** (Completed).

Удаление внешнего реестра

Примечание: При удалении записи о внешнем реестре загруженные из него образы среды исполнения остаются в Private Automation Hub. Инструкции по удалению образов приведены *здесь*.

Для удаления записи о внешнем реестре выполните следующие действия:

1. В таблице записей о внешних реестрах нажмите кнопку ⋮ в нужной строке.
2. В открывшемся меню выберите **Удалить** (Delete).
3. Подтвердите удаление.

Управление задачами

Окно **Управление задачами** (Task management) позволять выполнять следующие действия:

- *просматривать сведения о выполняющихся и выполненных задачах*;
- *останавливать выполняющиеся задачи*.

Для перехода в окно **Управление задачами** (Task management) выберите одноименный пункт на панели навигации.

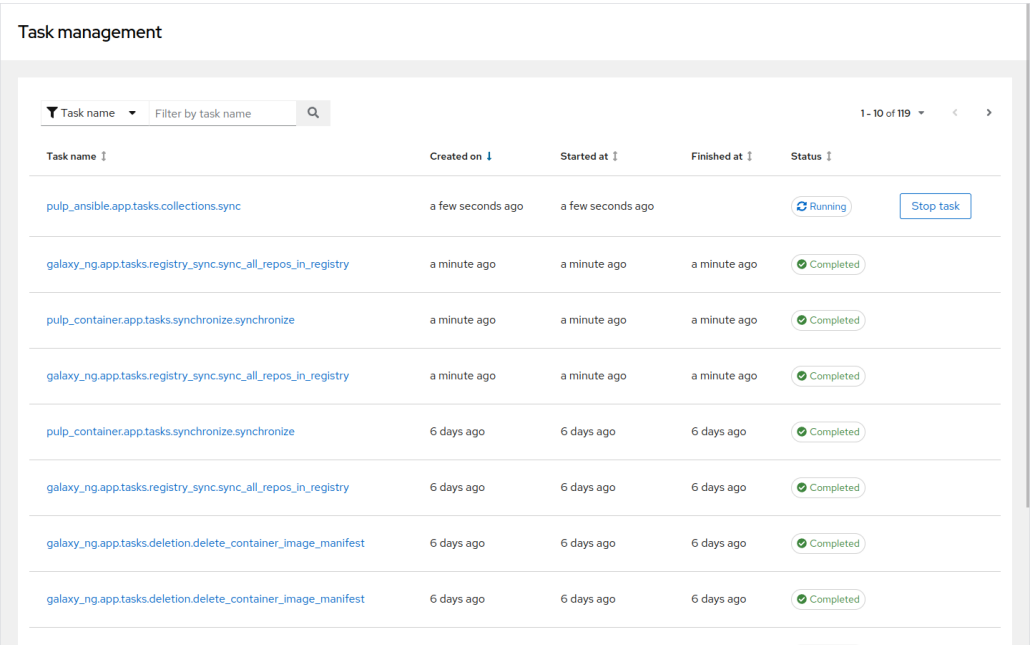


Таблица записей о задачах состоит из следующих колонок:

- **Название задачи** (Task name).

  При нажатии на ссылку происходит переход в окно просмотра подробных сведений о задаче.
- **Создано** (Created on) – время, прошедшее с момента создания задачи.
- **Начата** (Started at) – время, прошедшее с момента запуска задачи.
- **Закончена** (Finished at) – время, прошедшее с момента выполнения задачи.
- **Статус** (Status) – текущий статус задачи.
- Кнопка *Остановить задачу* (Stop task).

  Выводится только в строках с заданиями в статусе **Работающий** (Running).

Просмотр

Для получения подробных сведений о задаче нажмите на ссылку с ее названием в таблице задач. Окно просмотра сведений о задаче состоит из следующих панелей:

- **Подробная информация о задаче** (Task detail) – внутреннее и человекочитаемое названия, а также время, прошедшее с моментов запуска и завершения задачи.
- **Рабочие группы** (Task groups) – группы, к которым относится задача, ссылка на родительскую задачу и дочерние задачи.
- **Зарезервировать ресурсы** (Reserve resources) – типы связанных ресурсов, используемое расширение и название ресурса.
- **Сообщения о ходе исполнения** (Progress messages) – сведения о ходе выполнения задачи.

Остановка

Чтобы остановить выполнение задачи, выполните следующие действия:

1. В таблице задач нажмите кнопку *Остановить задачу* (Stop task) в строке с соответствующим заданием.
2. Подтвердите остановку задачи.

Доступ пользователя

Раздел **Доступ пользователя** (User Access) содержит подразделы, используемые для управления пользователями, группами и ролями пользователей.

Пользователи

Окно **Пользователи** (Users) используется для управления пользователями. Для перехода к нему выберите на панели навигации *Доступ пользователя* ▸ *Пользователи* (*User Access* ▸ *Users*).

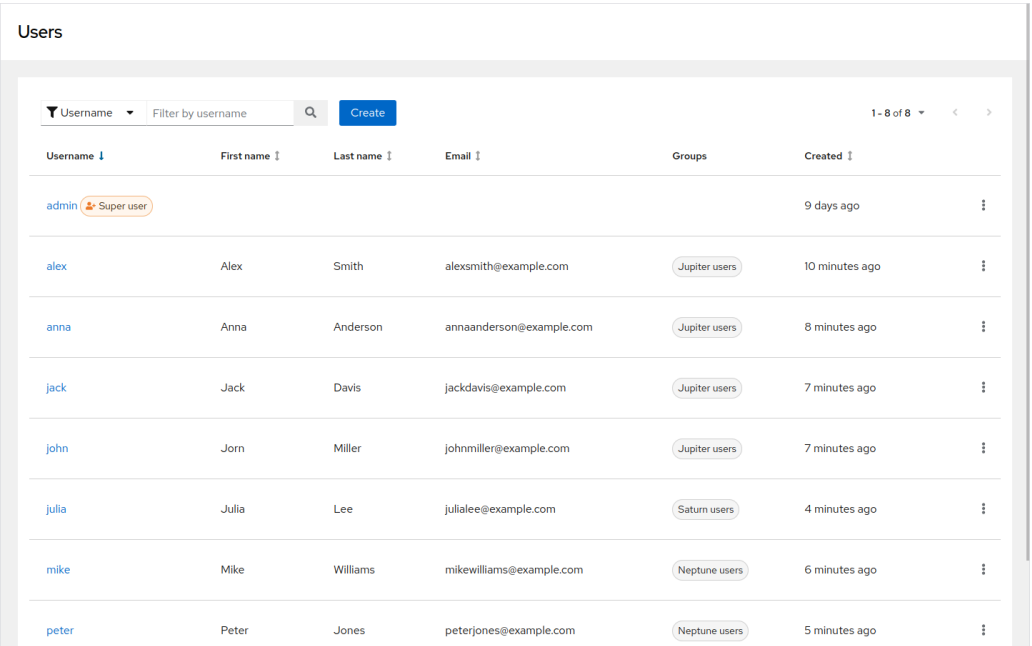


Таблица пользователей состоит из следующих колонок:

- **Название учетной записи** (Username) – ссылка с названием учетной записи, используемой для работы с Private Automation Hub. Если пользователь обладает флагом суперпользователя, в этой колонке выводится соответствующая отметка.

  При нажатии на ссылку происходит переход в окно просмотра подробной информации о пользователе.
- **Фамилия** (First name) – фамилия пользователя.
- **Имя** (Last name) – имя пользователя.
- **Электронная почта** (Email) – адрес электронной почты пользователя.
- **Группы** (Groups) – список групп, в которых состоит пользователь. При большом количестве групп выводятся названия только первых трех и ссылка, нажатие на которую отображает полный список.
- **Создан** (Created) – время, прошедшее с момента создания учетной записи пользователя.
- Кнопка быстрого вызова меню дополнительных действий:

- **Редактировать** (Edit) – переход в окно изменения свойств пользователя.
- **Удалить** (Delete) – удаление пользователя.

Создание пользователя

Для создания пользователя выполните следующие действия:

1. В окне **Пользователи** (Users) нажмите кнопку *Создать* (Create).
2. Заполните форму **Создать нового пользователя** (Create new user):
   - **Название учетной записи** – укажите название учетной записи пользователя.

     Требования к названию учетной записи:
     - Максимально допустимая длина – 150 символов.
     - Разрешено использование букв, цифр и символов `_`, `@`, `+`, `.`, `-`.
     - Уникальность на уровне Private Automation Hub.
   - **Фамилия** (First name) – фамилия пользователя.
   - **Имя** (Last name) – имя пользователя.
   - **Адрес электронной почты** (Email) – адрес электронной почты пользователя.
   - **Пароль** (Password) и **Подтверждение пароля** (Password confirmation) – укажите пароль и подтверждение пароля пользователя.

     Требования к паролю:
     - Длина не менее 9 символов.
     - Наличие хотя бы одного специального символа (`!`, `@`, `$`, `%` и тому подобное).
   - **Группы** (Groups) – выберите группы, в которые должен быть включен пользователь.
   - **Тип пользователя** (User type) – если переключатель включен, пользователю будет присвоен флаг суперпользователя.
3. Нажмите кнопку *Сохранить* (Save).

Настройка доступа

Для изменения привилегий пользователя выполните следующие действия:

1. *Создайте группу* с необходимыми ролями пользователей.
2. В окне **Пользователи** (Users) нажмите на ссылку с названием учетной записи нужного пользователя.
3. Нажмите кнопку *Редактировать* (Edit).
4. Используя поле **Группы** (Groups), измените список групп, в которых состоит пользователь.
5. Нажмите кнопку *Сохранить* (Save).

Смена пароля

Чтобы изменить пароль пользователя, выполните следующие действия:

1. В таблице пользователей нажмите на ссылку с названием учетной записи нужного пользователя.
2. Нажмите кнопку *Редактировать* (Edit).
3. Укажите новый пароль в полях **Пароль** (Password) и **Подтверждение** (Password confirmation).

   Требования к паролю:

   - Длина не менее 9 символов.
   - Наличие хотя бы одного специального символа (`!`, `@`, `$`, `%` и тому подобное).
4. Нажмите кнопку *Сохранить* (Save).

Смена типа пользователя

Чтобы изменить тип пользователя, выполните следующие действия:

1. В таблице пользователей нажмите на ссылку с названием учетной записи нужного пользователя.
2. Нажмите кнопку **Редактировать** (Edit).
3. В поле **Тип пользователя** (User type) переведите переключатель в нужное положение.
4. Нажмите кнопку *Сохранить* (Save).

Удаление пользователя

Чтобы удалить пользователя, выполните следующие действия:

1. В таблице пользователей нажмите на ссылку с названием учетной записи удаляемого пользователя.
2. Нажмите кнопку *Удалить* (Delete).
3. Подтвердите удаление.

Группы

Окно **Группы** (Groups) используется для управления группами. Для перехода к нему выберите на панели навигации *Доступ пользователя* ‣ *Группы* (*User Access* ‣ *Groups*).

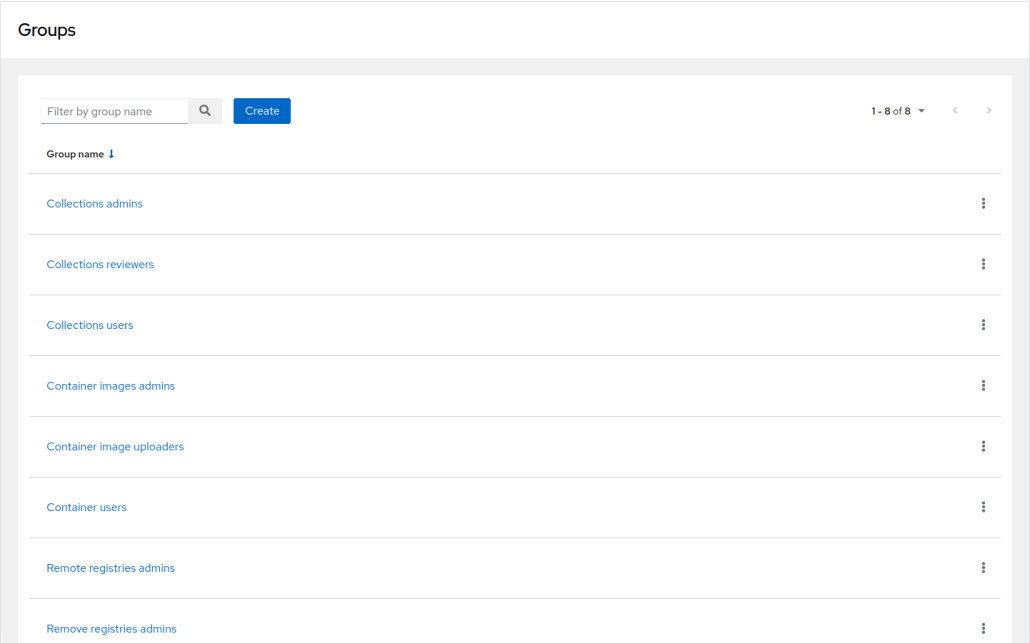


Таблица групп состоит из следующих колонок:

- **Название группы** (Group name) – ссылка с названием группы.

  При нажатии на ссылку происходит переход в окно просмотра подробной информации о группе.
- Кнопка вызова меню удаления группы.

Создание группы

Для создания группы выполните следующие действия:

1. В окне **Группы** (Groups) нажмите кнопку *Создать* (Create).
2. Заполните форму **Создать группу** (Create a group):

   В поле **Название** (Name) укажите название группы. Единственное требование к названию группы – уникальность на уровне Private Automation Hub.

   Важно: Будьте внимательны при выборе названия группы – его нельзя будет изменить.

3. Нажмите кнопку *Создать* (Create).
4. *Добавьте роли*.
5. *Добавьте пользователей*.

Управление ролями группы

Важно: Привилегии группы автоматически распространяются на всех ее участников.

Во вкладке **Доступ** (Access) выводится список связанных с группой пользовательских ролей.

Таблица ролей состоит из следующих колонок:

- Переключатель подробности вывода информации о роли.

При нажатии на переключатель изменяется видимость списка привилегий, предоставляемых ролью.

- **Роль** (Role) – название роли.
- **Описание** (Description) – краткое описание роли.
- Кнопка вызова меню удаления роли.

Добавление ролей

Чтобы добавить в группу роли, выполните следующие действия:

1. Во вкладке **Доступ** (Access) нажмите кнопку *Добавить роли* (Add roles). Будет открыто окно **Добавить роли** (Add roles).
2. Во вкладке **Выбрать роль (роли)** включите флаги пользовательских ролей, которые будут автоматически назначаться участникам группы.
3. Нажмите кнопку *Далее* (Next).
4. Во вкладке **Предпросмотр** (Preview) проверьте итоговый список ролей.
5. Нажмите кнопку *Добавить* (Add).

Удаление ролей

Предупреждение: При удалении роли предоставляемые ей привилегии отзываются у всех пользователей группы.

Чтобы удалить роли, предоставляемые группой, выполните следующие действия:

1. Во вкладке **Доступ** (Access) нажмите кнопку вызова дополнительных действий в строке с удаляемой ролью.
2. В открывшемся меню выберите **Удалить роль** (Remove role).
3. Подтвердите удаление роли.

Добавление пользователей

Чтобы добавить в группу пользователей, выполните следующие действия:

1. Во вкладке **Пользователи** (Users) нажмите кнопку *Добавить* (Добавить). Будет открыто окно **Добавить выбранных пользователей в группу** (Add selected users to group).
2. В поле **Выбрать пользователей** (Select users) выберите пользователей, которых необходимо добавить в группу.
3. Нажмите кнопку *Добавить* (Add).

Удаление пользователей

Чтобы удалить пользователя из группы, выполните следующие действия:

1. Во вкладке **Пользователи** (Users) нажмите кнопку вызова дополнительных действий в строке с удаляемым пользователем.
2. В открывшемся меню выберите **Удалить** (Remove).
3. Подтвердите удаление роли.

Удаление группы

> **Предупреждение:** При удалении группы у пользователей отзываются все привилегии, предоставляемые ее ролями.

Чтобы удалить группу, выполните следующие действия:

1. В таблице групп нажмите на ссылку с названием удаляемой группы.
2. Нажмите кнопку *Удалить* (Delete).
3. Подтвердите удаление.

Роли

Окно **Роли** (Roles) используется для управления ролями. Для перехода к нему выберите на панели навигации *Доступ пользователя* ▸ *Роли* (*User Access* ▸ *Roles*).

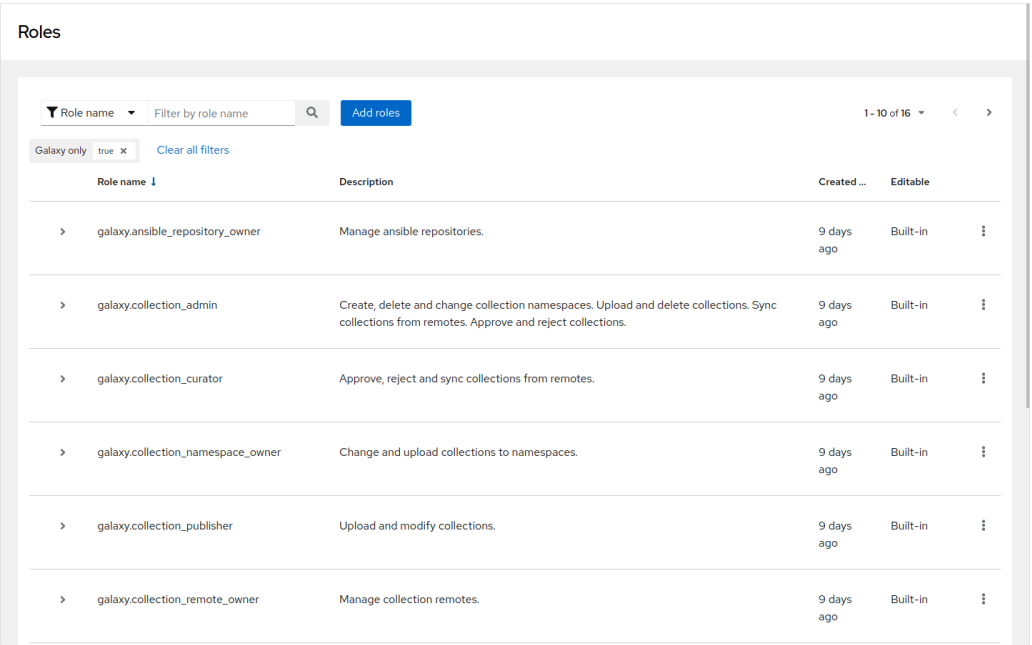


Таблица ролей состоит из следующих колонок:

- Переключатель подробности вывода информации о роли.

  При нажатии на переключатель изменяется видимость списка привилегий, предоставляемых ролью.
- **Название роли** (Role name) – название роли.
- **Описание** (Description) – краткое описание роли.
- **Создан** (Created) – количество времени, прошедшего с момента создания роли.

- **Редактируемый** (Editable) – возможность изменения свойств роли. Встроенные (built-in) роли нельзя изменить или удалить.
- Кнопка быстрого вызова меню дополнительных действий:
  - **Редактировать** (Edit) – переход в окно изменения свойств роли.
  - **Удалить** (Delete) – удаление роли.

Примечание: По умолчанию в таблице отображаются только роли с префиксом `galaxy.`. Для отображения всех существующих ролей нажмите на ссылку *Очистить все фильтры* (Clear all filters).

Создание роли

Для создания роли выполните следующие действия:

1. В окне **Роли** (Roles) нажмите кнопку *Добавить роли* (Add roles). Будет открыто окно **Создать новую роль** (Create a new role).
2. Заполните форму **Подробности** (Details):
   - **Название** (Name) – название роли.

     Требования к названию роли:
     - Начинается с префикса `galaxy.`.
     - Содержит только буквы, цифры латинского алфавита и символы подчеркивания.

     Важно: Будьте внимательны при выборе названия роли – его нельзя изменить.

   - **Описание** (Description) – краткое описание роли.
3. Заполните списки в форме **Привилегии** (Permissions):

Поле	Ресурс
Collection Namespaces	Пространства имен
Collections	Коллекции Ansible
Users	Пользователи Private Automation Hub
Groups	Группы Private Automation Hub
Collection Remotes	Внешние источники коллекций Ansible
Ansible Repository	Репозитории с кодом коллекций Ansible
Execution Environments	Среды исполнения
Container Registry Remotes	Внешние реестры образов
Task Management	Задачи Private Automation Hub

4. Нажмите кнопку *Сохранить* (Save).

Удаление роли

Предупреждение: При удалении роли у пользователей связанных с ней групп отзываются предоставляемые ролью привилегии.

Чтобы удалить роль, выполните следующие действия:

1. В таблице ролей нажмите на кнопку вызова меню дополнительных действий в строке с удаляемой ролью.
2. Выберите **Удалить** (Delete).
3. Подтвердите удаление.

6.5 Программный интерфейс (API)

Важно: Эта часть документации находится в стадии разработки.

6.6 Справочные данные

Встроенные в Private Automation Hub роли нельзя изменить или удалить. Набор предоставляемых встроенными ролями привилегий фиксирован.

6.6.1 galaxy.ansible_repository_owner

- `Add Ansible Repository` – создание репозиториев;
- `Change Ansible repository` – изменение репозиториев;
- `Delete Ansible repository` – удаление репозиториев;
- `Manage repository roles` – управление доступом к репозиториям;
- `Modify Ansible repo content` – управление содержимым репозиториев;
- `Repair Ansible repository` – восстановление репозиториев из резервных копий;
- `Sign collections` – подписывание коллекций;
- `View Ansible repository` – просмотр содержимого репозиториев.

6.6.2 galaxy.collection_admin

- `Add Ansible Repository` – создание репозиториев;
- `Add collection remote` – добавление записей о внешних репозиториях;
- `Add namespace` – создание пространств имен;
- `Change Ansible repository` – изменение репозиториев;
- `Change collection remote` – изменение записей о внешних репозиториях;
- `Change namespace` – изменение пространств имен;
- `Delete Ansible repository` – удаление репозиториев;
- `Delete collection` – удаление коллекций;
- `Delete collection remote` – удаление записей о внешних репозиториях;

- Delete namespace – удаление пространств имен;
- Manage remote roles – управление доступом к внешним репозиториям;
- Manage repository roles – управление доступом к репозиториям;
- Modify Ansible repo content – управление содержимым репозиториев;
- Repair Ansible repository – восстановление репозиториев из резервных копий;
- Sign collections – подписывание коллекций;
- Upload to namespace – загрузка версий коллекций в пространства имен;
- View Ansible repository – просмотр содержимого репозиториев;
- View collection remote – просмотр записей о внешних репозиториях.

6.6.3 galaxy.collection_curator

- Add Ansible Repository – создание репозиториев;
- Add collection remote – добавление записей о внешних репозиториях;
- Change Ansible repository – изменение репозиториев;
- Change collection remote – изменение записей о внешних репозиториях;
- Delete Ansible repository – удаление репозиториев;
- Delete collection remote – удаление записей о внешних репозиториях;
- Manage remote roles – управление доступом к внешним репозиториям;
- Manage repository roles – управление доступом к репозиториям;
- Modify Ansible repo content – управление содержимым репозиториев;
- Repair Ansible repository – восстановление репозиториев из резервных копий;
- Sign collections – подписывание коллекций;
- View Ansible repository – просмотр содержимого репозиториев;
- View collection remote – просмотр записей о внешних репозиториях.

6.6.4 galaxy.collection_namespace_owner

- Change namespace – изменение пространств имен;
- Upload to namespace – загрузка версий коллекций в пространства имен.

6.6.5 galaxy.collection_publisher

- Add namespace – создание пространств имен;
- Change namespace – изменение пространств имен;
- Upload to namespace – загрузка версий коллекций в пространства имен.

6.6.6 galaxy.collection_remote_owner

- Add collection remote – добавление записей о внешних репозиториях;
- Change collection remote – изменение записей о внешних репозиториях;
- Delete collection remote – удаление записей о внешних репозиториях;
- Manage remote roles – управление доступом к внешним репозиториям;
- View collection remote – просмотр записей о внешних репозиториях.

6.6.7 galaxy.content_admin

- Add Ansible Repository – создание репозиториев;
- Add collection remote – добавление записей о внешних репозиториях;
- Add namespace – создание пространств имен;
- Add remote registry – создание записей о внешних репозиториях;
- Change Ansible repository – изменение репозиториев;
- Change collection remote – изменение записей о внешних репозиториях;
- Change container namespace permissions – изменение разрешений на доступ к пространствам имен образов среды исполнения;
- Change containers – изменение свойств образов среды исполнения;
- Change image tags – изменение меток образов среды исполнения;
- Change namespace – изменение пространств имен;
- Change remote registry – изменение записей о внешних реестрах;
- Create new containers – создание образов среды исполнения;
- Delete Ansible repository – удаление репозиториев;
- Delete collection – удаление коллекций;
- Delete collection remote – удаление записей о внешних репозиториях;
- Delete container repository – удаление репозиториев с образами среды исполнения;
- Delete namespace – удаление пространств имен;
- Delete remote registry – удаление записей о внешних реестрах;
- Manage container namespace roles – управление доступом к пространствам имен образов среды исполнения;
- Manage remote roles – управление доступом к внешним репозиториям;
- Manage repository roles – управление доступом к репозиториям;
- Modify Ansible repo content – управление содержимым репозиториев;
- Push new containers – загрузка новых образов среды исполнения;
- Push to existing containers – обновление существующих образов среды исполнения;
- Repair Ansible repository – восстановление репозиториев из резервных копий;
- Sign collections – подписывание коллекций;
- Upload to namespace – загрузка версий коллекций в пространства имен;
- View Ansible repository – просмотр содержимого репозиториев;
- View collection remote – просмотр записей о внешних репозиториях.

6.6.8 galaxy.execution_environment_admin

- Add remote registry – создание записей о внешних репозиториях;
- Change container namespace permissions – изменение разрешений на доступ к пространствам имен образов среды исполнения;
- Change containers – изменение свойств образов среды исполнения;
- Change image tags – изменение меток образов среды исполнения;
- Change remote registry – изменение записей о внешних реестрах;
- Create new containers – создание образов среды исполнения;
- Delete container repository – удаление репозиториев с образами среды исполнения;
- Delete remote registry – удаление записей о внешних реестрах;
- Manage container namespace roles – управление доступом к пространствам имен образов среды исполнения;
- Push new containers – загрузка новых образов среды исполнения;
- Push to existing containers – обновление существующих образов среды исполнения.

6.6.9 galaxy.execution_environment_collaborator

- Change containers – изменение свойств образов среды исполнения;
- Change image tags – изменение меток образов среды исполнения;
- Push to existing containers – обновление существующих образов среды исполнения.

6.6.10 galaxy.execution_environment_namespace_owner

- Change container namespace permissions – изменение разрешений на доступ к пространствам имен образов среды исполнения;
- Change containers – изменение свойств образов среды исполнения;
- Change image tags – изменение меток образов среды исполнения;
- Manage container namespace roles – управление доступом к пространствам имен образов среды исполнения;
- Push new containers – загрузка новых образов среды исполнения;
- Push to existing containers – обновление существующих образов среды исполнения.

6.6.11 galaxy.execution_environment_publisher

- Change container namespace permissions – изменение разрешений на доступ к пространствам имен образов среды исполнения;
- Change containers – изменение свойств образов среды исполнения;
- Change image tags – изменение меток образов среды исполнения;
- Create new containers – создание образов среды исполнения;
- Manage container namespace roles – управление доступом к пространствам имен образов среды исполнения;
- Push new containers – загрузка новых образов среды исполнения;
- Push to existing containers – обновление существующих образов среды исполнения.

6.6.12 galaxy.group_admin

- `Add group` – создание групп;
- `Change group` – изменение групп;
- `Delete group` – удаление групп;
- `View group` – просмотр групп.

6.6.13 galaxy.synclist_owner

- `galaxy.add_synclist` – создание списков синхронизации коллекций;
- `galaxy.change_synclist` – изменение списков синхронизации коллекций;
- `galaxy.delete_synclist` – удаление списков синхронизации коллекций;
- `galaxy.view_synclist` – просмотр списков синхронизации коллекций.

6.6.14 galaxy.task_admin

- `Change task` – изменение задач;
- `Delete task` – удаление задач;
- `View all tasks` – просмотр всех задач.

6.6.15 galaxy.user_admin

- `Add user` – создание пользователей;
- `Change user` – изменение пользователей;
- `Delete user` – удаление пользователей;
- `View user` – просмотр пользователей.

ГЛАВА 7

Контроллер

Automation Controller (контроллер) – центральный компонент платформы Astra Automation, который управляет автоматизацией различных процессов с помощью следующих встроенных механизмов:

- Управление через графический интерфейс пользователя, CLI, API и Ansible.
- Наблюдение за различными аспектами процессов выполнения playbook с использованием средств фильтрации.
- Упрощение запуска процессов с минимальным количеством действий.
- Развитая система безопасности, базирующаяся на ролевой модели и аудите.
- Масштабирование системы управления.
- Обеспечение надежности с помощью кластерных решений и встроенной системы резервного копирования и восстановления.
- Расширение возможностей за счет интеграции с различными репозиториями коллекций Ansible.
- Статическая и динамическая инвентаризация.
- Отправка уведомлений по электронной почте и в различные средства обработки, анализа и распространения сообщений (Grafana, IRC, Mattermost, Slack, Twilio и другие).
- Автоматическое распределение заданий по группам управляемых узлов для снижения пиковой нагрузки.
- И многое другое.

7.1 Особенности архитектуры

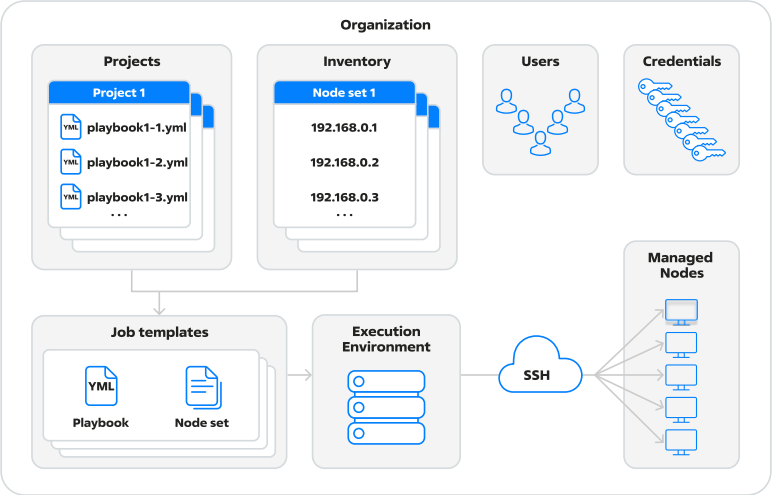
Важно: Эта часть документации находится в стадии разработки.

7.2 Структура управления

Для управления инфраструктурой контроллер использует следующие компоненты:

- playbooks;
- инвентарь (inventory);
- проекты (projects);
- шаблоны (templates);
- среды исполнения (execution environments);
- полномочия (credentials).

Взаимосвязи между компонентами показаны на схеме:



7.2.1 Проект

Проект состоит из одного или нескольких Ansible playbook. Контроллер позволяет импортировать существующие наборы playbook из различных источников:

- репозиторий Git;
- репозиторий Subversion;
- архивы форматов `.zip` и `.tar.gz`.

Также существующие наборы playbook можно вручную разместить в одном из каталогов локальной файловой системы контроллера, после чего указать путь к нужному каталогу в настройках проекта.

Подробности см. в документе *Проекты*.

7.2.2 Playbook

Для настройки управляемых узлов контроллер использует *Ansible playbook*.

7.2.3 Инвентарь

Контроллер предоставляет следующие возможности для управления *инвентарем Ansible*:

- заполнение списка управляемых узлов вручную или путем импорта из стороннего источника;
- использование статического и динамического инвентаря.

Подробности см. в разделе *Инвентарь*.

7.2.4 Шаблон

Шаблоны бывают двух типов:

- шаблоны заданий;
- шаблоны потоков заданий.

Шаблон задания связывает между собой набор управляемых узлов из инвентаря и один из playbook, доступных в проекте.

Шаблон потока заданий связывает между собой шаблоны заданий, а также предоставляет дополнительные функции, позволяющие настроить выполнение отдельных заданий при наступлении определенных условий.

Подробности о шаблонах см. в разделе *Шаблоны*.

7.2.5 Среда исполнения

Среда исполнения содержит все необходимое для запуска playbook – фиксированные версии ОС, приложений и библиотек, в том числе Ansible, Python и других. Использование среды исполнения позволяет сделать выполнение playbook предсказуемым и не зависящим от ОС, в которой развернут контроллер.

7.2.6 Организации, пользователи и команды

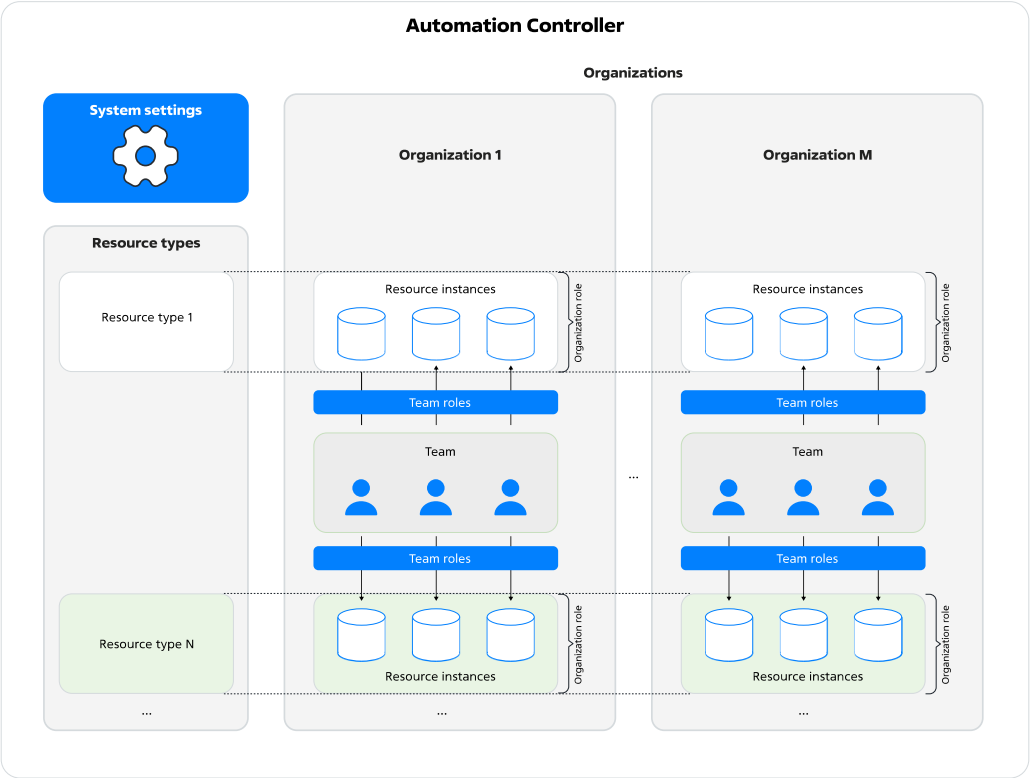
В Automation Controller используется управление доступом на основе ролей (*RBAC*), реализованное с помощью организаций, пользователей и команд.

Организационная структура

В Automation Controller используется разграничение доступа на основе ролей (*RBAC*). Набор привилегий пользователя на доступ к ресурсам определяется следующими параметрами:

- тип пользователя;
- назначенные роли.

Структура управления привилегиями показана на схеме:



Область действия предоставляемых ролью привилегий определяется уровнем, на котором она назначена:

1. Automation Controller – изменение настроек контроллера и доступ ко всем ресурсам и компонентам;
2. организация – все ресурсы организации;
3. команда – отдельные экземпляры ресурсов.

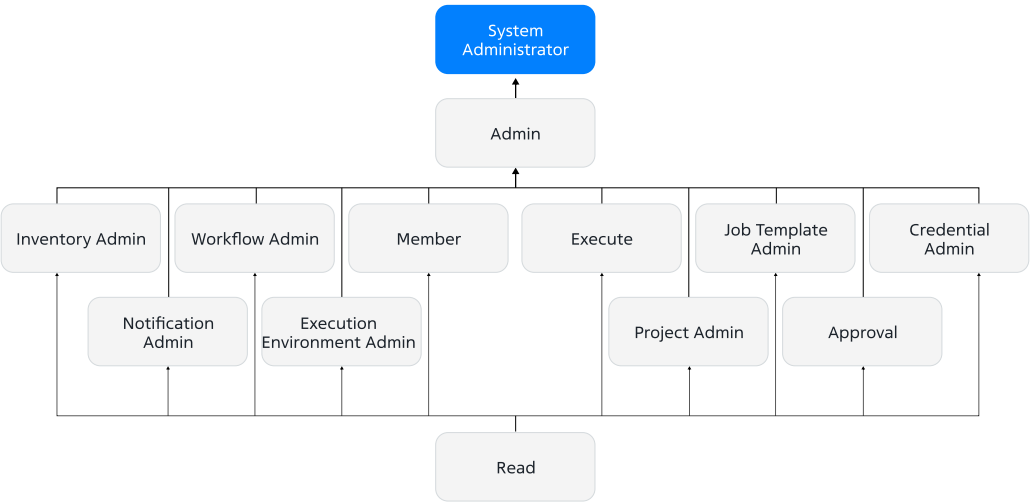
Роли

Роль – это набор привилегий на доступ к определенному ресурсу или типу ресурсов.

В Automation Controller роли имеют следующие особенности:

- Набор ролей фиксирован, создание дополнительных ролей не допускается.
- Роли имеют иерархическую структуру – при назначении пользователю родительской роли он получает все привилегии дочерних ролей.

Пример иерархии наследования ролей на уровне организации показан на схеме:



Здесь от роли «Read» наследуют свои привилегии следующие роли:

- Execute;
- Project Admin;
- Inventory Admin;
- Credential Admin;
- Workflow Admin;
- Notification Admin;
- Job Template Admin;
- Execution Environment Admin;
- Approval;
- Member.

Например, при назначении роли «Approval» пользователь получает также привилегии роли «Read», назначать ее отдельно нет необходимости.

Роль «Admin» наследует привилегии от всех ролей, указанных в списке выше – при ее назначении пользователь получает все привилегии дочерних ролей, в том числе «Read».

Роль «System Administrator» наследует привилегии от роли «Admin»: пользователь с ролью «System Administrator» получает все привилегии дочерних ролей.

Пользователи

Пользователю можно назначить различные роли, в том числе в разных организациях и командах.

Примечание: Чтобы пользователь считался участником организации или команды, ему должна быть назначена роль «Участник» (Member) на уровне организации или команды соответственно. Для получения привилегий на ресурсы назначение роли «Участник» не обязательно.

Типы пользователей

В Automation Controller доступны три типа пользователей (user types):

- системный администратор (System Administrator);
- системный аудитор (System Auditor);
- обычный пользователь (Normal User).

При необходимости тип пользователя можно изменить.

Системный администратор

При создании учетных записей пользователей этого типа им автоматически назначается системная роль «Системный администратор».

Системные администраторы имеют полный доступ ко всем компонентам и ресурсам контроллера, в том числе они имеют следующие привилегии:

- изменение системных настроек контроллера;
- создание, изменение и удаление следующих компонентов:
  - организаций;

  - команд;
  - пользователей, в том числе других системных администраторов;
- создание, изменение и удаление любых ресурсов.

При развертывании контроллера автоматически создается один пользователь типа «Системный администратор», который не входит ни в одну организацию. Имя учетной записи этого пользователя задается в файле `inventory` на этапе установки.

Системный аудитор

При создании учетных записей пользователей этого типа им автоматически назначается системная роль «Системный аудитор».

Системные аудиторы по умолчанию имеют привилегию на получение информации о любых ресурсах контроллера, а также создание полномочий.

Привилегии пользователей этого типа могут быть расширены путем назначения дополнительных ролей на уровне организаций и команд.

Обычный пользователь

При создании учетных записей пользователей этого типа им автоматически назначаются две роли:

- системная роль «Обычный пользователь» (Normal User);
- роль «Участник» (Member) на уровне указанной организации.

Привилегии пользователей этого типа расширяются путем назначения им дополнительных ролей на уровне организации и команды.

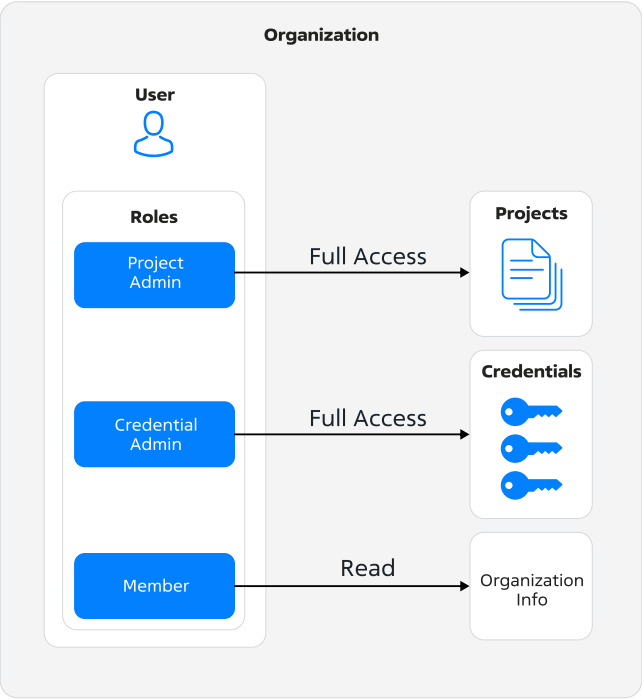
Организации

Организации используются для разграничения доступа к *типам ресурсов* контроллера.

На уровне организации роли могут быть назначены отдельным пользователям и командам.

Если на уровне организации роль назначена команде, то привилегии распространяются на всех членов команды, в том числе добавленных позднее.

Пусть пользователю на уровне организации назначены роли «Администратор проекта» (Project Admin), «Администратор учетных данных» (Credential Admin) и «Участник» (Member):



Назначенные роли предоставляют пользователю следующие привилегии на ресурсы организации:

- полный доступ ко всем проектам;
- полный доступ ко всем полномочиям;
- просмотр сведений об организации.

При развертывании контроллера в нем создается организация с названием «Default». Если другие организации отсутствуют, то все создаваемые ресурсы, пользователи и команды будут принадлежать этой организации.

Организационные роли

В таблице представлены привилегии, предоставляемые каждой организационной ролью на ресурсы организации.

Название роли	Может быть назначена пользователю	Может быть назначена команде	Описание
Администратор (Administrator)	Да	Нет	Полный доступ к ресурсам и свойствам организации
Администратор инвентаря (Inventory Admin)	Да	Да	Полный доступ к *инвентарю* и управляемым узлам
Администратор потока заданий (Workflow Admin)	Да	Да	Полный доступ к потокам заданий
Администратор проекта (Project Admin)	Да	Да	Полный доступ к *проектам*
Администратор среды исполнения (Execution Environment Admin)	Да	Да	Полный доступ к средам исполнения
Администратор уведомлений (Notification Admin)	Да	Да	Полный доступ к уведомлениям
Администратор учетных данных (Credential Admin)	Да	Да	Полный доступ к полномочиям
Администратор шаблона задания (Job Template Admin)	Да	Да	Полный доступ к *шаблонам заданий*
Аудитор (Auditor)	Да	Да	Получение сведений об организации и принадлежащих ей ресурсах
Выполнять (Execute)	Нет	Да	Запуск любых исполняемых ресурсов
Одобрить (Approve)	Да	Да	Подтверждение переходов между узлами потоков заданий
Участник (Member)	Да	Нет	Пользователь является членом организации
Чтение (Read)	Да	Да	Получение кратких сведений об организации

Связи с другими компонентами

Для некоторых компонентов связь с организацией обязательна:

Компонент	Связь с организацией
Шаблон задания	Обязательна Принадлежит организации через проект
Шаблон потока заданий	Не обязательна
Полномочия	Не обязательна
Проект	Обязательна
Инвентарь	Обязательна
Управляемый узел	Обязательна Принадлежит организации через инвентарь
Уведомления	Обязательна
Среда исполнения	Не обязательна
Задание	Не обязательна Задания связаны с организациями через шаблоны заданий
Команда	Обязательна
Пользователь	Обязательна при создании учетной записи При удалении организации связанные с ней пользователи сохраняются

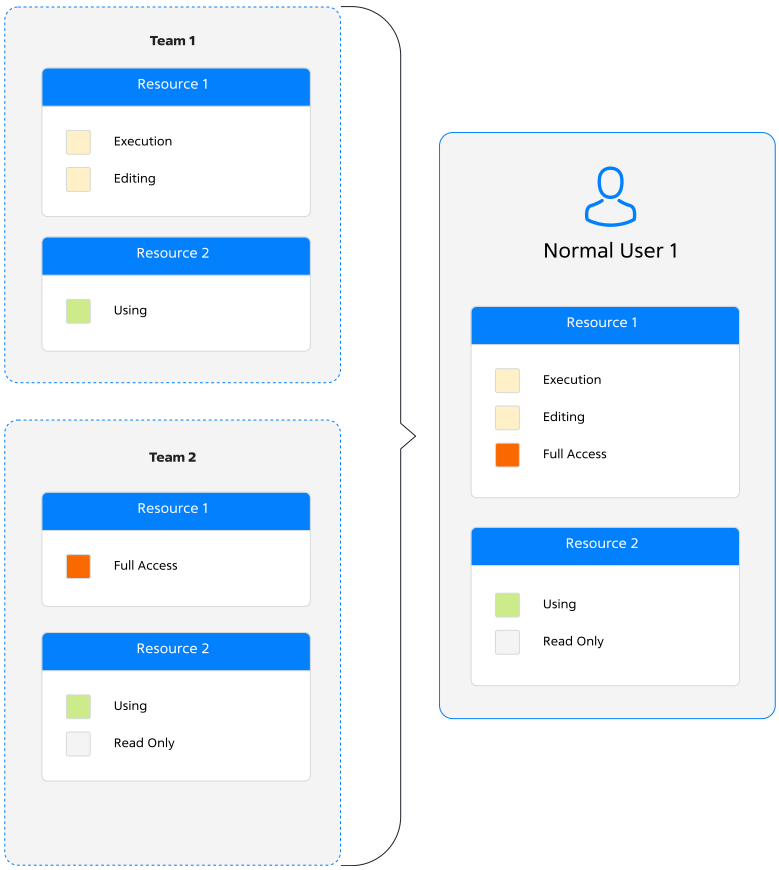
Команды

Команды существуют в рамках организаций и используются для разграничения доступа к *экземплярам ресурсов*. Это значит, что участники команды получают доступ не ко всем принадлежащим организации ресурсам определенного типа, а только к отдельным их экземплярам.

Например, если пользователю назначена командная роль «Администратор среды исполнения», то он может выполнять любые действия только с указанной средой исполнения: изменять, удалять и использовать ее.

Участниками команды могут быть пользователи из разных организаций.

Привилегии доступа к конкретному экземпляру ресурса определяются типом пользователя, а также объединением множеств привилегий, предоставляемых его пользовательскими и командными ролями во всех организациях и командах. Принцип объединения привилегий, предоставляемых ролями, показан на схеме:



Командные роли команды «Team 1» предоставляют к ресурсу «Resource 1» привилегии на исполнение (execution) и изменение (editing), а к ресурсу «Resource 2» – на использование (using). Командные роли команды «Team 2» предоставляют к ресурсу «Resource 1» привилегии полного доступа (full access), а к ресурсу «Resource 2» – на использование и только чтение (read only).

Все пользователи, входящие одновременно в команды «Team 1» и «Team 2», получают следующие привилегии на ресурсы:

- «Resource 1» – исполнение, изменение и полный доступ;
- «Resource 2» – использование и только чтение.

Командные роли

Командные роли определяют режим доступа **всех** участников команды к существующим ресурсам организации.

- шаблоны заданий;
- шаблоны потоков заданий;
- полномочия;
- инвентарь;
- проекты;
- организации;
- группы узлов контроллера.

На каждый ресурс можно назначить произвольную комбинацию доступных командных ролей.

Перечень доступных командных ролей определяется типом ресурса.

Шаблоны заданий (Job Templates)

Для предоставления привилегий на шаблоны заданий пользователям могут быть назначены следующие роли:

- Администратор (Admin) – полный доступ к шаблонам заданий.
- Выполнять (Execute) – запуск заданий из шаблона.
- Чтение (Read) – получение сведений о шаблонах заданий.

Шаблоны потоков заданий (Workflow Templates)

Для предоставления привилегий на шаблоны потоков заданий пользователям могут быть назначены следующие роли:

- Администратор (Admin) – полный доступ к шаблонам потоков заданий.
- Выполнять (Execute) – запуск шаблонов потоков заданий.
- Одобрить (Approval) – подтверждение перехода между заданиями в выбранных потоках заданий.
- Чтение (Read) – получение информации о шаблонах потоков заданий.

Полномочия (Credentials)

Для предоставления привилегий на полномочия пользователям могут быть назначены следующие роли:

- Администратор (Admin) – полный доступ к полномочиям.
- Использовать (Use) – использование указанных полномочий при создании ресурсов, например, проектов или шаблонов заданий.
- Чтение (Read) – получение сведений о полномочиях.

Инвентарь (Inventories)

Для предоставления привилегий на инвентарь пользователям могут быть назначены следующие роли:

- Администратор (Admin) – полный доступ к инвентарю.
- Использовать (Use) – использование инвентаря при создании шаблонов заданий и шаблонов потоков заданий.
- Обновление (Update) – обновление инвентарных списков, получаемых из внешних источников.
- Специальный (Ad Hoc) – запуск специальных (ad-hoc) команд на узлах из указанных инвентарных списков.
- Чтение (Read) – получение сведений об инвентарных списках.

Проекты (Projects)

Для предоставления привилегий на проекты пользователям могут быть назначены следующие роли:

- Администратор (Admin) – полный доступ к проектам.
- Использовать (Use) – использование проектов при создании шаблонов заданий.
- Обновление (Update) – изменение свойство проектов.
- Чтение (Read) – получение сведений о проектах.

Организации (Organizations)

Для предоставления привилегий на ресурсы организации пользователям могут быть назначены следующие роли:

- Администратор инвентаря (Inventory Admin) – полный доступ к инвентарю.
- Администратор потока заданий (Workflow Admin) – полный доступ к потокам заданий.
- Администратор проекта (Project Admin) – полный доступ к проектам.
- Администратор среды исполнения (Execution Environment Admin) – полный доступ к средам исполнения.
- Администратор уведомлений (Notification Admin) – полный доступ к уведомлениям.
- Администратор учетных данных (Credential Admin) – полный доступ к полномочиям.
- Администратор шаблона задания (Job Template Admin) – полный доступ к шаблонам заданий.
- Аудитор (Auditor) – получение сведений обо всех ресурсах организации.
- Выполнять (Execute) – запуск любых исполняемых ресурсов.
- Одобрить (Approve) – подтверждение перехода между заданиями потоков.
- Чтение (Read) – получение сведений об организации.

Группы узлов контроллера (Instance Groups)

Для предоставления привилегий на группы узлов контроллера пользователям могут быть назначены следующие роли:

- Администратор (Admin) – полный доступ к группам узлов.
- Использовать (Use) – использование групп узлов контроллера.
- Чтение (Read) – получение сведений о группах узлов контроллера.

Пользовательские роли

На уровне команды пользователю могут быть назначены индивидуальные роли, влияющие на доступ ко всем ресурсам команды:

- Администратор (Administrator) – полный доступ ко всем ресурсам команды.
- Участник (Member) – получение расширенных сведений об организации. Привилегии доступа к отдельным ресурсам определяются назначенными командными ролями.
- Чтение (Read) – получение сведений о команде.

Полномочия

Полномочия используются для доступа к внешним ресурсам и управления узлами. Полномочия могут содержать следующие данные:

- имена пользователей;
- пароли;
- токены;
- приватные ключи SSH и GPG;
- и другие данные, которые можно использовать для получения доступа к ресурсам и подключения к управляемым узлам.

Конфиденциальные данные, используемые полномочием, называются *секретами*. По умолчанию все секреты хранятся в базе данных Automation Controller. Также Automation Controller может использовать секреты, хранящиеся во *внешних системах управления секретами*.

При создании любых полномочий обязательно должны быть указаны их название и *тип*. Если ни один из существующих встроенных типов полномочий не подходит для выполнения поставленных задач, можно создать *собственный тип полномочий*.

Полномочие может быть связано с организацией, и тогда использовать его могут только пользователи этой организации. Если связи полномочия с организацией нет, то оно доступно для использования всеми пользователями контроллера.

Встроенные типы полномочий

В Automation Controller встроены следующие типы полномочий:

- *API-токен Ansible Galaxy/Automation Hub* (Ansible Galaxy/Automation Hub API Token);
- *Centrify Vault Credential Provider Lookup*;
- *CyberArk Central Credential Provider Lookup*;
- *CyberArk Conjur Secrets Manager Lookup*;
- *Google Compute Engine*;
- *Открытый ключ GPG* (GPG Public Key);
- *HashiCorp Vault Secret Lookup*;
- *HashiCorp Vault Signed SSH*;
- *Insights*;
- *Microsoft Azure Key Vault*;
- *OpenStack*;
- *Red Hat Satellite 6*;
- *Thycotic DevOps Secrets Vault*;
- *Thycotic Secret Server*;
- *Vault*;
- *VMware vCenter*;
- *Web-сервисы Amazon* (Amazon Web Services);
- *Виртуализация Red Hat* (Red Hat Virtualization);
- *Диспетчер ресурсов Microsoft Azure* (Microsoft Azure Resource Manager);
- *Личный токен доступа к GitHub* (GitHub Personal Access Token);
- *Машина* (Machine);

- *Платформа автоматизации Red Hat Ansible* (Red Hat Ansible Automation Platform);
- *Реестр контейнеров* (Container Registry);
- *Сеть* (Network);
- *Токен персонального доступа GitLab* (GitLab Personal Access Token);
- *Токен доступа OpenShift или Kubernetes API* (OpenShift or Kubernetes API Bearer Token);
- *Система управления версиями* (Source Control).

API-токен Ansible Galaxy/Automation Hub

Полномочия этого типа предоставляют доступ к репозиториям коллекций Ansible, таким, как Ansible Galaxy[71], *Automation Hub* и *Private Automation Hub*.

Они необходимы для загрузки и установки коллекций, которые требуются для проектов организации, при создании и синхронизации этих проектов. Для использования полномочия требуется выполнение следующих условий:

- наличие в корневом каталоге проекта файла `requirements.yml` со списком зависимостей;
- *связь* созданного полномочия с организацией.

Для создания полномочий этого типа необходимо предоставить следующие данные:

- URL репозитория.

  Важно: URL репозитория должен завершаться символом `/`.

  По умолчанию коллекции загружаются из репозитория `published`. В этом случае допускается указать сокращенный URL:

  ```
  https://hub.example.com/api/galaxy/
  ```

  Чтобы получить доступ к репозиторию, отличному от `published`, приведите URL к следующему виду:

  ```
  https://hub.example.com/api/galaxy/content/<repository>/
  ```

  Здесь <repository> – название репозитория в реестре коллекций.
- Токен доступа, созданный пользователем на стороне используемого реестра контента Ansible.
- Адрес сервера Keycloak для выдачи токена (при использовании SSO).

Чтобы использовать несколько репозиториев одного и того же реестра, создайте для каждого репозитория отдельное полномочие.

Centrify Vault Credential Provider Lookup

Полномочия этого типа используются для доступа к системе управления секретами Centrify Vault Credential Provider[72].

[71] https://galaxy.ansible.com/
[72] https://developer.delinea.com/docs/vault-functionality

CyberArk Central Credential Provider Lookup

Полномочия этого типа используются для доступа к системе управления секретами CyberArk Central Credential Provider[73]. Для использования данной интеграции требуется запущенный веб-сервис хранения секретов CyberArk Central Credential Provider.

Поддерживается защита подключения с помощью SSL.

CyberArk Conjur Secrets Manager Lookup

Полномочия этого типа используются для доступа к системе управления секретами CyberArk Conjur Secrets Manager[74].

Google Compute Engine

Полномочия этого типа используются для доступа к списку управляемых узлов в Google Compute Engine[75] с реквизитами сервисной учетной записи.

Для создания полномочий этого типа необходимо предоставить следующие данные о сервисной учетной записи Google Compute Cloud:

- адрес электронной почты;
- закрытый ключ RSA, связанный с указанным адресом электронной почты.

Совет: Для автоматического заполнения реквизитов сервисной учетной записи Google Compute Cloud можно использовать файл в формате JSON. Пошаговые инструкции по созданию такого файла см. в документации Google Compute Cloud[76].

Открытый ключ GPG

Полномочия этого типа используются для проверки подписи содержимого, полученного из *внешних источников*.

HashiCorp Vault Secret Lookup

Полномочия этого типа используются для доступа к учетным данным, хранящимся в хранилище секретов HashiCorp Vault[77].

Для создания полномочий этого типа необходимо предоставить следующие данные:

- URL сервера HashiCorp Vault.
- Учетные данные для доступа к серверу HashiCorp Vault.
- Номер версии API.

Подробности об использовании хранилища секретов HashiCorp Vault см. в *соответствующей секции*.

[73] https://docs.cyberark.com/credential-providers/latest/en/content/ccp/the-central%20-credential-provider.htm
[74] https://www.conjur.org/get-started/why-conjur/how-conjur-works/
[75] https://cloud.google.com/compute/vm-manager/docs/os-inventory/os-inventory-management
[76] https://cloud.google.com/iam/docs/keys-create-delete
[77] https://developer.hashicorp.com/vault/docs/what-is-vault

HashiCorp Vault Signed SSH

Полномочия этого типа используются для доступа к приватным ключам SSH, хранящимся в HashiCorp Vault[78].

Для создания полномочий этого типа необходимо предоставить следующие данные:

- URL сервера HashiCorp Vault.
- Учетные данные для доступа к серверу HashiCorp Vault.
- Номер версии API.

Подробности об использовании хранилища секретов HashiCorp Vault см. в *соответствующей секции*.

Insights

Полномочия этого типа используются для доступа к списку управляемых узлов в Red Hat Insights[79].

Microsoft Azure Key Vault

Полномочия этого типа используются для доступа к списку управляемых узлов в Microsoft Azure Key Vault[80].

OpenStack

Полномочия этого типа используются для доступа к списку управляемых узлов в OpenStack[81].

Поддерживается защита подключения с помощью SSL.

Red Hat Satellite 6

Полномочия этого типа используются для доступа к списку управляемых узлов в Red Hat Satellite 6[82].

Thycotic DevOps Secrets Vault

Полномочия этого типа используются для доступа к системе управления секретами Thycotic DevOps Secrets Vault.

[78] https://developer.hashicorp.com/vault/docs/secrets/ssh/signed-ssh-certificates
[79] https://docs.redhat.com/en/documentation/red_hat_insights/1-latest
[80] https://azure.microsoft.com/ru-ru/products/key-vault
[81] https://docs.openstack.org/2024.1/
[82] https://docs.redhat.com/en/documentation/red_hat_satellite/6.15

Thycotic Secret Server

Полномочия этого типа используются для доступа к серверу секретов Thycotic Secret Server.

Vault

Полномочия этого типа используются для доступа к списку управляемых узлов в Ansible Vault[83].

Если необходим доступ к нескольким хранилищам, необходимо предоставить соответствующие им идентификаторы Vault.

VMware vCenter

Полномочия этого типа используются для доступа к списку управляемых узлов в VMware vCenter[84].

Web-сервисы Amazon

Полномочия этого типа используются для доступа к списку управляемых узлов в Web-сервисах Amazon[85].

Для создания полномочий этого типа необходимо предоставить следующие данные Web-сервисов Amazon:

- публичный ключ доступа;
- секретный ключ.

Виртуализация Red Hat

Полномочия этого типа используются для доступа к расширению (plug-in) инвентаря `oVirt4.py`, которым управляют с помощью Виртуализации Red Hat[86].

Диспетчер ресурсов Microsoft Azure

Полномочия этого типа используются для доступа к списку управляемых узлов в Диспетчере ресурсов Microsoft Azure[87].

Личный токен доступа к GitHub

Полномочия этого типа используются для доступа к GitHub[88] с помощью личного токена доступа. Данный тип полномочий позволяет устанавливать API-соединение с GitHub для использования в заданиях и публикации обновлений состояния.

Пошаговые инструкции по созданию токена см. в документации GitHub[89].

[83] https://docs.ansible.com/ansible/latest/cli/ansible-vault.html
[84] https://docs.vmware.com/en/VMware-vSphere/index.html
[85] https://aws.amazon.com/documentation-overview/?nc2=h_ql_doc_do
[86] https://access.redhat.com/products/red-hat-virtualization
[87] https://learn.microsoft.com/ru-ru/azure/azure-resource-manager/management/overview
[88] https://github.com/
[89] https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/managing-your-personal-access-tokens

Машина

Полномочия этого типа используются для доступа к управляемым узлам по SSH. Поддерживается аутентификация по паролю и ключам SSH, созданным с применением различных алгоритмов.

При настройке полномочия можно выбрать один из следующих способов повышения привилегий:

- sudo;
- su;
- pbrun;
- pfexec;
- dzdo;
- pmrun;
- runas;
- enable;
- doas;
- ksu;
- machinectl;
- sesu.

Использование этой настройки эквивалентно выполнению команды `ansible-playbook` с параметром `--become-method=<method>`.

Платформа автоматизации Red Hat Ansible

Полномочия этого типа используются для доступа к Платформе автоматизации Red Hat Ansible[90].

Поддерживается защита подключения с помощью SSL.

Реестр контейнеров

Полномочия этого типа используются для доступа к различным хранилищам образов, например:

- *Private Automation Hub*;
- Docker[91];
- Quay[92];
- GitLab[93].

Поддерживается защита подключения с помощью SSL.

[90] https://docs.redhat.com/en/documentation/red_hat_ansible_automation_platform/2.4
[91] https://www.docker.com/
[92] https://quay.io/
[93] https://registry.gitlab.com/

Сеть

Полномочия этого типа используются для управления сетевым оборудованием. К настоящему времени, согласно документации Ansible[94], он устарел и не рекомендован к использованию. Вместо него рекомендуется использовать тип «Машина».

Токен персонального доступа GitLab

Полномочия этого типа используются для доступа к GitLab[95] с помощью личного токена доступа. Данный тип полномочий позволяет устанавливать API-соединение с GitLab для использования в заданиях и публикации обновлений состояния с помощью webhook.

Пошаговые инструкции по созданию токена см. в документации GitLab[96].

Токен доступа OpenShift или Kubernetes API

Полномочия этого типа используются для создания групп с доступом в контейнеры OpenShift или Kubernetes.

Для обеспечения доступа необходимо предоставить следующие данные:

- адрес точки доступа OpenShift или Kubernetes API для аутентификации;
- токен.

Поддерживается защита подключения с помощью SSL.

Система управления версиями

Полномочия этого типа используются для доступа к системам контроля версий на основе Git и Subversion.

Поддерживается доступ с помощью ключа SSH или с использованием имени пользователя и пароля.

Собственные типы полномочий

Automation Controller позволяет создавать собственные типы полномочий.

При создании собственного типа полномочий используются следующие понятия:

- *Входные данные* – данные, которые вводит пользователь при создании полномочия. В качестве входных данных могут использоваться имена пользователей, пароли, приватные ключи SSH и GPG, токены и тому подобное.
- *Инжектор* – механизм, выполняющий преобразование данных полномочия в нужный формат.
- *Инжекция* – процесс передачи данных полномочия в *среду исполнения* при выполнении задания.

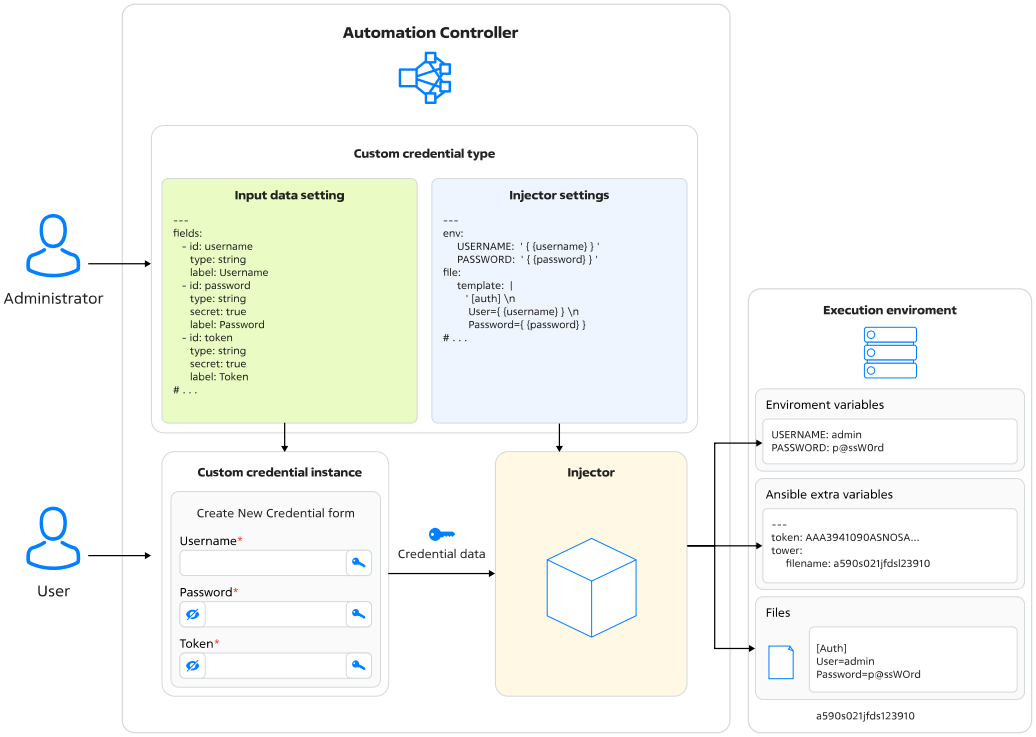
Создание собственного типа полномочий имеет следующие особенности:

- Каждый собственный тип полномочий должен содержать описание полей входных данных и настройки инжектора.
- Контроллер не проверяет наличие коллизий названий переменных окружения, дополнительных переменных Ansible и пространств имен.

[94] https://docs.ansible.com/ansible/devel/network/getting_started/network_differences.html#multiple-communication-protocols
[95] https://about.gitlab.com/
[96] https://docs.gitlab.com/ee/user/profile/personal_access_tokens.html

- Следует избегать использования названий переменных, начинающихся с `ANSIBLE_`, поскольку они зарезервированы.

Работа контроллера с собственными типами полномочий показана на схеме:



1. При создании собственного типа полномочий администратор задает настройки входных данных и инжектора.
2. Пользователи создают необходимые экземпляры полномочий собственного типа.
3. При запуске заданий введенные пользователями данные полномочия передаются в инжектор.
4. Инжектор передает данные полномочия в среду исполнения.

   Возможны следующие способы передачи данных полномочия в среду исполнения:

   - переменная окружения;
   - дополнительная переменная (extra vars) Ansible;
   - текстовый файл, в том числе с возможностью генерации содержимого с использованием шаблонизатора Jinja.

   Способ передачи данных определяется настройками инжектора. Одни и те же данные полномочия могут быть переданы в среду исполнения сразу несколькими способами, например, в переменной окружения, дополнительной переменной Ansible и содержимом текстового файла.

Настройка входных данных

Настройки входных данных типа полномочий определяют параметры полей ввода, используемых при вводе данных полномочия.

При создании поля необходимо указать его идентификатор и тип. Также можно указать дополнительные атрибуты, влияющие на поведение поля формы и обработку введенного пользователем значения.

Типовые настройки входных данных состоят из обязательного блока `fields` и опционального блока `required`.

В блоке `fields` содержится список записей с описанием полей ввода данных полномочия. Каждая запись может состоять из следующих полей:

- `id` – идентификатор поля. Значение, указанное в этом поле, используется для передачи данных в инжектор.
- `type` – тип поля:
  - `string` – строка;
  - `boolean` – логическое.
- `default` – значение по умолчанию.
- `format` – дополнительная проверка корректности значения, указанного в поле.

  Поддерживаются следующие значения:
  - `ssh_private_key` – приватный ключ SSH;
  - `url` – URI.
- `label` – название поля при заполнении сведений о полномочии через графический интерфейс.
- `help_text` – подсказка, которая выводится рядом с полем при заполнении сведений о полномочии через графический интерфейс.
- `secret` – если равно `true`, вводимое значение является секретом.

  Для защиты секретов в полях ввода используется маскировка символов, а в таблице базы данных Automation Controller – защитное преобразование.
- `multiline` – логическое значение для полей типа `string`. При значении `true` поле позволяет вводить многострочный текст.
- `choices` – массив доступных значений для полей типа `string`. Если массив не пуст, при заполнении поля через графический интерфейс значение нужно выбрать из списка, а не вводить вручную.

В блоке `required` содержится список идентификаторов полей, обязательных для заполнения.

Конфигурация инжектора

Для доступа к значению, полученному из входных данных, следует в двойных фигурных скобках указать идентификатор соответствующего поля, например:

```
---
extra_vars:
  auth_token: '{{ auth_token }}'
```

Дополнительной переменной Ansible `auth_token` присваивается значение поля входных данных с идентификатором `auth_token`.

Настройки инжектора могут содержать следующие блоки:

- `file` – шаблон, на основе которого генерируется содержимое временного файла.

  Имя файла генерируется случайным образом. Оно хранится в переменной Ansible `tower.filename`.

  Если используется несколько полей для загрузки файлов, их следует описать в отдельных переменных, названия которых начинаются с префикса `template.`. Для получения имени файла следует обратиться к соответствующему значению словаря `tower.filename`, например:

  ```
  ---
  file:
    template.cert_file: '{{ tls_cert }}'
    template.key_file: '{{ tls_key }}'
  env:
  ```

(continues on next page)

(продолжение с предыдущей страницы)

```
  TLS_CERT_FILE_NAME: '{{ tower.filename.cert_file }}'
  TLS_KEY_FILE_NAME: '{{ tower.filename.key_file }}'
```

Здесь для хранения сертификата и соответствующего ему ключа создаются два временных файла, имена которых сохраняются в переменных `tower.filename.cert_file` и `tower.filename.key_file` соответственно.

- env – список названий переменных окружения и соответствующих им значений.
- `extra_vars` – список названий дополнительных переменных Ansible и соответствующих им значений.

  Подробности о порядке присвоения значений переменным см. в документе *Переменные*.

Примеры

Пусть для работы с некоторым сервисом необходимы следующие учетные данные:

- Название учетной записи пользователя – строка, хранится в открытом виде.

  Значение передается в среду исполнения через переменную окружения `API_USERNAME`.
- Токен – строка, хранится в зашифрованном виде.

  Значение передается в среду исполнения через переменную окружения `API_TOKEN` в следующем формате:

```
Bearer-Token: <token>
```

- Уровень доступа – строка, может принимать значения `low`, `medium` и `high`. Значение по умолчанию – `low`.

  Значение передается в среду исполнения через дополнительную переменную Ansible `api_access_level`.
- Приватный ключ SSH для защиты соединения.

  Значение передается в среду исполнения в виде файла, имя которого хранится в переменной окружения `API_PRIVATE_SSH_KEY`.

Для хранения указанных данных в Automation Controller следует использовать поля со следующими свойствами:

Параметр	Идентификатор	Тип	Обязательное
Название учетной записи	`username`	`string`	Да
Токен	`token`	`string`	Да
Уровень доступа	`access_level`	`string`	Да
Ключ SSH	`ssh_key`	`string`	Нет

Настройки собственного типа полномочий в этом случае можно задать следующим образом:

Список 1: Настройки входных данных

```
---
fields:
  - id: username
    type: string
    label: Username

  - id: token
```

(continues on next page)

(продолжение с предыдущей страницы)

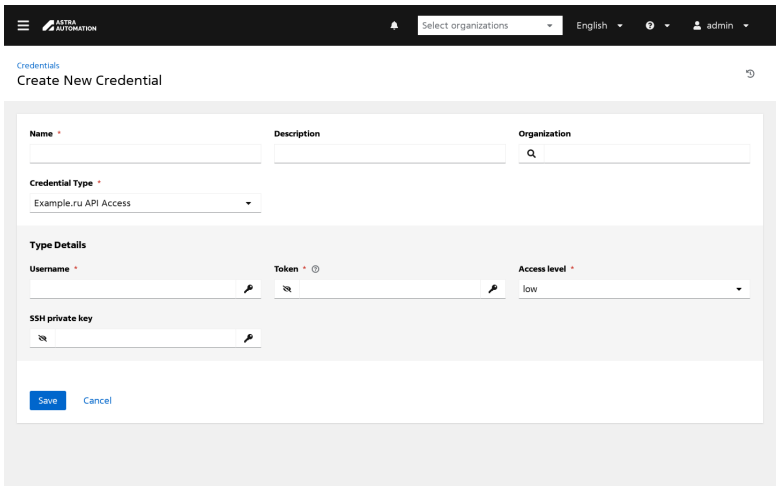
```
    type: string
    label: Token
    secret: true
    help_text: A string of 32 charactes long.

  - id: access_level
    type: string
    label: Access level
    choices:
      - low
      - medium
      - high
    default: low

 - id: ssh_key
   type: string
   label: SSH private key
   format: ssh_private_key
   secret: true

required:
  - username
  - token
  - access_level
```

Список 2: Настройки инжектора

```
---
env:
  API_USERNAME: '{{ username }}'
  API_TOKEN: 'Bearer-Token:: {{ token }}'
  API_PRIVATE_SSH_KEY: '{{ tower.filename.private_ssh_key }}'
extra_vars:
  api_access_level: '{{ access_level }}'
file:
  template.private_ssh_key: '{{ ssh_key }}'
```

Примечание: Два двоеточия в строке с параметром `API_TOKEN` необходимы для соблюдения корректности синтаксиса YAML. При передаче данных строка будет преобразована в указанный выше формат с одним двоеточием.

В графическом интерфейсе Automation Controller форма для создания полномочий этого типа имеет следующий вид:



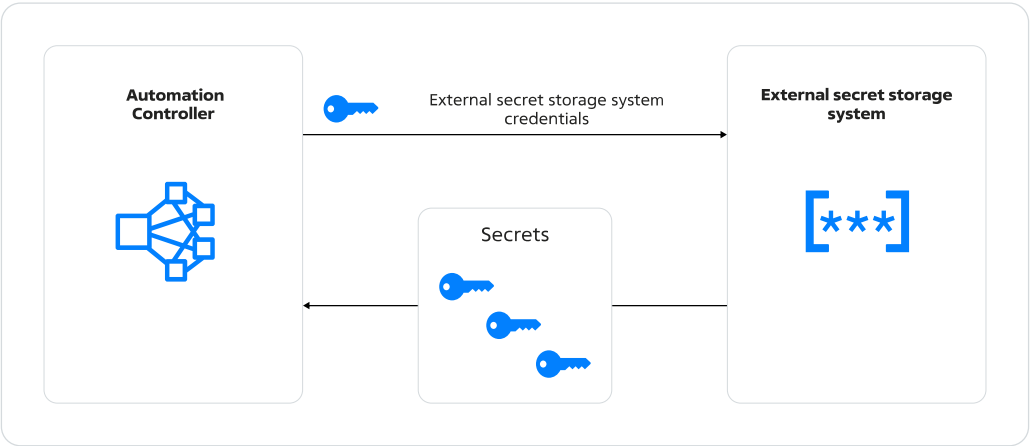
Внешние системы управления секретами

Внешней системой управления секретами называется сторонний сервис, используемый для хранения секретов и управления ими.

Automation Controller может использовать секреты, хранящиеся в следующих системах управления секретами:

- AWS Secrets Manager Lookup;
- Centrify Vault Credential Provider Lookup;
- CyberArk Central Credential Provider Lookup;
- CyberArk Conjur Secrets Manager Lookup;
- HashiCorp Vault;
- Microsoft Azure Key Vault;
- Thycotic DevOps Secrets Vault;
- Thycotic Secret Server.

В общем виде схема работы Automation Controller с секретами, хранящимися во внешней системе управления секретами, показана на схеме:



1. В Automation Controller создается полномочие для доступа к внешней системе управления секретами.
2. В Automation Controller создаются полномочия необходимых типов, используемые в заданиях. При этом вместо ввода значения секретов в соответствующих полях указывается созданное ранее полномочие для доступа к внешней системе управления секретами.
3. При запуске задания Automation Controller обращается к внешней системе управления секретами для получения значений необходимых секретов.

Ниже рассматриваются особенности использования различных внешних систем управления секретами.

AWS Secrets Manager Lookup

Важно: Эта часть документации находится в стадии разработки.

Centrify Vault Credential Provider Lookup

Важно: Эта часть документации находится в стадии разработки.

CyberArk Central Credential Provider Lookup

Важно: Эта часть документации находится в стадии разработки.

CyberArk Conjur Credential Provider Lookup

Важно: Эта часть документации находится в стадии разработки.

HashiCorp Vault

Для доступа к секретам Automation Controller может использовать API HashiCorp Vault версий v1 и v2.

Для аутентификации в HashiCorp Vault доступны следующие методы и средства:

- название учетной записи пользователя и пароль[97];
- AppRole[98];
- Kubernetes[99];
- сертификат TSL[100].
- LDAP[101] (только для HashiCorp Vault редакции Enterprise).

Данные аутентификации необходимо указывать при создании полномочий соответствующих типов:

- *HashiCorp Vault Secret Lookup*;
- *HashiCorp Vault Signed SSH*.

При аутентификации через AppRole для доступа к HashiCorp Vault можно использовать *собственный тип полномочий* со следующими свойствами:

- Настройка входных данных:

[97] https://developer.hashicorp.com/vault/docs/auth/userpass
[98] https://developer.hashicorp.com/vault/docs/auth/approle
[99] https://developer.hashicorp.com/vault/docs/auth/kubernetes
[100] https://developer.hashicorp.com/vault/docs/auth/cert
[101] https://developer.hashicorp.com/vault/docs/auth/ldap

```yaml
---
fields:
  - id: vault_server
    type: string
    label: URL for Vault Server
  - id: vault_role_id
    type: string
    label: Vault AppRole ID
  - id: vault_secret_id
    type: string
    label: Vault Secret ID
    secret: true
required:
  - vault_server
  - vault_role_id
  - vault_secret_id
```

- Конфигурация инжектора:

```yaml
---
extra_vars:
  ansible_hashi_vault_addr: '{{ vault_server }}'
  ansible_hashi_vault_role_id: '{{ vault_role_id }}'
  ansible_hashi_vault_secret_id: '{{ vault_secret_id }}'
  ansible_hashi_vault_auth_method: approle
```

Microsoft Azure Key Vault

Важно: Эта часть документации находится в стадии разработки.

Thycotic DevOps Secrets Vault

Важно: Эта часть документации находится в стадии разработки.

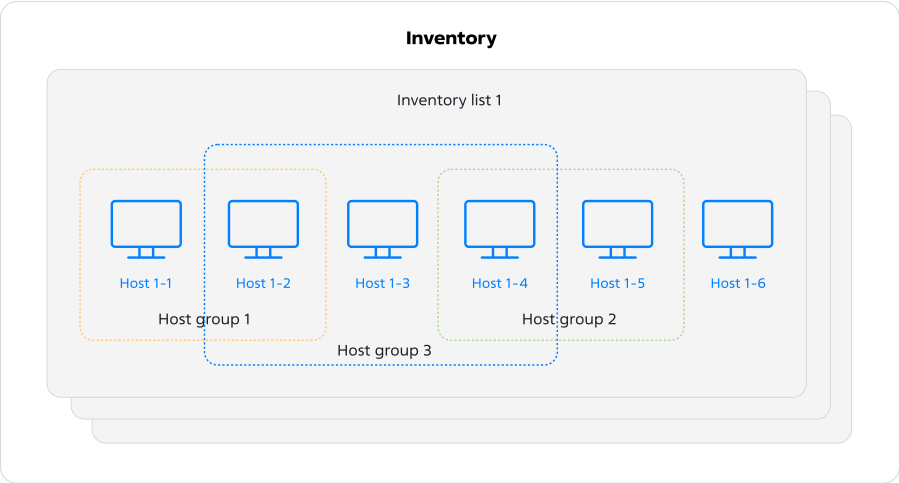
Thycotic Secret Server

Важно: Эта часть документации находится в стадии разработки.

Инвентарь

Инвентарь (inventory) в Automation Controller используется для хранения сведений об управляемых узлах. При создании шаблона задания управляемые узлы из выбранного инвентаря связываются с Ansible playbook из проекта.

Структура инвентаря показана на схеме:



Инвентарь состоит из инвентарных списков различного типа. Каждый инвентарный список содержит перечень управляемых узлов (hosts). Один и тот же узел может входить в несколько инвентарных списков.

Примечание: Далее для краткости инвентарные списки будут называться инвентарями.

Группы узлов

Группы узлов (inventory groups) используются для логического объединения узлов, входящих в один инвентарь. Один и тот же управляемый узел может входить в несколько групп одновременно.

Группы узлов обладают следующими свойствами:

- Для каждой группы узлов можно задать свои значения переменных Ansible[102].
- При импорте инвентаря из внешнего источника, содержащего группы узлов, одноименные группы в инвентаре контроллера создаются автоматически.
- Automation Controller позволяет выполнять специальные (ad-hoc) команды на узлах выбранной группы без создания шаблонов.

Типы инвентаря

В Automation Controller поддерживаются три типа инвентаря:

- обычный (standard);
- умный (smart);
- сборный (constructed).

Тип инвентаря влияет на способ добавления в него сведений об управляемых узлах. Подробности о каждом типе инвентаря приведены далее.

[102] https://docs.ansible.com/ansible/latest/playbook_guide/playbooks_variables.html

Особенности удаления

Удаление инвентарных списков имеет следующие особенности:

- Вместе с обычным инвентарным списком из контроллера удаляются данные обо всех связанных с ним управляемых узлах и группах узлов.
- При удалении обычных и сборных инвентарных списков состав связанных с ними сборных инвентарных списков **не** меняется. Однако, он может измениться, если запустить синхронизацию сборного инвентарного списка.
- Связанные с инвентарным списком шаблоны заданий не удаляются и могут быть связаны с другим инвентарным списком.

Состояния

Инвентарь может находиться в одном из состояний:

- **Успех** (Success) – синхронизация инвентаря прошла успешно.
- **Запрещен** (Disabled) – в инвентарь не добавлено ни одного источника сведений об управляемых узлах.
- **Ошибка** (Error) – последняя попытка синхронизации инвентаря с источником была неудачной.

Статусы управляемых узлов

Контроллер позволяет исключить использование в заданиях управляемых узлов без необходимости изменения существующих или создания новых инвентарных списков, а также внесения изменений в шаблоны заданий. Для этого каждой записи об управляемом узле присваивается *статус*, который указывает, может ли управляемый узел быть объектом выполняемых заданий Ansible:

- **Вкл** (On) – может. Этот статус используется по умолчанию при создании и импорте записей об управляемых узлах.
- **Выкл** (Off) – не может. Управляемые узлы с этим статусом исключаются из инвентаря при запуске заданий.

Обычный инвентарь

Обычный инвентарь, также называемый просто инвентарем (inventory), имеет следующие особенности:

- Сведения об управляемых узлах могут быть добавлены вручную или импортированы из внешнего источника, в том числе из файлов инвентаря Ansible, добавленных в проект.
- Поддерживает использование переменных Ansible и групп узлов.

Источники сведений об управляемых узлах

Импорт сведений об управляемых узлах из внешних источников имеет следующие особенности:

- Если в качестве источника выбран отдельный файл, импортируются узлы только из него.
- Если в качестве источника выбран каталог, импортируются узлы из всех размещенных в нем файлов, включая дочерние каталоги.
- Поддерживается фильтрация узлов по названию с помощью регулярных выражений. Этот фильтр применяется в последнюю очередь, после всех остальных фильтров, предоставляемых используемым расширением (plug-in) инвентаря.

  Пусть файл инвентаря содержит записи о тридцати управляемых узлах (диапазон IP-адресов от `10.1.0.1` до `10.1.0.30`):

  ```
  ---
  all:
    hosts:
      # Контроллеры домена
      dc1.example.com:
      dc2.example.com:
      # Сервер печати
      cups.example.com:
      # Клиенты домена
      client01.example.com:
      client02.example.com:
      client03.example.com:
      # ...
      client30.example.com:
  ```

  Для импорта только первых десяти клиентов домена (с `client01.example.com` по `client10.example.com` включительно) следует использовать регулярное выражение следующего вида:

  ```
  client([0][1-9]|10)\.example\.com
  ```

- В описании импортированных узлов указывается значение **imported**.

  Это значение можно изменить, добавив в файл инвентаря поле `_awx_description` для нужных узлов, например:

  ```
  ---
  domain-controllers:
    hosts:
      dc1.example.com:
        _awx_description: Primary domain controller
      dc2.example.com:
        _awx_description: First replica of domain controller
  ```

- Если в файле инвентаря узел принадлежит одной или нескольким группам, в контроллере эти группы будут автоматически созданы, а узел включен в каждую из них.

  Пусть файл инвентаря имеет следующий вид:

  ```
  ---
  domain-controllers:
    # Контроллеры домена
    hosts:
      dc1.example.com:
      dc2.example.com:
  clients:
    # Клиенты домена
    hosts:
  ```

(continues on next page)

(продолжение с предыдущей страницы)

```
    client-1.example.com:
    client-2.example.com:
    client-3.example.com:
    client-4.example.com:
```

При импорте инвентаря в нем будут автоматически созданы группы domain-controllers и clients, содержащие контроллеры домена и клиенты домена соответственно.

- Если в источнике группы узлов организованы иерархически, структура групп будет воссоздана при импорте.

Важно: Узлы дочерних групп не связываются с родительской группой автоматически – их нужно добавлять вручную.

Например, пусть есть инвентарь следующего вида:

```
---
all:
  hosts:
    node-[01:10].example.com:
controllers:
  hosts:
    node-01.example.com:
    node-02.example.com:
clients:
  hosts:
    node-[03:10].example.com:
production:
  children:
    controllers:
    clients:
```

При импорте такого инвентаря будут созданы три группы – production, clients и controllers. Управляемые узлы будут связаны только с группами clients и controllers.

- По умолчанию при повторной синхронизации инвентарного списка с источником импортированные ранее управляемые узлы, группы узлов и переменные не удаляются.
- Поддерживается использование сценариев динамического создания инвентаря.

  Подробности о динамическом создании инвентаря см. в документации Ansible[103].
- Поддерживается синхронизация с источником по расписанию.

[103] https://docs.ansible.com/ansible/latest/inventory_guide/intro_dynamic_inventory.html

Поддержка расширений

Automation Controller поддерживает следующие расширения инвентаря:

Тип источника	Ссылка на руководство
Amazon EC2	aws_ec2_inventory[104]
Google Compute Engine	gcp_compute_inventory[105]
Диспетчер ресурсов Microsoft Azure	azure_rm_inventory[106]
VMware vCenter	vmware_vm_inventory_inventory[107]
Red Hat Satellite 6	foreman_inventory[108]
OpenStack	oprenstack_inventory[109]
Виртуализация Red Hat	ovirt_inventory[110]
Платформа автоматизации Red Hat Tower	tower_inventory[111]
Red Hat Insights	insights_inventory[112]

Умный инвентарь

Умный инвентарь представляет собой список узлов, сформированный путем фильтрации всех управляемых узлов организации. При изменении списка управляемых узлов организации содержимое умного инвентарного списка обновляется автоматически, так как он хранит не копии сведений об управляемых узлах, а лишь ссылки на них.

Предупреждение: Умный инвентарь считается устаревшим и не поддерживается. В одной из следующих версий Automation Controller этот тип инвентаря станет недоступным для применения. В качестве замены следует использовать *сборный инвентарь*.

Умный инвентарь имеет следующие особенности:

- Использование переменных Ansible не поддерживается.
- Группы узлов импортируются из источников, но доступны только для просмотра – создать новые группы или изменить существующие нельзя.

Базовые настройки фильтрации узлов

В самом простом случае запрос на выборку узлов формируется путем одновременного применения одного или нескольких фильтров из указанного списка:

- `enabled` (Enabled / Подключен) – фильтр по статусу управляемого узла.

  Если значение фильтра равно `true`, отбираются только узлы со статусом **Вкл** (On).

- `group__name__icontains` (Group / Группа) – фильтр по названию группы узлов.

  Отбираются узлы, которые принадлежат группе, в названии которой есть указанная подстрока. Поиск группы по названию производится без учета регистра.

[104] https://docs.ansible.com/ansible/latest/collections/amazon/aws/aws_ec2_inventory.html
[105] https://docs.ansible.com/ansible/latest/collections/google/cloud/gcp_compute_inventory.html
[106] https://docs.ansible.com/ansible/latest/collections/azure/azcollection/azure_rm_inventory.html
[107] https://docs.ansible.com/ansible/latest/collections/community/vmware/vmware_vm_inventory_inventory.html
[108] https://docs.ansible.com/ansible/latest/collections/theforeman/foreman/foreman_inventory.html
[109] https://docs.ansible.com/ansible/latest/collections/openstack/cloud/openstack_inventory.html
[110] https://docs.ansible.com/ansible/latest/collections/ovirt/ovirt/ovirt_inventory.html
[111] https://docs.ansible.com/ansible/latest/collections/awx/awx/tower_inventory.html
[112] https://github.com/RedHatInsights/insights-host-inventory

- `instance_id` (Instance ID / Идентификатор узла контроллера) – фильтр по идентификаторам узлов контроллера.
- `inventory_id` (Inventory ID / Идентификатор инвентаря) – фильтр по идентификаторам инвентарных списков.

  Отбираются узлы, входящие в инвентарные списки с указанными идентификаторами.
- `last_job` (Last Job / Последнее задание) – фильтр по идентификатору последнего выполненного на узле задания.

  Отбираются узлы, последнее выполненное на которых задание имеет идентификатор из указанного списка.
- `name__icontains` (Name / Название) – фильтр по названию узла.

  Отбираются узлы, в названии которых есть указанная подстрока. Поиск подстроки производится без учета регистра.

Пусть имеется следующий список управляемых узлов:

```
---
controllers:
  hosts:
    dc1.example.com:
    dc2.example.com:
clients:
  hosts:
    client-1.example.com:
    client-2.example.com:
    client-3.example.com:
    client-4.example.com:
```

Необходимо создать умный инвентарь, в который входят только узлы клиентов домена. С помощью умного инвентаря выполнить эту задачу можно различными способами, например:

- Создать умный инвентарь с фильтром `name__icontains` со значением `client-`.
- Создать умный инвентарь с фильтром `group__name__icontains` со значением `clients`.
- Создать умный инвентарь сразу с двумя указанными выше фильтрами.

Расширенные настройки фильтрации узлов

Умный инвентарь поддерживает расширенные настройки фильтрации узлов. В этом случае выражение поиска строится по следующей схеме:

```
<type>__<key>__<search_type> <value>
```

Здесь:

- <type> – способ обработки нескольких условий фильтрации.

  Поддерживаются следующие значения:

  - `or` – это необязательное условие;
  - and – это обязательное условие.
- <key> – название поля в свойствах узла или одного из связанных с ним ресурсов.

  Поддерживаются следующие значения:

  - `ad_hoc_command_events` – название событий, произошедших на управляемых узлах при выполнении специальных (ad-hoc) команд.
  - `ad_hoc_commands` – название специальной команды.

- `ansible_facts_modified` – дата и время последнего обновления фактов Ansible.

  Совет: Как правило, факты Ansible об управляемом узле обновляются при выполнении заданий.

- `ansible_facts` – факт Ansible.
- `created_by` – имена пользователей, создавших записи об управляемых узлах.
- `created` – дата и время создания записи об управляемом узле.
- `description` – описание управляемого узла.
- `enabled` – выбор узлов по статусу в Automation Controller.
- `groups` – названия групп узлов.
- `id` – идентификатор управляемого узла.
- `inventory_sources` – источник инвентаря.
- `inventory` – идентификатор инвентаря, из которого должны быть выбраны управляемые узла.
- `jobs` – список идентификаторов заданий, выполненных на управляемом узле.
- `last_job_host_summary` – статус выполнения последнего задания, выполненного на управляемом узле.
- `last_job` – идентификатор последнего задания, выполненного на управляемом узле.
- `modified_by` – имена пользователей, последними изменявших данные об управляемом узле.
- `modified` – дата и время последнего изменения записи об управляемом узле.
- `name` – название управляемого узла.
- `smart_inventories` – названия умных инвентарных списков.
- `variables` – фильтр узлов по названиям указанных в их данных переменных.

• <search_type> – обработка значения, используемого в качестве фильтра.

  Поддерживаются следующие способы обработки указанного значения:

  - `exact` – точное совпадение с учетом регистра.

    Это значение используется по умолчанию.
  - `gt` – проверка условия «больше чем», >.
  - `gte` – проверка условия «больше или равно», ≥.
  - `in` – проверка на присутствие в указанном списке.
  - `iregex` – поиск по регулярному выражению без учета регистра.
  - `isnull` – проверка на пустое значение.
  - `lt` – проверка условия «меньше чем», <.
  - `lte` – проверка условия «меньше или равно», ≤.
  - `regex` – поиск по регулярному выражению с учетом регистра.

• <value> – значение, используемое в качестве фильтра.

  Особенности ввода значений фильтрации:

  - Для ввода дробных чисел в качестве разделителя следует использовать символ точки, например, `3.1415926`.
  - Для ввода дат следует использовать формат `YYYY-MM-DD`, где:

    * YYYY – четырехзначное значение года, например, 2024.
    * MM – двузначеное значение месяца. Отсчет месяцев начинается с 01.
    * DD – двузначное значение числа. Отсчет чисел начинается с 01.
  - Для ввода времени следует использовать формат HH:MM:ssuuuuuuTZ, где:
    * HH – двузначное значение часа в 24-часовом формате, например, 07.
    * MM – двузначное значение минут, от 00 до 59.
    * (опционально) ss – двузначное значение секунд, от 00 до 59.
    * (опционально) uuuuuu – микросекунды.
    * (опционально) TZ – название временной зоны.

Сборный инвентарь

Сборный инвентарь представляет собой список записей об управляемых узлах, сведения о которых получены путем копирования из обычных инвентарных списков. В этом документе обычные инвентарные списки, из которых копируются данные, называются источниками.

Сборный инвентарь имеет следующие особенности:

- Для формирования сборного инвентаря требуется хотя бы один источник.
- Сборный инвентарь хранит **копии** записей об управляемых узлах и группах управляемых узлов, а не ссылки на них. Это значит, что при обновлении источников сборный инвентарь не обновляется автоматически.
- Существующие группы узлов импортируются из источника независимо от того, будет ли в них хотя бы один узел после выполнения условий отбора узлов.
- Новые группы узлов создаются и заполняются автоматически в соответствии с параметрами, заданными в настройках расширения Ansible для работы с инвентарем.
- Сведения об управляемых узлах копируются в сборный инвентарь при синхронизации. Для первичного заполнения сборного инвентаря ее необходимо запустить вручную.

Фильтрация узлов

Параметры отбора управляемых узлов задаются в переменных source_vars и limit.

- source_vars – параметры, определяющие название используемого расширения Ansible и правила его использования.

  Примечание: Для создания сборного инвентаря в Automation Controller используется расширение Ansible ansible.builtin.constructed[113].

- limit – правила выборки записей из существующих или временных групп управляемых узлов, сформированных при обработке значения переменной source_vars.

В самом простом случае сборный инвентарь формируется путем копирования сведений обо всех управляемых узлах из источников. Значение переменной source_vars в этом случае задается следующим образом:

```
---
plugin: ansible.builtin.constructed
```

Здесь plugin – переменная, в которой указывается название расширения Ansible, используемого для работы с инвентарем. Это обязательная запись, значение которой должно быть равно ansible.builtin.constructed или constructed.

[113] https://docs.ansible.com/ansible/latest/collections/ansible/builtin/constructed_inventory.html

Примечание: Рекомендуется использовать полное название расширения – `ansible.builtin.constructed`.

В настройки фильтрации может быть добавлен логический параметр `strict`, значения которого интерпретируются следующим образом:

- `true` – считать любую ошибку обработки сведений об управляемых узлах критической и останавливать создание или обновление сборного инвентарного списка;
- `false` – пропускать управляемые узлы, при обработке сведений о которых возникли ошибки.

Примеры

В рассматриваемых далее примерах в качестве источника используются следующий инвентарь:

```
[controllers]
dc1.example.com
dc2.example.com      timezone=Novosibirsk
dc3.example.com      timezone=Chita
dc4.example.com      timezone=Chita        env=testing

[rubackup]
rb1.example.com
rb2.example.com
rb3.example.com      timezone=Chita
rb4.example.com                            env=testing

[clients]
client1.example.com
client2.example.com
client3.example.com  timezone=Novosibirsk
client4.example.com                        env=testing
client5.example.com  timezone=Novosibirsk  env=testing
client6.example.com  timezone=Chita
client7.example.com  timezone=Chita        env=testing
```

Здесь:

- `controllers` – группа узлов контроллера домена;
- `rubackup` – узлы кластера RuBackup;
- `clients` – клиенты домена;
- `timezone` – часовой пояс, в котором физически находится управляемый узел;
- `env` – окружение, к которому относится управляемый узел.

Пример 1

Пусть необходимо сформировать сборный инвентарь, узлы которого отбираются по следующим признакам:

- часовой пояс не задан или равен `Moscow`;
- окружение тестовое.

В этом случае параметры сборного инвентаря можно задать следующим образом:

- `source_vars`:

```yaml
---
plugin: ansible.builtin.constructed
strict: true
groups:
  timezone_is_moscow: timezone | default("Moscow") == "Moscow"
  env_is_testing: env | default("production") == "testing"
```

- `limit` – `timezone_is_moscow:&env_is_testing`.

Пример 2

Пусть необходимо сформировать сборный инвентарь, узлы которого отбираются по следующим признакам:

- окружение не тестовое;
- не входит в группу `rubackup`.

В этом случае параметры сборного инвентаря можно задать следующим образом:

- `source_vars`:

```yaml
---
plugin: ansible.builtin.constructed
strict: true
groups:
  env_is_not_testing: env | default("production") != "testing"
```

- `limit` – `env_is_not_testing:!rubackup`

Источники кода

Automation Controller позволяет при создании проектов и заполнении инвентаря использовать код из внешних источников. В этом разделе описаны доступные типы источников кода и особенности работы с ними.

При создании проекта может быть использован один из указанных источников:

- ручное управление (локальный каталог);
- репозиторий Git;
- репозиторий Subversion;
- внешний архив.

Для хранения локальной копии кода каждого проекта Automation Controller создает подкаталог в каталоге, используемом для хранения проектов (далее – каталог проектов). Путь к каталогу для хранения проектов хранится в параметре контроллера `PROJECTS_ROOT`. Его значение задается на этапе развертывания контроллера и по умолчанию равно `/var/lib/awx/projects/`.

При создании или удалении проекта с любым типом источника, кроме «Ручное управление», подкаталог проекта в каталоге проектов соответственно создается или удаляется автоматически.

Ручное управление

Для использования локального каталога в качестве источника кода проекта необходимо соблюдение следующих условий:

- код проекта размещен в одном из подкаталогов каталога проектов;
- владельцами каталога проекта являются пользователь и группа awx;
- на файлы и каталоги проекта установлены следующие режимы доступа:
  - файлы – `0744`;
  - каталоги – `0755`.

Репозиторий Git

Automation Controller поддерживает использование репозиториев Git в качестве источников кода проектов. Основной параметр, который при этом должен быть указан – URL репозитория в *VCS*. В каталоге для хранения проектов создается подкаталог с локальной копией указанного репозитория.

URL репозитория в зависимости от используемого протокола подключения может иметь следующий вид:

- HTTPS:

  ```
  https://example.org/project.git
  ```

- SSH

  ```
  ssh://git@example.org/project.git
  ```

- GIT

  ```
  git://example.org/project.git
  ```

Примечание: Поддержка протоколов и особенности их использования определяются возможностями сервиса, в котором размещен репозиторий.

Дополнительные настройки

Для репозитория Git можно указать дополнительные параметры системы управления исходными данными.

Название ветки, название тега или хеш коммита

По умолчанию используется последний коммит из основной ветки репозитория. Эта настройка позволяет использовать другие версии кода:

- Название ветки.

  Используется версия кода из последнего коммита в указанной ветке.

  Например, если в репозитории основная ветка называется `main`, а ветка с экспериментальными возможностями называется `development`, для использования кода из ветки `development` необходимо указать ее название в свойствах источника.

- Название тега.

Используется версия кода, отмеченная указанным тегом. Например, чтобы использовать коллекции версии `1.7.0`, соответствующее значение следует указать в настройках источника.

- Хеш коммита.

  Используется версия кода из коммита с указанным хешем. Хеш коммита может быть указан полностью или частично (первые несколько символов).

  Для использования коммита с хешем `d7e82ab6ebff598b928edd356803758a3f3435ce` допускается указать значение `d7e82ab6`.

- Произвольная ссылка на версию кода в локальной копии репозитория.

Важно: Некоторые хеши коммитов и ссылки могут быть недоступны без указания спецификации (refspec) для измененной ссылки.

Спецификация (refspec)

Эта настройка используется для управления ссылками в локальной копии репозитория на версии кода в удаленном репозитории.

Например, для получения из внешнего репозитория всех возможных ссылок в настройках спецификации необходимо указать следующую строку:

```
refs/*:refs/remotes/origin/*
```

Примечание: Для удаленного источника используется название `origin`.

Подробности использования спецификаций ссылок см. в документации Git[114].

Полномочия

Эта настройка позволяет указать полномочия, которые следует использовать для доступа к удаленному репозиторию. Для выбора доступны полномочия типа «Управление версиями».

Настройки управления версиями кода

В настройках Automation Controller можно указать дополнительные параметры, управляющие поведением при работе с репозиторием.

- **Очистить**

  Перед выполнением обновления удаляются все сделанные изменения в локальной копии репозитория.

- **Удалить**

  При обновлении локальная копия репозитория удаляется полностью, после чего загружается заново.

  Примечание: В зависимости от размера репозитория и скорости соединения процесс загрузки может занять длительное время.

  ---

[114] https://git-scm.com/book/ru/v2/Git-изнутри-Спецификации-ссылок

- **Отслеживание подмодулей**

  Если эта настройка включена, отслеживается состояние не только основного репозитория, но и имеющихся подмодулей. Использование этой настройки эквивалентно выполнению команд `git fetch` и `git clone` с параметром `--recurse-submodules`.

  Подробности об использовании подмодулей см. в документации Git[115].

- **Обновить версию при запуске**

  Если эта настройка включена, перед запуском любого задания, использующего код из этого источника, он будет автоматически обновлен.

- **Разрешить переопределение ветки**

  Если эта настройка включена, в шаблоне задания разрешается выбрать ветку или версию кода, отличную от заданной в свойствах проекта.

Репозиторий Subversion

Automation Controller поддерживает использование репозиториев Subversion в качестве источников кода проектов. Основной параметр, который при этом должен быть указан – URL репозитория в *VCS*.

URL репозитория в зависимости от используемого протокола подключения может иметь следующий вид:

- HTTP(S):

```
https://example.org/project
```

- SVN

```
svn@example.org/project
```

- SVN+SSH

```
svn+ssh://example.org/project
```

Примечание: Поддержка протоколов и особенности их использования определяются возможностями сервиса, в котором размещен репозиторий.

Дополнительные настройки

Для репозитория Subversion можно указать дополнительные параметры:

- номер ревизии;
- полномочия на систему управления исходными данными.

[115] https://git-scm.com/book/ru/v2/Инструменты-Git-Подмодули

Номер версии

По умолчанию используется самая свежая ревизия кода из основной ветки репозитория. Эта настройка позволяет использовать другие ревизии кода:

- Название ветки.

  Используется последняя ревизия кода в указанной ветке.

- Номер ревизии.

  Используется код из указанной ревизии.

Полномочия на систему управления исходными данными

Эта настройка позволяет указать полномочия типа «Управление версиями», которые следует использовать для доступа к удаленному репозиторию.

Настройки управления ревизиями кода

В настройках Automation Controller можно указать дополнительные параметры, управляющие поведением при работе с репозиторием.

- **Очистить**

  Перед выполнением обновления удаляются все сделанные изменения в локальной копии репозитория.

- **Удалить**

  При обновлении локальная копия репозитория удаляется полностью, после чего загружается заново.

  Примечание: В зависимости от размера репозитория и скорости соединения процесс загрузки может занять длительное время.

- **Обновить версию при запуске**

  Если эта настройка включена, перед запуском любого задания, использующего код из этого источника, он будет автоматически обновлен.

- **Разрешить переопределение ветки**

  Если эта настройка включена, в шаблоне задания разрешается выбрать ветку или версию кода, отличную от заданной в свойствах проекта.

Внешний архив

В качестве источника кода Automation Controller может использовать архивы, содержащие код самого проекта и его зависимостей. Основной параметр, который при этом должен быть указан – URL архива в источнике. Поддерживаются архивы форматов ZIP и TAR.GZ.

Доступ к архивам возможен по протоколам HTTP и HTTPS.

В каталоге для хранения проектов создается подкаталог, в который распаковывается содержимое загруженного архива.

Дополнительные настройки

Для внешнего архива доступны следующие дополнительные настройки:

- Полномочия

  Эта настройка позволяет указать полномочия, которые следует использовать для доступа к архиву в источнике. Для выбора доступны полномочия типа «Управление версиями».

- **Очистить**

  Перед выполнением обновления удаляются все сделанные изменения в локальной копии репозитория.

- **Удалить**

  При обновлении локальная копия репозитория удаляется полностью, после чего загружается заново.

  Примечание: В зависимости от размера репозитория и скорости соединения процесс загрузки может занять длительное время.

- **Обновить версию при запуске**

  Если эта настройка включена, перед запуском любого задания, использующего код из этого источника, он будет автоматически обновлен.

- **Разрешить переопределение ветки**

  Если эта настройка включена, в шаблоне задания разрешается выбрать ветку или версию кода, отличную от заданной в свойствах проекта.

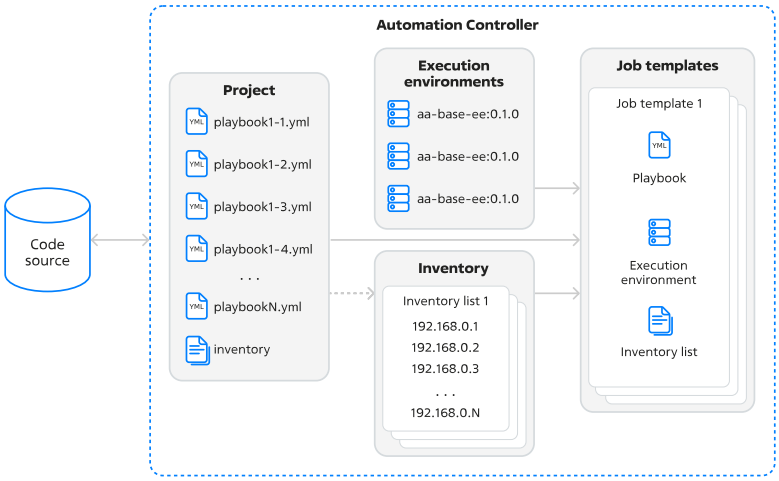
Проекты

В Automation Controller проект – это набор Ansible playbook. Также проект может содержать инвентарные списки, зависимости Ansible и другие файлы.

Ansible playbook из проектов используются при создании шаблонов заданий.

Файлы инвентаря из проектов можно использовать в качестве источников при заполнении *обычных инвентарных списков*.

Взаимодействие проектов с другими компонентами контроллера показано на схеме:



Типовая структура проекта

При создании проекта для использования его в Automation Controller следуйте приведенным здесь рекомендациям по структуре файлов и каталогов.

Рекомендуемая структура файлов и каталогов:

```
├── ansible.cfg
├── collections
│   └── requirements.yml
├── inventory
│   └── all.yml
└── playbooks
    ├── group_vars
    │   ├── vars1.yml
    │   ├── vars2.yml
    │   ├── ...
    │   └── varsM.yml
    ├── playbook1.yml
    ├── playbook2.yml
    ├── ...
    └── playbookN.yml
```

Здесь:

- `ansible.cfg` – файл с настройками Ansible.

  Подробное описание доступных настроек см. в документации Ansible[116].

- `collections/` – каталог с файлами используемых коллекций.
- `collections/requirements.yml` – файл зависимостей, содержащий сведения об используемых коллекциях. Указанные в этом файле коллекции автоматически загружаются в каталог `collections/` при синхронизации проекта.

  Примеры заполнения файла `requirements.yml` см. в документе *Версионирование коллекций*.

- `inventory/` – каталог с файлами инвентаря.

  Если инвентарь проекта состоит из одного файла, допускается разместить его в корневом каталоге.

- `playbooks/` – каталог с файлами playbook.

  Допускается размещение файлов playbook в других каталогах.

  Примечание: В проект должен быть хотя бы один playbook.

- `playbooks/group_vars/` – каталог с файлами переменных Ansible.

  Допускается размещение файлов с переменными в других каталогах, а также присвоение значений в playbook и инвентаре.

  Подробности о порядке присвоения значений переменным см. в документе *Переменные*.

[116] https://docs.ansible.com/ansible/latest/reference_appendices/config.html

Проект как источник файлов инвентаря

Automation Controller позволяет использовать проекты для заполнения обычных инвентарных списков.

Поддерживается импорт инвентарных списков из отдельных файлов и каталогов.

Дополнительные параметры проекта

В настройках проекта можно указать дополнительные параметры:

- Среда исполнения.

  Если при создании шаблона не будет выбрана другая среда исполнения, то вместо среды исполнения по умолчанию будет использоваться значение, указанное в настройках проекта.

- Учетные данные для проверки подписи содержимого.

  Указанное полномочие типа «*GPG Public Key*» используется для проверки аутентичности исходного кода проекта.

  Если хотя бы один из подписанных файлов не пройдет проверку, обновление кода проекта будет завершено со статусом **Отказ** (Fail). Также контроллер заблокирует запуск всех заданий на основе шаблонов, использующих код проекта.

- URL системы управления исходными данными.

  Эта настройка доступна для проектов со следующими источниками управления исходными данными:

  - Git
  - Subversion
  - Внешний архив

  Для источников типа «Git» и «Subversion» необходимо указать URL репозитория.

Шаблоны

В этом разделе рассматриваются шаблоны заданий (Job Template) и шаблоны потоков заданий (Workflow Template).

Шаблоны заданий

Шаблон задания (Job Template) связывает между собой *инвентарь* и Ansible playbook из выбранного *проекта*.

Для шаблонов заданий поддерживаются следующие дополнительные параметры:

- тип задания (job type);
- среда исполнения (execution environment);
- полномочия (credentials);
- метки (labels);
- переменные (variables);
- ответвления (forks);
- лимит на управляемые узлы (limit);
- степень подробности вывода (verbosity);
- деление на срезы (job slicing);

- таймаут (timeout);
- показ изменений (show changes);
- теги задания (job tags) и пропуск тегов (skip tags);
- повышение привилегий (privilege escalation);
- параллельные задания (concurrent jobs);
- обратные вызовы процесса обеспечения сервиса (provisioning callbacks);
- хранилище фактов (fact storage);
- запрет отката группы узлов управления (prevent instance group fallback);
- настройки webhook (webhook).

Для некоторых параметров доступна опция **Запрос при запуске**. Если она включена, значение для связанного параметра можно задать позже:

- Если задание на основе шаблона запускается вне потока заданий – в момент запуска задания.
- Если шаблон задания используется в шаблоне потока заданий – в момент добавления шаблона задания в шаблон потока заданий.

Тип задания

Поддерживаются задания двух типов:

- Исполнение (Run) – на управляемых узлах будут выполнены необходимые задания автоматизации. Этот тип заданий используется по умолчанию.
- Проверка (Check) – выполняется проверка синтаксиса playbook, настроек среды исполнения и окружения, в котором будут выполняться задания автоматизации. Изменения в конфигурации управляемых узлов не производятся.

  Использование этого значения эквивалентно запуску команды `ansible-playbook` с параметром `--check`.

Среда исполнения

В настройках шаблона задания можно выбрать *среду исполнения*, используемую для запуска заданий. Эта среда исполнения имеет более высокий приоритет, чем заданная на уровне контроллера, организации и проекта.

Полномочия

Полномочия, используемые для доступа к управляемым узлам.

Поддерживается одновременное использование полномочий разных типов, но не более одной записи каждого типа.

Метки

С помощью меток можно группировать задания, созданные на основе шаблона. Метки также удобно использовать для поиска нужных записей журнала выполнения заданий или их логической группировки.

Переменные

Переменные шаблона задания позволяют задать значения переменных во время выполнения заданий.

Использование этого параметра эквивалентно запуску команды `ansible playbook` с одним или несколькими аргументами `--extra-vars` (`-e`).

В переменных шаблона задания используется формат `ключ:значение`.

Поддерживаются форматы YAML и JSON.

Подробности о порядке присвоения значений переменным см. в документе *Переменные*.

Ответвления

Количество параллельных процессов, используемых для выполнения playbook.

Если для этого параметра задано значение меньше 1, используется значение по умолчанию, равное 5.

Максимальное значение этого параметра задается в значении *системной настройки контроллера* **Максимальное количество ответвлений задания** (Maximum number of forks per job) и по умолчанию равно `200`.

Лимит на управляемые узлы

Фильтр управляемых узлов по их названию.

Если значение не указано, либо равно `all` или `*`, действие playbook распространяется на все узлы из выбранного инвентарного списка.

Значение по умолчанию – пустая строка (фильтрация узлов не используется).

Подробности о шаблонах названий узлов см. в документации Ansible[117].

Степень подробности

Детализация журнала выполнения задания:

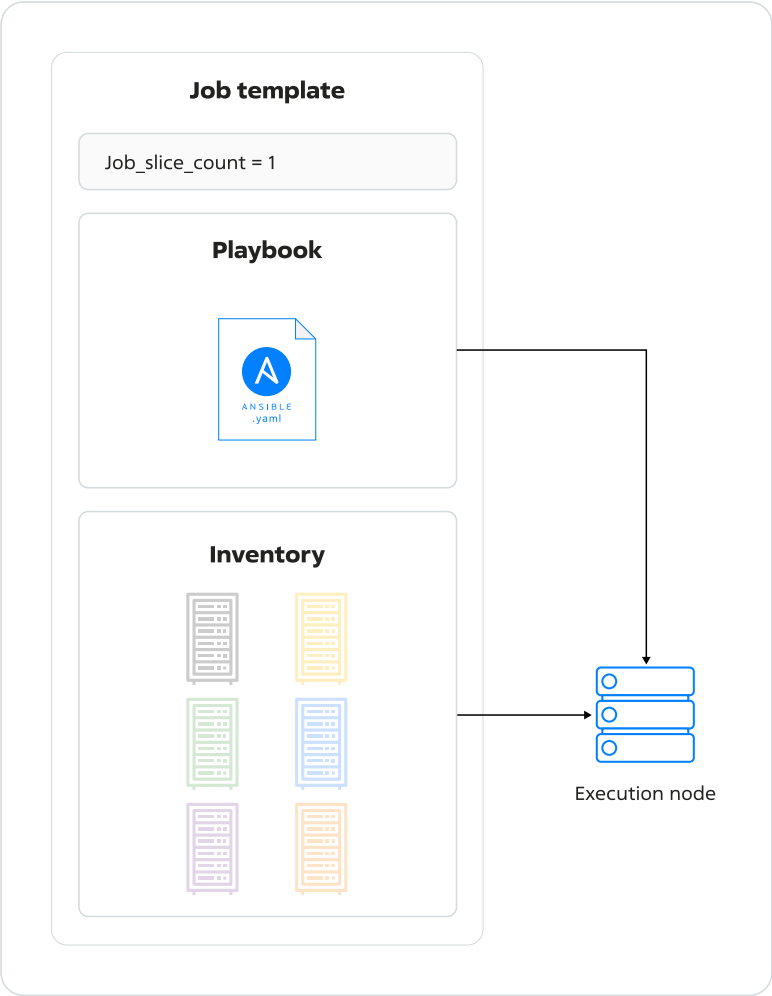
- `0` – нормальный (normal);
- `1` – подробный (verbose);
- `2` – более подробный (more verbose);
- `3` – отладка (debug);
- `4` – отладка подключения (connection debug);
- `5` – отладка WinRM (WinRM debug).

Значение по умолчанию – `0`.

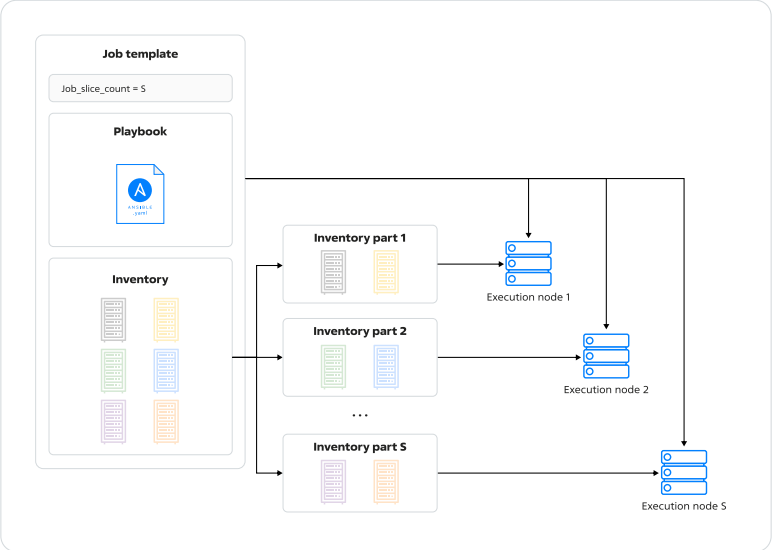
[117] https://docs.ansible.com/ansible/latest/inventory_guide/intro_patterns.html

Деление на срезы

По умолчанию количество срезов равно 1 – все задачи из playbook обычным образом запускаются с одним и тем же инвентарем на одном исполняющем узле.



Если количество срезов больше 1, контроллер делит список узлов исходного инвентаря на указанное количество частей (срезов) и вместо задания запускает поток заданий. Все задания в этом потоке планируются как выполняющиеся параллельно. Каждое задание запускается со своей частью управляемых узлов из исходного инвентаря.



Далее исходное задание называется *разделенным*, а задания потока, сформированного на

его основе, – *отдельными* заданиями.

При использовании деления на срезы контроллер реализует следующее поведение:

- Статус выполнения разделенного задания определяется так же как и статус выполнения потока заданий: если выполнение хотя бы одного отдельного задания завершилось ошибкой, неуспешным считается выполнение всего потока.

  Предупреждение: Не следует использовать деление на срезы для заданий, требующих взаимодействия между управляемыми узлами. Любое отдельное задание может завершиться ошибкой, но Automation Controller не будет пытаться обнаружить или учесть наборы сценариев (playbook), выполнение которых завершилось ошибкой при запуске разделенного задания.

- Каждое отдельное задание содержит информацию о шаблоне задания, инвентаре, количестве срезов и порядковом номере среза.
- Выполнение разделенного задания происходит в соответствии с обычным режимом планирования: учитываются настройки ответвлений, постановка в очередь и привязка к предпочтительным узлам.

  Примечание: Допускается установка крайне высоких значений количества срезов (порядка нескольких тысяч), однако, это может привести к снижению производительности, так как планировщик Automation Controller не предназначен для управления потоками заданий, задания которых одновременно исполняются на тысячах узлов. Для обеспечения высокой производительности рекомендуется использовать количество срезов, не превышающее количество исполняющих узлов.

- В отдельные задания передаются значения указанных пользователем и дополнительных переменных (extra vars), применяются все переменные и *лимиты на управляемые узлы*, указанные в шаблоне задания. Однако, если в отдельное задание переданы лимиты на управляемые узлы, при которых итоговый инвентарный список становится пустым, выполнение задания считается неуспешным.
- Отдельные задания могут выполняться на любом узле, при этом не всегда одновременно (например, из-за высокой нагрузки на систему).
- По умолчанию шаблоны заданий не допускают *параллельный запуск заданий*. Деление на срезы переопределяет это поведение и разрешает одновременное выполнение заданий, даже если эта настройка выключена в свойствах шаблона задания.

Таймаут

Время в секундах на выполнение задания. Если в течение указанного времени задание не выполняется, оно отменяется автоматически.

При значении `0` используется таймаут, заданный в системных настройках контроллера (значение по умолчанию – `0`, время выполнения заданий не ограничивается).

Отрицательное значение разрешает неограниченно долгое выполнение задания.

Значение по умолчанию – `0` (используется значение, заданное в глобальных настройках контроллера).

Показ изменений

Отображение изменений в конфигурации управляемых узлов, сделанных в результате выполнения заданий автоматизации.

Использование этого параметра эквивалентно запуску команды `ansible-playbook` с параметром `--diff`.

Примечание: Для отображения списка изменений необходима поддержка со стороны используемых модулей Ansible.

Теги

Эта настройка позволяет запускать или пропускать описанные в playbook задачи, отмеченные определенными тегами.

По умолчанию фильтрация по тегам не используется – выполняются все задачи, описанные в playbook.

Особенности настройки и применения:

- Названия тегов задаются одной строкой через запятую.
- Если заданы теги задания, то при запуске задания будут выполнены только задачи, отмеченные указанными тегами.

  Использование этого параметра эквивалентно запуску команды `ansible-playbook` с параметром `--tags`.
- Если заданы пропускаемые теги, то при запуске задания задачи с указанными тегами будут пропущены.

  Использование этого параметра эквивалентно запуску команды `ansible-playbook` с параметром `--skip-tags`.

Подробности о тегах см. в документации Ansible[118].

Повышение привилегий

Если эта настройка включена, playbook запускается с повышенными привилегиями.

Использование этого параметра эквивалентно запуску команды `ansible-playbook` с параметром `--become`.

Важно: Запрос пароля суперпользователя не производится. Убедитесь, что на управляемых узлах разрешено выполнение команды sudo без ввода пароля.

Инструкции по настройке приведены в секции *Использование sudo без ввода пароля*.

---

[118] https://docs.ansible.com/ansible/latest/playbook_guide/playbooks_tags.html

Параллельные задания

Если эта настройка включена, на основе шаблона можно запустить одновременное выполнение нескольких заданий с одним и тем же инвентарным списком.

Обратные вызовы процесса обеспечения сервиса

Если эта настройка включена, необходимо указать ключ конфигурации узла. Это URL, по которому узел может связаться с Automation Controller и запросить обновление конфигурации, используя шаблон задания.

Хранилище фактов

Если эта настройка включена, при запуске задания собранные об узлах *факты Ansible* сохраняются в кеше фактов. Затем собранные факты можно использовать, например, для создания *сборного инвентаря*.

Настройки webhook

Если эта настройка включена, запуск задания на основе шаблона будет происходить при наступлении определенного события в сервисе хранения исходного кода.

При включении этой настройки необходимо задать значения дополнительных параметров:

- Сервис – сервис хранения исходного кода.

  Возможные значения:

  - Bitbucket Data Center.
  - GitFlic.

    Added in version 1.1-upd1.
  - GitHub.
  - GitLab.
- Webhook URL – адрес, по которому доступен нужный webhook. Поле заполняется автоматически.
- Ключ webhook – ключ, используемый для подписи данных, отправляемых в Automation Controller. Значение из этого поля следует указать в параметрах webhook используемого сервиса хранения исходного кода.
- Полномочия webhook – полномочия, используемые для передачи данных обратно в webhook.

Шаблоны потоков заданий

Шаблоны потоков заданий (workflow templates) используются для реализации сложной логики управления выполнением заданий. Они поддерживают следующую функциональность:

- Запуск заданий при выполнении определенных условий[119] без необходимости их описания в playbook.
- Обновление кода проектов и инвентарных списков.
- Запуск заданий и потоков заданий только после одобрения пользователей (согласование).

[119] https://docs.ansible.com/ansible/latest/playbook_guide/playbooks_conditionals.html

- Параллельный запуск нескольких потоков заданий из одного потока.
- Синхронизация заданий, выполняемых параллельно.
- Запуск потоков заданий по расписанию.
- Рекурсивное использование шаблонов потоков заданий.

  По отношению к родительскому шаблону потока заданий такие шаблоны потоков заданий называются вспомогательными. Артефакты, собранные во время выполнения заданий вспомогательного шаблона, могут быть переданы в следующие узлы.

Шаблон потока заданий представляет собой направленный ациклический граф (*DAG*), начинающийся с корневого узла «НАЧАЛО» (START). Этот узел нельзя изменить или удалить.

Узлами графа могут выступать следующие компоненты и процессы:

- Согласование (approval).

  При переходе к узлу этого типа пользователь с ролью «Одобрить» (Approval) должен подтвердить или отклонить выполнение дальнейших действий.

- Обновление *инвентаря* (inventory update).

  Если инвентарь использует внешние источники, при переходе к узлу этого типа выполняется обновление сведений об управляемых узлах.

- *Шаблон задания* (job template).

  При переходе к узлу этого типа запускается задание на основе указанного шаблона.

- Обновление *проекта* (project update).

  При переходе к узлу этого типа выполняется обновление проекта, если его источником являются репозиторий Git, репозиторий Subversion или внешний архив.

- Шаблон потока заданий (workflow template).

  В роли узлов этого типа могут выступать уже существующие шаблоны потоков заданий.

- Служебное задание (management job).

  Служебные задания используются для выполнения действий, связанных с внутренним функционированием контроллера. Для узлов этого типа поддерживаются следующие служебные задания:

  - Cleanup Activity Stream – удаление записей в ленте активности;
  - Cleanup Expired OAuth 2 Tokens – очистка токенов OAuth 2 с истекшим сроком действия;
  - Cleanup Expired Sessions – очистка истекших сессий пользователей контроллера;
  - Cleanup Job Details – очистка сведений о выполнении задания.

Ветви графа могут быть одного из трех типов:

- успешное выполнение задания или согласование;
- ошибка при выполнении задания, его отмена или отказ в согласовании;
- безусловный переход, когда результат выполнения предыдущих этапов не имеет значения.

Особенности работы с инвентарем

Инвентарь может быть задан во время создания шаблона потока заданий или во время запуска потока заданий.

Если в настройках шаблона потока заданий включен выбор инвентаря при запуске, инвентарь может быть задан в настройках планировщика или узлов.

Если инвентарь указан в настройка шаблона задания, то значение из настроек шаблона потока заданий игнорируется.

Если в настройках шаблона задания включен выбор инвентаря при запуске, то используется инвентарь, заданный на уровне шаблона потока заданий.

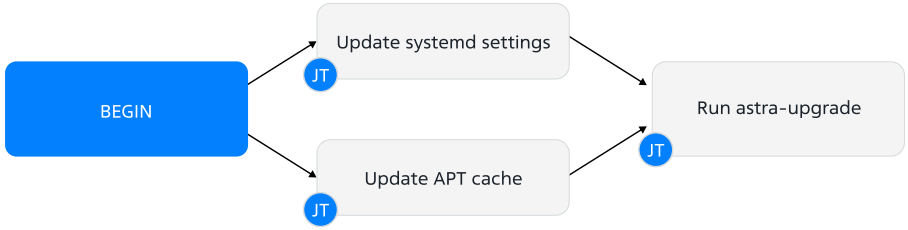
Конвергенция

Если узел имеет несколько входящих ветвей, настройки конвергенции определяют количество выполненных условий, необходимых для запуска задания узла. Поддерживаются следующие значения:

- **Любой** – достаточно успешного выполнения заданий на любом связанном узле;
- **Все** – должны быть успешно выполнены все условия на всех связанных узлах.

Пример

Пусть имеется шаблон потока заданий, показанный на схеме:



Узлы «Update systemd settings» и «Update APT cache» выполняют задания по обновлению настроек systemd и обновлению кеша APT соответственно. При этом указанные задания выполняются параллельно.

Задание «Run astra-upgrade» следует запускать только при успешном выполнении обоих предшествующих заданий. Для этого в настройках конвергенции следует выбрать значение «Все».

Важно: Порядок вычисления значения переменной `set_stats`, хранящей сведения о результатах выполнения задания, не определен. По этой причине для хранения сведений о результатах выполнения заданий рекомендуется использовать уникальные названия ключей.

Дополнительные переменные (extra_vars)

Значения переменных могут быть заданы следующими способами:

- Определение переменных и их значений в свойствах шаблона потока заданий.
- Интерактивный опрос пользователя при запуске потока заданий.
- Дополнительные переменные, переданные в момент запуска потока заданий.

Подробности о порядке присвоения значений переменным см. в документе *Переменные*.

Состояния

Шаблоны потоков заданий могут находиться в одном из следующих состояний:

- Ожидание (Waiting) – для запуска или продолжения выполнения заданий необходимо согласование.
- Выполнение (Running) – выполняются задания.
- Успех (Success) – выполнение задания завершено без ошибок.
- Отмена (Cancel) – выполнение задания отменено.
- Ошибка (Error) – при выполнении заданий произошли ошибки. Для обработки ошибок запущены соответствующие задания.
- Сбой (Failed) – при выполнении заданий произошли ошибки. Продолжение выполнения заданий невозможно.

Опросы

Опросом называется процесс получения значений переменных Ansible путем интерактивного взаимодействия с пользователем. Для этого пользователю, запустившему задание или поток заданий, предлагается ответить на ряд заранее подготовленных вопросов.

Состав вопросов и тип ожидаемых ответов определяется автором опроса.

Опросы обладают следующими свойствами:

- С каждым шаблоном может быть связан только один опрос.
- Каждый вопрос предусматривает получение ответа определенного типа.

  Поддерживаются следующие типы ответов:

  - **Текст** (Text) – строка символов.
  - **Текстовая область** (Textarea) – текст, состоящий из одной или более строк.
  - **Пароль** (Password) – защищенная строка символов.

    Для защиты ввода используется маскировка символов, для защиты значения в таблице базы данных контроллера – защитное преобразование.
  - **Множественный выбор (одиночное выделение)** (Multiple Choice (single select)) – выбор одного значения из предложенного списка.
  - **Множественный выбор (множественное выделение)** (Multiple Choice (multiple select)) – набор записей, выбранных из предложенного списка.
  - **Целое число** (Integer).
  - **Плавающее** (Float) – число с плавающей точкой, например, `3.1415926`.
- Для каждого вопроса можно задать значение ответа по умолчанию.

Подробности о порядке присвоения значений переменным см. в документе *Переменные*.

Задания и потоки заданий

В этом документе рассматриваются задания и потоки заданий.

Задания

Задания (jobs) бывают следующих типов:

- запуск playbook (run playbook) – выполнение playbook, указанного в *шаблоне задания*;
- служебное (management jobs) – используется для обслуживания самого контроллера;
- обновление проекта (source control update);
- синхронизация инвентаря (inventory sync);
- команда (command) – выполнение специальных команд (ad-hoc) на выбранных управляемых узлах.

Все задания контроллер ставит в очередь. Задания из очереди выполняются в том порядке, в котором они были в нее добавлены.

Если в шаблоне заданий включено ветвление, для выполнения задания будет использоваться указанное количество параллельных процессов.

Если в шаблоне задания разрешено деление на срезы, связанный с заданием инвентарь делится на части, и задания для управляемых узлов в каждой из них запускаются параллельно.

Запуск заданий может быть привязан к определенной группе узлов контроллера из плоскости исполнения. Для этого в настройках организации, шаблона заданий, шаблона потока заданий или инвентаря необходимо выбрать группу узлов контроллера.

Потоки заданий

Потоки заданий создаются на основе *шаблонов потоков заданий*. При запуске потока заданий обход направленного ациклического графа начинается с точки «НАЧАЛО» (START).

Служебные задания

В контроллере доступны следующие служебные задания:

- Cleanup Activity Stream – очистка истории активности;
- Cleanup Expired OAuth 2 Tokens – очистка токенов OAuth с истекшим сроком действия;
- Cleanup Expired Sessions – очистка данных истекших сессий;
- Cleanup Job Details – очистка истории выполнения заданий.

При выполнении потока заданий некоторые служебные задания вызываются неявно.

Задания синхронизации

Задания синхронизации используются для обновления локальной копии playbook и инвентаря и называются, соответственно, заданиями обновления проекта и заданиями обновления инвентаря.

При выполнении задания обновления проекта контроллер получает версию кода, указанную в настройках *источника*:

- если не указаны ветка, тег, ревизия или хеш коммита – последний коммит в основной ветке проекта;
- если указана ветка – последний коммит в указанной ветке;

- если указан тег – коммит, к которому привязан тег;
- если указана ревизия – указанную ревизию кода.

Если инвентарь использует один из файлов проекта в качестве источника сведений об управляемых узлах, то его автоматическое обновление при обновлении кода проекта не происходит. Для обновления сведений об управляемых узлах необходимо запустить задание обновления инвентаря.

Задания обновления могут находиться в одном из следующих статусов:

- **Назначено** (Pending) – задание создано, но еще не поставлено в очередь и не запущено;
- **Ожидание** (Waiting) – задание поставлено в очередь, но еще не запущено;
- **Выполнение** (Running) – задание выполняется;
- **Успех** (Successful) – задание выполнено без ошибок либо с ошибками, для которых предусмотрен обработчик;
- **Сбой** (Failed) – во время выполнения задания произошли ошибки, обработка которых не предусмотрена.

Журналирование

Контроллер позволяет просматривать вывод выполнения заданий и специальных команд в реальном времени. Журнал выполнения каждого задания сохраняется в контроллере.

Управление узлами

Группы узлов

Узлы можно объединять в группы узлов (instance groups). Один и тот же узел может входить в несколько групп узлов.

При развертывании контроллера все гибридные и исполняющие узлы автоматически включаются в группу узлов по умолчанию с названием `default`. Название этой группы нельзя изменить.

Выбор группы узлов для запуска заданий

Группы узлов, доступные на уровне контроллера, называются глобальными. По умолчанию задания запускаются на наиболее свободной глобальной группе узлов. Если все глобальные группы узлов заняты, задание ставится в очередь и запускается на узлах первой освободившейся глобальной группы.

Группы узлов могут быть *ассоциированы* с *организациями* (через проекты), *инвентарными списками* и *шаблонами заданий*. В этом случае для запуска заданий используются только ассоциированные группы узлов. Если они будут заняты, контроллер не будет использовать для запуска заданий глобальные группы узлов, а поставит задание в очередь и запустит его на первой освободившейся группе узлов из ассоциированных.

Если группы узлов ассоциированы сразу на нескольких уровнях, выбор групп узлов для запуска заданий выполняется в следующем порядке:

1. шаблон задания;
2. инвентарный список;
3. организация.

Например, пусть в контроллере существуют следующие группы узлов:

- `default`;

- south;
- north;
- east;
- west.

Пусть эти группы ассоциированы следующим образом:

1. шаблон задания – north;
2. инвентарный список – east и west;
3. организация – south.

В этом случае выбор группы узлов для запуска задания выполняется в следующем порядке:

1. Если свободна хотя бы одна группа, ассоциированная с шаблоном задания, задания запускаются на ее узлах.

   В данном случае проверяется статус группы north. Если она занята, выполняется проверка статуса групп узлов, ассоциированных с инвентарным списком.

2. Если свободна хотя бы одна группа, связанная с инвентарным списком, задания запускаются на ее узлах.

   В данном случае проверяется статус групп east и west. Если обе группы заняты, выполняется проверка статуса групп узлов, связанных с организацией.

3. Если свободна хотя бы одна группа, связанная с организацией, задания запускаются на ее узлах.

   В данном случае проверяется статус группы south. Если группа south тоже занята, задание ставится в очередь и выполняется на первой освободившейся группе узлов, ассоциированной с шаблоном задания, инвентарным списком или организацией.

Запрет отката группы узлов управления

Запрет отката группы узлов изменяет порядок выбора групп узлов для запуска заданий. В зависимости от уровня, на котором он включен, выбор групп узлов выполняется следующим образом:

- Шаблон задания.

  Если шаблон задания ассоциирован с группами узлов, контроллер будет ожидать их освобождения. Если список ассоциированных групп узлов пуст, используются глобальные группы узлов. При этом игнорируются группы, ассоциированные с инвентарным списком и организацией.

- Инвентарный список.

  Если инвентарный список ассоциирован с группами узлов, контроллер будет ожидать их освобождения. Если список ассоциированных групп узлов пуст, используются глобальные группы узлов. При этом игнорируются группы, ассоциированные с организацией.

Переменные

В этом документе описывается порядок разрешения значений переменных Ansible в Automation Controller.

Правило присвоения значения

При присвоении значений переменным используется следующий порядок (значения, указанные на более поздних этапах, заменяют значения, указанные ранее):

1. Значения по умолчанию, заданные в роли (role defaults).
2. Переменные динамического инвентаря (dynamic inventory variables).
3. Переменные инвентаря (inventory variables).
4. Переменные группы узлов (group variables).
5. Переменные управляемого узла (node variables).
6. `group_vars` в playbook.
7. `host_vars` в playbook.
8. Факты Ansible (facts).
9. Зарегистрированные переменные (registered variables).
10. Значения, заданные с помощью `set_facts`.
11. Переменные сценария (play variables)
12. Переменные из файлов переменных сценария (play `vars_files`).
13. Переменные роли (role variables) и подключенных (include) файлов.
14. Переменные блока (block variables).
15. Переменные задачи (task variables).
16. Дополнительные переменные из свойств шаблона задания (Job Template extra variables).
17. Переменные, полученные путем интерактивного опроса пользователя при запуске задания на основе шаблона (Job Template Survey). Если опрос пользователя выключен, используются значения по умолчанию.
18. Дополнительные переменные, заданные при запуске задания (Job Launch extra variables).
19. Артефакты задания.
20. Дополнительные переменные из свойств шаблона потока заданий (Workflow Job Template extra variables).
21. Переменные, полученные путем интерактивного опроса пользователя при запуске потока заданий на основе шаблона (Workflow Job Template Survey). Если опрос пользователя выключен, используются значения по умолчанию.
22. Дополнительные переменные, заданные при запуске потока заданий (Workflow Job Launch extra variables).

Пример

Пусть значение переменной `astra_linux_version` задано в нескольких местах:

- инвентарь – `1.7.1`;
- playbook – `1.7.5`;
- шаблон задания – `1.7.3`;
- шаблон потока заданий – `1.7.4`.

Шаблон потока заданий связывает вместе указанные инвентарь, playbook и шаблон задания. Согласно описанному выше порядку, для переменной `astra_linux_version` принимается значение `1.7.4`, заданное последним, на уровне шаблона потока заданий. Это же значение передается на уровни всех остальных компонентов, связанных с шаблоном потока заданий. Таким образом, при запуске задания и выполнении сценария для переменной `astra_linux_version` также будет использоваться значение `1.7.4`.

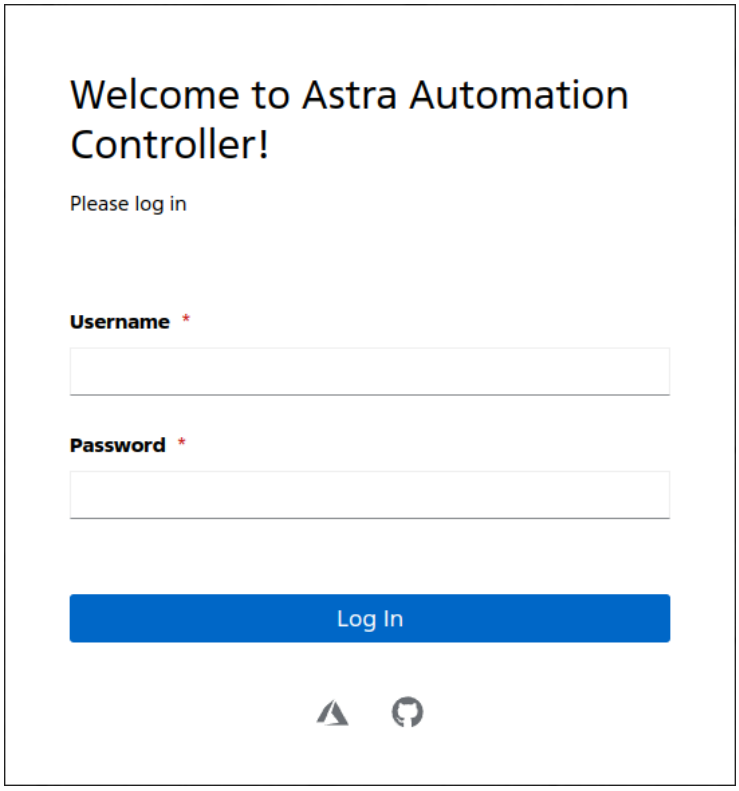
Методы аутентификации

Automation Controller может использовать для аутентификации пользователей внешние системы, называемые *поставщиками идентификационных данных* (Identity Provider, IdP). Далее с целью сокращения они называются провайдерами.

При первоначальной настройке необходимо установить доверительные отношения между контроллером и провайдером. Набор данных, используемых для идентификации контроллера на стороне провайдера аутентификации, зависит от типа провайдера и его собственных настроек.

После настройки аутентификация пользователя Automation Controller через провайдер выполняется по следующему алгоритму:

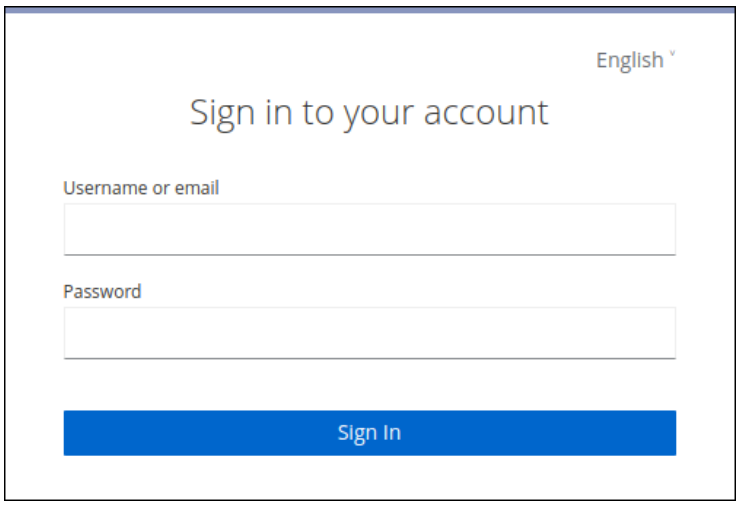
1. В Automation Controller на странице аутентификации становится доступна кнопка аутентификации с использованием провайдера.



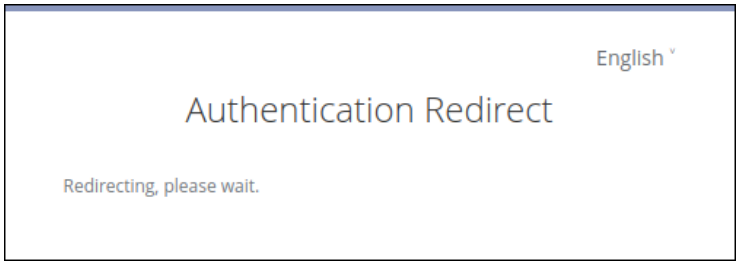
Примечание: Количество кнопок и их внешний вид зависят от количества и типов

настроенных провайдеров.

2. При нажатии на кнопку Automation Controller перенаправляет пользователя на страницу аутентификации, заданную в настройках провайдера.



3. Пользователь вводит свои учетные данные на указанной странице. Если идентификация успешна, провайдер перенаправляет пользователя на страницу аутентификации Automation Controller.



   Также провайдер передает в Automation Controller сведения о пользователе и токен, подтверждающий подлинность данных.

4. Automation Controller проверяет токен и наличие в своей базе учетной записи пользователя.

   Если такой записи нет, она создается на основе данных, полученных от провайдера аутентификации. Запись о пользователе Automation Controller может содержать следующие поля:

   - `username` – название учетной записи;
   - `first_name` – фамилия;
   - `last_name` – имя;
   - `email` – адрес электронной почты.

   Если в настройках Automation Controller заданы правила ассоциации пользователей с организациями и командами, Automation Controller выполняет следующие действия:

   1. Создает организации и команды, отсутствующие в Automation Controller.
   2. Включает пользователя в организации и команды в соответствии с заданными настройками.

      На уровне организации и команды пользователю автоматически назначается роль «Администратор» или «Участник». Прочие роли при необходимости должны быть назначены вручную.

Список провайдеров аутентификации, которые может использовать Automation Controller для реализации *SSO*:

- *Azure AD*;
- *GitHub*;
- *Google OAuth2*;
- *LDAP*;
- *RADIUS*;
- *SAML*;
- *TACACS+*;
- *OIDC*.

Azure AD

Для использования этого провайдера аутентификации необходимы ключ и секрет Azure AD. Пошаговые инструкции по их получению приведены в документации Azure[120].

Пошаговые инструкции по настройке провайдера через графический интерфейс Automation Controller приведены в секции *Azure AD*.

Ассоциация с организациями

Настройки ассоциации пользователей Azure AD с организациями контроллера необходимо задавать в виде словаря, в котором ключи – названия организаций Automation Controller, а значения – вложенные словари с настройками.

Словарь с настройками может содержать следующие ключи:

- `admins` – правила назначения роли «Администратор»;
- `remove_admins` – управление отзывом роли «Администратор»;
- `users` – правила назначения роли «Участник»;
- `remove_users` – управление отзывом роли «Участник».

Обработка значений параметров `admins` и `users` выполняется следующим образом:

- Если значение не задано – список администраторов и участников организации не меняется.
- `true` – всем пользователям Azure AD, выполнившим аутентификацию в Automation Controller, автоматически назначается роль «Администратор» или «Участник» соответственно.
- `false` – пользователям Azure AD автоматически назначается организационная роль «Администратор» или «Участник» соответственно. Аутентификация в Automation Controller не требуется.
- Если в значении параметра указана строка или массив строк, то на соответствие этим строкам проверяются названия учетных записей и адреса электронной почты пользователей Azure AD.

  Строки, в начале и конце которых стоит символ `/`, обрабатываются как регулярные выражения. Для регулярных выражений поддерживаются модификаторы, которые ставятся после закрывающей косой черты `/`:

  - `i` – игнорирование регистра символов;
  - `m` – многострочное выражение.

Обработка значений параметров `remove_admins` и `remove_users` выполняется следующим образом:

[120] https://learn.microsoft.com/en-us/entra/identity-platform/quickstart-register-app

- true – организационная роль отзывается.
- false – организационная роль не отзывается.

Пример заполнения словаря:

```
{
  "Jupiter": {
    "admins": "admin@jupiter.example.com",
    "users": [
      "/.*?@users\.example\.com$/i",
      "/.*?@devops\.example\.com$/i"
    ],
    "remove_admins": true,
    "remove_users": true
  }
}
```

Если организация «Jupiter» не существует, она будет создана.

Пользователю с адресом электронной почты admin@jupiter.example.com назначается роль «Администратор». У прочих пользователей Azure AD эта роль автоматически отзывается.

Пользователям, адрес электронной почты которых заканчивается на @users.example.com или @devops.example.com (регистр символов в адресе электронной почты игнорируется), назначается организационная роль «Участник». У прочих пользователей Azure AD эта роль автоматически отзывается.

Ассоциация с командами

Настройки ассоциации пользователей Azure AD с командами контроллера необходимо задавать в виде словаря, где ключи – названия команды в контроллере, а значения – вложенные словари с настройками.

Словарь с настройками может содержать следующие ключи:

- organization – название организации, которой принадлежит команда.
  - Если организация с указанным названием не существует, она будет создана.
  - Если в организации не существует команда с указанным названием, она будет создана.
- users – параметр, определяющий членство пользователей в команде.
  - Если значение параметра не задано, список участников команды не меняется.
  - Если значение параметра равно true, пользователям назначается командная роль «Участник».
  - Если значение параметра равно false, у пользователей отзывается командная роль «Участник».
  - Если в значении параметра указана строка или массив строк, то на соответствие этим строкам проверяются названия учетных записей и адреса электронной почты пользователей Azure AD.

  Строки, в начале и конце которых стоит символ /, обрабатываются как регулярные выражения. Для регулярных выражений поддерживаются модификаторы, которые ставятся после закрывающей косой черты /:
  - i – игнорирование регистра символов;
  - m – многострочное выражение.
- remove – если значение этого параметра равно true, у не соответствующих критериям отбора пользователей Azure AD автоматически отзывается роль «Участник».

Пример заполнения словаря:

```
{
  "DevOps Team": {
    "organization": "Jupiter",
    "users": "/.*?@jupiter\.example\.com$/",
    "remove": true
  }
}
```

Если организация «Jupiter» не существует, она будет создана.

Если в организации «Jupiter» не существует команда «DevOps Team», она будет создана.

Пользователям, адрес электронной почты которых заканчивается на `@jupiter.example.com`, автоматически назначается роль «Участник».

GitHub

Важно: Эта часть документации находится в стадии разработки.

Google OAuth2

Важно: Эта часть документации находится в стадии разработки.

LDAP

Automation Controller поддерживает использование нескольких конфигураций провайдера аутентификации LDAP одновременно.

Для использования метода аутентификации LDAP необходимы следующие данные:

- URI серверов LDAP;
- пароли привязки к LDAP;
- запрос для поиска пользователей в LDAP;
- запрос для поиска групп в LDAP.

Для получения этих данных обратитесь к документации используемого провайдера аутентификации LDAP.

Автоматическая синхронизация пользователей LDAP с Automation Controller не выполняется.

При успешной аутентификации в LDAP для пользователя автоматически создается учетная запись в Automation Controller. У созданных таким образом учетных записей нельзя изменить название учетной записи, пароль, имя и фамилию пользователя.

По умолчанию Automation Controller использует для подключения к серверам LDAP настройки, обеспечивающие корректную работу с реализацией протокола в Microsoft Active Directory. Для прочих реализаций может быть использован один из обработчиков:

- `ActiveDirectoryGroupType`;
- `GroupOfNamesType`;
- `GroupOfUniqueNamesType`;
- `MemberDNGroupType`;
- `NestedActiveDirectoryGroupType`;

- NestedGroupOfNamesType;
- NestedGroupOfUniqueNamesType;
- NestedMemberDNGroupType;
- NestedOrganizationalRoleGroupType;
- OrganizationalRoleGroupType;
- PosixGroupType;
- PosixUIDGroupType.

Подробное описание каждого типа обработчика см. в документации[121] библиотеки `django-auth-ldap`.

Ассоциация данных пользователей

Настройки ассоциации данных пользователей LDAP с данными пользователей Automation Controller необходимо задавать в виде словаря, в котором ключи – названия полей данных пользователя Automation Controller, а значения – названия полей данных пользователя LDAP, например:

```
{
  "first_name": "givenName",
  "last_name": "sn",
  "email": "mail"
}
```

Назначение ролей «Системный администратор» и «Системный аудитор»

Возможно автоматическое назначение пользователям LDAP типов «Системный администратор» (System Administrator) и «Системный аудитор» (System Auditor). Настройки необходимо задавать в виде словаря, в котором ключи – `is_superuser` и `is_system_auditor` соответственно, а значения – запросы к LDAP для фильтрации пользователей.

Пример заполнения словаря:

```
{
  "is_superuser": "cn=superuser,ou=groups,dc=example,dc=com",
  "is_system_auditor": "cn=auditors,ou=groups,dc=example,dc=com"
}
```

Ассоциация с организациями

Настройки ассоциации пользователей LDAP с организациями контроллера необходимо задавать в виде словаря, в котором ключи – названия организаций Automation Controller, а значения – вложенные словари с настройками.

Вложенный словарь может содержать следующие ключи:

- `admins` – правила назначения роли «Администратор»;
- `remove_admins` – управление отзывом роли «Администратор»;
- `users` – правила назначения роли «Участник»;
- `remove_users` – управление отзывом роли «Участник».

Обработка значение параметров `admins` и `users` выполняется следующим образом:

- Если значение не задано – список администраторов и участников организации не меняется.

[121] https://django-auth-ldap.readthedocs.io/en/stable/groups.html#types-of-groups

- `true` – всем пользователям LDAP, выполнившим аутентификацию в Automation Controller, автоматически назначается роль «Администратор» или «Участник» соответственно.
- `false` – пользователям LDAP автоматически назначается роль «Администратор» или «Участник» соответственно.
- Если в значении параметра указана строка или массив строк, они интерпретируются как запрос к LDAP. Найденным пользователям автоматически назначается роль «Администратор» или «Участник» соответственно.

Пример заполнения словаря:

```
{
  "Jupiter": {
    "admins": "CN=admins,OU=groups,DC=jupiter,DC=example,DC=com",
    "users": [
      "CN=devops,OU=groups,DC=jupiter,DC=example,DC=com",
      "CN=testers,OU=groups,DC=jupiter,DC=example,DC=com"
    ],
    "remove_admins": true,
    "remove_users": true
  }
}
```

Если организация «Jupiter» не существует, она будет создана.

Пользователям, найденным при выполнении запроса `CN=admins,OU=groups,DC=jupiter,DC=example,DC=com`, назначается роль «Администратор». У прочих пользователей LDAP эта роль автоматически отзывается.

Пользователям, найденным при выполнении запросов `CN=devops,OU=groups,DC=jupiter,DC=example,DC=com` и `CN=terster,OU=groups,DC=jupiter,DC=example,DC=com`, назначается роль «Участник». У прочих пользователей LDAP эта роль автоматически отзывается.

Ассоциация с командами

Настройки ассоциации пользователей LDAP с командами контроллера необходимо задавать в виде словаря, где ключи – названия команд в контроллере, а значения – вложенные словари с настройками.

Словарь с настройками может содержать следующие ключи:

- `organization` – название организации, которой принадлежит команда.
  - Если организация с указанным названием не существует, она будет создана.
  - Если в организации не существует команда с указанным названием, она будет создана.
- `users` – параметр, определяющий назначение пользователю роли «Участник»:
  - Если значение параметра не задано, список участников команды не меняется.
  - Если значение параметра равно `true`, пользователю назначается роль «Участник».
  - Если значение параметра равно `false`, у пользователя отзывается роль «Участник».
  - Если в значении параметра указана строка или массив строк, то они интерпретируются как запросы к LDAP. Найденным пользователям назначается роль «Участник».
- `remove` – если значение этого параметра равно `true`, не соответствующие критериям отбора пользователи LDAP удаляются из списка участников команды.

Пример заполнения словаря:

```
{
  "DevOps Team": {
    "organization": "Jupiter",
    "users": "CN=devops,OU=groups,DC=jupiter,DC=example,DC=com",
    "remove": true
  }
}
```

Если организация «Jupiter» не существует, она будет создана.

Если в организации «Jupiter» не существует команда «DevOps Team», она будет создана.

Пользователям, найденным в LDAP при выполнении запроса `CN=devops,OU=groups,DC=jupiter,DC=example,DC=com`, автоматически назначается роль «Участник».

RADIUS

> **Предупреждение:** Этот провайдер аутентификации считается устаревшим и в одной из следующих версий Automation Controller станет недоступным для применения.

SAML

В Astra Automation *поставщиком услуг* является кластер Automation Controller, а поставщиком идентификационных данных – группа серверов SAML, предоставляющих услугу *SSO*.

В качестве идентификатора объекта поставщика услуг SAML следует указать базовый URL Automation Controller. Если для доступа к кластеру используется балансировщик нагрузки, следует указать его URL, например:

```
https://aac.example.com/
```

Открытый сертификат и закрытый ключ должны быть указаны с маркерами начала и конца содержимого:

Открытый сертификат

```
-----BEGIN CERTIFICATE-----
AAAAA231jfdsalfkjkl........
-----END CERTIFICATE-----
```

Закрытый ключ

```
-----BEGIN PRIVATE KEY-----
PPPPP2183910das0921........
-----END PRIVATE KEY-----
```

Информацию об организации поставщика услуг SAML необходимо указывать в виде словаря, где ключи – названия локалей, а значения – вложенные словари с настройками.

Словарь с настройками может содержать следующие ключи:

- `url` – URL кластера Automation Controller.
- `displayname` – человекочитаемое название кластера Automation Controller.
- `name` – внутреннее название кластера Automation Controller. Используется в качестве идентификатора на стороне провайдера SAML, позволяя ему отличать один кластер от другого.

Пример заполнения словаря:

```
{
  "ru-RU": {
    "url": "https://aac.example.com",
    "displayname": "Кластер Automation Controller",
    "name": "aac-cluster"
  }
}
```

Атрибуты пользователей

По умолчанию Automation Controller ожидает от провайдера SAML данные пользователя в следующем формате:

```
Username(urn:oid:0.9.2342.19200300.100.1.1)
Email(urn:oid:0.9.2342.19200300.100.1.3)
FirstName(urn:oid:2.5.4.42)
LastName(urn:oid:2.5.4.4)
```

Если на стороне провайдера SAML данные о пользователях хранятся в полях с другими названиями, необходимо задать настройки ассоциации атрибутов пользователей SAML с атрибутами пользователей Automation Controller.

Настройки ассоциации атрибутов пользователей SAML с атрибутами пользователей Automation Controller необходимо задавать в виде словаря, где ключи – идентификаторы провайдеров идентификации, а значения – вложенные словари с настройками.

Словарь с настройками может содержать следующие ключи:

- `attr_user_permanent_id` – название атрибута пользователя SAML, в котором хранится уникальный идентификатор пользователя.
- `attr_username` – название атрибута SAML, в котором хранится название учетной записи пользователя.
- `attr_first_name` – название атрибута SAML, в котором хранится имя пользователя.
- `attr_last_name` – название атрибута SAML, в котором хранится фамилия пользователя.
- `attr_email` – название атрибута SAML, в котором хранится адрес электронной почты пользователя.

  > **Предупреждение:** Использование одного и того же адреса электронной почты, в том числе для пользователей, созданных средствами Automation Controller, запрещено.

- `entity_id` – идентификатор профиля, используемого для аутентификации Automation Controller. Предоставляется провайдером SAML.
- `url` – URL страницы аутентификации SAML. Automation Controller будет перенаправлять пользователей на эту страницу при аутентификации через SAML. Предоставляется провайдером SAML.
- `x509_cert` – сертификат провайдера идентификации SAML.

  При заполнении этого поля необходимо убрать из сертификата маркеры начала (`-----BEGIN CERTIFICATE`) и конца (`-----END CERTIFICATE----`), а оставшиеся строки объединить в одну.

Пример заполнения словаря:

```
{
  "keycloak": {
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    "entity_id": "https://saml.example.com/keycloak/b2912kdsjkl1k",
    "url": "https://saml.example.com/keycloak/sso/",
    "x509cert": "MIID0DCCAj12309jklwqW...3123jklfdsajkl213",
    "attr_user_permanent_id": "user_id",
    "attr_username": "Email",
    "attr_email": "Email",
    "attr_first_name": "FirstName",
    "attr_last_name": "LastName"
  }
}
```

Ассоциация атрибутов организаций

Настройки ассоциации атрибутов SAML с организациями необходимо задавать в виде словаря, который может содержать следующие ключи:

- `saml_attr` – если в данных пользователя SAML есть атрибут с указанным названием, этому пользователю автоматически назначается роль «Участник» в *соответствующих организациях контроллера*.
- `saml_admin_attr` – если в данных пользователя SAML есть атрибут с указанным названием, этому пользователю автоматически назначается роль «Администратор» в *соответствующих организациях контроллера*.
- `remove` – если значение этого параметра равно `true`, у не соответствующих критериям отбора пользователей SAML автоматически отзывается роль «Участник».
- `remove_admins` – если значение этого параметра равно `true`, у не соответствующих критериям отбора пользователей SAML автоматически отзывается роль «Администратор».

Пусть имеется следующее описание атрибутов SAML:

```
<saml2:AttributeStatement>
  <saml2:Attribute FriendlyName="is-member"
                   Name="is-member"
                   NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">

    <saml2:AttributeValue>org.jptr</saml2:AttributeValue>
    <saml2:AttributeValue>org.strn</saml2:AttributeValue>
  </saml2:Attribute>

  <saml2:Attribute FriendlyName="is-admin"
                   Name="is-admin"
                   NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">

    <saml2:AttributeValue>org.jptr</saml2:AttributeValue>
  </saml2:Attribute>
</saml2:AttributeStatement>
```

Здесь:

- атрибут `is-member` указывает, что пользователь является участником организаций SAML `org.jptr` и `org.strn`;
- атрибут `is-admin` указывает, что пользователь является администратором организации SAML `org.jptr`.

Соответствующий словарь ассоциаций атрибутов организаций имеет следующий вид:

```
{
  "saml_attr": "is-member",
  "saml_admin_attr": "is-admin",
  "remove": true,
```

(continues on next page)

(продолжение с предыдущей страницы)

```
  "remove_admins": true
}
```

Пользователям SAML, обладающим атрибутом `is-member`, назначается роль «Участник» в соответствующих организациях контроллера. У прочих пользователей SAML эта роль автоматически отзывается.

Пользователям SAML, обладающим атрибутом `is-admin`, назначается роль «Администратор» в соответствующих организациях контроллера. У прочих пользователей SAML эта роль автоматически отзывается.

Ассоциация атрибутов команд

Настройки ассоциации атрибутов SAML с командами необходимо задавать в виде словаря, который может содержать следующие ключи:

- `saml_attr` – если в данных пользователя SAML есть атрибут с указанным названием, этому пользователю автоматически назначается роль «Участник» в *соответствующих командах контроллера*.
- `team_org_map` – массив словарей, содержащих настройки ассоциации пользователей SAML с командами контроллера.

  Каждый словарь в этом массиве может содержать следующие ключи:

  - `team` – название команды.
  - `team_alias` – псевдоним команды.
  - `organization` – название организации, которой принадлежит команда.
  - `organization_alias` – псевдоним организации, которой принадлежит команда.
    - Если организация контроллера с указанным названием не существует, она будет создана.
    - Если в организации контроллера не существует команда с указанным названием, она будет создана.

  Примечание: Ключи `team_alias` и `organization_alias` не обязательны для заполнения, но они могут быть полезны, если на стороне поставщика услуг SAML используются сложные названия организаций и команд. Если значения ключей `team_alias` и `organization_alias` не заданы, организации и команды будут созданы в контроллере с названиями, полученными из атрибутов, указанных в поле `team` и `organization` соответственно.

- `remove` – если значение этого параметра равно `true`, у не соответствующих критериям отбора пользователей SAML автоматически отзывается роль «Участник».

Пусть имеется следующее описание атрибутов SAML:

```
<saml:AttributeStatement>
  <saml:Attribute xmlns:x500="urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
                  x500:Encoding="LDAP"
                  NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
                  Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.1"
                  FriendlyName="team">
    <saml:AttributeValue xsi:type="xs:string">data:team:devOps</saml:AttributeValue>
    <saml:AttributeValue xsi:type="xs:string">data:team:auditors</saml:AttributeValue>
  </saml:Attribute>
</saml:AttributeStatement>
```

Здесь атрибут `team` указывает, что пользователь является участником команд SAML `data:team:devOps` и `data:team:auditors`.

Соответствующий словарь ассоциаций атрибутов команды имеет следующий вид:

```
{
  "saml_attr": "team",
  "remove": true,
  "team_org_map": [
    {
      "team": "data:team:devOps",
      "team_alias": "DevOps Team"
      "organization": "org.jptr"
    }, {
      "team": "data:team:auditors",
      "team_alias": "Auditors Team",
      "organization": "org.jptr"
    }
  ]
}
```

Если в контроллере не существует организация `org.jptr`, она будет создана.

Если в организации контроллера `org.jptr` не существуют команды «DevOps Team» и «Auditors Team», они будут созданы. Этим командам контроллера соответствуют команды SAML `data:team:devOps` и `data:team:auditors` соответственно.

Пользователям SAML, обладающим атрибутом `team`, хотя бы одно из значений которого равно `data:team:devOps`, автоматически назначается роль «Участник» в команде «DevOps Team». У прочих пользователей эта роль автоматически отзывается.

Пользователям SAML, обладающим атрибутом `team` со значением, равным `data:team:auditors`, автоматически назначается роль «Участник» в команде «Auditors Team». У прочих пользователей эта роль автоматически отзывается.

Ассоциация с организациями

Настройки ассоциации пользователей SAML с организациями контроллера необходимо задавать в виде словаря, в котором ключи – названия организаций Automation Controller, а значения – вложенные словари с настройками.

Словарь с настройками может содержать следующие ключи:

- `admins` – правила назначения роли «Администратор»;
- `remove_admins` – управление отзывом роли «Администратор»;
- `users` – правила назначения роли «Участник»;
- `remove_users` – управление отзывом роли «Участник».

Обработка значений параметров `admins` и `users` выполняется следующим образом:

- Если значение не задано – список администраторов и участников организации не меняется.
- `true` – всем пользователям SAML, выполнившим аутентификацию в Automation Controller, автоматически назначается роль «Администратор» или «Участник» соответственно.
- `false` – пользователям SAML автоматически назначается организационная роль «Администратор» или «Участник» соответственно. Аутентификация в Automation Controller не требуется.
- Если в значении параметра указана строка или массив строк, то на соответствие этим строкам проверяются названия учетных записей пользователей SAML.

  Строки, в начале и конце которых стоит символ `/`, обрабатываются как регулярные выражения. Для регулярных выражений поддерживаются модификаторы, которые ставятся после закрывающей косой черты `/`:

  - `i` – игнорирование регистра символов;

  - m – многострочное выражение.

Обработка значений параметров remove_admins и remove_users выполняется следующим образом:

- true – организационная роль отзывается.
- false – организационная роль не отзывается.

Пример заполнения словаря:

```
{
  "Jupiter": {
    "admins": "admin@jupiter.example.com",
    "users": [
      "/.*?@users\.example\.com$/i",
      "/.*?@devops\.example\.com$/i"
    ],
    "remove_admins": true,
    "remove_users": true
  }
}
```

Если организация «Jupiter» не существует, она будет создана.

Пользователю с названием учетной записи admin@jupiter.example.com назначается роль «Администратор». У прочих пользователей SAML эта роль автоматически отзывается.

Пользователям, название учетной записи которых заканчивается на @users.example.com или @devops.example.com (без учета регистра), назначается организационная роль «Участник». Пользователи, не удовлетворяющие указанному условию, автоматически удаляются из списка участников организации «Jupiter».

Ассоциация с командами

Настройки ассоциации пользователей SAML с командами контроллера необходимо задавать в виде словаря, где ключи – названия команды в контроллере, а значения – вложенные словари с настройками.

Словарь с настройками может содержать следующие ключи:

- organization – название организации, которой принадлежит команда.
  - Если организация с указанным названием не существует, она будет создана.
  - Если в организации не существует команда с указанным названием, она будет создана.
- users – параметр, определяющий членство пользователей в команде.
  - Если значение параметра не задано, список участников команды не меняется.
  - Если значение параметра равно true, пользователям назначается командная роль «Участник».
  - Если значение параметра равно false, у пользователей отзывается командная роль «Участник».
  - Если в значении параметра указана строка или массив строк, то на соответствие этим строкам проверяются названия учетных записей SAML. В *настройках ассоциации атрибутов пользователей* указывается, какой из атрибутов SAML считать названием учетной записи.

  Строки, в начале и конце которых стоит символ /, обрабатываются как регулярные выражения. Для регулярных выражений поддерживаются модификаторы, которые ставятся после закрывающей косой черты /:

  - i – игнорирование регистра символов;
  - m – многострочное выражение.

- remove – если значение этого параметра равно true, у не соответствующих критериям отбора пользователей SAML автоматически отзывается роль «Участник».

Пример заполнения словаря:

```
{
  "DevOps Team": {
    "organization": "Jupiter",
    "users": "/.*?@jupiter\.example\.com$/",
    "remove": true
  }
}
```

Если организация «Jupiter» не существует, она будет создана.

Если в организации «Jupiter» не существует команда «DevOps Team», она будет создана.

Пользователям, название учетной записи которых заканчивается на @jupiter.example.com, автоматически назначается роль «Участник» на уровне команды «DevOps Team».

Ассоциация атрибутов и ролей SAML с ролями контроллера

Настройки ассоциации атрибутов и ролей SAML с типами пользователей Automation Controller необходимо задавать в виде словаря, который может содержать следующие ключи:

- is_superuser_role – название роли или ролей SAML, при наличии которых пользователю Automation Controller назначается роль «Системный администратор».
- is_superuser_attr – название атрибута SAML, используемого для принятия решения о назначении пользователю Automation Controller роли «Системный администратор».
- is_superuser_value – значение атрибута SAML, необходимое для назначения пользователю Automation Controller роли «Системный администратор».
- remove_superusers – управление автоматическим отзывом роли «Системный администратор» у пользователей, не соответствующим критериям:
  - true – роль отзывается;
  - false – роль не отзывается.
- is_system_auditor_role – название роли или ролей SAML, при наличии которых пользователю Automation Controller назначается роль «Системный аудитор».
- is_system_auditor_attr – название атрибута SAML, используемого для принятия решения о назначении пользователю Automation Controller роли «Системный аудитор».
- is_system_auditor_value – значение атрибута SAML, необходимое для назначения пользователю Automation Controller роли «Системный аудитор».
- remove_system_auditors – управление автоматическим отзывом роли «Системный аудитор» у пользователей, не соответствующим критериям:
  - true – роль отзывается;
  - false – роль не отзывается.

Эти настройки имеют следующие особенности:

- Поля с role в названии и поля с value являются списками, и при обработке их значений используется логическое «ИЛИ». Это значит, что для назначения соответствующей роли пользователь SAML должен иметь *хотя бы одну* роль из списка, заданного в настройках ассоциаций.
- Если одновременно заданы значения параметров с role и attr в названии, то настройки с attr имеют более высокий приоритет перед role.

- Назначение ролей «Системный администратор» и «Системный аудитор» происходит при каждом входе пользователя в Automation Controller. Если назначить пользователю роль «Системный администратор» или «Системный аудитор» через графический интерфейс Automation Controller, а не через настройки SAML, то при следующем входе пользователя в систему назначенная роль будет отозвана.

  Чтобы предотвратить отзыв ролей «Системный администратор» или «Системный аудитор», укажите в ключах `remove_superusers` и `remove_system_auditors` соответственно значение `false`.

  Логика работы Automation Controller при отзыве и назначении ролей «Системный администратор» и «Системный аудитор» показана в таблице:

Наличие роли SAML	Наличие атрибута SAML	Наличие нужного значения атрибута SAML	Де
Нет	Нет	—	tr
Нет	Нет	—	fa
Нет	Нет	—	tr
Нет	Нет	—	fa
Да	Нет	—	tr
Да	Нет	—	fa
Да	Нет	—	tr
Да	Нет	—	fa
Нет	Да	Да	tr
Нет	Да	Да	fa
Нет	Да	Да	tr
Нет	Да	Да	fa
Нет	Да	Нет	tr
Нет	Да	Нет	fa
Нет	Да	Нет	tr
Нет	Да	Нет	fa
Нет	Да	Не задано	tr
Нет	Да	Не задано	fa
Нет	Да	Не задано	tr
Нет	Да	Не задано	fa
Да	Да	Да	tr
Да	Да	Да	fa
Да	Да	Да	tr
Да	Да	Да	fa
Да	Да	Нет	tr
Да	Да	Нет	fa
Да	Да	Нет	tr
Да	Да	Нет	fa
Да	Да	Не задано	tr
Да	Да	Не задано	fa
Да	Да	Не задано	tr
Да	Да	Не задано	fa

- Если значение ключа `remove_superusers` или `remove_system_auditors` равно `false`, соответствующая роль не будет отозвана у пользователя SAML никогда.

Пример заполнения словаря:

```
{
  "is_superuser_role": [
    "admin",
    "root"
  ],
  "is_superuser_attr": "groups",
  "is_superuser_value": [
    "admin",
    "root",
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    "astra-admin"
  ],
  "is_system_auditor_role": "auditor",
  "is_system_auditor_attr": "groups",
  "is_system_auditor_value": ["auditor"]
}
```

Здесь:

- Роль «Системный администратор» назначается пользователям SAML, в отношении которых выполняется хотя бы одно из условий:
  - наличие роли `admin` или `root`;
  - наличие атрибута `groups`, хотя бы одним из значений которого является `admin`, `root` или `astra-admin`.
- Роль «Системный аудитор» назначается пользователям SAML, в отношении которых выполняется хотя бы одно из условий:
  - наличие роли `auditor`;
  - наличие атрибута `groups`, хотя бы одним из значений которого является `auditor`.
- Значения ключей `remove_superusers` и `remove_system_auditors` не заданы, поэтому используются значения по умолчанию – `true`. Это значит, что у всех пользователей SAML, не удовлетворяющих указанным требованиям, роли «Системный администратор» и «Системный аудитор» будут отозваны при первом входе в систему независимо от того, каким образом они были назначены – через настройки SAML, API или графический интерфейс Automation Controller.

TACACS+

Предупреждение: Этот провайдер аутентификации считается устаревшим и в одной из следующих версий Automation Controller станет недоступным для применения.

OIDC

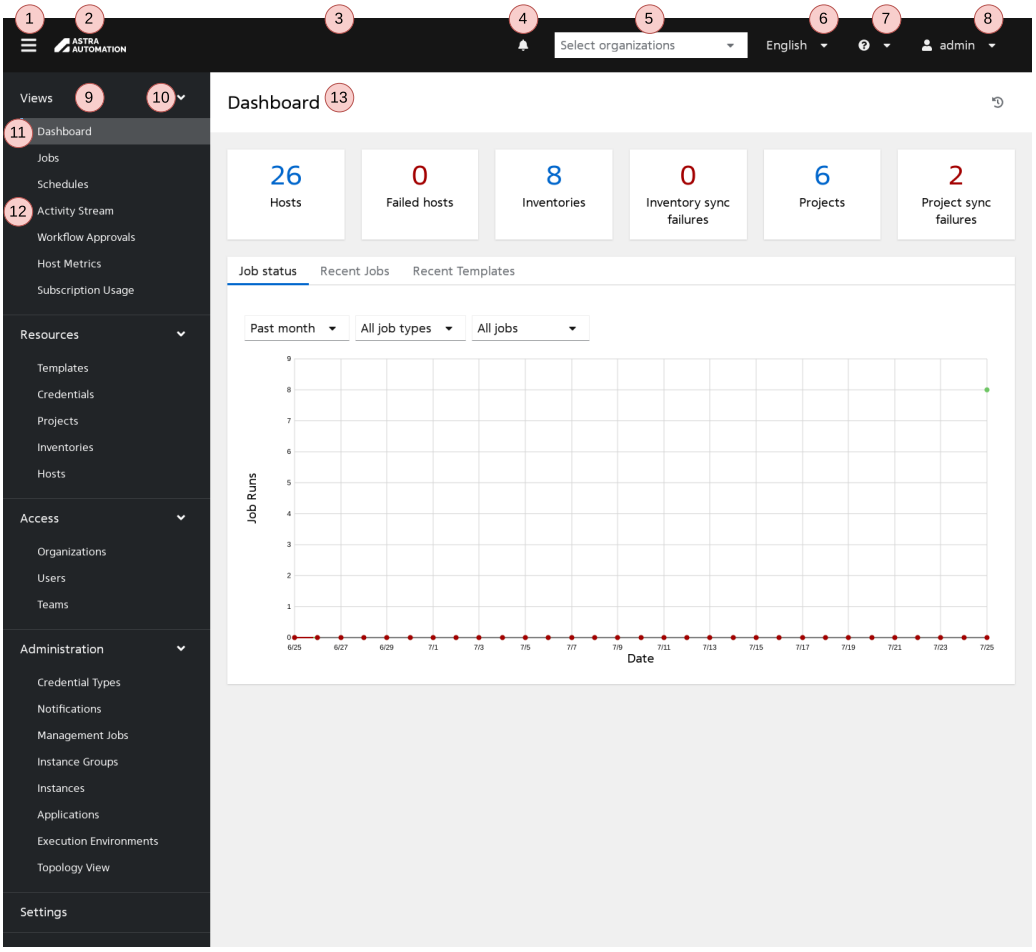
Важно: Эта часть документации находится в стадии разработки.

7.3 Графический интерфейс

Automation Controller предоставляет удобный веб-интерфейс для управления его работой. Расположенная в левой части панель навигации предоставляет доступ к управлению всеми ресурсами.

7.3.1 Основные компоненты

На схеме представлены основные компоненты пользовательского интерфейса контроллера:



Здесь:

1. Кнопка переключения видимости панели навигации.
2. Логотип.
3. Панель инструментов.
4. Кнопка-индикатор, показывающая количество уведомлений.
5. Глобальный фильтр по организациям.
6. Кнопка смены языка.
7. Кнопка вызова справки.
8. Пользовательское меню.
9. Панель навигации.
10. Кнопка переключения видимости раздела навигации.
11. Активный раздел навигации.
12. Неактивные разделы навигации.
13. Заголовок окна.

7.3.2 Режимы просмотра

Раздел *Режимы просмотра* (Views) содержит ссылки на окна, содержащие различные сведения о работе контроллера и выполнении заданий:

- **Информационная панель** (Dashboard) содержит краткую сводку о состоянии самых важных ресурсов контроллера – проектах, инвентарных списках и управляемых узлах.

  В верхней части окна размещаются панели:

  - **Управляемые узлы** (Hosts) – общее количество управляемых узлов во всех инвентарных списках.
  - **Узлы с ошибками** (Failed Hosts) – количество узлов, при выполнении заданий на которых произошли ошибки.
  - **Инвентарь** (Inventories) – количество существующих инвентарных списков.
  - **Ошибки синхронизации инвентаря** (Inventory Sync failures) – количество ошибок при выполнении заданий синхронизации инвентарных списков.
  - **Проекты** (Projects) – количество проектов.
  - **Проекты с ошибками синхронизации** (Project sync failures) – количество существующих проектов.

  Нажатие на панель приводит к переходу к соответствующему разделу на панели навигации.

  Нижнюю часть окна занимает панель с вкладками:

  - **Статус задания** (Job status) – график, отображающий количество выполненных заданий и заданий с ошибками.
  - **Последние задания** (Recent Jobs) – таблица со списком заданий, статусом их выполнения, временем начала и завершения.
  - **Последние шаблоны** (Recent Templates) – таблица со списком последних использованных *шаблонов заданий*.

- **Задания** (Jobs) содержит таблицу со списком запущенных заданий. Поддерживается фильтрация записей по различным критериям, запуск заданий напрямую из таблицы и переход к журналам выполнения заданий.
- **Расписание** (Schedules) содержит таблицу заданий, выполняемых по расписанию. Поддерживается фильтрация записей по различным критериям, управление запуском заданий и переход к редактированию их свойств.
- **Лента активности** (Activity Stream) содержит таблицу со списком событий, произошедших в контроллере. По умолчанию отображаются все произошедшие события. Поддерживается фильтрация записей по различным критериям и просмотр подробной информации о каждом событии.
- **Согласования потоков заданий** (Workflow Approvals) содержит таблицу согласований и кнопки для управления ими.
- **Метрики узлов** (Host Metrics) используется для отображения сведений об управляемых узлах и связанных событиях автоматизации.
- **Расход ресурсов подписки** (Subscription Usage) содержит график использования ресурсов подписки.

7.3.3 Ресурсы

Раздел *Ресурсы* (Resources) содержит ссылки на разделы, используемые для управления следующими ресурсами:

- **Шаблоны** (Templates) – *шаблоны заданий* и *шаблоны потоков заданий*;
- **Полномочия** (Credentials) – полномочия, используемые для доступа к различным ресурсам, в том числе управляемым узлам, источникам инфраструктурного кода и облачным сервисам;
- **Проекты** (Projects) – *проекты*;
- **Инвентарь** (Inventories) – *инвентарные списки*;
- **Управляемые узлы** (Hosts) – сведения об управляемых узлах.

7.3.4 Доступ

Раздел *Доступ* (Access) содержит ссылки на разделы, используемые для настройки разграничения доступа пользователей к ресурсам:

- **Организации** (Organizations)

  Управление *организациями*:

  - создание;
  - удаление;
  - назначение и отзыв ролей пользователям и командам.
- **Пользователи** (Users)

  Управление пользователями контроллера, в том числе создание учетных записей *системных администраторов и системных аудиторов*.
- **Команды** (Teams)

  Управление *командами*:

  - создание;
  - удаление;
  - назначение командных и индивидуальных ролей.

7.3.5 Администрирование

Раздел *Администрирование* (Administration) содержит ссылки на разделы, используемые для управления расширенными функциями контроллера:

- **Типы полномочий** (Credential Types) – управление нестандартными типами *полномочий*.
- **Уведомления** (Notifications) – настройка способов отправки оповещений о различных событиях.
- **Служебные задания** (Management Jobs) – настройка и запуск заданий по обслуживанию самого контроллера: очистка истории, истекших токенов, пользовательских сессий и истории выполнения заданий.
- **Группы узлов контроллера** (Instance Groups) – управление группами узлов контроллера.
- **Узлы контроллера** (Instances) – просмотр сведений о состоянии отдельных узлов контроллера и запуск диагностических тестов.
- **Приложения** (Applications) – управление приложениями.
- **Среды исполнения** (Execution Environments) – управление *средами исполнения*.

- **Топология** – отображение в реальном времени состояния узлов, используемых контроллером.

7.3.6 Настройки

Раздел *Настройки* используется для управления общими настройками контроллера:

- Аутентификация с помощью внешних провайдеров.
- Общие настройки выполнения заданий.
- Общесистемные настройки.
- Настройки пользовательского интерфейса.
- Отладка работы контроллера.

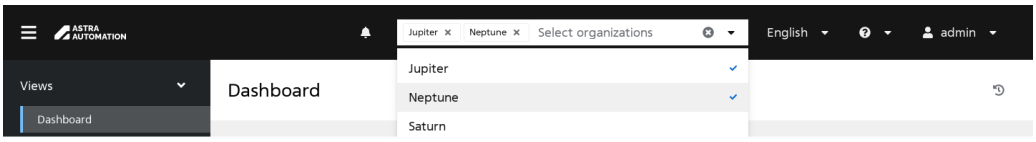
7.3.7 Глобальный фильтр по организациям

Глобальный фильтр позволяет включить фильтрацию ресурсов, связанных с организациями, сразу во всех разделах интерфейса, где это возможно.

Включение фильтрации

Чтобы включить фильтрацию, выполните следующие действия:

1. Нажмите на фильтр организаций на панели инструментов.
2. Выберите организации, которые необходимо добавить в фильтр.



Отключение фильтрации

Фильтрацию можно отключить полностью или удалить из фильтра определенную организацию. На снимке экрана показан пример фильтра по двум организациям.

Здесь:

- 1, 2 – кнопки удаления определенной организации.
- 3 – кнопка полного отключения фильтрации.

7.3.8 Запрос при запуске

Для некоторых полей доступна опция **Запрос при запуске** (Prompt on launch). Если соответствующий флаг включен, связанное поле меняет свое поведение следующим образом:

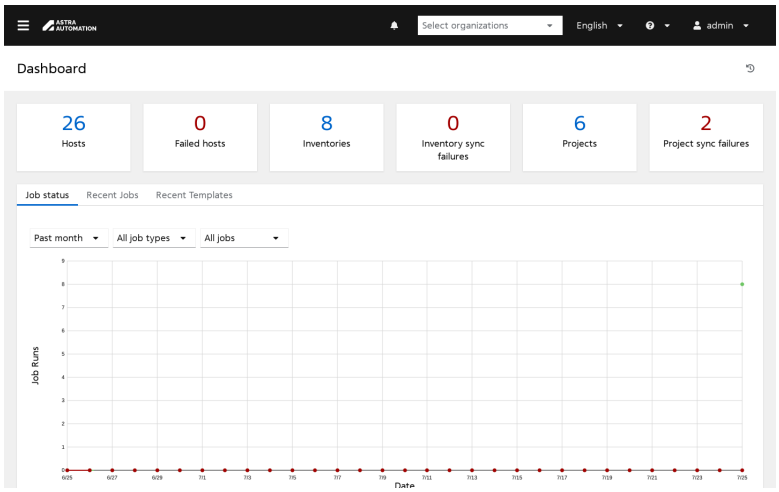
- Обязательное поле становится не обязательным для заполнения прямо сейчас. Однако, его все равно нужно будет заполнить позже.
- Если поле заполнено, указанное значение используется как значение по умолчанию.

Режимы просмотра

В этом разделе описывается просмотр сведений о состоянии заданий.

Информационная панель

Окно **Информационная панель** (Dashboard) используется для вывода сводной информации об управляемых узлах, статусах запущенных заданий, заданий синхронизации инвентаря и заданий синхронизации проектов.



Окно состоит из панелей и группы вкладок.

Панели, слева направо:

- **Управляемые узлы** (Hosts) – общее количество управляемых узлов.

  При расчете значения учитываются в том числе управляемые узлы, находящиеся в *статусе* «Выкл» (Off).

- **Узлы с ошибками** (Failed Hosts) – количество управляемых узлов, для которых статус выполнения последнего связанного задания – «Отказ» (Failed).
- **Инвентарь** (Inventories) – количество *инвентарных списков* всех типов.
- **Ошибки синхронизации инвентаря** (Inventory sync failures) – количество инвентарных списков, в которых выполнение задания синхронизации списка узлов с источником завершилось ошибкой.
- **Проекты** (Projects) – количество *проектов*.
- **Проекты с ошибками синхронизации** (Project sync failures) – количество проектов, задание обновления которых завершилось ошибкой.

Нажатие на любую панель приводит к переходу в соответствующее окно.

Во вкладках под панелями отображается следующая информация:

- **Статус задания** (Job Status) – график запуска заданий и статус их выполнения за последний месяц.

  График отображает количество запущенных заданий за каждые сутки. Сведения о количестве успешно выполненных заданий отображаются зеленым цветом, а о количестве заданий с ошибками или отказами – красным.

  Поддерживаются следующие настройки отображения:

  - Размер периода:
    - **Последний месяц** (Past month);
    - **Последние две недели** (Past two weeks);
    - **Последняя неделя** (Past week);

    * **Последние 24 часа** (Past 24 hours).
  - Тип заданий:
    * **Все типы заданий** (All job types);
    * **Синхронизация инвентаря** (Inventory sync);
    * **Обновлением SCM** (SCM update);
    * **Запуск playbook** (Playbook run).
  - Статус заданий – может принимать следующие значения:
    * **Все задания** (All jobs);
    * **Успешные задания** (Successful jobs);
    * **Задания с ошибками** (Failed jobs).

  При изменении значения любого из параметров график обновляется автоматически.

- **Последние задания** (Recent Jobs) – информация о последних запущенных заданиях.

  Таблица состоит из следующих колонок:
  - Переключатель подробности вывода основных сведений о задании.

    Набор отображаемых полей зависит от типа задания.
  - Флаги для выбора нескольких записей.
  - **Название** (Name) – ссылка для перехода в окно просмотра журнала выполнения задания или графа потока заданий.
  - **Статус** (Status)
    * **Успех** (Successful) – не было ошибок;
    * **Ошибка** (Error) – возникли ошибки, которые были успешно обработаны;
    * **Сбой** (Failed) – возникли необрабатываемые ошибки, выполнение задания прервано;
    * **Отменено** (Canceled) – задание отменено пользователем;
    * **Ожидание** (Pending) – задание находится в очереди;
    * **Выполнение** (Running) – задание выполняется.
  - **Время начала** (Start Time) – время запуска задания.
  - **Время завершения** (Finish Time) – время завершения, отмены или прерывания задания.
  - **Действия** (Actions) – кнопки для быстрого вызова часто выполняемых действий:
    * отмена задания;
    * перезапуск задания.

- **Последние шаблоны** (Recent Templates) – информация о последних использованных шаблонах заданий и шаблонах потоков заданий.

  Таблица состоит из следующих колонок:
  - Переключатель подробности вывода основных сведений о шаблоне.

    Набор отображаемых полей зависит от типа шаблона.
  - Флаги для выбора нескольких записей.
  - **Название** (Name) – ссылка для перехода в окно просмотра подробных сведений о шаблоне задания или шаблоне потока заданий.
  - **Тип** (Type) – тип шаблона.
  - **Организация** (Organization) – ссылка для перехода в окно просмотра подробных сведений об организации, которой принадлежит шаблон.

- **Последний запуск** (Last Ran) – дата и время последнего запуска заданий на основе шаблона заданий или потока заданий.
- **Действия** (Actions) – кнопки для быстрого вызова часто выполняемых действий:
  * переход в окно редактирования графа шаблона потока заданий (эта кнопка доступна только для шаблонов потоков заданий);
  * запуск задания на основе шаблона задания или потока заданий на основе шаблона потока заданий;
  * переход в окно редактирования свойств шаблона;
  * копирование шаблона.

Задания

Окно **Задания** (Jobs) используется для просмотра сведений о запущенных и выполненных заданиях. Для этого необходимо перейти в соответствующий раздел с помощью панели навигации: *Режимы просмотра* ▸ *Задания* (*Views* ▸ *Jobs*).

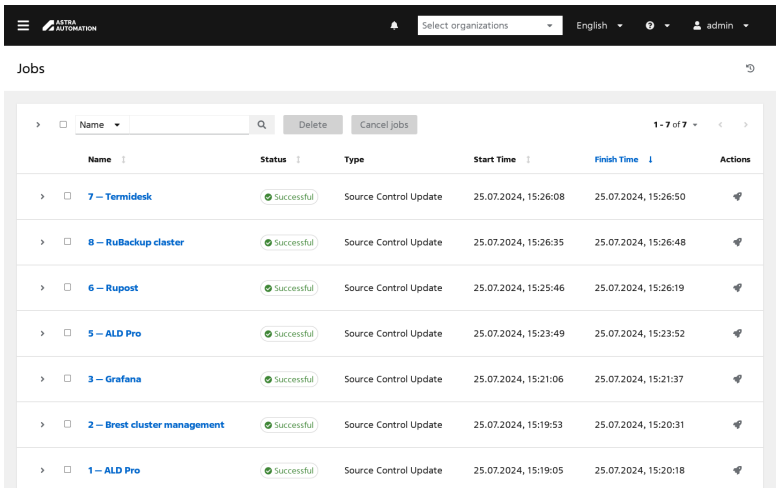


Таблица заданий состоит из следующих колонок:

- Переключатель подробности вывода основных сведений о задании.

  При нажатии на переключатель выводится дополнительная информация о задании:

  Примечание: Набор полей зависит от типа задания.

  - **Запущено** (Launched By) – ссылка на профиль пользователя, запустившего задание;
  - **Проект** (Project) – ссылка на *проект*;
  - **Шаблон задания** (Job Template) – ссылка на *шаблон задания*;
  - **Среда исполнения** (Execution Environment) – ссылка на среду исполнения, используемую для выполнения задания;
  - **Инвентарь** (Inventory) – ссылка на инвентарь;
  - **Срез задания** (Job Slice) – номер среза и *общее количество срезов*, используемых для выполнения задания;
  - **Расписание** (Schedule) – ссылка на расписание, по которому выполняется запуск задания;
  - **Источник** (Source) – тип *источника*, на основе которого создано задание синхронизации проекта или инвентарного списка.
- Флаги для выбора нескольких записей.

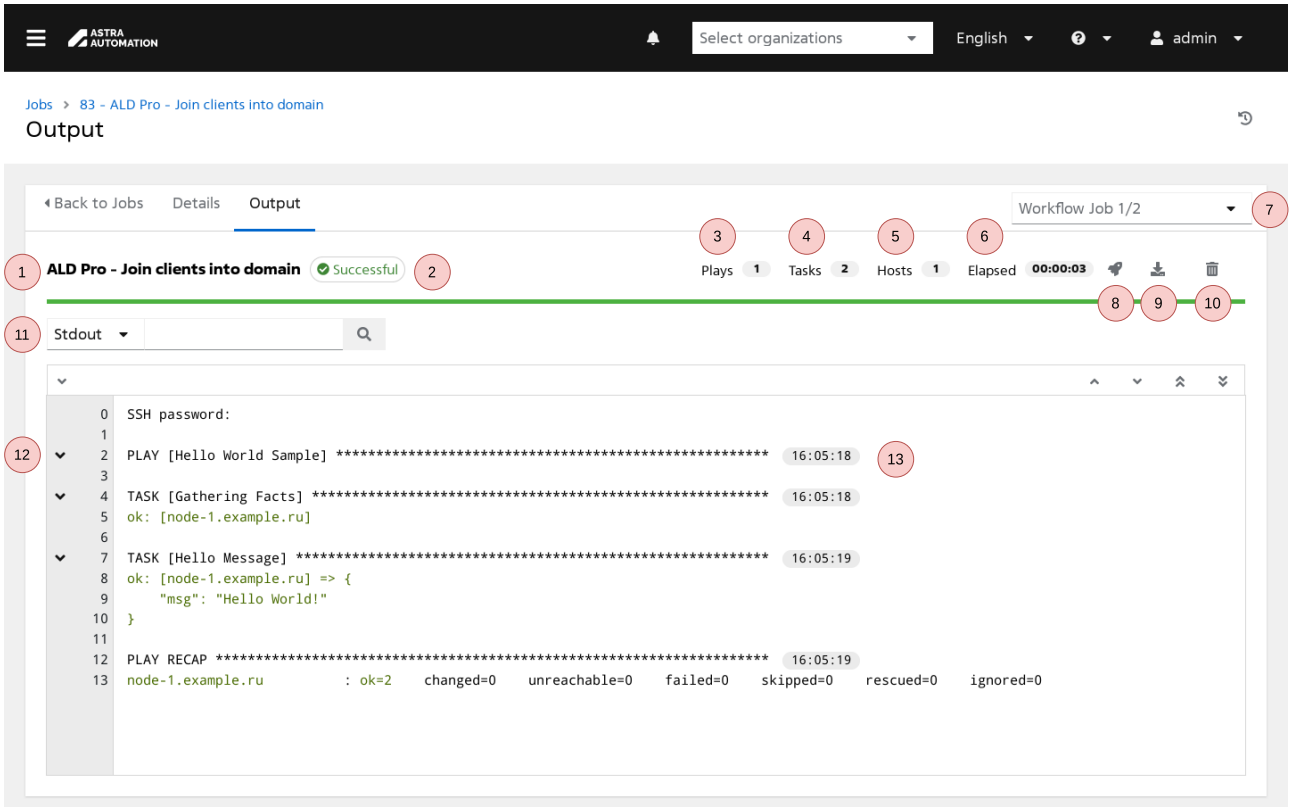
- **Название** (Name) – ссылка для перехода в окно просмотра журнала выполнения задания или окно просмотра графа потока заданий.
- **Статус** (Status) – текущий статус выполнения задания:
  - **Успех** (Successful) – не было ошибок;
  - **Ошибка** (Error) – возникли ошибки, которые были успешно обработаны;
  - **Сбой** (Failed) – возникли необрабатываемые ошибки, выполнение задания прервано;
  - **Отменено** (Canceled) – задание отменено пользователем;
  - **Ожидание** (Pending) – задание находится в очереди;
  - **Выполнение** (Running) – задание выполняется.
- **Тип** (Type) – тип задания.
- **Время начала** (Start Time) – время запуска задания.
- **Время завершения** (Finish Time) – время завершения, отмены или прерывания задания.
- **Действия** (Actions) – кнопки для быстрого вызова часто выполняемых действий:
  - отмена задания;
  - перезапуск задания.

Просмотр вывода Ansible

Для просмотра вывода Ansible нажмите на ссылку с названием задания в таблице заданий. Для заданий в статусе «Выполнение» (Running) используется вывод в реальном времени.

На снимке экрана представлены основные компоненты вкладки **Вывод** (Output).

Примечание: Состав компонентов зависит от типа задания.



Здесь:

1. Название задания.
2. Статус выполнения задания.
3. Количество выполненных сценариев.
4. Общее количество выполненных задач (на всех узлах).
5. Количество управляемых узлов.
6. Общее время выполнения задания.
7. Поле выбора заданий потока.

   Чтобы открыть вывод одного из заданий потока, выберите его в этом списке.
8. Кнопка перезапуска задания.
9. Кнопка сохранения на локальный компьютер файла с выводом Ansible.
10. Кнопка удаления задания из истории.
11. Поле для настройки фильтров.

    Поле настройки фильтров позволяет отфильтровать строки в выводе Ansible по различным критериям.
12. Кнопки сворачивания / разворачивания части вывода. Позволяют скрыть часть вывода для более удобного просмотра оставшихся строк.
13. Вывод Ansible.

Удаление

Чтобы удалить сведения о задании, выполните следующие действия:

1. В таблице заданий нажмите на ссылку с названием удаляемого задания.
2. Во вкладке **Подробности** (Details) нажмите кнопку *Удалить* (Delete).
3. Подтвердите удаление.

Расписание

Окно **Расписание** (Schedules) используется для просмотра сведений о заданиях, выполняемых по расписанию, и управления ими. Для этого необходимо перейти в соответствующий раздел с помощью панели навигации: *Режимы просмотра* ▸ *Расписание* (*Views* ▸ *Schedules*).

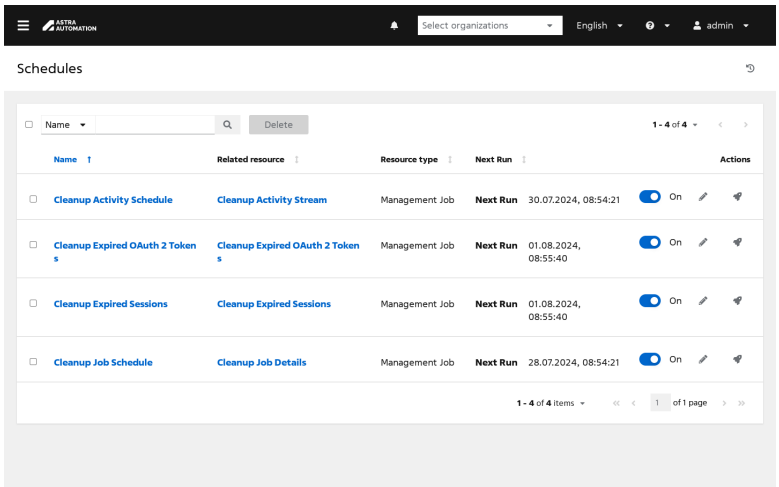


Таблица с расписанием запуска заданий состоит из следующих колонок:

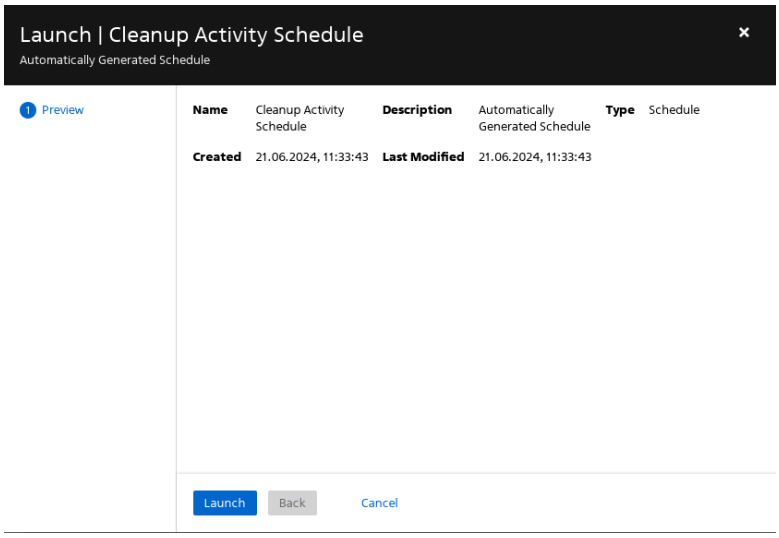
- флаги для выбора нескольких записей;

- **Название** (Name) – ссылка на расписание, по которому выполняется запуск задания;
- **Связанный ресурс** (Related resource) – ссылка на ресурс, связанный с заданием;
- **Тип ресурса** (Resource type) – тип задания;
- **Следующий запуск** (Next Run) – дата и время ближайшего запуска задания;
- **Действия** (Actions) – кнопки для быстрого вызова часто выполняемых действий:
  - переключение статуса;
  - редактирование;
  - немедленный запуск задания или потока заданий.

Немедленный запуск задания или потока заданий

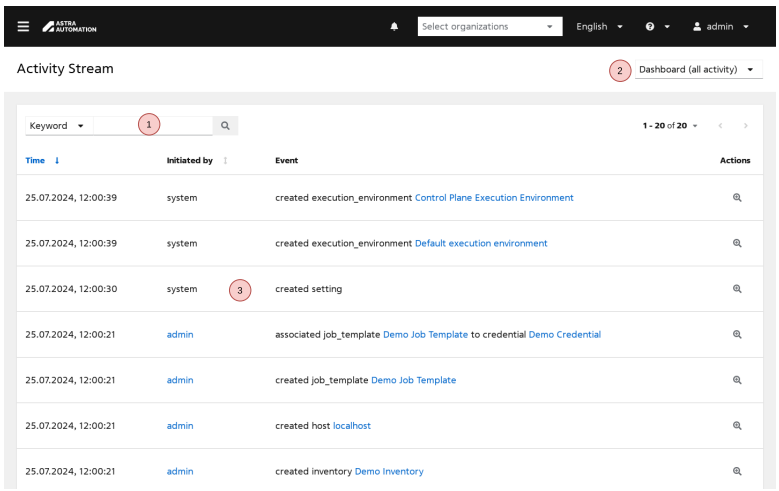
Чтобы запустить запланированное задание или поток заданий немедленно, выполните следующие действия:

1. В строке с необходимым расписанием в колонке **Действия** (Actions) нажмите кнопку *Запустить сейчас* (Run now).
2. В открывшемся окне **Запуск** (Launch) нажмите на кнопку *Запустить* (Launch) для подтверждения запуска.



Лента активности

Окно **Лента активности** (Activity Stream) используется для просмотра информации о действиях, выполненных внутри Automation Controller. Для этого необходимо перейти в соответствующий раздел с помощью панели навигации: *Режимы просмотра* ▸ *Лента активности* (*Views* ▸ *Activity Stream*).



Окно состоит из следующих элементов:

1. Поле поиска по строкам.

   Используется для фильтрации записей в таблице событий на основе вхождения в значение одного из полей указанной строки. Поддерживается создание сложных выражений фильтрации с использованием логических условий.

2. Поле фильтрации записей по типу связанных ресурсов или компонентов.

3. Таблица событий.

   Таблица событий состоит из следующих колонок:

   - **Время** (Time) – дата и время события.
   - **Инициировано** (Initiated by) – инициатор события. Если инициатором события был один из пользователей контроллера, в этой колонке выводится ссылка на его профиль.
   - **Событие** (Event) – название события.

     Если событие связано с одним из ресурсов или компонентов контроллера (проект, шаблон, поток заданий и так далее), в этом поле выводится ссылка на него.
   - **Действия** (Actions) – кнопка просмотра подробной информации о событии.

     При нажатии на кнопку открывается диалоговое окно **Подробная информация о событии** (Event detail), в котором в дополнение ко времени, инициатору и типу события выводится список сделанных изменений.

Согласования потоков заданий

Окно **Согласования потоков заданий** (Workflow Approvals) используется для подтверждения или отмены действий, связанных с узлами типа «Согласование» (Approval) *шаблонов потоков заданий*. Для просмотра согласований и принятия решений по ним необходимо перейти в соответствующий раздел с помощью панели навигации: *Режимы просмотра* ▸ *Согласование потоков заданий* (*Views* ▸ *Workflow Approvals*).

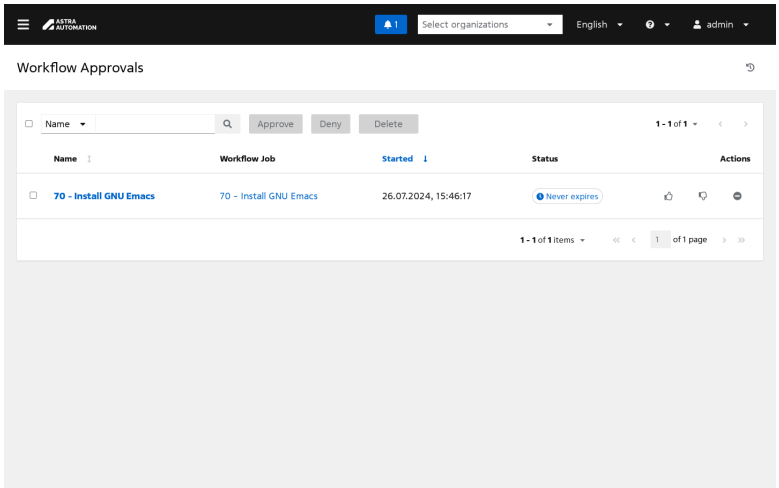


Таблица согласований состоит из следующих колонок:

- флаги для выбора нескольких записей;
- **Название** (Name) – ссылка на окно просмотра сведений о согласовании.
- **Задание потока** (Workflow Job) – ссылка на окно просмотра сведений о задании потока, которое будет запущено при согласовании.
- **Начато** (Started) – дата и время запуска согласования.
- **Статус** (Status) – текущий статус согласования.

  Если время на согласование ограничено, для нерешенных согласований в этой колонке выводятся дата и время, до наступления которых нужно принять решение.

  Для согласований, по которым решение принято, в этой колонке выводится их статус – «Согласовано» (Approved) и «Отказано» (Denied) соответственно.
- **Действия** (Actions) – кнопки управления согласованиями:
  - принятие положительного решения по согласованию;
  - отказ в согласовании;
  - остановка выполнения потока заданий.

Просмотр

Для просмотра подробных сведений о согласовании нажмите на ссылку с его названием в таблице согласований.

Окно подробных сведений о согласовании состоит из вкладки **Подробности** (Details), на которой выводится следующая информация:

- **Название** (Name) – название узла в графе потока заданий.
- **Создано** (Created) – дата и время создания согласования, а также ссылка на профиль связанного пользователя.
- **Истекает** (Expires) – если время на согласование ограничено, в этом поле выводятся дата и время, до наступления которых нужно принять решение.

- **Последнее изменение** (Last Modified) – дата и время последнего изменения согласования.
- **Шаблон потока заданий** (Workflow Job Template) – ссылка на окно просмотра подробных сведений о шаблоне потока заданий.
- **Время выполнения** (Elapsed) – время, прошедшее с момента запуска потока заданий.
- **Сведения о задании потока** (Workflow job details) – набор полей, связанных с заданием:
  - **Задание потока** (Workflow Job) – ссылка на задание, которое будет запущено при положительном решении о согласовании.
  - **Инвентарь** (Inventory) – ссылка на инвентарный список, связанный с заданием.
  - **Переменные** (Variables) – значения переменных Ansible, заданные на уровне потока заданий.

    Подробности о порядке присвоения значений переменным см. в документе *Переменные*.

Метрики узлов

Окно **Метрики узлов** (Host Metrics) используется для вывода статистической информации о запуске заданий автоматизации на управляемых узлах.

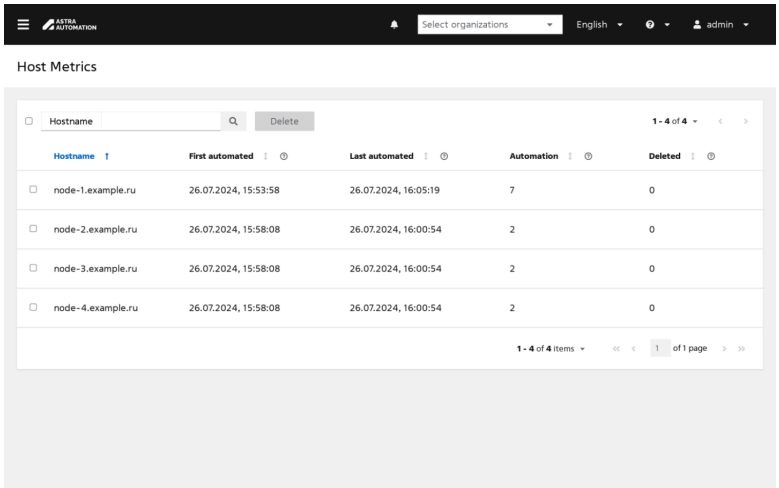


Таблица с информацией о выполнении заданий автоматизации на управляемых узлах состоит из следующих колонок:

- флаги для выбора нескольких записей;
- **Наименование узла** (Hostname) – название управляемого узла;
- **Первый автоматизированный** (First automated) – дата и время первого запуска задания автоматизации;
- **Последняя автоматизация** (Last automated) – дата и время последнего запуска задания автоматизации;
- **Автоматизация** (Automation) – общее количество выполненных заданий автоматизации;
- **Инвентарь** (Inventories) – количество инвентарных списков, в которые входит управляемый узел;
- **Удалено** (Deleted) – количество раз, которые управляемый узел был удален из списка управляемых узлов.

Поддерживается фильтрация и сортировка записей по различным параметрам.

Удаление записей

Чтобы удалить сводную информацию об определенных управляемых узлах, выполните следующие действия:

1. В таблице со статистической информацией включите флаги в строках с управляемыми узлами, сведения о которых следует удалить.
2. Нажмите кнопку *Удалить* (Delete).
3. Подтвердите удаление записей.

Расход ресурсов подписки

Важно: Эта часть документации находится в стадии разработки.

Ресурсы

В этом разделе описывается управление ресурсами контроллера, в которые входят:

- шаблоны заданий и шаблоны потоков заданий;
- полномочия;
- проекты;
- инвентарь;
- управляемые узлы.

Шаблоны

Окно **Шаблоны** (Templates) используется для управления *шаблонами заданий* и *шаблонами потоков заданий*. Для создания шаблонов и управления ими необходимо перейти в соответствующий раздел с помощью панели навигации: *Ресурсы* ▸ *Шаблоны* (*Resources* ▸ *Templates*).

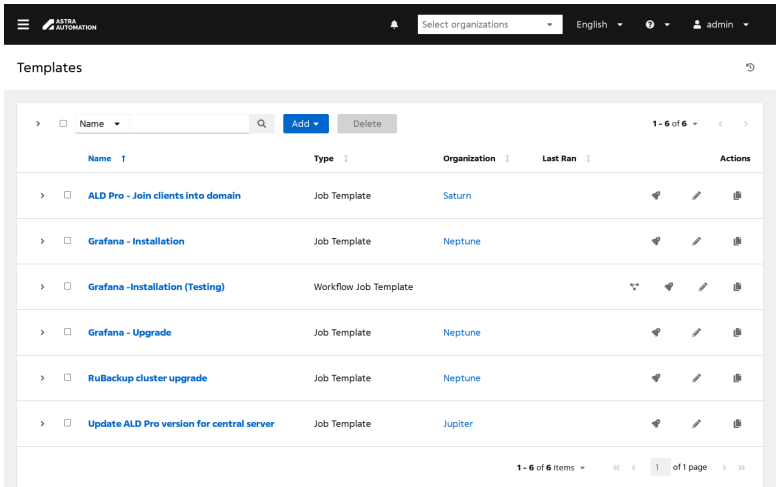


Таблица шаблонов состоит из следующих колонок:

- Флаги для выбора нескольких записей.
- **Название** (Name).
- **Тип** (Type).

- **Организация** (Organization) – ссылка на окно просмотра свойств *организации*, которой принадлежит *проект*, связанный с шаблоном.
- **Последний запуск** (Last Ran).

  Дата и время последнего запуска задания или шаблона потока заданий.
- **Действия** (Actions) – кнопки для быстрого вызова часто выполняемых действий:
  - переход в окно визуализатора графа шаблона потока заданий;
  - запуск задания или потока заданий на основе шаблона;
  - переход в окно редактирования свойств шаблона;
  - создание копии шаблона потока заданий.

Просмотр

Для получения подробных сведений о шаблоне нажмите на ссылку с его названием в таблице шаблонов. Окно просмотра сведений о шаблоне состоит из следующих вкладок:

- **Подробности** (Details) – общие сведения о шаблоне.
- **Доступ** (Access) – таблица пользователей и назначенных им пользовательских и командных *ролей*.
- **Уведомления** (Notifications) – таблица уведомлений.
- **Расписание** (Schedules) – расписание запуска заданий на основе шаблона.
- **Визуализатор** (Vizualizer) – графический редактор узлов потока заданий.

  Примечание: Эта вкладка доступна только для шаблонов потоков заданий.

- **Задания** (Jobs) – таблица заданий, созданных на основе шаблона.
- **Опрос** (Survey) – таблица параметров, которые должны быть заданы при запуске заданий на основе шаблона.

Настройка доступа

Во вкладке **Доступ** (Access) выводится список пользователей и назначенных им *ролей*. Нажатие на ссылку с именем пользователя приводит к переходу в окно просмотра сведений о пользователе.

Назначение ролей отдельным пользователям

Чтобы настроить доступ к шаблону для отдельных пользователей, выполните следующие действия:

1. Во вкладке **Доступ** (Access) нажмите кнопку *Добавить* (Add).
2. Выберите тип объектов *Пользователи* (Users) и нажмите кнопку *Продолжить* (Next).
3. Выберите пользователей, которым хотите назначить роли, и нажмите кнопку *Продолжить* (Next).
4. Выберите роли, которые хотите назначить выбранным на предыдущем шаге пользователям.
5. Нажмите кнопку *Сохранить* (Save).

Назначение командных ролей

Чтобы настроить доступ к шаблону для всех участников одной или нескольких команд, выполните следующие действия:

1. Во вкладке **Доступ** (Access) нажмите кнопку *Добавить* (Add).
2. Выберите тип объектов *Команды* (Teams) и нажмите кнопку *Продолжить* (Next).
3. Выберите команды, участникам которых хотите назначить роли, и нажмите кнопку *Продолжить* (Next).
4. Выберите роли и нажмите кнопку *Сохранить* (Save).

Отзыв ролей

Чтобы отозвать пользовательскую или командную роль, выполните следующие действия:

1. Во вкладке **Доступ** (Access) в строке с нужным пользователем нажмите кнопку × рядом с названием отзываемой роли.
2. Подтвердите отзыв роли.

Примечание: Роли «Системный администратор» (System Administrator) и «Системный аудитор» (System Auditor) нельзя отозвать через настройку доступа на уровне проекта, однако, можно *изменить тип пользователя*.

Запуск

Для запуска задания или потока заданий выполните следующие действия:

1. В таблице шаблонов нажмите на ссылку с названием нужного шаблона.
2. Во вкладке **Подробности** (Details) нажмите кнопку *Запустить* (Run).
3. Если в шаблоне настроен и включен опрос – в открывшемся окне **Запуск** (Run) заполните форму во вкладке **Опрос** (Survey) и нажмите кнопку *Продолжить* (Next).

   Заданные значения передаются в задание или поток заданий как переменные.

   Подробности о порядке присвоения значений переменным см. в документе *Переменные*.
4. Во вкладке **Предварительный просмотр** (Preview) проверьте корректность введенных данных. Если все значения указаны верно, нажмите кнопку *Запустить* (Run).

Удаление

Чтобы удалить шаблон, выполните следующие действия:

1. В таблице шаблонов нажмите на ссылку с названием удаляемого шаблона.
2. Во вкладке **Подробности** (Details) нажмите кнопку *Удалить* (Delete).
3. Подтвердите удаление.

Шаблоны заданий

В этой секции рассматривается создание *шаблонов заданий* и запуск заданий на их основе через веб-интерфейс.

Создание шаблона заданий

Для создания шаблона задания выполните следующие действия:

1. В окне **Шаблоны** (Templates) нажмите кнопку *Добавить* (Add) и в открывшемся меню выберите **Добавить шаблон задания** (Add job template).
2. Заполните форму **Создать новый шаблон задания** (Create New Job Template):
   - **Название** (Name).

     Особенности заполнения поля:

     - В рамках одной организации не допускается создание шаблонов с одним и тем же названием.
     - Название шаблона не может состоять из одних пробельных символов (пробелы, табуляции и так далее).
     - Допускается использования символов кириллицы и специальных символов.
   - **Описание** (Description) – укажите описание шаблона, например, его назначение или особенности использования.
   - **Тип задания** (Job Type) – выберите, будет ли выполнение задания вносить фактические изменения в конфигурацию управляемых узлов:

     - **Выполнение** (Run) – да;
     - **Проверка** (Check) – нет.

     Подробности см. в секции *Тип задания*.
   - **Инвентарь** (Inventory) – выберите *инвентарный список*, управляемые узлы из которого настраиваются.
   - **Проект** (Project) – выберите *проект*.
   - **Среда исполнения** (Execution Environment) – выберите среду исполнения для запуска заданий.

     Примечание: Это поле становится доступным после заполнения поля **Проект** (Project).

   - **Playbook** – выберите файл playbook из состава проекта.

     Примечание: Это поле становится доступным после заполнения поля **Проект** (Project).

   - **Полномочия** (Credentials) – укажите полномочия, которые следует использовать для доступа к управляемым узлам.

     Подробности см. в секции *Полномочия*.
   - **Метки** (Labels).

     С помощью меток можно группировать задания, созданные на основе шаблона. Их удобно использовать для поиска нужных записей журнала выполнения заданий или их логической группировки.

- **Переменные** (Variables) – укажите дополнительные переменные, которые следует использовать при запуске заданий на основе шаблона.

  Подробности см. в секции *Переменные*.

- **Ответвления** (Forks) – укажите количество параллельных процессов, используемых для запуска заданий на основе шаблона.

  Подробности см. в секции *Ответвления*.

- **Лимит** (Limit) – укажите дополнительные фильтры названий управляемых узлов.

  Подробности см. в секции *Лимит на управляемые узлы*.

- **Степень подробности** (Verbosity) – укажите степень детализации журнала выполнения заданий на основе шаблона.

  Подробности см. в секции *Степень подробности*.

- **Деление заданий на срезы** (Job Slicing) – количество частей, на которые следует разделить исходный инвентарный список.

  Подробности см. в секции *Деление на срезы*.

- **Таймаут** (Timeout) – укажите ограничение по времени (в секундах) на выполнение задания на основе шаблона.

  Подробности см. в секции *Таймаут*.

- **Показать изменения** (Show Changes).

  Если эта настройка включена, при запуске задания на основе шаблона в журнал выводится список изменений, выполненных на управляемых узлах.

  Значение по умолчанию – Выкл (`Off`).

  Подробности см. в секции *Показ изменений*.

- **Группы узлов контроллера** (Instance Groups) – укажите группы узлов контроллера, которые следует использовать для запуска заданий на основе шаблона.

- **Теги задания** (Job Tags) – укажите одной строкой через запятую названия тегов, задачи с которыми следует выполнять при запуске заданий на основе шаблона. Прочие задачи будут пропущены.

  Подробности см. в секции *Теги*.

- **Пропускать теги** (Skip Tags) – укажите одной строкой через запятую названия тегов, задачи с которыми следует пропустить при запуске заданий на основе шаблона. Прочие задачи будут выполнены.

  Подробности см. в секции *Теги*.

- **Повышение привилегий** (Privilege Escalation) – укажите, следует ли использовать повышение привилегий при выполнении заданий на основе шаблона.

  Подробности см. в секции *Повышение привилегий*.

- **Обратные вызовы процесса обеспечения сервиса** (Provisioning Callbacks).

  Если эта настройка включена:

  - при выполнении заданий на основе шаблона управляемые узлы могут связаться с контроллером и запросить обновление конфигурации;
  - становится доступным для заполнения поле **Ключ конфигурации узла** (Host Config Key).

- **Ключ конфигурации узла** (Host Config Key).

  URL, используемый управляемыми узлами для запроса в контроллере обновления конфигурации.

- **Включить webhook** (Enable Webhook).

  Если эта настройка включена, разрешено создание заданий на основе шаблона путем запроса к Webhook. Также становятся доступны для заполнения следующие поля:

  - **Сервис webhook** (Webhook Service).

    Сервис хранения исходного кода, используемый для работы с Webhook.

    Возможные значения:

    * BitBucket Data Center.
    * GitFlic.

      Added in version 1.1-upd1.
    * GitHub.
    * GitLab.

  - **Webhook URL**.

    URL для работы с Webhook.

    Значение в этом поле формируется автоматически при создании шаблона.

  - **Ключ webhook** (Webhook Key).

    Ключ для работы с Webhook.

    Значение в этом поле формируется автоматически при создании шаблона.

  - **Полномочия webhook** (Webhook Credential).

    Полномочия, используемые для доступа к выбранному сервису хранения исходного кода.

- **Параллельные задания** (Concurrent Jobs).

  Если эта настройка включена, на основе шаблона можно запустить одновременное выполнение нескольких заданий с одним и тем же инвентарным списком.

- **Включить хранилище фактов** (Enable Fact Storage) – укажите, следует ли сохранить в кеш Automation Controller факты, полученные при выполнении заданий на основе шаблона.

  Подробности см. в секции *Хранилище фактов*.

- **Запретить откат группы узлов управления** (Prevent Instance Group Fallback).

  Если эта настройка включена, для запуска заданий будут использоваться группы узлов, указанные в поле **Группы узлов контроллера** (Instance Groups). Если группы узлов контроллера не выбраны, используются глобальные группы узлов.

  Подробности см. в секции *Запрет отката группы узлов управления*.

3. Нажмите кнопку *Сохранить* (Save).

Запуск задания на основе шаблона

Чтобы запустить задание на основе шаблона, выполните следующие действия:

1. В окне **Шаблоны** (Templates) нажмите на ссылку с названием нужного шаблона задания.
2. Во вкладке **Подробности** (Details) нажмите на кнопку *Запустить* (Run).
3. Если в настройках шаблона включен опрос, откроется диалоговое окно **Запуск** (Run).
   1. Во вкладке **Опрос** (Survey) заполните значения полей.
   2. Нажмите кнопку *Продолжить* (Next).

3. Во вкладке **Предварительный просмотр** (Preview) проверьте настройки запуска задания. Если все значения заданы верно, нажмите кнопку *Запустить* (Run).

Шаблоны потоков заданий

В этом документе рассматриваются следующие операции, связанные с *шаблонами потоков заданий*:

- создание;
- запуск.

Создание шаблонов потоков заданий

Для создания шаблона потока заданий выполните следующие действия:

1. В окне **Шаблоны** (Templates) нажмите кнопку *Добавить* (Add) и в открывшемся меню выберите **Добавить шаблон потока заданий** (Add workflow template).
2. Заполните форму **Создать новый шаблон потока заданий** (Create New Workflow Template):
   - **Название** (Name).

     Особенности заполнения поля:

     - В рамках одной организации не допускается создание шаблонов с одним и тем же названием.
     - Название шаблона не может состоять из одних пробельных символов (пробелы, табуляции и так далее).
     - Допускается использование символов кириллицы и специальных символов.
   - **Описание** (Description) – укажите описание шаблона, например, его назначение или особенности использования.
   - **Организация** (Organization) – укажите организацию, которой принадлежит шаблон потока заданий.
   - **Инвентарь** (Inventory) – укажите инвентарный список.

     Важно: Значение в этом поле переопределяет значения, заданные на уровне шаблонов заданий.

   - **Лимит** (Limit) – укажите дополнительные фильтры названий управляемых узлов.

     Подробности см. в секции *Лимит на управляемые узлы*.
   - **Ветка системы управления исходными данными** (Source control branch) – если при запуске заданий следует использовать ветку, отличную от основной ветки проекта, укажите ее название в этом поле.

     Важно: Название ветки, указанное в этом поле, применяется ко всем шаблонам заданий, используемым в потоке.

   - **Метки** (Labels).

     С помощью меток можно группировать задания, созданные при выполнении потока. Их удобно использовать для поиска нужных записей журнала выполнения заданий или их логической группировки.

- **Переменные** (Variables) – укажите дополнительные переменные, которые следует использовать при запуске заданий потока.

  Подробности о порядке разрешения переменных см. в документе *aac_management_workflow_templates_extra_vars*.

- **Теги задания** (Job Tags) – укажите одной строкой через запятую названия тегов, задачи с которыми следует выполнять при запуске заданий на основе шаблона. Прочие задачи будут пропущены.

  Подробности см. в секции *Теги*.

- **Пропускать теги** (Skip Tags) – укажите одной строкой через запятую названия тегов, задачи с которыми следует пропустить при запуске заданий на основе шаблона. Прочие задачи будут выполнены.

  Подробности см. в секции *Теги*.

- **Включить webhook** (Enable Webhook).

  Если эта настройка включена, разрешено создание потока заданий на основе шаблона путем запроса к Webhook. Также становятся доступны для заполнения следующие поля:

  – **Сервис Webhook** (Webhok Service).

    Сервис хранения исходного кода, используемый для работы с Webhook. Возможные значения:

    * BitBucket Data Center.
    * GitFlic.

      Added in version 1.1-upd1.

    * GitHub.
    * GitLab.

  – **Webhook URL** (Webhook URL)

    URL для работы с Webhook.

      Значение в этом поле формируется автоматически при создании шаблона.

  – **Ключ webhook** (Webhook Key)

    Ключ для работы с Webhook.

    Значение в этом поле формируется автоматически при создании шаблона.

- **Разрешить одновременные задания** (Enable Concurrent Jobs).

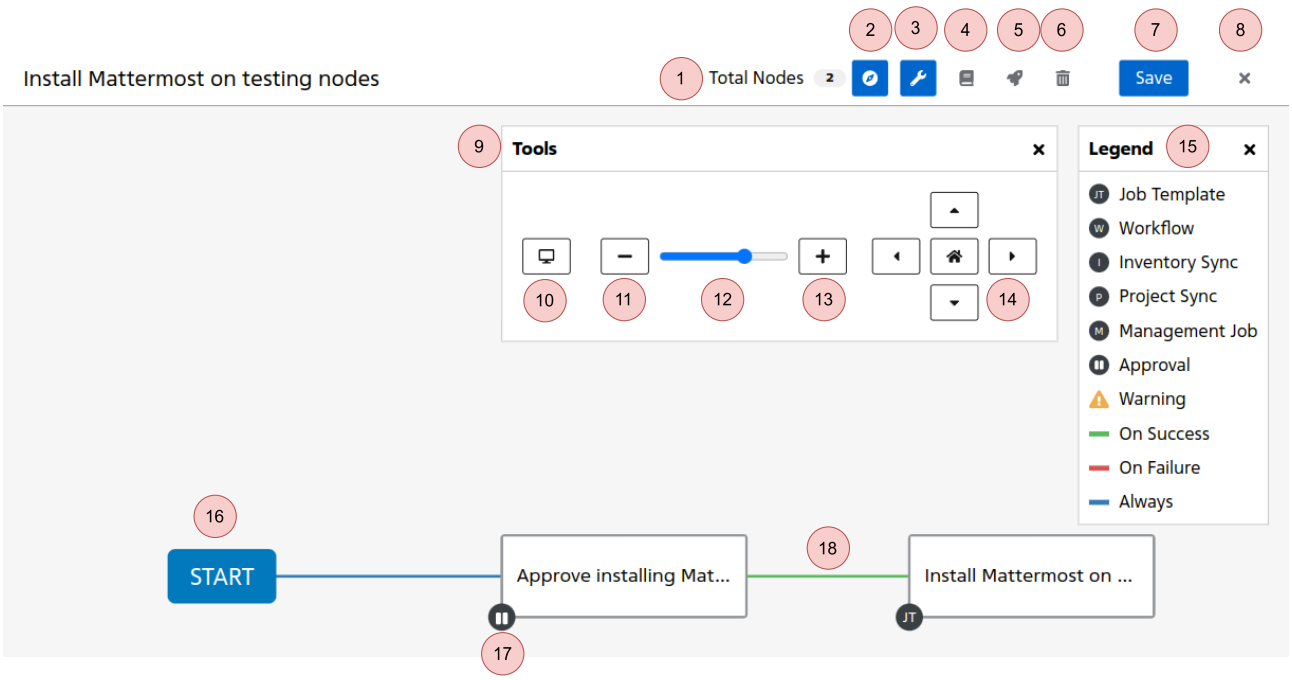
  Если эта настройка включена, на основе шаблона можно запустить одновременно несколько потоков заданий с одним и тем же инвентарным списком.

3. Нажмите кнопку *Сохранить* (Save).

Визуализатор шаблонов потоков заданий

После создания шаблона потока заданий визуализатор запускается автоматически. Для его запуска для существующего шаблона потока заданий выполните следующие действия:

1. В таблице шаблонов нажмите на ссылку с названием нужного шаблона потока заданий.
2. Выберите вкладку **Визуализатор** (Visializer).



Окно визуализатора содержит следующие элементы:

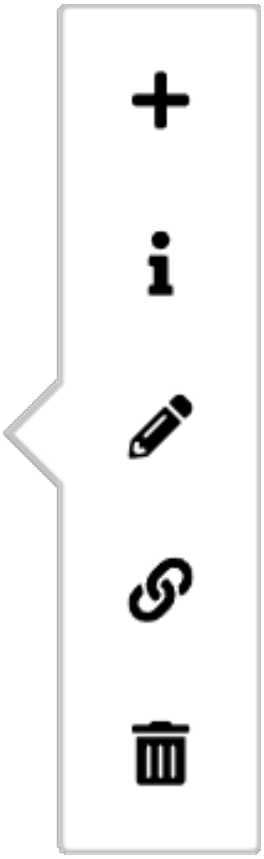
1. Общее количество узлов. При расчете значения узел «НАЧАЛО» (START) не учитывается.
2. Переключатель отображения легенды.
3. Переключатель отображения кнопок управления масштабом.
4. Кнопка вызова справочной информации.
5. Кнопка запуска потока заданий.
6. Кнопка удаления всех узлов шаблона потока заданий.
7. Кнопка сохранения изменений в шаблоне.
8. Кнопка закрытия визуализатора без сохранения внесенных изменений.
9. Панель управления масштабом.
10. Кнопка изменения масштаба под размер экрана.
11. Кнопка увеличения масштаба.
12. Ползунок изменения масштаба.
13. Кнопка уменьшения масштаба.
14. Кнопки управления положением графа на экране.
15. Панель легенды.
16. Узел «НАЧАЛО» (START).
17. Индикатор типа узла. В данном случае – «Согласование» (Approval).
18. Ссылка на следующий узел.

    Цвет ссылки указывает на условие, при котором выполняется переход:

    - красный – ошибки при выполнении задания или отказ в согласовании;
    - зеленый – успешное выполнение задания или согласование;
    - синий – безусловный переход.

При нажатии на узел или ссылку появляется контекстное меню.

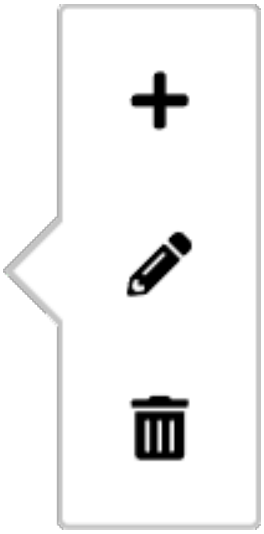
Контекстное меню узла:



Контекстное меню узла содержит кнопки для выполнения следующих действий:

- создание нового узла и связь с ним;
- просмотр сведений об узле;
- редактирование узла;
- создание ссылки на существующий узел;
- удаление узла.

Контекстное меню ссылки:



Контекстное меню ссылки содержит кнопки для выполнения следующих действий:

- вставка узла между двумя существующими узлами;
- редактирование ссылки;
- удаление ссылки.

Начало работы

Чтобы начать работу с графом нового шаблона потока заданий, выполните следующие действия:

1. Нажмите кнопку **Пуск** (Start).
2. В диалоговом окне **Добавить узел** (Add Node) выберите тип узла.
3. Заполните поля, специфичные для выбранного типа узла. В зависимости от типа узла могут быть доступны дополнительные настройки.
4. Нажмите кнопку *Сохранить* (Save).

Первый добавленный в граф узел связывается с узлом «НАЧАЛО» (START), который имеет следующие особенности:

- его нельзя изменить или удалить;
- тип ссылки на другие узлы – только «Всегда» (Always).

Создание узлов

Чтобы добавить в граф новый узел, выполните следующие действия:

1. Вызовите контекстное меню узла или ссылки.
2. Нажмите кнопку создания нового узла. Это приводит к открытию диалогового окна **Добавить узел** (Add Node).
3. Во вкладке **Тип выполнения** (Run type) выберите условие перехода к создаваемому узлу:
   - **При успехе** (On Success) – согласование или успешное выполнение задания;
   - **При неудаче** (On Failure) – отказ в согласовании или ошибки при выполнении задания;
   - **Всегда** (Always) – переход выполняется в любом случае.
4. Нажмите кнопку *Продолжить* (Next).
5. Во вкладке **Тип узла** (Node type) выберите тип создаваемого узла.
6. Заполните поля, соответствующие выбранному типу узла:
   - **Согласование** (Approval).

     Узлы этого типа используются для создания согласований – запросов к пользователю на подтверждение перехода к следующему заданию в потоке.

     Укажите параметры согласования.

     - **Название** (Name) – название узла.

       Указанное название выводится в таблице согласований.
     - **Описание** (Description) – описание узла.
     - **Таймаут** (Timeout) – время на согласование.

       Если в полях для минут и секунд указаны значения `0`, время на согласование не ограничено.
   - **Источники синхронизации инвентаря** (Inventory Source Sync).

     Узлы этого типа создают задание синхронизации инвентарного списка с источником.

     - **Название** (Name) – выберите источник сведений об управляемых узлах.

- **Шаблон задания** (Job Template).

  Узлы этого типа используются для запуска заданий на основе выбранного шаблона задания.

  Если в настройках шаблона задания включен опрос, вместо кнопки *Сохранить* (Save) в форме выводится кнопка *Продолжить* (Next). При ее нажатии происходит переход к шагу опроса.

  Подробности об опросах см. в документе *Опросы*.

  - **Название** (Name) – выберите шаблон задания.

- **Синхронизация проекта** (Project Sync).

  Узлы этого типа создают задание синхронизации кода проекта с источником.

  - **Название** (Name) – выберите проект.

- **Шаблон потока заданий** (Workflow Job Template).

  Узлы этого типа запускают поток заданий на основе выбранного шаблона потока заданий.

  - **Название** (Name) – выберите шаблон потока заданий.

- **Служебное задание** (Management Job).

  Узлы этого типа создают служебное задание.

  При выборе заданий «Cleanup Activity Stream» и «Cleanup Job Details» вместо кнопки *Сохранить* (Save) в форме выводится кнопка *Продолжить* (Next). При ее нажатии происходит переход к шагу заполнения поля **Количество дней, в течение которых будут храниться данные** (Days of data to be retained). Все записи старше указанного периода будут удалены.

  Значение по умолчанию – `30`.

  Подробности о служебных заданиях см. в секции *Служебные задания*.

7. Заполните поля **Конвергенция** (Convergence) и **Псевдоним узла** (Alias):

   - **Конвергенция** (Convergence) – укажите количество условий, которые должны быть выполнены для перехода к этому узлу потока заданий.

     Подробности см. в секции *Конвергенция*.

   - **Псевдоним узла** (Node Alias) – если в этом поле указано какое-либо значение, при просмотре графа потока заданий оно выводится вместо значения, указанного в поле **Название** (Name).

8. Нажмите кнопку *Сохранить* (Save).

Удаление узлов и ссылок

Чтобы удалить узел или ссылку, выполните следующие действия:

1. Вызовите контекстное меню узла или ссылку.
2. Нажмите кнопку удаления.
3. Подтвердите удаление.

Запуск потока заданий

Чтобы запустить поток заданий, выполните следующие действия:

1. В таблице шаблонов нажмите на ссылку с названием нужного шаблона потоков заданий.
2. Во вкладке **Подробности** (Details) нажмите кнопку *Запустить* (Launch).

Опросы

Для управления *опросами* Automation Controller предоставляет интуитивно понятный пользовательский интерфейс.

При просмотре подробных сведений о шаблоне задания или шаблоне потока заданий во вкладке **Опрос** (Survey) выводится список вопросов, задаваемых пользователю при запуске заданий или потока заданий.

Примечание: В этом документе для краткости шаблоны заданий и шаблоны потоков заданий именуются общим термином «шаблоны».

Таблица вопросов состоит из следующих колонок:

- Флаги для выбора нескольких записей.
- **Название** (Name) – текст вопроса, задаваемого пользователю.
- **Тип** (Type).
- **По умолчанию** (Default).

  Значение по умолчанию. Если ответ имеет тип «Пароль» (Password), в этой колонке выводится значение `ЗАШИФРОВАНО` (`ENCRYPTED`).
- **Действия** (Actions) – кнопка для быстрого перехода в окно редактирования вопроса.

Переключение состояния

Чтобы при запуске задания или потока заданий пользователю были заданы вопросы, связанные с шаблоном, необходимо включить соответствующий опрос. Для изменения статуса опроса выполните следующие действия:

1. В таблице шаблонов нажмите на ссылку с названием нужного шаблона.
2. Выберите вкладку **Опрос** (Survey).
3. Переведите переключатель состояния опроса в нужное положение.

Создание вопросов

Чтобы добавить вопросы в опросник, выполните следующие действия:

1. В таблице шаблонов нажмите на ссылку с названием нужного шаблона.
2. Во вкладке **Опрос** (Survey) нажмите кнопку *Добавить* (Add).
3. Заполните форму **Добавить вопрос** (Add Question):
   - **Вопрос** (Question) – укажите текст вопроса, который выводится в форме при запуске задания или потока заданий.
   - **Описание** (Description) – укажите справочную информацию, которая поможет пользователю принять решение при ответе на вопрос.

- **Имя переменной ответа** (Answer variable name) – укажите имя переменной Ansible, значение которой пользователь задает при ответе на вопрос.

  Требования к имени переменной:

  - Уникальность на уровне шаблона.
  - Может состоять из символов латинского алфавита, цифр и знака _.
  - Не может содержать пробельные символы.
- **Необходимо** (Required) – если эта настройка включена, ответ пользователя на вопрос обязателен.
- **Тип ответа** (Answer type) – выберите тип ответа, который может ввести пользователь.

  Подробности о поддерживаемых типах ответов см. в документе *Опросы*.

4. В зависимости от выбранного типа ответа заполните соответствующие поля:

   - **Текст** (Text).

     Совет: Для ввода конфиденциальных данных рекомендуется использовать ответ типа «Пароль» (Password).

     Поддерживаются следующие опциональные поля:

     - **Минимальная длина** – минимальное количество символов в строке.

       Значение по умолчанию – `0`.
     - **Максимальная длина** – максимальное количество символов в строке.

       Значение по умолчанию – `1024`.
     - **Ответ по умолчанию** (Default answer)
   - **Текстовая область** (Textarea).

     Поддерживаются следующие опциональные поля:

     - **Минимальная длина** – минимальное количество символов во введенном тексте.

       Значение по умолчанию – `0`.
     - **Максимальная длина** – максимальное количество символов во введенном тексте.

       Значение по умолчанию – `1024`.
     - **Ответ по умолчанию** (Default answer)
   - **Пароль** (Password).

     Поддерживаются следующие опциональные поля:

     - **Минимальная длина** – минимальное количество символов в строке.

       Значение по умолчанию – `0`.
     - **Максимальная длина** – максимальное количество символов в строке.

       Значение по умолчанию – `1024`.
     - **Ответ по умолчанию** (Default answer)
   - **Множественный выбор (одиночное выделение)** (Multiple Choice (single select)).

     Чтобы создать элемент списка, введите значение в поле **Варианты множественного выбора** (Muptiple Choice Options) и нажмите `Enter`.

     Чтобы сделать один из элементов списка значением по умолчанию, включите флаг в соответствующей строке.

- **Множественный выбор (множественное выделение)** (Multiple Choice (multiple select)).

  Чтобы создать элемент списка, введите значение в поле **Варианты множественного выбора** (Multiple Choice Options) и нажмите `Enter`.

  Чтобы сделать пункты выбранными по умолчанию, включите флаги в соответствующих строках.

- **Целое число** (Integer).

  Поддерживаются следующие опциональные поля:

  - **Минимум** (Minimum) – минимально допустимое значение.

    Значение по умолчанию – `0`.

  - **Максимум** (Maximum) – максимально допустимое значение.

    Значение по умолчанию – `1024`.

  - **Ответ по умолчанию** (Default answer).

- **Плавающее** (Float).

  Поддерживаются следующие опциональные поля:

  - **Минимум** (Minimum) – минимально допустимое значение.

    Значение по умолчанию – `0`.

  - **Максимум** (Maximum) – максимально допустимое значение.

    Значение по умолчанию – `1024`.

  - **Ответ по умолчанию** (Default answer).

5. Нажмите кнопку *Сохранить* (Save).

Удаление вопросов

Чтобы удалить вопросы, выполните следующие действия:

1. В таблице шаблонов включите флаги на удаляемых опросах.
2. Нажмите кнопку *Удалить* (Delete).
3. Подтвердите удаление.

Полномочия

Окно **Полномочия** (Credentials) используется для управления *полномочиями*. Для создания полномочий и управления ими необходимо перейти в соответствующий раздел с помощью панели навигации: *Ресурсы* ▸ *Полномочия* (*Resources* ▸ *Credentials*).

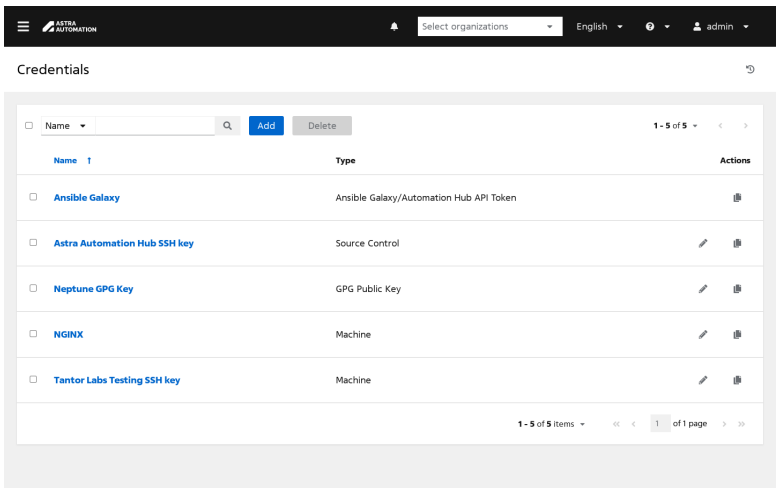


Таблица полномочий состоит из следующих колонок:

- флаги для выбора нескольких записей;
- **Название** (Name);
- **Тип** (Type) – тип полномочий;
- **Действия** (Actions) – кнопки для быстрого вызова часто выполняемых действий:
  - переход в окно редактирования сведений о полномочии (не поддерживается для некоторых типов полномочий);
  - копирование полномочия.

Просмотр сведений о полномочиях

Для получения подробных сведений о полномочии нажмите на ссылку с его названием в таблице полномочий.

Внешний вид окна со сведениями о полномочии зависит от их типа и может состоять из следующих вкладок:

- **Подробности** (Details) – общие сведения о полномочии;
- **Доступ** (Access) – таблица пользователей и назначенных им пользовательских и командных *ролей*;
- **Шаблоны заданий** (Job Templates) – таблица *шаблонов заданий*, использующих полномочие.

Создание полномочия

Для создания полномочия выполните следующие действия:

1. В окне **Полномочия** (Credentials) нажмите кнопку *Добавить* (Add).
2. Заполните форму **Создать новые учетные данные** (Create New Credential):
   - **Название** (Name) – название полномочия.
   - **Описание** (Description) – дополнительная информация о полномочии, например, цель его создания или правила использования.
   - **Организация** (Organization) – организация, которой принадлежит полномочие.

     Если поле не заполнено, полномочие принадлежит всем организациям контроллера.
   - **Тип полномочия** (Credential Type) – *тип полномочия*.

     При выборе значения в этом поле в форме появляются соответствующие поля.

3. Заполните поля, соответствующие выбранному типу полномочия.

   Особенности заполнения полей для каждого типа полномочия приведены в следующих секциях.

4. Нажмите кнопку *Сохранить* (Save).

API-токен Ansible Galaxy/Automation Hub

При создании полномочия этого типа доступны следующие поля:

- **URL сервера Galaxy** (Galaxy Server URL) – URL репозитория в реестре коллекций Ansible.
- **Токен API** (API Token) – токен, используемый для аутентификации в реестре коллекций.

Подробности о полномочиях этого типа см. в секции *API-токен Ansible Galaxy/Automation Hub*.

Centrify Vault Credential Provider Lookup

Важно: Эта часть документации находится в стадии разработки.

CyberArk Central Credential Provider Lookup

Важно: Эта часть документации находится в стадии разработки.

CyberArk Conjur Secrets Manager Lookup

Важно: Эта часть документации находится в стадии разработки.

Google Compute Engine

При создании полномочия этого типа доступны следующие поля:

- **Файл JSON сервисной учетной записи** (Service account JSON file) – файл с реквизитами сервисной учетной записи Google Compute Cloud. При выборе файла JSON, содержащего необходимые сведения, остальные поля формы заполняются автоматически.

  Пошаговые инструкции по созданию такого файла см. в документации Google Compute Cloud[122].

- **Адрес электронной почты сервисной учетной записи** (Service Account Email Address) – адрес электронной почты, связанный с сервисной учетной записью Google Compute Engine.
- **Project** – идентификатор проекта в Google Compute Engine.
- **Закрытый ключ RSA** (RSA Private Key) – содержимое приватного ключа SSH, используемого для доступа к Google Compute Engine с реквизитами сервисной учетной записи.

Подробности об этом типе полномочий см. в подразделе *Google Compute Engine*.

[122] https://cloud.google.com/iam/docs/keys-create-delete

HashiCorp Vault Secret Lookup

При создании полномочия этого типа доступны следующие поля:

- **URL сервера** (Server URL) – URL сервера HashiCorp Vault, используемого для хранения секретов.
- **Токен** (Token) – токен аутентификации на сервере HashiCorp Vault.
- **Сертификат CA** (CA Certificate) – сертификат, используемый для защиты подключения к серверу HashiCorp Vault.
- **AppRole role_id** – идентификатор AppRole, используемой для аутентификации на сервере HashiCorp Vault.
- **AppRole secret_id** – секрет AppRole, используемые для аутентификации на сервере HashiCorp Vault.
- **Клиентский сертификат** (Client certificate) – сертификат в формате PEM, используемый при аутентификации TLS. Должен включать в себя в том числе все необходимые промежуточные сертификаты, необходимые для аутентификации в HashiCorp Vault.
- **Ключ клиентского сертификата** (Client Certificate Key) – приватный ключ в формате PEM, используемый при аутентификации TLS.
- **Роль аутентификации TLS** (TLS Authentication Role) – название роли или сертификата, используемых при аутентификации TLS в HashiCorp Vault.
- **Название пространства имен (только для Vault Enterprise)** (Namespace name (Vault Enterprise only)) – название пространства имен в HashiCorp Vault.

  Примечание: Для использования значения, указанного в этом поле, требуется Hashicorp Vault редакции Enterprise.

- **Роль Kubernetes** (Kubernetes role) – название роли, используемой для аутентификации Kubernetes.
- **Имя пользователя** (Username) – название учетной записи пользователя HashiCorp Vault.
- **Пароль** (Password) – пароль пользователя HashiCorp Vault.
- **Путь к авторизации** (Path to Auth) – значение в этом поле следует указывать, если путь к авторизации на сервере HashiCorp Vault отличается от значения `/approle`, используемого по умолчанию.
- **Версия API** (API Version) – версия API HashiCorp Vault.
  - v1 – статичный поиск;
  - v2 – поиск с версионированием.

Подробности об этом типе полномочий см. в секции *HashiCorp Vault Secret Lookup*.

HashiCorp Vault Signed SSH

Важно: Эта часть документации находится в стадии разработки.

Insights

Важно: Эта часть документации находится в стадии разработки.

Microsoft Azure Key Vault

Важно: Эта часть документации находится в стадии разработки.

OpenStack

При создании полномочия этого типа доступны следующие поля:

- **Имя пользователя** (Username) – название учетной записи пользователя OpenStack.
- **Пароль (ключ API)** (Password (API Key)) – пароль или ключ API OpenStack.
- **Узел (URL для аутентификации)** (Host (Authentication URL)) – URL, используемый для аутентификации в OpenStack, например:

  ```
  https://openstack.example.com/auth/
  ```

- **Проект (имя арендатора)** (Project (Tenant Name)) – название или идентификатор проекта (тенанта) OpenStack. Как правило, совпадает с именем пользователя OpenStack.
- **Проект (доменное имя)** (Project (Domain Name)) – доменное имя проекта (тенанта) OpenStack.
- **Доменное имя** (Domain Name) – FQDN сервера OpenStack.

  Заполнение этого поля требуется только при использовании сервера аутентификации Keystone v3.
- **Название региона** (Region Name) – название региона OpenStack.
- **Проверка SSL** (Verify SSL) – если эта настройка включена, при подключении к OpenStack проверяется корректность сертификата SSL.

  Значение по умолчанию – включено.

Red Hat Satellite 6

Важно: Эта часть документации находится в стадии разработки.

Thycotic DevOps Secrets Vault

Важно: Эта часть документации находится в стадии разработки.

Thycotic Secret Server

Важно: Эта часть документации находится в стадии разработки.

Vault

Важно: Эта часть документации находится в стадии разработки.

VMware vCenter

Важно: Эта часть документации находится в стадии разработки.

Web-сервисы Amazon

Важно: Эта часть документации находится в стадии разработки.

Виртуализация Red Hat

Важно: Эта часть документации находится в стадии разработки.

Диспетчер ресурсов Microsoft Azure

Важно: Эта часть документации находится в стадии разработки.

Личный токен доступа к GitHub

При создании полномочий этого типа укажите в поле **Токен** (Token) значение токена, созданного в профиле GitHub.

Подробности о полномочиях этого типа см. в секции *Личный токен доступа к GitHub*.

Машина

При создании полномочия этого типа доступны следующие поля:

- **Имя пользователя** (Username) – название учетной записи пользователя, с реквизитами которого контроллер должен подключаться к управляемому узлу.
- **Пароль** (Password) – пароль пользователя, с реквизитами которого контроллер должен подключаться к управляемому узлу.
- **Закрытый ключ SSH** (SSH Private Key) – содержимое приватного ключа SSH.
- **Подписанный сертификат SSH** (Signed SSH Certificate) – содержимое сертификата, используемого для защиты подключения SSH.
- **Парольная фраза закрытого ключа** (Private Key Passphrase) – если приватный ключ SSH защищен паролем, укажите его в этом поле.
- **Способ повышения привилегий** (Privilege Escalation Method) – метод повышения привилегий.
- **Имя пользователя с повышенными привилегиями** (Privilege Escalation Username) – если для повышения привилегий команды следует выполнять от имени пользователя, отличающегося от указанного в поле **Имя пользователя** (Username), укажите его в этом поле.
- **Пароль с повышенными привилегиями** (Privilege Escalation Password) – пароль пользователя, название учетного записи которого указано в поле **Имя пользователя с повышенными привилегиями** (Privilege Escalation Username).

Подробности о полномочиях этого типа см. в секции *Машина*.

Открытый ключ GPG

При создании полномочия этого типа укажите в поле **Открытый ключ GPG** (GPG Public Key) содержимое ключа GPG, который контроллер должен использовать для проверки подписи содержимого.

Подробности о полномочиях этого типа см. в секции *Открытый ключ GPG*.

Платформа автоматизации Red Hat Ansible

Важно: Эта часть документации находится в стадии разработки.

Реестр контейнеров

Важно: Эта часть документации находится в стадии разработки.

Сеть

Важно: Эта часть документации находится в стадии разработки.

Система управления версиями

При создании полномочия этого типа доступны следующие поля:

- **Имя пользователя** (Username) – название учетной записи пользователя, с реквизитами которого контроллер должен подключаться к источнику кода.
- **Пароль** (Password) – пароль пользователя, с реквизитами которого контроллер должен подключаться к источнику кода.
- **Закрытый ключ SCM** (SCM Private Key) – содержимое приватного ключа SSH, используемого для подключения к источнику кода.
- **Парольная фраза закрытого ключа** (Private Key Passphrase) – если приватный ключ SSH защищен паролем, укажите его в этом поле.

Подробности о полномочиях этого типа см. в секции *Система управления версиями*.

Токен доступа OpenShift или Kubernetes API

Важно: Эта часть документации находится в стадии разработки.

Токен персонального доступа GitLab

При создании полномочий этого типа укажите в поле **Токен** (Token) значение токена, созданного в профиле GitLab.

Подробности о полномочиях этого типа см. в секции *Токен персонального доступа GitLab*.

Удаление полномочия

Чтобы удалить полномочие, выполните следующие действия:

1. В таблице полномочий нажмите на ссылку с названием удаляемого полномочия.
2. Во вкладке **Подробности** (Details) нажмите кнопку *Удалить* (Delete).
3. Подтвердите удаление.

Проекты

Окно **Проекты** (Projects) используется для управления *проектами*. Для создания проектов и управления ими необходимо перейти в соответствующий раздел с помощью панели навигации: *Ресурсы* ▸ *Проекты* (*Resources* ▸ *Projects*).

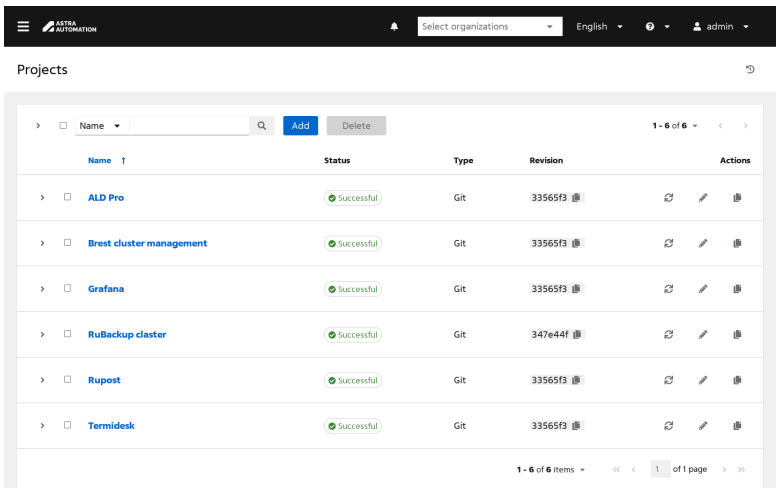


Таблица проектов состоит из следующих колонок:

- флаги для выбора нескольких записей;
- **Название** (Name);
- **Статус** (Status) – текущее состояние проекта;
- **Тип** (Type) – тип источника кода проекта;
- **Версия** (Revision) – версия кода проекта;
- **Действия** (Actions) – кнопки для быстрого вызова часто выполняемых действий:
  - синхронизация кода проекта с источником;
  - переход в окно редактирования свойств проекта;
  - создание копии проекта.

Просмотр

Для получения подробных сведений о проекте нажмите на ссылку с его названием в таблице проектов.

Окно сведений о проекте состоит из следующих вкладок:

- **Подробности** (Details) – общие сведения о проекте;
- **Доступ** (Access) – таблица пользователей и назначенных им пользовательских и командных *ролей*;
- **Шаблоны заданий** (Job Templates) – таблица *шаблонов заданий*, использующих код проекта;
- **Уведомления** (Notifications) – таблица уведомлений;
- **Расписание** (Schedules) – таблица заданий по обновлению кода проекта из *источника*.

Примечание: Эта вкладка недоступна для локальных проектов.

Создание

Для создания проекта выполните следующие действия:

1. В окне **Проекты** (Projects) нажмите кнопку *Добавить* (Add).
2. Заполните форму **Создать новый проект** (Create New Project):
   - **Название** (Name) – укажите название проекта.

     Особенности заполнения поля:

     - В рамках одной организации не допускается создание проектов с одним и тем же названием.
     - Название проекта не может состоять из одних пробельных символов (пробелы, табуляции и так далее).
     - Допускается использование символов кириллицы и специальных символов.
   - **Описание** (Description) – укажите описание проекта, например, его назначение или особенности использования.
   - **Организация** (Organization) – выберите организацию, которой принадлежит проект.
   - **Среда исполнения** (Execution Environment) – выберите *среду исполнения*, используемую для запуска заданий, созданных на основе проекта.

     Примечание: Это поле становится доступным для редактирования после выбора организации. Среда исполнения, выбранная на уровне проекта, имеет более высокий приоритет, чем среды исполнения, выбранные на уровне контроллера и организации.

   - **Тип системы управления исходными данными** (Source Control Type) – выберите тип источника кода проекта.

     Особенности работы контроллера с различными типами источников описаны в разделе *Источники кода*.
   - **Учетные данные для проверки подписи содержимого** (Content Signature Validation Credential) – если для проверки аутентичности кода проекта следует использовать GPG, выберите соответствующие полномочия типа «Публичный ключ GPG».
3. Заполните поля, соответствующие выбранному источнику кода.

Ручное управление (Manual)

- **Базовый путь проекта** (Project Base Path)

  Значение в этом поле недоступно для изменения. Оно указывает на каталог, в котором выполняется поиск проекта.
- **Каталог playbook** (Playbook Directory) – выберите локальный каталог с кодом проекта.

Git

- **Тип системы управления исходными данными** (Source Control Type) – укажите ссылку на репозиторий Git, в котором хранится код проекта.
- **Ветка/тег/коммит системы управления исходными данными** (Source Control Branch/Tag/Commit) – если требуется использовать ревизию кода, отличную от последнего коммита в основной ветке, укажите в этом поле название нужной ветки, название тега или хеш коммита.
- **Спецификация (Refspec) системы управления исходными данными** (Source Control Refspec) – если требуется использовать специфичную ревизию кода, укажите в этом поле правила получения необходимых ссылок из репозитория Git.
- **Полномочия на систему управления исходными данными** (Source Control Credential)

  Если для доступа к репозиторию требуется аутентификация, выберите в этом поле соответствующие полномочия типа «Управление версиями» (Source Control).
- **Очистить** (Clean)

  Если эта настройка включена, перед обновлением будут сброшены все локальные изменения кода.
- **Удалить** (Delete)

  Если эта настройка включена, при обновлении локальная копия кода проекта будет удалена, после чего загружена заново.
- **Отслеживание подмодулей** (Track Submodules)

  Если эта настройка включена, при обновлении проекта будут также обновлены все его субмодули.
- **Обновить версию при запуске** (Update Revision on Launch)

  Если эта настройка включена, перед запуском заданий на основе проекта его код будет обновляться автоматически.
- **Разрешить переопределение ветки** (Allow Branch Override)

  Если эта настройка включена, в шаблоне задания можно выбрать ветку или версию кода, отличные от указанных в свойствах проекта.

Subversion

- **URL системы управления исходными данными** (Source Control URL) – укажите ссылку на репозиторий Subversion, в котором хранится код проекта.
- **Версия №** (Revision #) – если требуется использовать ревизию кода, отличную от последнего коммита в основной ветке, укажите в этом поле номер нужной ревизии.
- **Полномочия на систему управления исходными данными** (Source Control Credential)

  Если для доступа к репозиторию требуется аутентификация, выберите в этом поле соответствующие полномочия типа «Управление версиями» (Source Control).
- **Очистить** (Clean)

  Если эта настройка включена, перед обновлением будут сброшены все локальные изменения кода.
- **Удалить** (Delete)

  Если эта настройка включена, при обновлении локальная копия кода проекта будет удалена, после чего загружена заново.

- **Обновить версию при запуске** (Update Revision on Launch)

  Если эта настройка включена, перед запуском заданий на основе проекта его код будет обновляться автоматически.

- **Разрешить переопределение ветки** (Allow Branch Override)

  Если эта настройка включена, в шаблоне задания можно выбрать ветку или версию кода, отличные от указанных в свойствах проекта.

Внешний архив (Remote Archive)

- **URL системы управления исходными данными** (Source Control URL) – укажите ссылку на архив, в котором хранится код проекта.
- **Полномочия на систему управления исходными данными** (Source Control Credential)

  Если для доступа к репозиторию требуется аутентификация, выберите в этом поле соответствующие полномочия типа «Управление версиями» (Source Control).

- **Очистить** (Clean)

  Если эта настройка включена, перед обновлением будут сброшены все локальные изменения кода.

- **Удалить** (Delete)

  Если эта настройка включена, при обновлении локальная копия кода проекта будет удалена, после чего загружена заново.

- **Обновить версию при запуске** (Update Revision on Launch)

  Если эта настройка включена, перед запуском заданий на основе проекта его код будет обновляться автоматически.

- **Разрешить переопределение ветки** (Allow Branch Override)

  Если эта настройка включена, в шаблоне задания можно выбрать ветку или версию кода, отличные от указанных в свойствах проекта.

4. Нажмите кнопку *Сохранить* (Save).

После создания проекта его настройки можно изменять с помощью различных групп параметров, описанных в последующих секциях.

Настройка доступа

Во вкладке **Доступ** (Access) выводится список пользователей и назначенных им *ролей*. Нажатие на ссылку с именем пользователя приводит к переходу в окно просмотра сведений о пользователе.

Назначение ролей отдельным пользователям

Чтобы настроить доступ к проекту для отдельных пользователей, выполните следующие действия:

1. Во вкладке **Доступ** (Access) нажмите кнопку *Добавить* (Add).
2. Выберите тип объектов *Пользователи* (Users) и нажмите кнопку *Продолжить* (Next).
3. Выберите пользователей, которым хотите назначить роли, и нажмите кнопку *Продолжить* (Next).
4. Выберите роли, которые хотите назначить выбранным на предыдущем шаге пользователям.
5. Нажмите кнопку *Сохранить* (Save).

Назначение командных ролей

Чтобы настроить доступ к проекту для всех участников одной или нескольких команд, выполните следующие действия:

1. Во вкладке **Доступ** (Access) нажмите кнопку *Добавить* (Add).
2. Выберите тип объектов *Команды* (Teams) и нажмите кнопку *Продолжить* (Next).
3. Выберите команды, участникам которых хотите назначить роли, и нажмите кнопку *Продолжить* (Next).
4. Выберите роли и нажмите кнопку *Сохранить* (Save).

Отзыв ролей

Чтобы отозвать пользовательскую или командную роль, выполните следующие действия:

1. Во вкладке **Доступ** (Access) в строке с нужным пользователем нажмите кнопку × рядом с названием отзываемой роли.
2. Подтвердите отзыв роли.

Примечание: Роли «Системный администратор» (System Administrator) и «Системный аудитор» (System Auditor) нельзя отозвать через настройку доступа на уровне проекта, однако, можно *изменить тип пользователя*.

Удаление

Удаление проектов имеет следующие особенности:

- При удалении проекта связанные с ним шаблоны заданий и инвентарные списки не удаляются и могут быть связаны с новым проектом.
- При удалении проекта, использующего в качестве источника локальный каталог, файлы проекта не удаляется с диска.

Чтобы удалить проект, выполните следующие действия:

1. В таблице проектов нажмите на ссылку с названием удаляемого проекта.
2. Во вкладке **Подробности** (Details) нажмите кнопку *Удалить* (Delete).
3. Подтвердите удаление.

Обновление

Если проект использует внешний источник кода, для поддержания локальной копии в актуальном состоянии необходимо обновлять проект. Automation Controller позволяет обновлять код проектов вручную или автоматически, при этом автоматическое обновление может выполняться при запуске связанных с проектом заданий или по расписанию.

Чтобы обновить проект вручную, во вкладке **Подробности** (Details) нажмите кнопку *Синхронизировать* (Sync).

Совет: Для быстрого запуска обновления проекта можно использовать кнопку *Синхронизировать проект* (Sync Project), доступную в колонке **Действия** (Actions) таблицы проектов.

Чтобы проект автоматически обновлялся перед запуском связанных с ним заданий, в настройках источника включите соответствующие флаги в блоке **Опции** (Options).

Чтобы проект обновлялся по расписанию, настройте его следующим образом:

1. Во вкладке **Расписание** (Schedules) нажмите кнопку *Добавить* (Add).
2. Заполните форму **Создать новое расписание** (Create New Schedule):
   - **Название** (Name) – укажите название расписания.
   - **Описание** (Description) – укажите описание расписания.
   - **Дата и время начала** (Start date/time) – укажите дату и время первого запуска задания.
   - **Локальный часовой пояс** (Local time zone) – выберите часовой пояс, по которому рассчитывается время.

     Значение по умолчанию – UTC.
   - **Частота повторения** (Repeat frequency) – выберите частоту повторения запусков задания.

     Если выбрано значение **Нет (запускается один раз)** (None (run once)), задание будет выполнено один раз.

     Включение флагов **Мин** (Min), **Час** (Hour), **День** (Day), **Неделя** (Week), **Месяц** (Month) и **Год** (Year) приводит к добавлению в форму наборов полей, позволяющих выполнить настройку соответствующих временных интервалов. Для всех типов интервалов доступны следующие настройки:

     – **Выполнять каждые** (Run every) – количество выбранных временных интервалов между запусками задания.

     – **Завершено** (End) – условие прекращения запуска задания по расписанию. Доступны следующие настройки:

       ∗ **Никогда** (Never) – не прекращать.

       ∗ **После указанного количества вхождений** (After number of occurences) – после указанного количества запусков.

       ∗ **На дату** (On date) – в указанные день и время.

     Для недельных интервалов доступна дополнительная настройка **В дни** (Days). Она позволяет выбрать дни недели, по которым следует запускать задание.

     Для месячных интервалов доступна дополнительная настройка **Выполнить в** (Run on). Она позволяет выбрать определенную дату или относительную позицию дня в месяце. Например, при выборе даты и указании числа «20» задание будет запускаться двадцатого числа каждого месяца. При выборе относительной позиции нет привязки к конкретной дате. Вместо этого настраивается запуск в дни, обозначаемые как, например, «второй понедельник» или «последняя пятница». Также поддерживаются значения, позволяющие выбрать рабочие или нерабочие дни.

     Для годовых интервалов доступна дополнительная настройка **Выполнить в** (Run on). Она позволяет выбрать месяц и число либо относительную позицию дня в году. Например, при выборе месяца «Январь» и числа «10» задание будет запускаться каждый год десятого января. При выборе относительной позиции дня нет привязки к конкретной дате. Вместо этого настраивается запуск в дни, обозначаемые как, например, «первый четверг мая» или «последний день февраля». Также поддерживаются значения, позволяющие выбрать рабочие или нерабочие дни.

     – **Исключения** (Exceptions) – настройки, предотвращающие запуск задания в указанные минуты, часы, дни, месяцы и годы. Эти настройки работают по тем же принципам, что и настройки частоты повторений.
3. Нажмите кнопку *Сохранить* (Save).

Инвентарь

Окно **Инветарь** (Inventories) используется для управления *инвентарными списками*. Для создания инвентарных списков и управления ими необходимо перейти в соответствующий раздел с помощью панели навигации: *Ресурсы* ▸ *Инвентарь* (*Resources* ▸ *Inventories*).

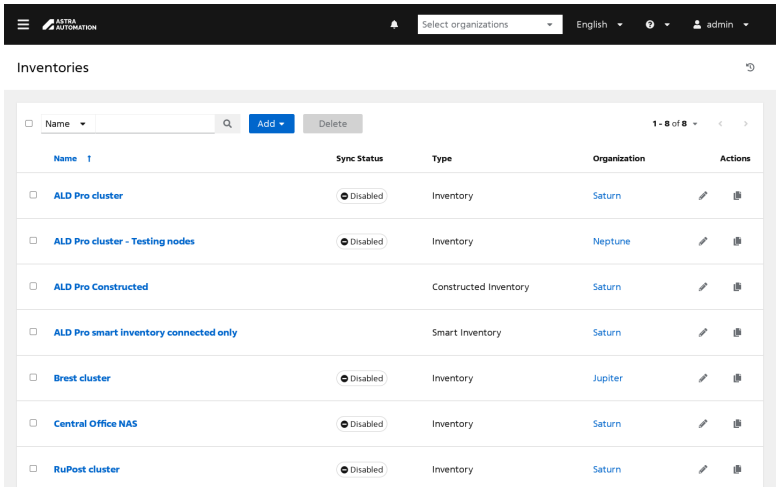


Таблица инвентарных списков состоит из следующих колонок:

- Флаги для выбора нескольких записей.
- **Название** (Name).
- **Состояние синхронизации** (Sync Status).

  Если для формирования инвентаря используется хотя бы один внешний источник, в этом поле выводится статус синхронизации. В противном случае выводится статус **Отключено** (Disabled).

  Для умных и сборных инвентарных списков понятие синхронизации не используется.
- **Тип** (Type) – *тип инвентарного списка*.
- **Организация** (Organization) – организация-владелец.
- **Действия** (Actions) – кнопки для быстрого вызова часто выполняемых действий:
  - переход в окно редактирования свойств инвентарного списка;
  - создание копии инвентарного списка.

В этом документе приводятся инструкции, общие для инвентарных списков всех типов. Инструкции по управлению инвентарными списками различных типов описаны в соответствующих документах.

Просмотр

Для получения подробных сведений об инвентарном списке нажмите на ссылку с его названием в таблице инвентарных списков. Окно сведений об инвентарном списке состоит из следующих вкладок:

- **Подробности** (Details) – общие сведения об инвентарном списке.
- **Доступ** (Access) – таблица пользователей и назначенных им ролей.
- **Группы** (Groups) – таблица групп узлов.

Примечание: Эта вкладка недоступна при просмотре сведений об умных инвентарных списках.

- **Управляемые узлы** (Hosts) – список управляемых узлов, входящих в инвентарный список.
- **Источники** (Sources) – список источников сведений об управляемых узлах.

Примечание: Эта вкладка доступна только при просмотре сведений об обычных инвентарных списках.

- **Задания** (Jobs) – список заданий, связанных с инвентарным списком.
- **Шаблоны заданий** (Job Templates) – список шаблонов заданий, связанных с инвентарным списком.

Настройка доступа

Во вкладке **Доступ** (Access) выводится таблица пользователей и назначенных им *ролей*. Нажатие на ссылку с именем пользователя приводит к переходу в окно просмотра сведений о пользователе.

Назначение ролей отдельным пользователям

Чтобы настроить доступ к инвентарному списку для отдельных пользователей, выполните следующие действия:

1. Во вкладке **Доступ** (Access) нажмите кнопку *Добавить* (Add).
2. Выберите тип объектов *Пользователи* (Users) и нажмите кнопку *Продолжить* (Next).
3. Выберите пользователей, которым хотите назначить роли, и нажмите кнопку *Продолжить* (Next).
4. Выберите роли, которые хотите назначить выбранным на предыдущем шаге пользователям.
5. Нажмите кнопку *Сохранить* (Save).

Назначение командных ролей

Чтобы настроить доступ к инвентарному списку для всех участников одной или нескольких команд, выполните следующие действия:

1. Во вкладке **Доступ** (Access) нажмите кнопку *Добавить* (Add).
2. Выберите тип объектов *Команды* (Teams) и нажмите кнопку *Продолжить* (Next).
3. Выберите команды, участникам которых хотите назначить роли, и нажмите кнопку *Продолжить* (Next).
4. Выберите роли и нажмите кнопку *Сохранить* (Save).

Отзыв ролей

Чтобы отозвать пользовательскую или командную роль, выполните следующие действия:

1. Во вкладке **Доступ** (Access) в строке с нужным пользователем нажмите кнопку × рядом с названием отзываемой роли.
2. Подтвердите отзыв роли.

Примечание: Роли «Системный администратор» (System Administrator) и «Системный аудитор» (System Auditor) нельзя отозвать через настройку доступа на уровне инвентарного списка, однако, можно *изменить тип пользователя*.

Выполнение специальных команд (Ad Hoc)

Для выполнения специальных команд выполните следующие действия:

1. Во вкладке **Управляемые узлы** (Hosts) выберите узлы или во вкладке **Группы** (Groups) выберите группы управляемых узлов, на которых хотите выполнить специальную команду.

   Отфильтровать узлы по названию можно позже.

2. Нажмите кнопку *Выполнить команду* (Run Command).
3. В открывшемся окне заполните форму во вкладке **Подробности** (Details):
   - **Модуль** (Module) – выберите модуль Ansible, используемый для выполнения команды.

     Поддерживаются следующие модули (в алфавитном порядке):

     - apt[123];
     - apt_key[124];
     - apt_repository[125];
     - apt_rpm[126];
     - command[127];
     - group[128];
     - mount[129];
     - ping[130];
     - selinux[131];
     - service[132];
     - setup[133];
     - shell[134];
     - user[135];
     - win_group[136];
     - win_ping[137];

[123] https://docs.ansible.com/ansible/latest/collections/ansible/builtin/apt_module.html
[124] https://docs.ansible.com/ansible/latest/collections/ansible/builtin/apt_key_module.html
[125] https://docs.ansible.com/ansible/latest/collections/ansible/builtin/apt_repository_module.html
[126] https://docs.ansible.com/ansible/latest/collections/community/general/apt_rpm_module.html
[127] https://docs.ansible.com/ansible/latest/collections/ansible/builtin/command_module.html
[128] https://docs.ansible.com/ansible/latest/collections/ansible/builtin/group_module.html
[129] https://docs.ansible.com/ansible/latest/collections/ansible/posix/mount_module.html
[130] https://docs.ansible.com/ansible/latest/collections/ansible/builtin/ping_module.html
[131] https://docs.ansible.com/ansible/latest/collections/ansible/posix/selinux_module.html
[132] https://docs.ansible.com/ansible/latest/collections/ansible/builtin/service_module.html
[133] https://docs.ansible.com/ansible/latest/collections/ansible/builtin/setup_module.html
[134] https://docs.ansible.com/ansible/latest/collections/ansible/builtin/shell_module.html
[135] https://docs.ansible.com/ansible/latest/collections/ansible/builtin/user_module.html
[136] https://docs.ansible.com/ansible/latest/collections/ansible/windows/win_group_module.html
[137] https://docs.ansible.com/ansible/latest/collections/ansible/windows/win_ping_module.html

  - win_service[138];
  - win_updates[139];
  - win_user[140];
  - yum.

- **Аргументы** (Arguments) – укажите аргументы, с которыми должна быть выполнена команда из выбранного модуля.

  Примечание: Для некоторых модулей заполнение этого поля обязательно.

- **Степень подробности** (Verbosity) – выберите степень детализации журнала выполнения задания.

  Поддерживаются следующие значения:

  - `0 (Нормальный) (0 Normal)`;
  - `1 (Подробный) (1 Verbose)`;
  - `2 (Более подробный) (2 More Verbose)`;
  - `3 (Отладка) (3 Debug)`;
  - `4 (Отладка подключения) (4 Connection Debug)`;
  - `5 (Отладка WinRM) (5 WinRM Debug)`.

  Значение по умолчанию – `0 (Нормальный) (0 Normal)`.

- **Лимит** (Limit) – укажите шаблон названий управляемых узлов, на которых следует выполнить команду. В качестве разделителя названий узлов используется запятая.

  Если значение не указано, либо равно `all` или `*`, команда выполняется на всех узлах, входящих в инвентарный список.

  Подробности о шаблонах названий узлов см. в документации Ansible[141].

- **Ответвления** (Forks) – укажите количество параллельных процессов, используемых для выполнения задания.

  Если для этого параметра задано значение меньше 1, используется значение по умолчанию, равное 5.

  Максимальное значение этого параметра задается в значении *системной настройки контроллера* **Максимальное количество ответвлений задания** (Maximum number of forks per job).

- **Показать изменения** (Show changes).

  Если эта настройка включена, в журнал выполнения задания выводятся изменения, сделанные на управляемых узлах при выполнении команды (если это поддерживается выбранным модулем или командой).

  Эквивалентно выполнению команды `ansible-playbook` с параметром `--diff`.

- **Включить эскалацию привилегий** (Enable privilege escalation).

  Если эта настройка включена, задание запускается с повышенными привилегиями.

  Эквивалентно запуску команды `ansible-playbook` с параметром `--become`.

[138] https://docs.ansible.com/ansible/latest/collections/ansible/windows/win_service_module.html
[139] https://docs.ansible.com/ansible/latest/collections/ansible/windows/win_updates_module.html
[140] https://docs.ansible.com/ansible/latest/collections/ansible/windows/win_user_module.html
[141] https://docs.ansible.com/ansible/latest/inventory_guide/intro_patterns.html

Важно: Запрос пароля суперпользователя не производится. Убедитесь, что на управляемых узлах разрешено выполнение команды sudo без ввода пароля.

Инструкции по настройке приведены в секции *Использование sudo без ввода пароля*.

- **Дополнительные переменные** (Extra variables) – укажите дополнительные переменные, значения которых следует использовать при выполнении задания.

  Подробности о порядке присвоения значений переменным см. в документе *Переменные*.

4. Нажмите кнопку *Продолжить* (Next).
5. Выберите *среду исполнения*, в которой следует запустить задание.
6. Нажмите кнопку *Продолжить* (Next).
7. Выберите полномочия, используемые для доступа к управляемым узлам, и нажмите кнопку *Продолжить* (Next).
8. Во вкладке **Предварительный просмотр** (Preview) проверьте корректность введенных данных и нажмите кнопку *Запустить* (Run).

Удаление

Предупреждение: Особенности удаления инвентарных списков различного типа описаны в секции *Особенности удаления*.

Чтобы удалить инвентарный список, выполните следующие действия:

1. В таблице инвентарных списков нажмите на ссылку с названием удаляемого инвентарного списка.
2. Во кладке **Подробности** (Details) нажмите кнопку *Удалить* (Delete).
3. Подтвердите удаление.

Обычный инвентарный список

В этой секции приводятся инструкции по работе с *обычными инвентарными списками* через веб-интерфейс.

Создание обычного инвентарного списка

Для создания *обычного инвентарного списка* выполните следующие действия:

1. В окне **Инветарь** (Inventories) нажмите кнопку *Добавить* (Add) и в открывшемся меню выберите пункт **Добавить инвентарь** (Add Inventory).
2. Заполните форму **Создать новый инвентарь** (Create new inventory):
   - **Название** (Name) – укажите название инвентарного списка.

     Особенности заполнения поля:

     - Название инвентарного списка должно быть уникальным на уровне организации.
     - Название инвентарного списка не может состоять из одних пробельных символов (пробелы, табуляции и так далее).
     - Допускается использование символов кириллицы и специальных символов.

- **Описание** (Description) – укажите дополнительные данные об инвентарном списке, например, цель его создания или особенности использования.
- **Организация** (Organization) – выберите организацию, которой принадлежит инвентарный список.
- **Группы узлов контроллера** (Instance Groups) – группы узлов, используемые для запуска заданий, связанных с инвентарным списком.

  Подробности о порядке выбора узлов для запуска заданий см. в секции *Выбор группы узлов для запуска заданий*.
- **Метки** (Labels).

  С помощью меток можно группировать задания, связанные с инвентарным списком. Также метки удобно использовать для поиска нужных записей журнала выполнения заданий или их логической группировки.
- **Запретить откат группы узлов управления** (Prevent Instance Group Fallback).

  Если эта настройка включена, при запуске заданий, связанных с инвентарным списком, выбор предпочтительных групп узлов исполнения работает следующим образом:
  - Нельзя добавить группы, указанные как предпочтительные в свойствах организации.
  - Если в инвентарном списке список предпочтительных групп узлов пуст, используются глобальные настройки предпочтительных групп узлов.
- **Переменные** (Variables) – укажите переменные, действующие на уровне инвентарного списка.

  Подробности о порядке присвоения значений переменным см. в документе *Переменные*.

3. Нажмите кнопку *Сохранить* (Save).

После создания инвентарного списка его настройки можно изменять с помощью различных групп параметров, описанных в последующих секциях.

Управление группами узлов

Обычный инвентарный список имеет полную поддержку групп управляемых узлов и позволяет выполнять с ними следующие операции:

- создание;
- изменение;
- удаление;
- заполнение управляемыми узлами;
- построение иерархии.

Примечание: Группы узлов из источников создаются автоматически при синхронизации.

Для перехода к списку групп узлов и управления ими выполните следующие действия:

1. В таблице инвентарных списков нажмите на ссылку с названием нужного обычного инвентарного списка.
2. Выберите вкладку **Группы** (Groups).

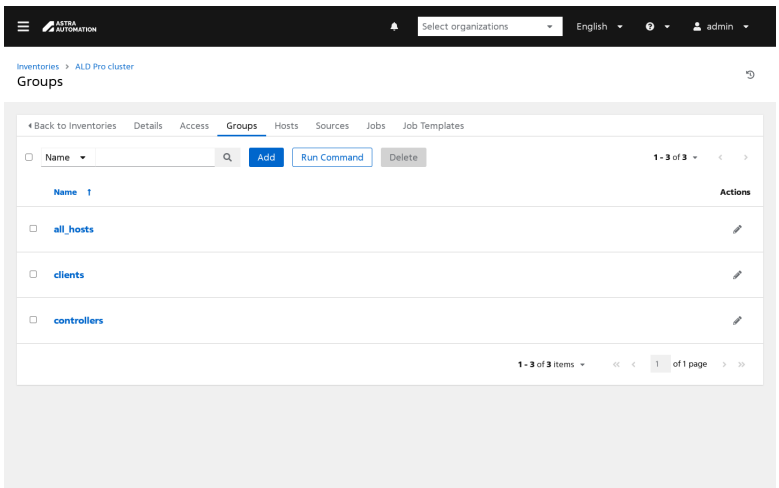


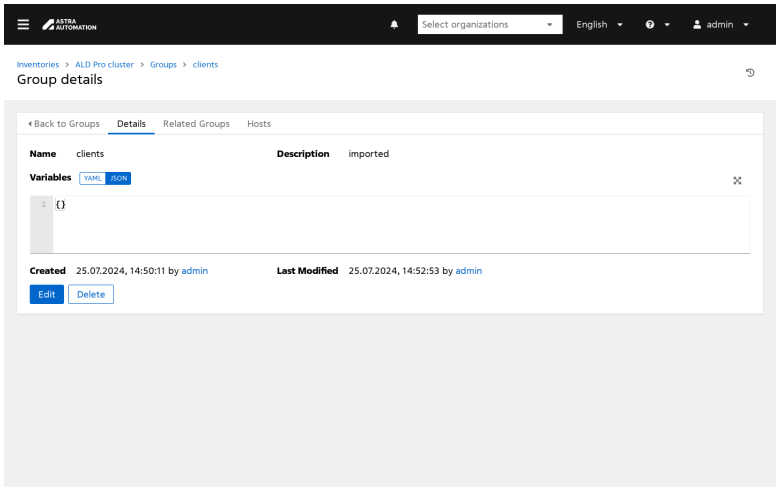
Таблица групп узлов обычного инвентарного списка состоит из следующих колонок:

- флаги для выбора нескольких записей;
- **Название** (Name);
- **Действия** (Actions) – кнопка перехода в окно редактирования свойств группы узлов.

Просмотр информации о группах узлов

Для просмотра информации о группе управляемых узлов выполните следующие действия:

1. В таблице инвентарных списков нажмите на ссылку с названием нужного обычного инвентарного списка.
2. Выберите вкладку **Группы** (Groups).
3. Нажмите на ссылку с названием нужной группы узлов.



Окно просмотра информации о группе управляемых узлов состоит из следующих вкладок:

- **Подробности** (Details) – общие сведения о группе управляемых узлов:
  - **Название** (Name) – название группы.
  - **Переменные** (Variables) – переменные группы.

    Подробности о порядке присвоения значений переменным см. в документе *Переменные*.
  - **Создано** (Created) – дата и время создания группы узлов, а также ссылка на профиль выполнившего его пользователя.

  - **Последнее изменение** (Last Modified) – дата и время последнего изменения группы узлов, а также ссылка на профиль выполнившего его пользователя.
- **Связанные группы** (Related Groups) – таблица дочерних групп узлов.

  Таблица дочерних групп узлов состоит из следующих колонок:
  - флаги для выбора нескольких записей;
  - **Название** (Name) – ссылка на дочернюю группу;
  - **Действия** (Actions) – кнопка изменения свойств дочерней группы.
- **Управляемые узлы** (Hosts) – таблица управляемых узлов, входящих в группу.

  Таблица управляемых узлов группы состоит из следующих колонок:
  - Флаги для выбора нескольких записей.
  - **Название** (Name) – ссылки для перехода в окно просмотра свойств управляемых узлов.
  - **Описание** (Description).
  - **Активность** (Activity).

    Для управляемых узлов, сведения о которых добавлены из внешних источников, в этом поле выводится значение `imported`.
  - **Действия** (Actions) – кнопки вызова часто выполняемых действий:
    * изменение *статуса управляемого узла*;
    * переход в окно редактирования свойств управляемого узла.

Создание группы узлов

Чтобы создать группу управляемых узлов, выполните следующие действия:

1. Во вкладке **Группы** (Groups) нажмите кнопку *Добавить* (Add).
2. Заполните форму **Создать новую группу** (Create new group):
   - **Название** (Name) – укажите название группы управляемых узлов.

     > **Предупреждение:** Название группы управляемых узлов должно быть уникально в пределах инвентарного списка.

   - **Описание** (Description) – укажите описание группы управляемых узлов.
   - **Переменные** (Variables) – укажите значения переменных, действующих на уровне группы управляемых узлов.

     Подробности о порядке присвоения значений переменным см. в документе *Переменные*.
3. Нажмите кнопку *Сохранить* (Save).

Создание иерархии групп узлов

Чтобы создать иерархические отношения между группами управляемых узлов, выполните следующие действия:

1. Во вкладке **Группы** (Groups) нажмите на ссылку с названием группы узлов, которая должна быть родительской. Далее эта группа называется родительской группой.
2. В окне просмотра информации о родительской группе выберите вкладку **Связанные группы** (Related Groups).
3. Нажмите кнопку *Добавить* (Add) и в открывшемся меню выберите соответствующий пункт:
   - **Добавить существующую группу** (Add existing group) – чтобы сделать дочерними уже существующие группы узлов;
   - **Добавить новую группу** (Add new group) – чтобы создать новую дочернюю группу узлов.
4. Если вы выбрали добавление существующих групп узлов, в открывшемся окне **Выберите группы** (Select Groups) включите флаги в строках с названиями нужных групп узлов и нажмите кнопку *Сохранить* (Save).
5. Если вы выбрали создание новой дочерней группы узлов, заполните форму **Создать новую группу** (Create new group) и нажмите кнопку *Сохранить* (Save).

Удаление группы узлов

Чтобы удалить группу узлов, выполните следующие действия:

1. В таблице инвентарных списков нажмите на ссылку с названием нужного обычного инвентарного списка.
2. Выберите вкладку **Группы** (Groups).
3. В таблице групп управляемых узлов нажмите на ссылку с названием удаляемой группы.
4. В диалоговом окне **Удалить группу?** (Delete Group?) выполните следующие действия:
   1. Выберите действие, которое следует выполнить с дочерними группами и входящими в удаляемую группу узлами:
      - **Удалить все группы и узлы** (Delete All Groups and Hosts) – удалить;
      - **Добавить дочерние узлы и группы** (Promote Child Groups and Hosts) – сохранить.
   2. Нажмите кнопку *Удалить* (Delete).

Управление источниками

Обычный инвентарный список поддерживает получение сведений об управляемых узлах из одного или нескольких внешних источников.

Примечание: Чтобы в контроллере появились сведения об управляемых узлах из источника, после его добавления необходимо запустить синхронизацию.

Для перехода к списку источников и управления ими выполните следующие действия:

1. В таблице инвентарных списков нажмите на ссылку с названием нужного обычного инвентарного списка.
2. Выберите вкладку **Источники** (Sources).

Добавление источника

Чтобы добавить в обычный инвентарный список источник сведений об управляемых узлах, выполните следующие действия:

1. Во вкладке **Источники** (Sources) нажмите кнопку *Добавить* (Add).
2. Заполните форму **Создать новый источник** (Create new source):
   - **Название** (Name) – укажите название источника сведений об управляемых узлах.
   - **Описание** (Description) – укажите описание источника, например, название связанного проекта.
   - **Среда исполнения** (Execution Environment) – укажите среду исполнения, используемую для работы с источником.

     Указанное здесь значение имеет более высокий приоритет, чем среда исполнения, указанная на уровне организации, команды или инвентаря.
   - **Источник** (Source) – выберите тип источника сведений об управляемых узлах.
3. Заполните поля, соответствующие выбранному источнику.

   Примечание: Поля **Проект** (Project) и **Инвентарный файл** (Inventory file) доступны только для источников типа «Взято из проекта» (Source from a Project). При этом значение в поле **Инвентарный файл** (Inventory file) можно выбрать только после выбора проекта.

   - **Полномочия** (Credential) – если для доступа к источнику требуется аутентификация, выберите полномочие соответствующего типа в этом поле.
   - **Проект** (Project) – выберите проект, служащий источником сведений об управляемых узлах.
   - **Инвентарный файл** (Inventory file) – выберите файл инвентаря из состава проекта.

     При выборе значения `/ (project root)` источниками сведений об управляемых узлах считаются все файлы инвентаря, которые контроллер найдет в проекте. Это значение рекомендуется выбирать только в том случае, когда инвентарь в проекте хранится в виде иерархической структуры из файлов и каталогов.
   - **Степень подробности** (Verbosity) – укажите степень подробности вывода Ansible в журнал выполнения задания синхронизации инвентаря.

     Поддерживаются следующие значения:

     - `0 (Предупреждение) (0 Warnning)` – выводится информация о событиях со степенью важности не ниже «Предупреждение».
     - `1 (Информация) (1 Info)` – выводится информация обо всех событиях, происходящих при синхронизации инвентаря.
     - `2 (Отладка) (2 Debug)` – выводится вся доступная информация обо всех событиях, происходящих при обработке инвентаря.

     Значение по умолчанию – `1 Информация (1 Info)`.
   - **Фильтр узлов** (Hosts Filter) – укажите регулярное выражение, используемое для фильтрации узлов по названию. Этот фильтр применяется последним, после всех остальных фильтров, предоставляемых используемым расширением инвентаря.
   - **Включенная переменная** (Enabled Variable) – укажите название переменной из данных об управляемом узле, значение которой следует использовать для определения статуса узла после импорта.
   - **Разрешенное значение** (Enabled Value).

Чтобы управляемый узел импортировался со статусом «Вкл» (On), должны выполниться два условия:

- в свойствах управляемого узла есть переменная с названием, указанным в поле **Включенная переменная** (Enabled Variable);
- значение этой переменной совпадает с указанным в поле **Разрешенное значение** (Enabled Value).

Если такой переменной нет или ее значение отличается от указанного в этом поле, узел импортируется со статусом «Выкл» (Off).

- **Перезаписать** (Overwrite).

  Если эта настройка включена, синхронизация с источником работает следующим образом:

  - Импортированные ранее узлы и группы, удаленные в источнике, удаляются из инвентарного списка.
  - Узлы и группы, не привязанные к источнику, перемещаются в первую найденную группу, созданную вручную. Если такой группы нет, все не привязанные к источнику узлы и группы остаются в группе по умолчанию (`all`).

  Если эта настройка выключена, при синхронизации из инвентарного списка **не** удаляются узлы и группы, удаленные в источнике.

> **Предупреждение: Известная проблема**
>
> При удалении узла из источника инвентаря синхронизация не приводит к удалению его из инвентаря в контроллере. Поэтому такой узел необходимо удалить вручную из инвентаря контроллера.

- **Перезаписывать переменные** (Overwrite variables).

  Если эта настройка включена, при синхронизации с источником значения переменных инвентаря, хранящихся в контроллере, удаляются и заменяются значениями из источника.

  Если эта настройка выключена, при синхронизации с источником хранящиеся в контроллере значения переменных объединяются со значениями в источнике.

Примечание: Для одноименных переменных значение из источника заменяет значение, хранящееся в контроллере.

- **Обновление при запуске** (Update on launch).

  Если эта настройка включена, при запуске связанного с инвентарем задания автоматически запускается задание синхронизации инвентаря с источником.

- **Таймаут кэширования (в секундах)** (Cache timeout, seconds).

Примечание: Эта настройка доступна только если включен флаг **Обновление при запуске** (Update on launch)

Время, в течение которого с момента последней синхронизации данные об управляемых узлах считаются свежими.

Эта настройка может быть полезна, если задания обновления инвентаря создаются быстрее, чем выполняются. Если данные в кеше свежие, задание синхронизации считается успешно выполненным без запуска.

При значении `0` задания синхронизации запускаются всегда.

Значение по умолчанию – `0`.

- **Исходные переменные** (Source variables).

  При использовании источника типа «Взято из проекта» (Sourced from a Project) переменные объединяются с переменными инвентаря.

  При использовании источников других типов в этом поле задаются настройки соответствующего расширения инвентаря.

  Ссылки на документацию расширений инвентаря см. в секции *Поддержка расширений*.

4. Нажмите кнопку *Сохранить* (Save).

Синхронизация списка узлов с источником

Синхронизация используется для первичного импорта сведений об управляемых узлах в контроллер и для актуализации сведений об импортированных ранее узлах.

Чтобы запустить синхронизацию инвентарного списка с источниками, во вкладке **Источники** (Sources) выполните следующие действия:

- Если нужно синхронизировать инвентарный список со всеми источниками, нажмите кнопку *Синхронизировать все* (Sync all).
- Если нужно синхронизировать инвентарный список только с определенным источником, выполните следующие действия:
  1. В таблице со списком источников нажмите на ссылку с названием нужного источника.
  2. В окне свойств источника нажмите кнопку *Синхронизировать* (Sync).
  3. Чтобы отслеживать ход выполнения задания синхронизации, нажмите на ссылку в поле **Статус** (Status).

Настройка расписания синхронизации списка узлов с источником

Чтобы синхронизация с источником выполнялась по расписанию, во вкладке **Источники** (Sources) выполните следующие действия:

1. В таблице со списком источников нажмите на ссылку с названием нужного источника.
2. Во вкладке **Расписание** (Schedules) нажмите кнопку *Добавить* (Add).
3. Заполните форму **Создать новое расписание** (Create New Schedule):
   - **Название** (Name) – укажите название расписания.
   - **Описание** (Description) – укажите описание расписания.
   - **Дата и время начала** (Start date/time) – укажите дату и время первого запуска задания.
   - **Локальный часовой пояс** (Local time zone) – выберите часовой пояс, по которому рассчитывается время.

     Значение по умолчанию – `UTC`.
   - **Частота повторения** (Repeat frequency) – выберите частоту повторения запусков задания.

     Если выбрано значение **Нет (запускается один раз)** (None (run once)), задание будет выполнено один раз.

     Включение флагов **Мин** (Min), **Час** (Hour), **День** (Day), **Неделя** (Week), **Месяц** (Month) и **Год** (Year) приводит к добавлению в форму наборов полей, позволяющих выполнить настройку соответствующих временных интервалов. Для всех типов интервалов доступны следующие настройки:

- **Выполнять каждые** (Run every) – количество выбранных временных интервалов между запусками задания.
- **Завершено** (End) – условие прекращения запуска задания по расписанию. Доступны следующие настройки:
  * **Никогда** (Never) – не прекращать.
  * **После указанного количества вхождений** (After number of occurences) – после указанного количества запусков.
  * **На дату** (On date) – в указанные день и время.

Для недельных интервалов доступна дополнительная настройка **В дни** (Days). Она позволяет выбрать дни недели, по которым следует запускать задание.

Для месячных интервалов доступна дополнительная настройка **Выполнить в** (Run on). Она позволяет выбрать определенную дату или относительную позицию дня в месяце. Например, при выборе даты и указании числа «20» задание будет запускаться двадцатого числа каждого месяца. При выборе относительной позиции нет привязки к конкретной дате. Вместо этого настраивается запуск в дни, обозначаемые как, например, «второй понедельник» или «последняя пятница». Также поддерживаются значения, позволяющие выбрать рабочие или нерабочие дни.

Для годовых интервалов доступна дополнительная настройка **Выполнить в** (Run on). Она позволяет выбрать месяц и число либо относительную позицию дня в году. Например, при выборе месяца «Январь» и числа «10» задание будет запускаться каждый год десятого января. При выборе относительной позиции дня нет привязки к конкретной дате. Вместо этого настраивается запуск в дни, обозначаемые как, например, «первый четверг мая» или «последний день февраля». Также поддерживаются значения, позволяющие выбрать рабочие или нерабочие дни.

- **Исключения** (Exceptions) – настройки, предотвращающие запуск задания в указанные минуты, часы, дни, месяцы и годы. Эти настройки работают по тем же принципам, что и настройки частоты повторений.

4. Нажмите кнопку *Сохранить* (Save).

Удаление источника

> **Предупреждение:** При удалении источника удаляются все импортированные из него управляемые узлы.

Чтобы удалить источник, выполните следующие действия:

1. В таблице инвентарных списков нажмите на ссылку с названием нужного обычного инвентарного списка.
2. Выберите вкладку **Источники** (Sources).
3. В таблице источников нажмите на ссылку с названием удаляемого источника.
4. В окне свойств источника нажмите кнопку *Удалить* (Delete).
5. Подтвердите удаление источника.

Умный инвентарный список

В этой секции приводятся инструкции по работе с *умными инвентарными списками* через веб-интерфейс.

Важно: Эта часть документации находится в стадии разработки.

Сборный инвентарный список

В этой секции приводятся инструкции по работе со *сборными инвентарными списками* через веб-интерфейс.

Создание сборного инвентарного списка

Для создания сборного инвентаря выполните следующие действия:

1. В окне **Инветарь** (Inventories) нажмите на кнопку *Добавить* (Add) и в открывшемся меню выберите пункт **Добавить сборный инвентарь** (Add constructed inventory).
2. Заполните форму **Создать новый сборный инвентарь** (Create new constructed inventory):
   - **Название** (Name) – укажите название инвентарного списка.

     Особенности заполнения поля:

     - Название инвентарного списка должно быть уникальным на уровне организации.
     - Название инвентарного списка не может состоять из одних пробельных символов (пробелы, табуляции и так далее).
     - Допускается использование символов кириллицы и специальных символов.
   - **Описание** (Description) – укажите дополнительные данные об инвентарном списке, например, цель его создания или особенности использования.
   - **Организация** (Organization) – выберите организацию, которой принадлежит инвентарный список.
   - **Группы узлов контроллера** (Instance Groups) – группы узлов, используемые для запуска заданий, связанных с инвентарным списком.

     Подробности о порядке выбора узлов для запуска заданий см. в секции *Выбор группы узлов для запуска заданий*.
   - **Входные данные инвентаря** (Input Inventories) – выберите инвентарные списки, которые хотите использовать для получения сведений об управляемых узлах.
   - **Таймаут кэширования (в секундах)** (Cache timeout (seconds)).

     При синхронизации сборного инвентаря автоматически создается источник сведений об управляемых узлах. Эта настройка задает период времени, в течение которого данные в источнике считаются свежими.

     Эта настройка может быть полезна, если задания обновления инвентаря создаются быстрее чем выполняются. Если данные в кеше свежие, задание синхронизации считается успешно выполненным без запуска.

     При значении `0` задания синхронизации запускаются всегда.

     Значение по умолчанию – `0`.

- **Степень подробности** (Verbosity) – выберите степень детализации журнала выполнения задания синхронизации инвентарного списка.

  Поддерживаются следующие значения:

  - `0 (Нормальный) (0 Normal)`;
  - `1 (Подробный) (1 Verbose)`;
  - `2 (Более подробный) (2 More Verbose)`;
  - `3 (Отладка) (3 Debug)`;
  - `4 (Отладка подключения) (4 Connection Debug)`;
  - `5 (Отладка WinRM) (5 WinRM Debug)`.

  Значение по умолчанию – `0 (Нормальный) (0 Normal)`.

- **Лимит** (Limit) – укажите выражение, используемое для фильтрации управляемых узлов.
- **Исходные переменные** (Source vars) – укажите параметры используемого расширения Ansible и привила его использования.

3. Нажмите кнопку *Сохранить* (Save).
4. Во вкладке **Подробности** (Details) нажмите кнопку *Синхронизировать* (Sync).

Управляемые узлы

Окно **Управляемые узлы** (Hosts) используется для управления сведениями об управляемых узлах и запуска на них специальных (Ad-Hoc) команд. Для этого необходимо перейти в соответствующий раздел с помощью панели навигации: *Ресурсы* ▸ *Управляемые узлы* (*Resources* ▸ *Hosts*).

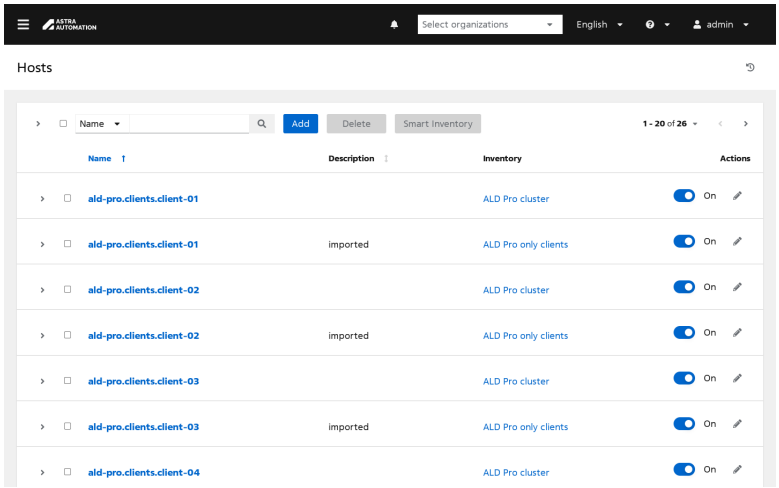


Таблица управляемых узлов состоит из следующих колонок:

- Переключатели вывода информации о статусах заданий, связанных с управляемым узлом.

  При нажатии на переключатель выводятся иконки, показывающие статус выполнения заданий, связанных с управляемым узлом. Нажатие на иконку приводит к переходу в окно просмотра вывода Ansible при выполнении задания.

- Флаги для выбора нескольких записей.
- **Название** (Name) – ссылка на окно просмотра свойств управляемого узла.
- **Описание** (Description).

  Для управляемых узлов, сведения о которых добавлены из внешних источников, в этом поле выводится значение `imported`.

- **Инвентарь** (Inventory) – ссылка на окно просмотра свойств *инвентарного списка*, в который входит управляемый узел.
- **Действия** (Actions) – кнопки для быстрого вызова часто выполняемых действий:
  - переключатель *статуса управляемого узла*;
  - кнопка перехода в окно редактирования свойств управляемого узла.

Просмотр

Для получения подробных сведений об управляемом узле нажмите на ссылку с его названием в таблице управляемых узлов. Окно сведений об управляемом узле состоит из следующих вкладок:

- **Подробности** (Details) – общие сведения об управляемом узел.
- **Факты** (Facts) – факты Ansible.

  Примечание: Факты доступны для просмотра после сбора.

- **Группы** (Groups) – таблица групп узлов, в которые входит управляемый узел. Состоит из следующих колонок:
  - флаги для выбора нескольких записей;
  - **Название** (Name) – ссылка на окно просмотра свойств группы узлов;
  - **Действия** (Actions) – кнопка перехода в окно редактирования свойств группы узлов.
- **Задания** – таблица заданий, связанных с управляемым узлом. Состоит из следующих колонок:
  - переключатель подробности вывода основных сведений о задании;
  - флаги для выбора нескольких записей;
  - **Название** (Name) – ссылка на окно просмотра журнала выполнения задания;
  - **Статус** (Status) – текущий статус выполнения задания;
  - **Время начала** (Start Time) – время запуска задания;
  - **Время завершения** (Finish Time) – время завершения, отмены или прерывания задания;
  - **Действия** (Actions) – кнопка перезапуска задания.

Создание

Для создания записи об управляемом узле выполните следующие действия:

1. В окне **Управляемые узлы** (Hosts) нажмите кнопку *Добавить* (Add).
2. Заполните форму **Создать новый узел** (Create New Host):
   - **Название** (Name) – укажите название управляемого узла.
   - **Описание** (Description) – укажите дополнительные сведения об управляемом узле.
   - **Инвентарь** (Inventory) – выберите *обычный инвентарный список*, в состав которого добавляется управляемый узел.
   - **Переменные** (Variables) – укажите переменные, действующие на уровне управляемого узла.

Подробности о порядке присвоения значений переменным см. в документе *Переменные*.

3. Нажмите кнопку *Сохранить* (Save).

Включение в группу

Управляемые узлы можно включать только в существующие группы в том же инвентарном списке, которому принадлежат сами узлы. Пошаговые инструкции по управлению группами управляемых узлов представлены в секции *Управление группами узлов*.

Чтобы включить управляемый узел в группы узлов, выполните следующие действия:

1. Во вкладке **Группы** (Groups) нажмите кнопку *Добавить* (Add).
2. В открывшемся окне **Выберите группы** (Select Groups) включите флаги в строках с теми группами узлов, в которые хотите включить управляемый узел.
3. Нажмите кнопку *Сохранить* (Save).

Исключение из группы

Чтобы исключить управляемый узел из групп узлов, выполните следующие действия:

1. Во вкладке **Группы** (Groups) включите флаги для групп, из которых хотите исключить управляемый узел.
2. Нажмите кнопку *Отвязать* (Disassociate).
3. Подтвердите исключение узла из групп узлов.

Удаление

Чтобы удалить запись об управляемом узле, выполните следующие действия:

1. В таблице управляемых узлов нажмите на ссылку с названием удаляемого управляемого узла.
2. Во вкладке **Подробности** (Details) нажмите кнопку *Удалить* (Delete).
3. Подтвердите удаление записи об управляемом узле.

Доступ

Раздел **Доступ** (Access) содержит подразделы, используемые для управления организациями, пользователями и командами.

Организации

Окно **Организации** (Organizations) используется для управления *организациями*. Организация объединяет в себе множество пользователей и команд, а также является владельцем различных ресурсов. Для создания организаций и управления ими необходимо перейти в соответствующий раздел с помощью панели навигации: *Доступ* ‣ *Организации* (*Access* ‣ *Organizations*).

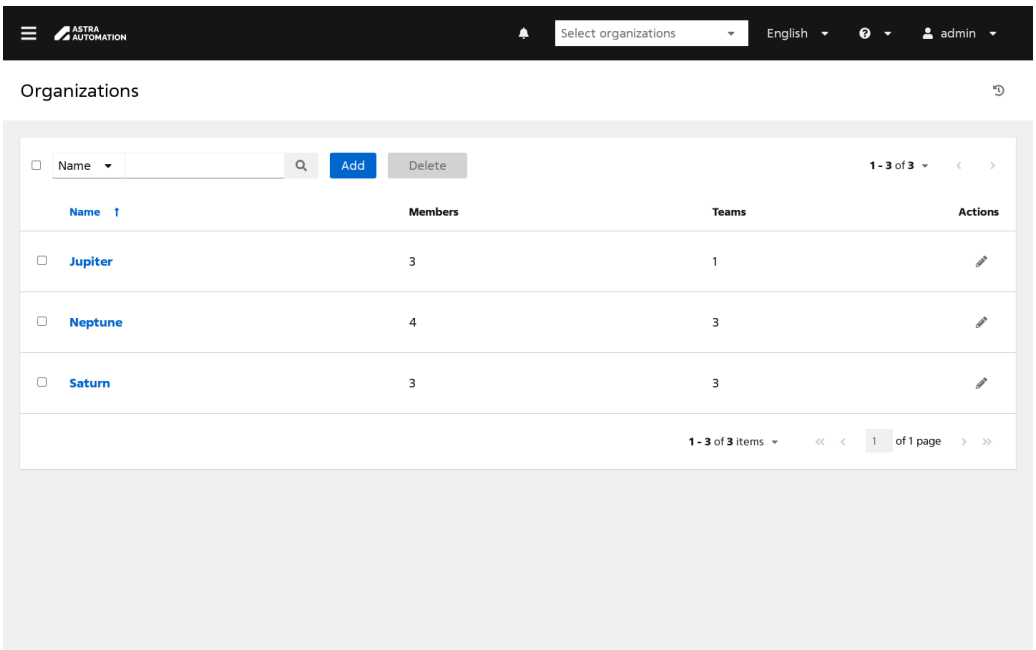


Таблица организаций состоит из следующих колонок:

- Флаги для выбора нескольких записей.
- **Название** (Name).
- **Участники** (Members).

  При расчете числа участников не учитываются пользователи, которым не назначена роль «Участник» (Member). Однако, эти пользователи имеют на управление ресурсами организации привилегии, предоставляемые остальными назначенными ролями.
- **Команды** (Teams) – количество команд, связанных с организацией.
- **Действия** (Actions) – кнопка перехода в окно редактирования свойств организации.

Просмотр

Для получения подробных сведений об организации нажмите на ссылку с ее названием в таблице организаций.

Окно сведений об организации состоит из следующих вкладок:

- **Подробности** (Details)– общие сведения об организации;
- **Доступ** (Access) – список пользователей и назначенных им ролей;
- **Команды** (Teams) – список команд организации;
- **Среды исполнения** (Execution Environments) – список сред исполнения, принадлежащих организации;
- **Уведомления** (Notifications) – список уведомлений, связанных с организацией.

Создание

Для создания организации выполните следующие действия:

1. В окне **Организации** (Organizations) нажмите кнопку *Добавить* (Add).
2. Заполните форму **Создать новую организацию** (Create New Organization):
   - **Название** (Name) – название организации.

     Особенности заполнения поля:

     - Не допускается создание двух организаций с одним и тем же названием.
     - Название организации не может состоять из одних пробельных символов (пробелы, табуляции и тому подобное).
     - Допускается использование символов кириллицы и специальных символов.
   - **Описание** (Description).

     В описании можно указать дополнительные данные об организации, используемые при администрировании контроллера. Например, контакты ответственных лиц или назначение организации.
   - **Группы узлов контроллера** (Instance Groups) – группы узлов контроллера, которые следует использовать для запуска заданий, созданных пользователями организации.

     Подробности о порядке выбора узлов для запуска заданий см. в секции *Выбор группы узлов для запуска заданий*.
   - **Среда исполнения** – *среда исполнения*, которую по умолчанию следует использовать на уровне организации. Эта настройка имеет более высокий приоритет чем та, которая задана *на уровне контроллера*.
   - **Учетные данные Galaxy** (Galaxy Credentials) – список полномочий типа *API-токен Ansible Galaxy/Automation Hub*.

     По умолчанию в этом поле указано только полномочие для доступа к реестру коллекций Ansible Galaxy[142]. Для доступа к другим реестрам коллекций Ansible, например, Automation Hub и Private Automation Hub, создайте полномочия указанного типа и выберите их в этом поле.
3. Нажмите кнопку *Сохранить* (Save).

Изменение

Для изменения свойств организации выполните следующие действия:

1. В окне **Организации** (Organizations) нажмите на ссылку с названием организации.
2. Нажмите кнопку *Редактировать* (Edit).
3. Внесите необходимые изменения в форме **Редактировать детали** (Edit Details):
   - **Название** (Name) – название организации.

     Особенности заполнения поля:

     - Не допускается создание двух организаций с одним и тем же названием.
     - Название организации не может состоять из одних пробельных символов (пробелы, табуляции и тому подобное).
     - Допускается использование символов кириллицы и специальных символов.

[142] https://galaxy.ansible.com/

- **Описание** (Description).

  В описании можно указать дополнительные данные об организации, используемые при администрировании контроллера. Например, контакты ответственных лиц или назначение организации.

- **Группы узлов контроллера** (Instance Groups) – группы узлов контроллера, которые следует использовать для запуска заданий, созданных пользователями организации.

  Подробности о порядке выбора узлов для запуска заданий см. в секции *Выбор группы узлов для запуска заданий*.

- **Среда исполнения** – *среда исполнения*, которую по умолчанию следует использовать на уровне организации. Эта настройка имеет более высокий приоритет чем та, которая задана *на уровне контроллера*.

- **Учетные данные Galaxy** (Galaxy Credentials) – список полномочий типа *API-токен Ansible Galaxy/Automation Hub*.

  По умолчанию в этом поле указано только полномочие для доступа к реестру коллекций Ansible Galaxy[143]. Для доступа к другим реестрам коллекций Ansible, например, Automation Hub и Private Automation Hub, создайте полномочия указанного типа и выберите их в этом поле.

4. Нажмите кнопку *Сохранить* (Save).

Удаление

> **Предупреждение:** Вместе с организацией удаляются связанные с ней ресурсы. Принадлежащие организации пользователи и команды при этом не удаляются, а открепляются от удаляемой организации и могут быть в дальнейшем привязаны к другой.

Чтобы удалить организацию, выполните следующие действия:

1. В окне **Организации** (Organizations) нажмите на ссылку с названием удаляемой организации.
2. Во вкладке **Подробности** (Details) нажмите кнопку *Удалить* (Delete).
3. Подтвердите удаление.

Настройка доступа

Во вкладке **Доступ** (Access) выводится список связанных с организацией *пользователей* и назначенных им *ролей*. Нажатие на ссылку с именем пользователя приводит к переходу в окно просмотра сведений о пользователе.

Назначение ролей

Чтобы назначить пользователям или командам роли на уровне организации, выполните следующие действия:

1. Во вкладке **Доступ** (Access) нажмите кнопку *Добавить* (Add).
2. Выберите тип объектов, которым назначаются роли:
   - *Пользователи* (Users) – если вы хотите назначить роли одному или нескольким пользователям;
   - *Команды* (Teams) – если вы хотите назначить роли всем пользователям команд.

[143] https://galaxy.ansible.com/

3. Нажмите кнопку *Продолжить* (Next).
4. Выберите пользователей или команды, которым хотите назначить роли, и нажмите кнопку *Продолжить* (Next).
5. Выберите роли, которые хотите назначить выбранным на предыдущем шаге пользователям или командам.

   Примечание: При *связывании пользователя и команды* выбранные командные роли будут назначены пользователю автоматически.

6. Нажмите кнопку *Сохранить* (Save).

Отзыв ролей

Чтобы отозвать роль у пользователя, выполните следующие действия:

1. Во вкладке **Доступ** (Access) нажмите кнопку × рядом с названием отзываемой роли.
2. Подтвердите отзыв.

Примечание: Роли «Системный администратор» (System Administrator) и «Системный аудитор» (System Auditor) нельзя отозвать через настройку доступа на уровне организации. Используйте для этого изменение свойств пользователя.

Уведомления

Во вкладке **Уведомления** (Notifications) выводится список уведомлений, связанных с организацией. Для каждого уведомления доступны переключатели, управляющие типом событий, при наступлении которых должно быть отправлено уведомление:

- **Согласование** (Approval) – согласован переход к очередному заданию в потоке заданий;
- **Пуск** (Start) – запущено задание или поток заданий;
- **Успех** (Success) – успешно выполнено задание или поток заданий;
- **Сбой** (Failure) – при выполнении задания произошла ошибка, для которой не предусмотрен обработчик.

Примечание: Если уведомления для организации не настроены, отображается пустой список. Для создания уведомлений и управления ими выберите на панели навигации *Администрирование* ▸ *Уведомления* (*Administration* ▸ *Notifications*).

Примеры

Изучите управление организациями на примерах.

Назначение роли отдельному пользователю

Пусть пользователю `alex` необходимо назначить роль, позволяющую управлять всеми проектами организации. Для этого в окне организации выполните следующие действия:

1. Во вкладке **Доступ** (Access) нажмите кнопку *Добавить* (Add).
2. В открывшемся окне выберите *Пользователи* (Users) и нажмите кнопку *Продолжить* (Next).
3. Из списка пользователей выберите `alex` и нажмите кнопку *Продолжить* (Next).
4. Из списка ролей выберите **Администратор проекта** (Project Admin).
5. Нажмите кнопку *Сохранить* (Save).

Назначение ролей команде

Пусть для управления инфраструктурой используется команда с названием `DevOps Team`. Всем ее участникам необходимо предоставить следующие привилегии:

- управление средами исполнения;
- управление проектами.

Для этого в окне организации выполните следующие действия:

1. Во вкладке **Доступ** (Access) нажмите кнопку *Добавить* (Add).
2. В открывшемся окне выберите *Команды* (Teams) и нажмите кнопку *Продолжить* (Next).
3. Из списка команд выберите `DevOps Team` и нажмите кнопку *Продолжить* (Next).
4. Из списка ролей выберите **Администратор проекта** (Project Admin) и **Администратор среды исполнения** (Execution Environment Admin).
5. Нажмите кнопку *Сохранить* (Save).

Пользователи

Окно **Пользователи** (Users) используется для управления *пользователями*. Для создания пользователей и управления ими необходимо перейти в соответствующий раздел с помощью панели навигации: *Доступ* ▸ *Пользователи* (*Access* ▸ *Users*).

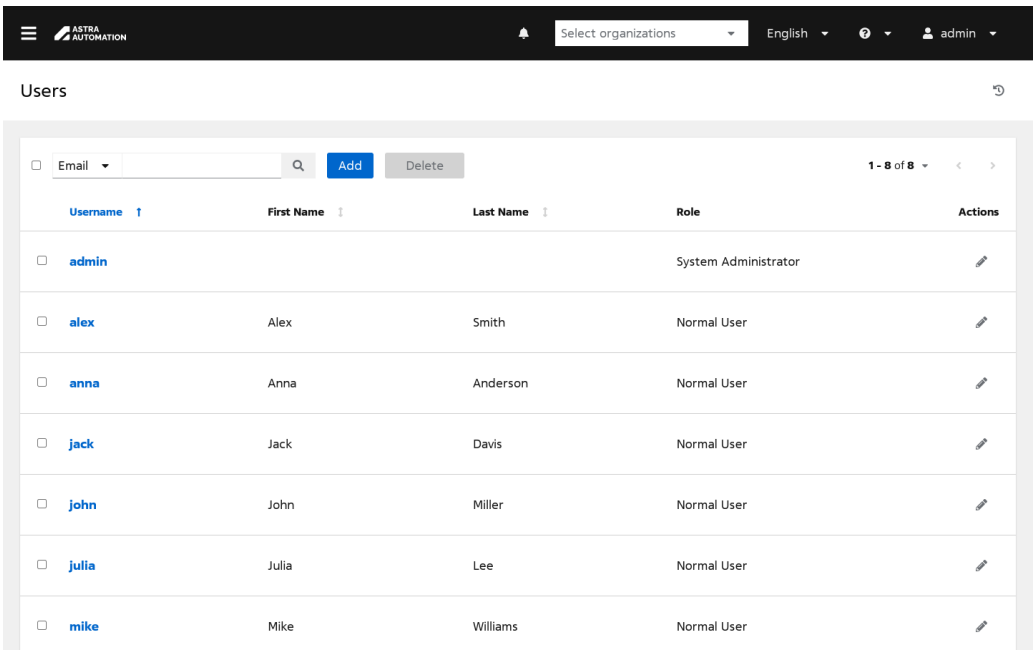


Таблица пользователей состоит из следующих колонок:

- флаги для выбора нескольких записей;
- **Имя пользователя** (Username) – название учетной записи пользователя, используемой для работы с контроллером;
- **Имя** (First Name) – имя пользователя;
- **Фамилия** (Last Name) – фамилия пользователя;
- **Роль** (Role) – тип пользователя;
- **Действия** (Actions) – кнопка перехода в окно редактирования сведений о пользователе.

Просмотр

Для получения подробных сведений о пользователе нажмите на ссылку с названием его учетной записи в таблице пользователей.

Окно сведений о пользователе состоит из следующих вкладок:

- **Подробности** (Details) – общие сведения о пользователе.
- **Организации** (Organizations) – список организаций, участником которых является пользователь.

  Чтобы сделать пользователя участником организации, *назначьте ему роль* «Участник» (Member) на ее уровне.
- **Команды** (Teams) – список команд, участником которых является пользователь.

  Чтобы сделать пользователя участником команды, *свяжите его с нужной командой*.
- **Токены** (Tokens) – список токенов доступа, созданных пользователем.

  Примечание: Эта вкладка доступна только при просмотре собственной учетной записи.

Создание

Для создания пользователя выполните следующие действия:

1. В окне **Пользователи** (Users) нажмите кнопку *Добавить* (Add).
2. Заполните форму **Создать нового пользователя**:
   - **Имя** (First Name) – укажите имя пользователя.
   - **Фамилия** (Last Name) – укажите фамилию пользователя.
   - **Электронная почта** (Email) – укажите адрес электронной почты пользователя.
   - **Имя пользователя** (Username) – укажите название учетной записи пользователя, используемой для работы с контроллером.

     Требования к названию учетной записи:

     - Максимально допустимая длина – 150 символов.
     - Разрешено использование букв, цифр и символов `_`, `@`, `+`, `.`, `-`.
     - Уникальность на уровне контроллера.
   - **Пароль** (Password) и **Подтвердить пароль** (Confirm Password) – укажите пароль и подтверждение пароля пользователя.

     Единственное требование к паролю и его подтверждению – длина не менее 8 символов.

- **Тип пользователя** (User Type) – выберите тип пользователя.

  Подробности о типах пользователей см. в секции *Типы пользователей*.

- **Организация** (Organization) – выберите организацию, которой принадлежит пользователь.

3. Нажмите кнопку *Сохранить* (Save).

Удаление

Чтобы удалить пользователя, выполните следующие действия:

1. В таблице пользователей нажмите на ссылку с названием учетной записи удаляемого пользователя.
2. Во вкладке **Подробности** (Details) нажмите кнопку *Удалить* (Delete).
3. Подтвердите удаление.

Настройка доступа

Окно **Пользователи** (Users) предоставляет функциональность, позволяющую выполнять с пользователями следующие действия:

- связывать с командами;
- назначение индивидуальные роли на доступ к экземплярам ресурсов.

Связывание с командой

Чтобы связать пользователя с командой, выполните следующие действия:

1. Во вкладке **Команды** (Teams) нажмите кнопку *Связать* (Associate).
2. Выберите команды, с которыми хотите связать пользователя, и нажмите кнопку *Сохранить* (Save).

Назначение индивидуальных ролей

Чтобы назначить пользователю роль, выполните следующие действия:

1. Во вкладке **Роли** (Roles) нажмите кнопку *Добавить* (Add).
2. Выберите тип объектов.

   Поддерживаются следующие значения:

   - *шаблоны заданий (job templates)*;
   - *шаблоны потока заданий (workflow job templates)*;
   - полномочия (credentials);
   - *инвентарь (inventories)*;
   - *проекты (projects)*;
   - *организации (organizations)*;
   - группы узлов контроллера (instance groups).
3. Нажмите кнопку *Продолжить* (Next).
4. Выберите экземпляры объектов и нажмите кнопку *Продолжить* (Next).
5. Выберите роли. Список доступных ролей зависит от выбранного ранее типа объектов.
6. Нажмите кнопку *Сохранить* (Save).

Отзыв ролей

Чтобы отозвать у пользователя роль, выполните следующие действия:

1. Во вкладке **Роли** (Roles) нажмите кнопку × рядом с названием отзываемой роли.
2. Подтвердите отзыв.

Смена пароля

Чтобы изменить пароль пользователя, выполните следующие действия:

1. В таблице пользователей нажмите на ссылку с названием учетной записи нужного пользователя.
2. Во вкладке **Подробности** (Details) нажмите кнопку *Редактировать* (Edit).
3. Укажите новый пароль в полях **Пароль** (Password) и **Подтвердить пароль** (Confirm Password).
4. Нажмите кнопку *Сохранить* (Save).

Смена типа пользователя

Чтобы изменить *тип пользователя*, выполните следующие действия:

1. В таблице пользователей нажмите на ссылку с названием учетной записи нужного пользователя.
2. Во вкладке **Подробности** (Details) нажмите кнопку *Редактировать* (Edit).
3. В поле **Тип пользователя** (User Type) выберите нужное значение.
4. Нажмите кнопку *Сохранить* (Save).

Управление токенами

Токены позволяют сторонним приложениям получить доступ к контроллеру от имени выбранного пользователя. Каждый пользователь создает свои токены самостоятельно.

Для управления токенами выполните следующие действия:

1. Войдите в веб-интерфейс контроллера от имени нужного пользователя.
2. На панели навигации выберите *Доступ* ‣ *Пользователи* (*Access* ‣ *Users*).
3. Нажмите на ссылку с названием учетной записи активного пользователя.

   Совет: Для быстрого перехода в нужное окно также можно выбрать в пользовательском меню пункт **Сведения о пользователе** (User Details).

4. Выберите вкладку **Токены** (Tokens).

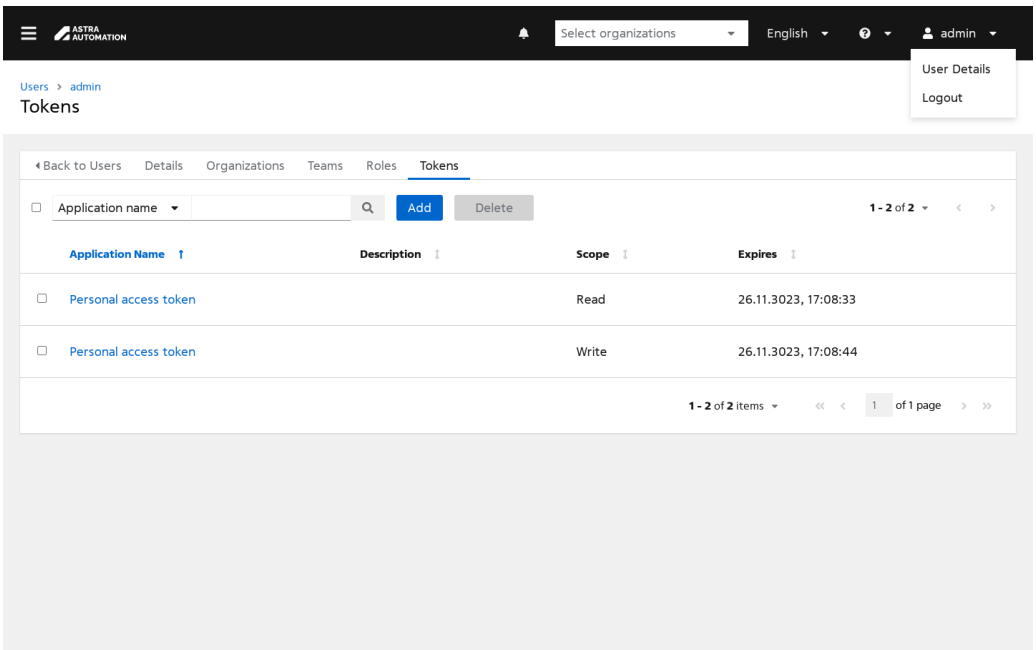


Таблица токенов состоит из следующих колонок:

- **Название приложения** (Application Name) – название приложения, связанного с токеном. Если с токеном не связано ни одно приложение, выводится название **Личный токен доступа** (Personal access token).
- **Описание** (Description) – описание токена.
- **Область** (Scope) – режим доступа к данным контроллера, предоставляемый токеном.
- **Истекает** (Expires) – дата и время истечения срока действия токена.

  Для изменения срока действия токенов измените значение настройки **Истечение срока действия токена доступа** (Access Token Expiration) в *общих параметрах аутентификации*.

Создание токена

Для создания токена выполните следующие действия:

1. Во вкладке **Токены** (Tokens) нажмите кнопку *Добавить* (Add).
2. Заполните форму **Создать пользовательский токен** (Create user token):
   - **Приложение** (Application) – выберите приложение, для которого создается токен.

     Примечание: Приложение должно быть предварительно добавлено в список приложений.

   - **Описание** (Description) – добавьте краткое описание токена, например, название приложения, для которого он создается.
   - **Область** (Scope) – выберите режим доступа к данным контроллера, предоставляемый токеном:
     - **Чтение** (Read);
     - **Запись** (Write).
3. Нажмите кнопку *Сохранить* (Save).
4. В открывшемся диалоговом окне скопируйте значение из поля **Токен** (Token) в буфер обмена и сохраните в надежное место

Важно: После закрытия диалогового окна получить доступ к значению токена невозможно.

5. Закройте окно **Информация о токене** (Token information).

Удаление токена

Предупреждение: При удалении токена все приложения, которые его используют, теряют доступ к контроллеру.

Для удаления пользовательского токена выполните следующие действия:

1. Во вкладке **Токены** (Tokens) нажмите на ссылку с названием удаляемого токена.
2. Нажмите кнопку *Удалить* (Delete).
3. Подтвердите удаление.

Команды

Окно **Команды** (Teams) используется для управления *командами*. Команда объединяет в себе множество пользователей и используется для разграничения доступа к существующим ресурсам. Для создания команд и управления ими необходимо перейти в соответствующий раздел с помощью панели навигации: *Доступ* ▸ *Команды* (*Access* ▸ *Teams*).

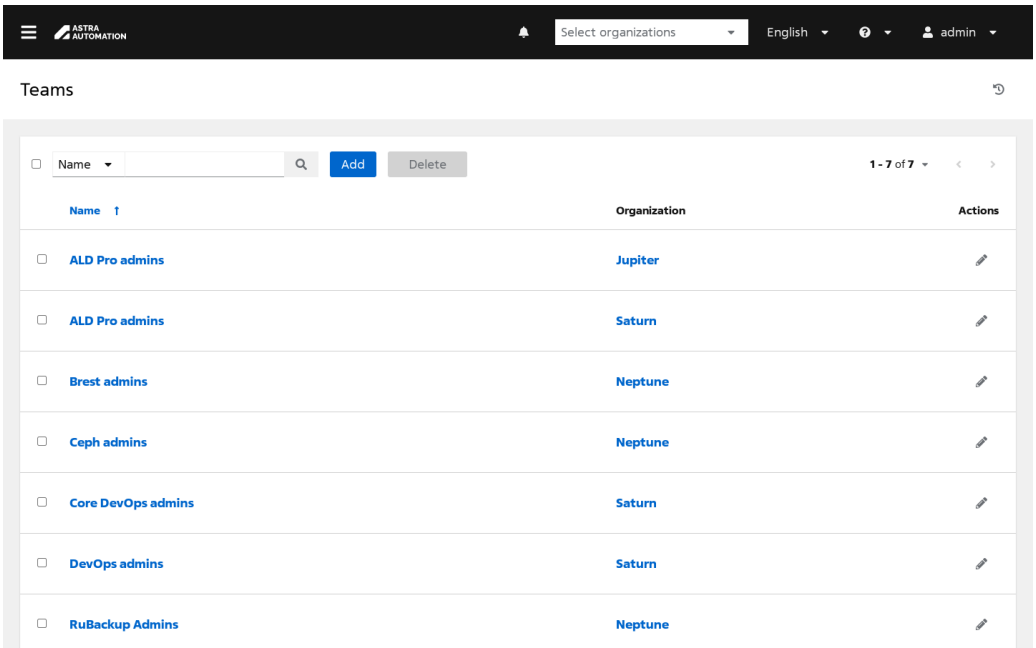


Таблица команд состоит из следующих колонок:

- флаги для выбора нескольких записей;
- **Название** (Name);
- **Организация** (Organization) – название организации, которой принадлежит команда;
- **Действия** (Actions) – кнопка перехода в окно редактирования свойств команды.

Просмотр

Для получения подробных сведений о команде нажмите на ссылку с ее названием в таблице команд.

Окно сведений о команде состоит из следующих вкладок:

- **Подробности** (Details) – общие сведения о команде;
- **Доступ** (Access) – список пользователей и назначенных им ролей.
- **Роли** (Roles) – список командных ролей. Указанные здесь роли автоматически назначаются всем участникам команды.

Создание

Для создания команды выполните следующие действия:

1. В окне **Команды** (Teams) нажмите кнопку *Добавить* (Add).
2. Заполните форму **Создать новую команду** (Create New Team):
   - **Название** (Name) – укажите название команды.

     Особенности заполнения поля:

     - Название команды должно быть уникальным на уровне организации.
     - Название команды не может состоять из одних пробельных символов (пробелы, табуляции и так далее).
     - Допускается использование символов кириллицы и специальных символов.
   - **Описание** (Description) – укажите дополнительные сведения о команде, например, контакты ответственных лиц или назначение команды.
   - **Организация** (Organization) – выберите организацию, которой принадлежит команда.
3. Нажмите кнопку *Сохранить* (Save).

Удаление

> **Предупреждение:** При удалении команды ее участники теряют предоставленные командой привилегии.

Чтобы удалить команду, выполните следующие действия:

1. В таблице команд нажмите на ссылку с названием удаляемой команды.
2. Во вкладке **Подробности** (Details) нажмите кнопку *Удалить* (Delete).
3. Подтвердите удаление

Настройка доступа

Во вкладке **Доступ** (Access) выводится список *пользователей* и назначенных им *ролей*. Нажатие на ссылку с именем пользователя приводит к переходу в окно просмотра сведений о пользователе.

Назначение ролей отдельным пользователям

Чтобы назначить отдельным пользователям роль на уровне команды, выполните следующие действия:

1. Во вкладке **Доступ** (Access) нажмите кнопку *Добавить* (Add).
2. Выберите тип объектов *Пользователи* (Users) и нажмите кнопку *Продолжить* (Next).
3. Выберите пользователей, которым хотите назначить роли, и нажмите кнопку *Продолжить* (Next).
4. Выберите роли, которые хотите назначить выбранным на предыдущем шаге пользователям.
5. Нажмите кнопку *Сохранить* (Save).

Назначение командных ролей

Чтобы предоставить участникам команды доступ к существующим ресурсам, выполните следующие действия:

1. Во вкладке **Роли** (Roles) нажмите кнопку *Добавить* (Add).
2. Выберите тип объектов и нажмите кнопку *Продолжить* (Next).
3. Выберите экземпляры ресурсов и нажмите кнопку *Продолжить* (Next).
4. Выберите роли и нажмите кнопку *Сохранить* (Save).

Отзыв ролей

Чтобы отозвать пользовательскую или командую роль, выполните следующие действия:

1. Во вкладке **Доступ** (Access) или **Роли** (Roles) соответственно нажмите кнопку × рядом с названием отзываемой роли.
2. Подтвердите отзыв.

Примечание: Роли «Системный администратор» (System Administrator) и «Системный аудитор» (System Auditor) нельзя отозвать через настройку доступа на уровне команды, однако, можно *изменить тип пользователя*.

Администрирование

Раздел *Администрирование* (Administration) содержит подразделы, используемые для управления нестандартными типами полномочий, уведомлениями, служебными заданиями, отдельными узлами контроллера и их группами, приложениями, средами исполнения и топологией.

Типы полномочий

Окно **Типы полномочий** (Credential Types) используется для управления пользовательскими (нестандартными) типами *полномочий*. Для этого необходимо перейти в соответствующий раздел с помощью панели навигации: *Администрирование* ▸ *Типы полномочий* (*Administration* ▸ *Credential Types*).

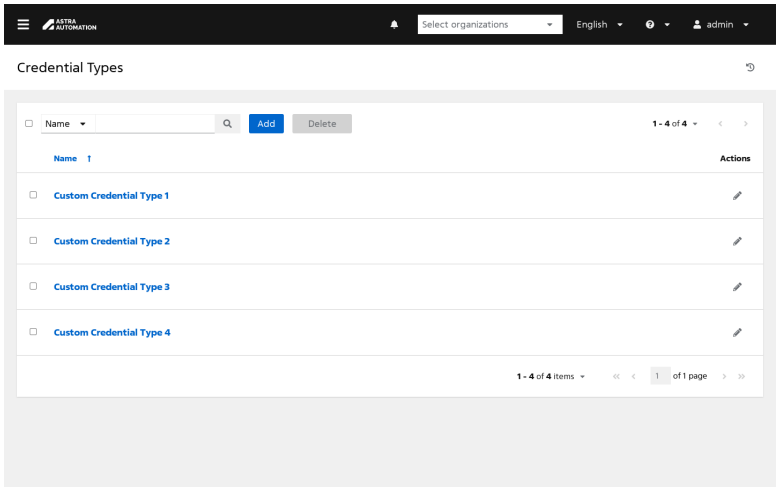


Таблица типов полномочий состоит из следующих колонок:

- флаги для выбора нескольких записей;
- **Название** (Name);
- **Действия** (Actions) – кнопка перехода в окно изменения свойств типа полномочий.

Примечание: Для управления пользовательскими типами полномочий необходимы привилегии администратора контроллера.

Создание

Для создания нового типа полномочий выполните следующие действия:

1. В окне **Типы полномочий** (Credential Types) нажмите кнопку *Добавить* (Add).
2. Заполните форму **Создать новый тип учетных данных** (Create new credential Type):
   - **Название** (Name) – укажите название типа полномочий.

     Название типа полномочий должно быть уникальным на уровне контроллера.
   - **Описание** (Description) – укажите описание типа полномочий, например, их назначение или особенности использования.
   - **Настройка входных данных** (Input configuration) – укажите параметры входных данных, которые необходимо указать при использовании этого типа полномочий.
   - **Конфигурация инжектора** (Injector configuration) – укажите данные, которые передаются во внешнюю систему аутентификации при использовании полномочий этого типа.

   Примечание: Особенности и примеры заполнения полей **Настройка входных данных** (Input configuration) и **Конфигурация инжектора** (Injector configuration) см. в секции *Собственные типы полномочий*.

   ---

3. Нажмите кнопку *Сохранить* (Save).

Удаление

Чтобы удалить тип полномочий, выполните следующие действия:

1. *Удалите* все полномочия соответствующего типа.
2. В таблице типов полномочий нажмите на ссылку с названием удаляемого типа полномочий.
3. Нажмите кнопку *Удалить* (Delete).
4. Подтвердите удаление.

Уведомления

Окно **Уведомления** (Notifications) используется для управления уведомлениями. Для создания уведомлений и управления ими необходимо перейти в соответствующий раздел с помощью панели навигации: *Администрирование* ▸ *Уведомления* (*Administration* ▸ *Notifications*).

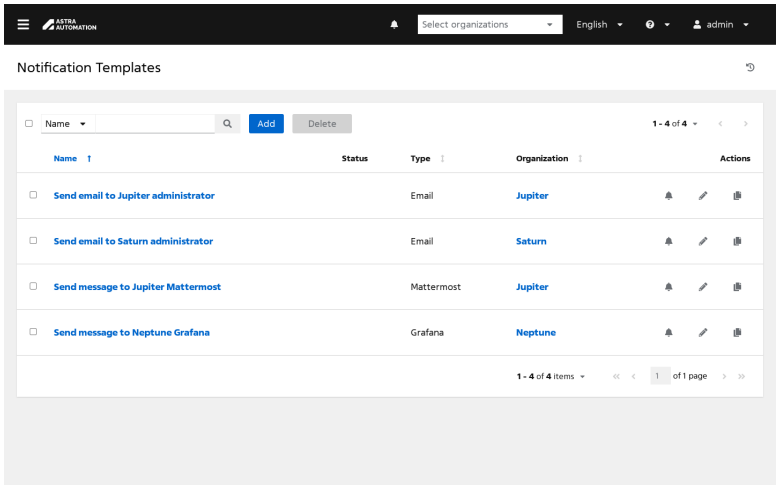


Таблица уведомлений состоит из следующих колонок:

- флаги для выбора нескольких записей;
- **Название** (Name);
- **Статус** (Status);
- **Тип** (Type) – способ доставки уведомлений;
- **Действия** (Actions) – кнопки для быстрого вызова часто выполняемых действий:
  - проверка корректности настроек уведомления;
  - переход в окно редактирования свойств уведомления;
  - копирование уведомления.

Просмотр

Для получения подробных сведений об уведомлении нажмите на ссылку с его названием в таблице уведомлений.

Внешний вид окна со сведениями об уведомлении зависит от типа уведомления.

Создание

Для создания уведомления выполните следующие действия:

1. В окне **Уведомления** (Notifications) нажмите кнопку *Добавить* (Add).
2. Заполните форму **Создать новый шаблон уведомления** (Create New Notification Template):
   - **Название** (Name) – укажите название уведомления.
   - **Описание** (Description) – укажите описание уведомления.
   - **Организация** (Organization) – выберите организацию, для которой настраивается уведомление.
   - **Тип** (Type) – выберите тип создаваемого уведомления.

     После выбора типа в форме отображаются соответствующие поля ввода.

     #### Электронная почта (Email)

     – **Имя пользователя** (Username) – укажите название учетной записи пользователя для доступа к почтовому серверу.

     – **Пароль** (Password) – укажите пароль пользователя для доступа к почтовому серверу.

     – **Узел** (Host) – укажите IP-адрес или *FQDN* почтового сервера.

     – **Список получателей** (Recipient list) – укажите адреса электронной почты получателей, по одному на строку.

     – **Адрес электронной почты отправителя** (Sender e-mail) – укажите адрес электронной почты, с которого следует отправлять письма.

     – **Порт** (Port) – укажите номер порта для подключения к почтовому серверу.

     – **Таймаут** (Timeout) – укажите период ожидания почтового сервера (в секундах). Если в течение указанного времени контроллер не сможет связаться с почтовым сервером, отправка уведомления будет отменена.

       Минимальное значение – 1.

       Максимальное значение – 120.

     – **Использовать SSL** (Use SSL) – защита подключения к почтовому серверу с помощью SSL.

     – **Использовать TLS** (Use TLS) – защита подключения к почтовому серверу с помощью TLS.

Grafana

- **URL Grafana** (Grafana URL) – укажите URL сервера Grafana без пути `/api/annotations/`. При создании уведомлений он добавляется к указанному URL автоматически.
- **Ключ API Grafana** (Grafana API key) – укажите ключ доступа к Grafana API.
- **Идентификатор информационной панели (необязательно)** (ID of the dashboard (optional)) – укажите идентификатор информационной панели Grafana для вывода уведомлений.
- **Идентификатор панели (необязательно)** (ID of the panel (optional)) – укажите идентификатор панели Grafana для вывода уведомлений.
- **Теги для аннотации (необязательно)** (Tags for the annotation (optional)) – укажите тэги без команд, по одному на строку.
- **Отключить проверку SSL** (Disable SSL verification) – укажите, должен ли контроллер проверять сертификат SSL сервера Grafana.

IRC

- **Пароль на сервере IRC** (IRC server password) – укажите пароль для подключения к серверу IRC.
- **Порт сервера IRC** (IRC server port) – укажите номер порта для подключения к серверу IRC.

  Значение по умолчанию – `80`.
- **Адрес сервера IRC** (IRC server address) – укажите IP-адрес или *FQDN* сервера IRC.
- **IRC Nick** (IRC nick) – укажите название учетной записи пользователя для доступа к IRC.
- **Целевые каналы или пользователи** (Destination channels or users) – укажите названия каналов и имена пользователей, которым должны быть отправлены сообщения с уведомлениями, по одному на строку.

  Важно: При вводе названий каналов и имен пользователей не указывайте символы `@` и `#` в начале строки.

- **Отключить проверку SSL** (Disable SSL verification) – укажите, должен ли контроллер проверять сертификат SSL сервера IRC.

Mattermost

- **Целевой URL** (Target URL) – укажите URL сервера Mattermost.
- **Имя пользователя** (Username) – укажите название учетной записи пользователя для доступа к серверу Mattermost.
- **Канал** (Channel) – укажите название канала Mattermost для отправки уведомлений.
- **URL значка** (Icon URL) – укажите ссылку на иконку, используемую как аватар пользователя Mattermost, от имени которого отправляются сообщения.
- **Отключить проверку SSL** (Disable SSL verification) – укажите, должен ли контроллер проверять сертификат SSL сервера Mattermost.

Pagerduty

- **API Token** (Токен API) – укажите токен для подключения к PagerDuty.
- **Поддомен Pagerduty** (Pagerduty subdomain) – укажите поддомен PagerDuty.
- **Ключ к сервису API/интеграции** (API service/intergration key) – укажите ключ к сервису API или ключ интеграции.
- **Идентификатор клиента** (Client identifier) – укажите идентификатор клиента, от имени которого контроллер будет отправлять уведомления в PagerDuty.

Rocket.Chat

- **Целевой URL** (Target URL) – укажите *FQDN* для подключения к Rocket.Chat.
- **Имя пользователя** (Username) – укажите название учетной записи пользователя для доступа к Rocket.Chat.
- **URL значка** (Icon URL) – укажите ссылку на иконку, используемую как аватар пользователя, от имени которого отправляются сообщения.
- **Отключить проверку SSL** (Disable SSL verification) – укажите, должен ли контроллер проверять сертификат SSL сервера Rocket.Chat.

Slack

- **Целевые каналы** (Destination channels) – укажите названия каналов Slack для отправки в них уведомлений, по одному каналу на строку.

  Важно: Название канала должно начинаться с символа #.

  Чтобы ответить на определенное сообщение или запустить тред, укажите идентификатор родительского сообщения – строку из 16 цифр. После 10-й цифры добавьте символ `.`, например, `#dev-ops, 0000000555.111111`.

  Подробности см. в документации Slack[144].
- **Токен** (Token) – укажите токен, используемый для подключения к Slack.
- **Цвет уведомления** (Notification color) – укажите цвет уведомления в шестнадцатеричном формате, например, `#12F` или `FF0000`.

Twilio

- **Токен учетной записи** (Account token) – укажите токен для доступа к серверу Twilio.
- **Исходный номер телефона** (Source phone number) – укажите номер телефона, с которого отправляются сообщения.

  Важно: Допускается использование цифр и символа + в начале строки.

- **Номер(а) для получения SMS** (Destination SMS number(s)) – укажите номера телефонов для отправки на них SMS с уведомлениями, по одному на строку.

  Важно: Допускается использование цифр и символа + в начале строки.

[144] https://api.slack.com/messaging/retrieving#individual_messages

Подробности см. в документации Twilio[145].

- **Идентификатор сеанса связи** (Account SID) – укажите идентификатор сеанса связи.

Webhook

- **Имя пользователя** (Username) – укажите название учетной записи пользователя для доступа к сервису WebHook.
- **Пароль basic auth** (Basic auth password) – укажите пароль пользователя для доступа к сервису WebHook.
- **Целевой URL** (Target URL) – укажите URL WebHook.
- **Отключить проверку SSL** (Disalbe SSL verification) – укажите, должен ли контроллер проверять сертификат SSL сервера WebHook.
- **Заголовки HTTP** (HTTP Headers) – укажите в формате JSON дополнительные заголовки HTTP, которые должны быть добавлены в запрос.
- **Метод HTTP** (HTTP Method) – выберите метод HTTP, используемый для отправки запросов к WebHook. Поддерживаются методы P0ST и PUT.

• **Настройка сообщений...** (Customize messages...) – переведите переключатель во включенное состояние, чтобы настроить шаблоны, используемые для формирования сообщений о событиях.

Если переключатель включен, в форме отображаются дополнительные поля:

Поле	Событие
Начальное сообщение (Start message)	Запуск задания
Сообщение об успехе (Success message)	Успешное выполнение задания
Сообщение об ошибке (Error message)	Ошибка при выполнении задания
Сообщение о согласовании потока заданий (Workflow approved message)	Согласование этапа потока заданий
Сообщение об отказе в выполнении потока заданий (Workflow denied message)	Отклонение этапа потока заданий
Сообщение о состоянии ожидания потока заданий (Workflow pending message)	Ожидание согласования этапа потока заданий
Сообщение об истечении времени ожидания потока заданий (Workflow timed out message)	Истечение времени ожидания согласования этапа потока заданий

Примечание: Подробности о синтаксисе шаблонов и доступных переменных см. в документе *Отправка уведомлений в Mattermost*.

3. Нажмите кнопку *Сохранить* (Save).

[145] https://www.twilio.com/docs/messaging

Удаление

Чтобы удалить уведомление, выполните следующие действия:

1. В таблице уведомлений нажмите на ссылку с названием удаляемого уведомления.
2. Во вкладке **Подробности** (Details) нажмите кнопку *Удалить* (Delete).
3. Подтвердите удаление.

Служебные задания

Окно **Служебные задания** (Management Jobs) используется для управления запуском *служебных заданий*. Чтобы запустить служебное задание немедленно, настроить его запуск по расписанию или получать уведомления, необходимо перейти в соответствующий раздел с помощью панели навигации: *Администрирование* ▸ *Служебные задания* (*Administration* ▸ *Management Jobs*).

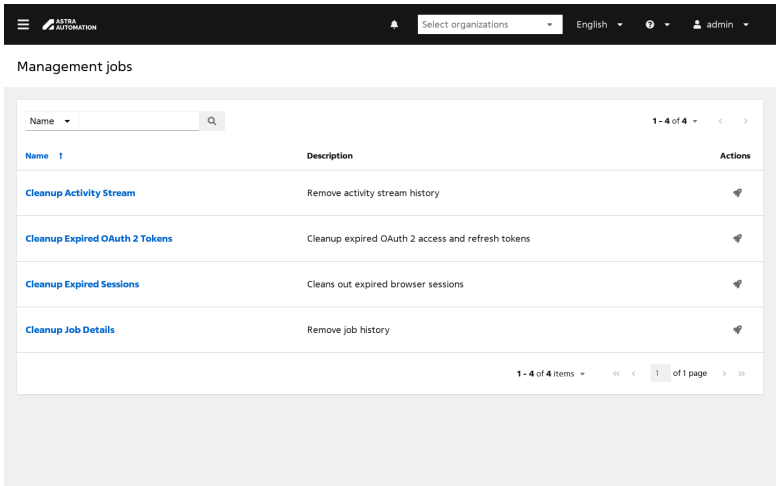


Таблица служебных заданий состоит из следующих колонок:

- **Название** (Name);
- **Описание** (Description);
- **Действия** (Actions) – кнопка запуска задания.

Запуск

Для запуска служебного задания нажмите кнопку запуска в колонке **Действия** (Actions).

Для заданий типа «Cleanup Activity Stream» и «Cleanup Job Details» при этом открывается диалоговое окно **Запуск служебного задания** (Launch management job).

1. Укажите период хранения записей (количество дней). Все записи старше указанного периода будут удалены.

   Значение по умолчанию – `30`.
2. Для задания «Cleanup Job Details» выберите типы ресурсов для удаления. По умолчанию выбраны все типы.
3. Нажмите кнопку *Запустить* (Run).

Настройка расписания

При развертывании контроллера расписание запуска служебных заданий настраивается автоматически.

> **Предупреждение:** Некорректные настройки расписания запуска служебных заданий могут привести к отказу контроллера.

Чтобы изменить расписание запуска служебного задания, выполните следующие действия:

1. В таблице служебных заданий нажмите на ссылку с названием нужного задания.
2. Во вкладке **Расписание** (Schedules) нажмите на ссылку с названием существующего расписания.
3. Во вкладке **Подробности** (Details) нажмите кнопку *Редактировать* (Edit).
4. Измените значения в форме **Редактировать детали** (Edit Details):
   - **Название** (Name) – укажите название расписания.
   - **Описание** (Description) – укажите описание расписания.
   - **Дата и время начала** (Start date/time) – укажите дату и время первого запуска задания.
   - **Локальный часовой пояс** (Local time zone) – выберите часовой пояс, по которому рассчитывается время.

     Значение по умолчанию – UTC.
   - **Частота повторения** (Repeat frequency) – выберите частоту повторения запусков задания.

     Если выбрано значение **Нет (запускается один раз)** (None (run once)), задание будет выполнено один раз.

     Включение флагов **Мин** (Min), **Час** (Hour), **День** (Day), **Неделя** (Week), **Месяц** (Month) и **Год** (Year) приводит к добавлению в форму наборов полей, позволяющих выполнить настройку соответствующих временных интервалов. Для всех типов интервалов доступны следующие настройки:

     - **Выполнять каждые** (Run every) – количество выбранных временных интервалов между запусками задания.
     - **Завершено** (End) – условие прекращения запуска задания по расписанию. Доступны следующие настройки:
       * **Никогда** (Never) – не прекращать.
       * **После указанного количества вхождений** (After number of occurences) – после указанного количества запусков.
       * **На дату** (On date) – в указанные день и время.

     Для недельных интервалов доступна дополнительная настройка **В дни** (Days). Она позволяет выбрать дни недели, по которым следует запускать задание.

     Для месячных интервалов доступна дополнительная настройка **Выполнить в** (Run on). Она позволяет выбрать определенную дату или относительную позицию дня в месяце. Например, при выборе даты и указании числа «20» задание будет запускаться двадцатого числа каждого месяца. При выборе относительной позиции нет привязки к конкретной дате. Вместо этого настраивается запуск в дни, обозначаемые как, например, «второй понедельник» или «последняя пятница». Также поддерживаются значения, позволяющие выбрать рабочие или нерабочие дни.

Для годовых интервалов доступна дополнительная настройка **Выполнить в** (Run on). Она позволяет выбрать месяц и число либо относительную позицию дня в году. Например, при выборе месяца «Январь» и числа «10» задание будет запускаться каждый год десятого января. При выборе относительной позиции дня нет привязки к конкретной дате. Вместо этого настраивается запуск в дни, обозначаемые как, например, «первый четверг мая» или «последний день февраля». Также поддерживаются значения, позволяющие выбрать рабочие или нерабочие дни.

  - **Исключения** (Exceptions) – настройки, предотвращающие запуск задания в указанные минуты, часы, дни, месяцы и годы. Эти настройки работают по тем же принципам, что и настройки частоты повторений.

- **Типы ресурсов** (Resources types).

  Примечание: Эта настройка доступна только для задания «Cleanup Job Details».

  Выберите типы ресурсов, данные о которых необходимо удалить из истории выполнения заданий.

- **Дни хранения данных** (Days of Data to Keep)

  Примечание: Эта настройка доступна только для заданий «Cleanup Activity Stream» и «Cleanup Job Details».

  Период (количество дней), в течение которого должны храниться данные о действиях пользователей и выполнении заданий соответственно. Все записи, созданные ранее этого периода, будут удалены.

  Значения по умолчанию:

  - Cleanup Activity Schedule – 355;
  - Cleanup Job Schedule – 120.

5. Нажмите кнопку *Сохранить* (Save).

Группы узлов контроллера

Окно **Группы узлов контроллера** (Instance Groups) используется для управления группами узлов контроллера и группами контейнеров. Для этого необходимо перейти в соответствующий раздел с помощью панели навигации: *Администрирование* ▸ *Группы узлов контроллера* (*Administration* ▸ *Instance Groups*).

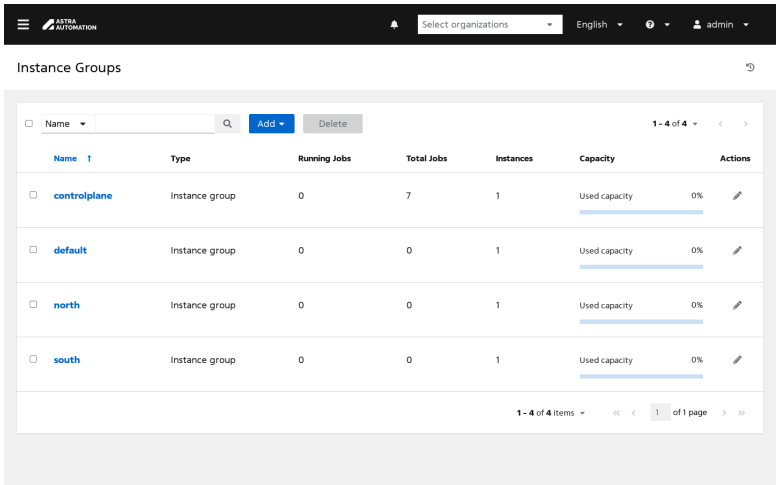


Таблица групп состоит из следующих колонок:

- Флаги для выбора нескольких записей.

- **Название** (Name) – ссылка для перехода в окно просмотра подробных сведений о группе узлов или группе контейнеров.
- **Тип** (Type) – тип записи (группа узлов контроллера или группа контейнеров).
- **Запущенные задания** (Running Jobs) – количество заданий, запущенных на узлах группы или группе контейнеров в настоящее время.
- **Всего задач** (Total Jobs) – общее количество заданий, запущенных на узлах группы или группе контейнеров когда-либо.
- **Узлы контроллера** (Instances) – количество узлов или контейнеров в группе.

  Примечание: Один и тот же узел может входить в несколько групп узлов.

- **Ресурсы** (Capacity) – степень утилизации ресурсов группы узлов. Чем выше значение в этой колонке, тем выше нагрузка на узлы группы.
- **Действия** (Actions) – кнопка перехода в окно редактирования группы узлов или контейнеров.

Просмотр группы узлов

Для получения подробных сведений о группе узлов контроллера нажмите на ссылку с ее названием в таблице групп. Окно сведений о группе узлов контроллера состоит из следующих вкладок:

- **Подробности** (Details) – общие сведения о группе.
- **Узлы контроллера** (Instances) – таблица узлов группы и кнопки управления ими.

  Таблица состоит из следующих колонок:

  - Переключатели вывода подробной информации об узлах.
  - Флаги для выбора нескольких записей.
  - **Название** (Name) – ссылка для перехода в окно просмотра подробных сведений об узле.
  - **Статус** (Status) – текущий статус узла.

    Для проверки статуса узла выполните *проверку его работоспособности*.
  - **Тип узла** (Node Type).

    Подробности о типах узлов см. в документе *Особенности архитектуры*.
  - **Настройка ресурсов** (Capacity Adjustment) – ползунок изменения максимального количества ветвлений для всех процессов, запущенных на узле.

    Максимальное значение этого параметра зависит от количества ядер CPU и объема оперативной памяти на узле.
  - **Потребляемые ресурсы** (User capacity) – индикатор утилизации ресурсов узла. Чем выше значение в этом поле, тем больше нагрузка на узел.
  - **Действия** (Actions) – переключатели статусов узлов.
- **Задания** (Jobs) – таблица заданий, запущенных на узлах группы.

  Таблица состоит из следующих колонок:

  - переключатели вывода подробной информации о задании;
  - флаги для выбора нескольких записей;
  - **Название** (Name) – ссылка для перехода в окно просмотра вывода задания;
  - **Статус** (Status) – текущий статус выполнения задания;
  - **Тип** (Type) – тип задания;

- **Время начала** (Start Time) – время запуска задания;
- **Время завершения** (Finish Time) – время завершения, отмены или прерывания задания;
- **Действия** (Actions) – кнопка перезапуска задания.

Создание группы узлов

Для создания группы узлов выполните следующие действия:

1. В окне **Группы узлов контроллера** (Instance Groups) нажмите кнопку *Добавить* (Add).
2. В открывшемся меню выберите **Добавить группу узлов контроллера** (Add instance group).
3. Заполните форму **Создать новую группу узлов управления** (Create new instance group):
   - **Название** (Name) – укажите название группы узлов.

     Требования к названию:

     - название должно быть уникальным;
     - запрещено создание группы с названием `default`.
   - **Политика минимального количества** (Policy instance minimum) – укажите минимальное количество узлов, которые должны быть включены в состав группы при подключении к кластеру новых узлов.

     Значение по умолчанию – `0`.
   - **Политика процентного количества** (Policy instance percentage) – укажите минимальный процент узлов от общего количества, которые должны быть включены в состав группы при подключении к кластеру новых узлов.

     Значение по умолчанию – `0`.
   - **Максимальное количество одновременных заданий** (Max concurrent jobs) – укажите максимальное количество заданий, которые могут быть одновременно запущены на узлах группы.

     При значении `0` количество заданий не ограничено.

     Значение по умолчанию – `0`.
   - **Максимальное количество ветвлений процесса** (Max forks) – максимальное количество ответвленных процессов всех заданий, запущенных на узлах группы.

     При значении `0` количество ветвлений не ограничено.

     Значение по умолчанию – `0`.
4. Нажмите кнопку *Сохранить* (Save).

Включение узлов в группу

Чтобы включить узлы в группу, выполните следующие действия:

1. В таблице групп узлов и контейнеров нажмите на ссылку с названием нужной группы узлов.
2. Выберите вкладку **Узлы контроллера** (Instances).
3. Нажмите кнопку *Связать* (Associate).
4. В диалоговом окне **Выберите узлы контроллера** (Select Instances) включите флаги в строках с нужными узлами и нажмите кнопку *Сохранить* (Save).

Отвязка узлов от группы

Чтобы исключить узлы из группы, выполните следующие действия:

1. В таблице групп узлов и контейнеров нажмите на ссылку с названием нужной группы узлов.
2. Выберите вкладку **Узлы контроллера** (Instances).
3. Включите флаги в строках с исключаемыми узлами.
4. Нажмите кнопку *Отвязать* (Disassociate).
5. Подтвердите исключение узлов из группы.

Проверка работоспособности узлов

Чтобы проверить работоспособность узлов группы, выполните следующие действия:

1. В таблице групп узлов и контейнеров нажмите на ссылку с названием нужной группы узлов.
2. Выберите вкладку **Узлы контроллера** (Instances).
3. Включите флаги в строках с узлами, работоспособность которых хотите проверить.
4. Нажмите кнопку *Выполнить проверку работоспособности* (Run health check).
5. Перезагрузите страницу.

Создание группы контейнеров

Для создания группы контейнеров выполните следующие действия:

1. В окне **Группы узлов контроллера** (Instance Groups) нажмите кнопку *Добавить* (Add).
2. В открывшемся меню выберите **Добавить группу контейнеров** (Add container group).
3. Заполните форму **Создать новую группу узлов контейнеров** (Create new instance container):
   - **Название** (Name) – укажите название группы контейнеров.
   - **Полномочия** (Credential) – если для доступа к группе контейнеров требуется аутентификация, выберите соответствующее *полномочие* типа «Токен доступа OpenShift или Kubernetes API» (OpenShift or Kubernetes API Bearer Token).
   - **Максимальное количество одновременных заданий** (Max concurrent jobs) – максимальное количество заданий, которые могут быть одновременно запущены на группе контейнеров.

     При значении `0` количество заданий не ограничено.

     Значение по умолчанию – `0`.
   - **Максимальное количество ветвлений процесса** (Max forks) – максимальное количество ответвленных процессов всех заданий, запущенных на группе контейнеров.

     При значении `0` количество ветвлений не ограничено.

     Значение по умолчанию – `0`.
   - **Настройка спецификации pod** (Customize pod specification). Если эта настройка включена, становятся доступными для изменения настройки спецификации подов. Укажите их в поле **Измененная спецификация pod** (Custom pod spec).
4. Нажмите кнопку *Сохранить* (Save).

Удаление группы узлов или группы контейнеров

Чтобы удалить группу узлов или группу контейнеров, выполните следующие действия:

1. В таблице групп узлов и контейнеров нажмите на ссылку с названием удаляемой группы.
2. Во вкладке **Подробности** (Details) нажмите кнопку *Удалить* (Delete).
3. Подтвердите удаление группы.

Узлы контроллера

Окно **Узлы контроллера** (Instances) используется для просмотра статуса узлов контроллера и запуска диагностических команд. Для этого необходимо перейти в соответствующий раздел с помощью панели навигации: *Администрирование* ▸ *Узлы контроллера* (*Administration* ▸ *Instances*).

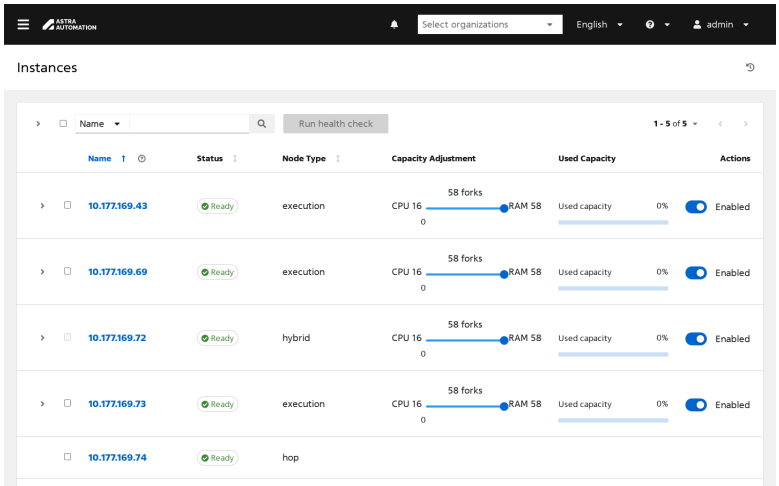


Таблица узлов состоит из следующих колонок:

- Переключатели вывода подробной информации об узлах.
- Флаги для выбора нескольких записей.
- **Название** (Name) – ссылка для перехода в окно просмотра подробных сведений об узле.
- **Статус** (Status) – текущий статус узла.

  Для проверки и обновления статуса узла выполните *проверку его работоспособности*.
- **Тип узла** (Node Type) – тип узла.

  Подробности о типах узлов см. в документе *Особенности архитектуры*.
- **Настройка ресурсов** (Capacity Adjustment) – ползунок изменения максимального количества ветвлений для всех процессов, запущенных на узле.

  Максимальное значение этого параметра зависит от количества ядер CPU и объема оперативной памяти на узле.
- **Потребляемые ресурсы** (Used Capacity) – индикатор утилизации ресурсов узла. Чем выше значение в этом поле, тем больше нагрузка на узел.
- **Действия** (Actions) – переключатели статусов узлов.

Проверка работоспособности узлов

Чтобы проверить работоспособность узлов, выполните следующие действия:

1. В таблице узлов включите флаги в строках с теми узлами, работоспособность которых хотите проверить.
2. Нажмите кнопку *Выполнить проверку работоспособности* (Run health check).
3. Обновите страницу браузера.

Приложения

Окно **Приложения** (Applications) используется для управления приложениями и их токенами. Для этого необходимо перейти в соответствующий раздел с помощью панели навигации: *Администрирование* ▸ *Приложения* (*Administration* ▸ *Applications*).

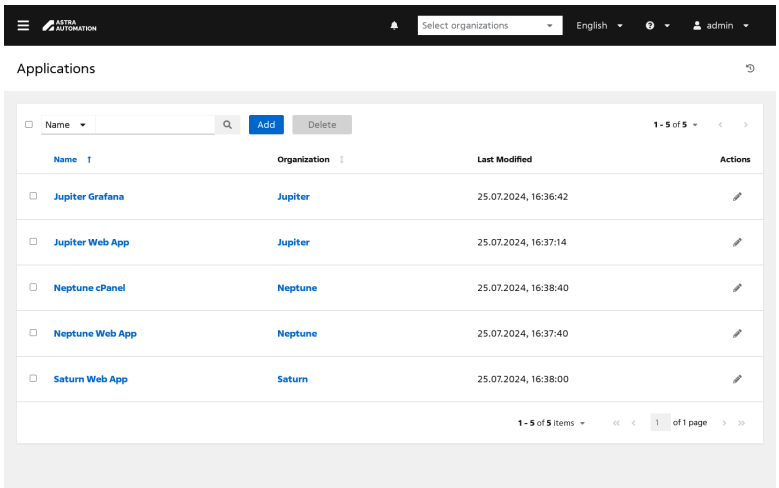


Таблица приложений состоит из следующих колонок:

- флаги для выбора нескольких записей;
- **Название** (Name) – ссылка для перехода в окно просмотра сведений о приложении;
- **Организация** (Organization) – ссылка для перехода в окно просмотра сведений об организации-владельце;
- **Последнее изменение** (Last Modified) – дата и время последнего изменения записи;
- **Действия** (Actions) – кнопка перехода в окно редактирования свойств приложения.

Просмотр сведений о приложении

Для получения подробных сведений о приложении нажмите на ссылку с его названием в таблице приложений.

Окно сведений о приложении состоит из следующих вкладок:

- **Подробности** (Details) – общие сведения о приложении.
- **Токены** (Tokens) – таблица связанных с приложением токенов.

  Таблица токенов состоит из следующих колонок:

  - флаги для выбора нескольких записей;
  - **Название** (Name) – ссылка для перехода в окно просмотра сведений о пользователе-владельце токена;
  - **Область** (Scope) – режим доступа к данным контроллера, предоставляемый токеном;

- **Истекает** (Expires) – дата и время истечения срока действия токена.

  Для изменения срока действия токенов измените значение настройки **Истечение срока действия токена доступа** (Access Token Expiration) в *общих параметрах аутентификации*.

Создание приложения

Для создания приложения выполните следующие действия:

1. В окне **Приложения** (Applications) нажмите кнопку *Добавить* (Add).
2. Заполните форму **Создать новое приложение** (Create New Application):
   - **Название** (Name) – должно быть уникальным в рамках организации.
   - **Описание** (Description) – укажите описание приложения.
   - **Организация** (Organization) – выберите организацию-владельца.
   - **Тип предоставляемых полномочий** (Authorization grant type) – выберите тип доступа к контроллеру.

     Поддерживаются следующие значения:

     - **Код авторизации** (Authorization code);
     - **Владелец ресурса с доступом по паролю** (Resource owner password-based).
   - **URI перенаправления** (Redirect URIs) – укажите список разрешенных URI. В качестве разделителя используйте символ пробела.

     Примечание: Это поле обязательно для заполнения, если в поле **Тип предоставляемых полномочий** (Authorization grant type) выбрано значение **Код авторизации** (Authorization code).

   - **Тип клиента** (Client type) – укажите уровень защищенности приложения, которому предоставляете доступ.

     Поддерживаются следующие значения:

     - **Конфиденциальный** (Confidential) – приложение имеет механизмы защиты доступа к своим данным;
     - **Публичный** (Public) – данные приложения публично доступны.
3. Нажмите кнопку *Сохранить* (Save).
4. В открывшемся окне **Информация о приложении** (Application information) выводятся следующие данные, необходимые для использования приложения:
   - **Идентификатор клиента** (Client ID);
   - **Секретная фраза клиента** (Client secret) (только если при создании приложения выбран тип клиента «Конфиденциальный»).

   Важно: Сохраните показанные значения в надежном месте – после закрытия диалогового окна получить к ним доступ повторно невозможно.

5. Создайте токены согласно инструкций, приведенных в секции *Создание токена*.

Удаление приложения

> **Опасно:** Вместе с приложением удаляются все связанные с ним токены. Для восстановления доступа приложений к контроллеру все токены нужно будет создать заново.

Чтобы удалить приложение, выполните следующие действия:

1. В таблице приложений нажмите на ссылку с названием удаляемого приложения.
2. Во вкладке **Подробности** (Details) нажмите кнопку *Удалить* (Delete).
3. Подтвердите удаление.

Удаление токенов приложения

> **Предупреждение:** При удалении токена использующие его приложения теряют доступ к контроллеру.

Чтобы удалить токен приложения, выполните следующие действия:

1. В таблице приложений нажмите на ссылку с названием нужного приложения.
2. Во вкладке **Токены** (Tokens) включите флаги в строках с удаляемыми токенами.
3. Нажмите кнопку *Удалить* (Delete).
4. Подтвердите удаление.

Среды исполнения

Окно **Среды исполнения** (Execution Environments) используется для управления *средами исполнения*. Для управления средами исполнения необходимо перейти в соответствующий раздел с помощью панели навигации: *Администрирование* ▸ *Среды исполнения* (*Administration* ▸ *Execution Environments*).

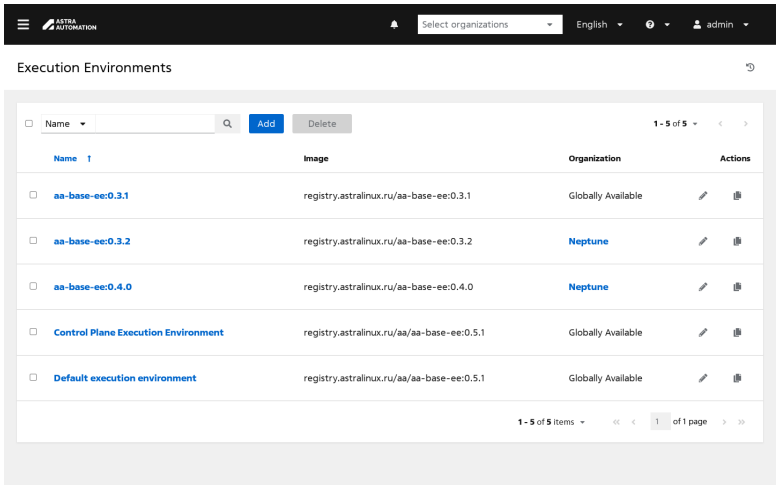


Таблица сред исполнения состоит из следующих колонок:

- флаги для выбора нескольких записей;
- **Название** (Name);
- **Образ** (Image) – ссылка на образ в реестре образов;

- **Организация** (Organization) – ссылка на организацию, которой принадлежит среда исполнения, или текст «Доступно глобально» (Globally Available), если среда исполнения доступна глобально.
- **Действия** (Actions) – кнопки для быстрого вызова часто выполняемых действий:
  - переход в окно редактирования свойств среды исполнения;
  - создание копии среды исполнения.

Просмотр

Для получения подробных сведений о среде исполнения нажмите на ссылку с ее названием в таблице сред исполнения.

Окно сведений о среде исполнения состоит из двух вкладок:

- **Подробности** (Details) – общие сведения о среде исполнения.
- **Шаблоны** (Templates) – *шаблоны заданий*, связанные со средой исполнения.

Добавление

Для добавления среды исполнения выполните следующие действия:

1. В окне **Среды исполнения** (Execution Environments) нажмите кнопку *Добавить* (Add).
2. Заполните форму **Создайте новую среду исполнения** (Create new execution environment):
   - **Название** (Name) – укажите название среды исполнения.

     Примечание: Название должно быть уникальным на уровне контроллера.

   - **Образ** (Image) – укажите ссылку на образ в реестре образов.
   - **Выбрать** (Pull) – укажите *параметры загрузки образа*.
   - **Описание** (Description) – укажите описание среды исполнения, например, версии компонентов.
   - **Организация** (Organization) – выберите *организацию*, которой принадлежит среда исполнения.

     Если это поле не заполнено, среду исполнения могут использовать все организации.

   - **Полномочия на реестр** (Registry credential) – если для доступа к реестру образов требуется аутентификация, выберите в этом поле *полномочия* типа «Реестр контейнеров» (Container Registry).

     Инструкции по созданию полномочий этого типа приведены в секции *Реестр контейнеров*.
3. Нажмите кнопку *Сохранить* (Save).

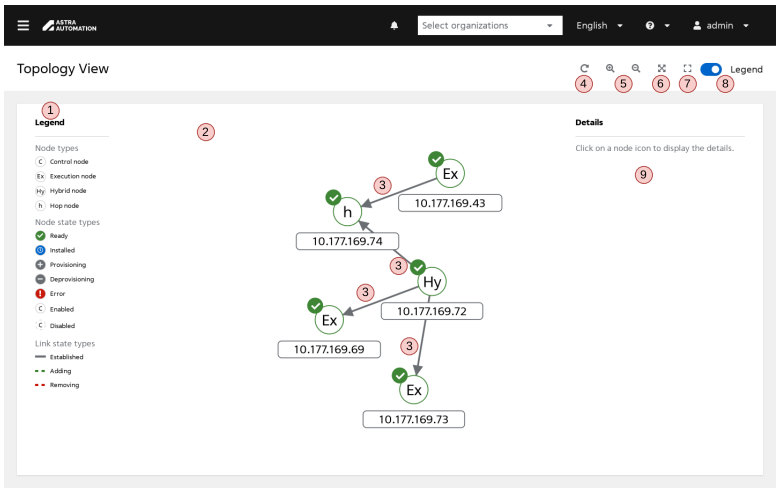
Удаление

Чтобы удалить среду исполнения, выполните следующие действия:

1. В таблице сред исполнения нажмите на ссылку с названием удаляемой среды исполнения.
2. Во вкладке **Подробности** (Details) нажмите кнопку *Удалить* (Delete).
3. Подтвердите удаление среды исполнения.

Топология

Окно **Топология** (Topology View) используется для просмотра топологии узлов кластера. Для этого необходимо перейти в соответствующий раздел с помощью панели навигации: *Администрирование* ▸ *Топология* (*Administration* ▸ *Topology View*).

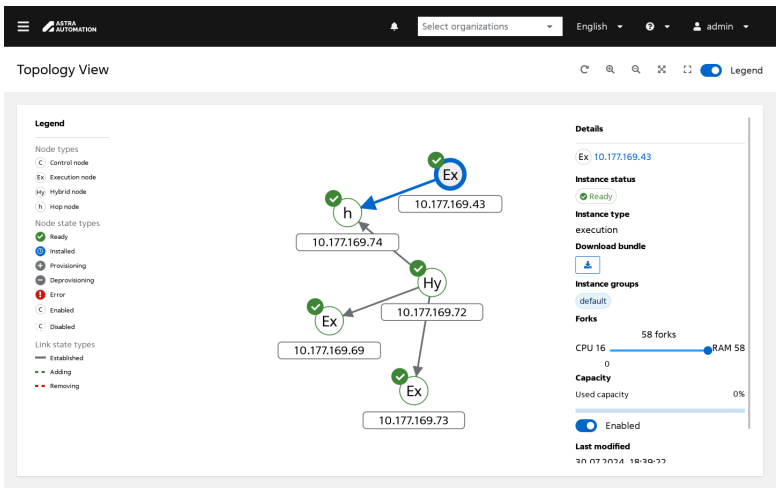


Окно **Топология** (Topology View) состоит из следующих элементов:

1. легенда;
2. схема узлов;
3. связи между узлами;
4. кнопка обновления;
5. кнопки управления масштабом;
6. кнопка подгонки масштаба под размеры экрана;
7. кнопка сброса масштаба к значению по умолчанию;
8. переключатель видимости панели легенды;
9. панель сведений о выбранном узле.

   Содержимое панели **Подробности** (Details) зависит от типа выбранного узла.

Для просмотра сведений об узле выделите его.



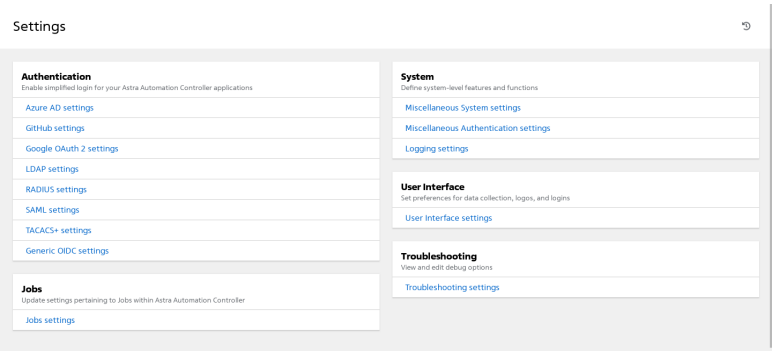
Здесь представлена топология контроллера, состоящего из следующих узлов:

- гибридный узел `10.177.169.72`, выполняющий роль управляющего узла;
- исполняющие узлы `10.177.169.69` и `10.177.169.73`, соединенные с плоскостью управления (в данном случае состоящей из одного узла `10.177.169.72`) напрямую;
- промежуточный узел `10.177.169.74`;
- исполняющий узел `10.177.169.43`, соединенный с плоскостью управления через промежуточный узел `10.177.169.74`.

Подробности о типах узлов см. в документе *Особенности архитектуры*.

Настройки

Окно **Настройки** (Settings) предназначено для изменения общих настроек контроллера.



Окно **Настройки** (Settings) состоит из четырех панелей:

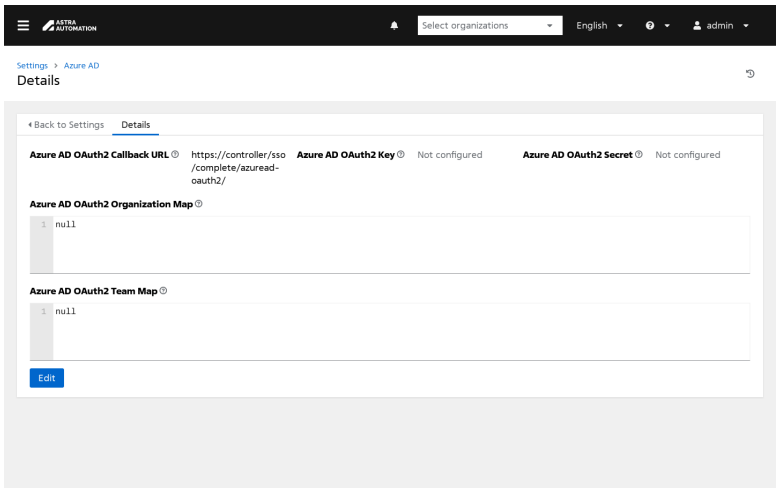
- **Аутентификация** (Authentication) – настройки аутентификации с использованием внешних провайдеров.
- **Задания** (Jobs) – настройки выполнения заданий.
- **Система** (System) – настройки встроенного провайдера аутентификации, общесистемные настройки и настройки журналирования.
- **Пользовательский интерфейс** (User Interface) – настройки внешнего вида страницы аутентификации.
- **Диагностика проблем** (Troubleshooting) – настройки диагностики.

Аутентификация

Панель **Аутентификация** (Authentication) содержит ссылки на страницы настроек внешних провайдеров аутентификации.

Azure AD

Окно **Azure AD** используется для настройки аутентификации с помощью провайдера Microsoft Azure Active Directory.



Форма **Редактировать** (Edit) содержит следующие поля:

- **Ключ Azure AD OAuth2** (Azure AD OAuth2 Key) – код авторизации приложения в Azure AD.
- **Секрет Azure AD OAuth2** (Azure AD OAuth2 Secret) – секретный ключ авторизации приложения в Azure AD.
- **Ассоциация организаций в контроллере с Azure AD с помощью OAuth2** (Azure AD OAuth2 Organization Map) – настройки ассоциации пользователей Azure AD с организациями контроллера.

Пример

Пусть настройки ассоциации пользователей Azure AD с организациями контроллера заданы следующим образом:

```
{
  "Jupiter": {
    "admins": [
      "/^admin_.*?@jupiter\.example\.com$/"
    ],
    "users": [
      "/.*?@jupiter\.example\.com$/"
    ],
    "remove_admins": true,
    "remove_users": true
  },
  "Saturn": {
    "admins": [
      "/^admin_.*?@saturn\.example\.com$/"
    ],
    "users": true,
    "remove_admins": true,
    "remove_users": true
```

(continues on next page)

(продолжение с предыдущей страницы)

```
  }
}
```

В список пользователей организации «Jupiter» будут добавлены все пользователи Azure AD, названия учетных записей которых заканчиваются на `@jupiter.example.com`. В администраторы организации «Juputer» будут добавлены только те пользователи Azure AD, названия учетных записей которых начинаются с `admin_` и заканчиваются на `@jupiter.example.com`. Все пользователи Azure AD, не удовлетворяющие указанным условиям, будут удалены из списка администраторов организации «Jupiter» и списка ее участников.

Аналогичные настройки используются для организации «Saturn», с той лишь разницей, что названия учетных записей пользователей этой организации должны заканчиваться на `@saturn.example.com`.

- **Ассоциация команд в контроллере с Azure AD с помощью OAuth2** (Azure AD OAuth2 Team Map) – настройки ассоциации пользователей Azure AD с командами контроллера.

Пример

Пусть настройки ассоциации пользователей Azure AD с командами контроллера заданы следующим образом:

```
{
  "DevOps Team": {
    "organization": "Jupiter",
    "users": [
        "/.*?@jupiter\.example\.ru$/"
    ],
    "remove": true
  }
}
```

Если организация «Jupiter» не существует, она будет создана.

Если в организации «Jupiter» не существует команда «DevOps Team», она будет создана.

В список участников команды «DevOps Team» будут добавлены все пользователи Azure AD, названия учетных записей которых заканчиваются на `@jupiter.example.ru`. Все пользователи Azure AD, не удовлетворяющие указанным условиям, будут удалены из списка участников команды «DevOps Team».

Подробности об ассоциации пользователей Azure AD с командами контроллера см. в секции *Ассоциация с командами*.

GitHub

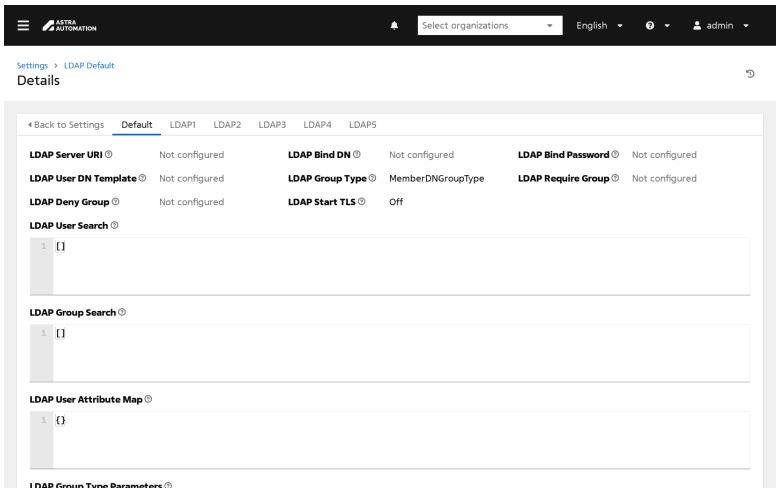
Важно: Эта часть документации находится в стадии разработки.

Google OAuth 2

Важно: Эта часть документации находится в стадии разработки.

LDAP

Окно **LDAP** используется для настройки аутентификации с помощью *провайдера LDAP*.



Оно состоит из вкладки **По умолчанию** (Default), содержащей настройки для сервера аутентификации LDAP по умолчанию, и пяти вкладок для настройки дополнительных серверов аутентификации LDAP.

Все вкладки содержат одинаковые наборы настроек.

Форма **Редактировать** (Edit) содержит следующие поля:

- **URI сервера LDAP** (LDAP Server URI) – URI сервера LDAP в следующем формате:

  ```
  <protocol>://<domain>:<port>
  ```

  где:

  - <protocol> – тип протокола для подключения, одно из значений:
    - `ldap` – подключение без защиты;
    - `ldaps` – защита подключения с помощью TLS.
  - <domain> – *FQDN* сервера LDAP.
  - <port> – порт для подключения. Как правило, порт 389 используется для подключения по протоколу `ldap`, а порт 636 – `ldaps`.

  Если необходимо ввести URI нескольких серверов, используйте символы пробела или запятой в качестве разделителя.

  Пустое значение в этом поле отключает аутентификацию через LDAP.

- **Пароль привязки к LDAP** (LDAP Bind Password) – пароль для подключения к серверу LDAP.

- **Тип группы LDAP** (LDAP Group Type) – тип обработчика, используемого для групп LDAP.

  Может принимать одно из следующих значений:

  - `ActiveDirectoryGroupType`;
  - `GroupOfNamesType`;
  - `GroupOfUniqueNamesType`;

- MemberDNGroupType;
- NestedActiveDirectoryGroupType;
- NestedGroupOfNamesType;
- NestedGroupOfUniqueNamesType;
- NestedMemberDNGroupType;
- NestedOrganizationalRoleGroupType;
- OrganizationalRoleGroupType;
- PosixGroupType;
- PosixUIDGroupType.

Подробное описание каждого типа см. в документации[146] библиотеки `django-auth-ldap`.

Значение по умолчанию – `MemberDNGroupType`.

- **Запуск LDAP с TLS** (LDAP Start TLS) – использование TLS для подключения к серверу LDAP по протоколу `ldap` (без защиты соединения).

  Значение по умолчанию – Выкл (Off).

- **LDAP Bind DN** – отличительное наименование пользователя (Distinguished Name) для привязки ко всем поисковым запросам, например:

```
CN=users,DC=example,DC=com
```

- **Шаблон уникального наименования (DN) пользователя в LDAP** (LDAP User DN Template) – если все пользователи в LDAP имеют одно и то же отличительное наименование, укажите в этом поле шаблон для более эффективного поиска.

Важно: Указанное в этом поле значение используется вместо значения, заданного в поле **LDAP Bind DN**.

- **Обязательная группа LDAP** (LDAP Require Group) – отличительное наименование группы LDAP, в которую должны входить пользователи. Если пользователь не является членом указанной группы, его аутентификация в контроллере невозможна.

  Если поле не заполнено, все пользователи LDAP, соответствующие поисковому запросу, смогут аутентифицироваться в контроллере.

  Поддерживается только одна обязательная группа.

- **Запрещенная группа LDAP** (LDAP Deny Group) – отличительное наименование группы LDAP, членам который запрещено аутентифицироваться в контроллере.

  Поддерживается только одна запрещенная группа.

- **LDAP-запрос по поиску пользователей** (LDAP User Search) – параметры запроса, используемого для поиска пользователей в LDAP, например:

```
[
  "DC=example,DC=com",
  "SCOPE_SUBTREE",
  "(cn=%(user)s)"
]
```

  Здесь:

  - `DC=example,DC=com` – поиск в домене `example.com`;
  - `SCOPE_SUBTREE` – ограничение поиска доменами, дочерними по отношению к `example.com`;

[146] https://django-auth-ldap.readthedocs.io/en/stable/groups.html#types-of-groups

- `(CN=%(user)s)` – фильтрация найденных пользователей по свойству `CN`, равному значению параметра `user`.

Поддерживается использование LDAPSearchUnion для создания сложных запросов.

Подробности см. в **RFC 4517**[147].

- **Групповой поиск по LDAP** (LDAP Group Search) – настройки поиска групп в LDAP, например:

```
[
  "DC=example,DC=com",
  "SCOPE_SUBTREE",
  "(ObjectClass=group)"
]
```

Здесь:

- `DC=example,DC=com` – поиск в домене `example.com`;
- `SCOPE_SUBTREE` – ограничение поиска доменами, дочерними по отношению к `example.com`;
- `(ObjectClass=group)` – поиск только объектов класса `group`.

Использование LDAPSearchUnion в этом поле не поддерживается.

Подробности см. в **RFC 4517**[148].

- **Ассоциация пользовательских атрибутов с LDAP** (LDAP User Attribute Map) – правила ассоциирования данных пользователя в LDAP с данными пользователя в контроллере, например:

```
{
  "first_name", "givenName",
  "last_name", "sn",
  "email": "mail"
}
```

По умолчанию используются настройки для работы с Microsoft Active Directory.

- **Параметры типа группы LDAP** (LDAP Group Type Parameters) – дополнительные параметры типа группы LDAP.

Список поддерживаемых параметров зависит от значения, выбранного в поле **Тип группы LDAP** (LDAP Group Type). Подробности см. в документации[149] библиотеки `django-auth-ldap`.

Значение по умолчанию:

```
{
  "member_attr": "member",
  "name_attr": "cn"
}
```

- **Пользовательские флаги LDAP по группам** (LDAP User Flags By Group) – критерии, при выполнении которых пользователям LDAP в контроллере будут автоматически присваиваться *типы* «Системный администратор» (System Administrator) и «Системный аудитор» (System Auditor), например:

```
{
  "is_superuser": "cn=superusers,ou=groups,db=example,dc=com",
  "is_system_auditor": "cn=auditors,ou=groups,dc=example,dc=com"
}
```

[147] https://datatracker.ietf.org/doc/html/rfc4517.html
[148] https://datatracker.ietf.org/doc/html/rfc4517.html
[149] https://django-auth-ldap.readthedocs.io/en/stable/groups.html#types-of-groups

- **Ассоциация организации с LDAP** (LDAP Organization Map) – параметры ассоциации организаций LDAP с *организациями* контроллера.

  Пример

  Пусть настройки ассоциации пользователей LDAP с организациями контроллера заданы следующим образом:

```
{
  "Jupiter": {
    "admins": "CN=admins,OU=groups,DC=jupiter,DC=example,DC=com",
    "users": [
      "CN=devops,OU=groups,DC=jupiter,DC=example,DC=com",
      "CN=testers,OU=groups,DC=jupiter,DC=example,DC=com"
    ],
    "remove_admins": true,
    "remove_users": true
  },
  "Saturn": {
    "admins": "CN=admins,OU=groups,DC=saturn,DC=example,DC=com",
    "users": true,
    "remove_admins": true,
    "remove_users": false
  }
}
```

  Если организации «Jupiter» и «Saturn» не существуют, они будут созданы.

  В список администраторов организации «Jupiter» будут добавлены все пользователи LDAP, которые удовлетворяют следующим условиям:

  - состоят в группе LDAP `admins`;
  - имеют учетную запись в домене `jupiter.example.com`.

  Все пользователи LDAP, не удовлетворяющие указанным условиям, будут удалены из списка администраторов организации «Jupiter».

  В список участников организации «Jupiter» будут добавлены все пользователи LDAP, которые удовлетворяют следующим условиям:

  - состоят в группах LDAP `devops` или `testers`;
  - имеют учетную запись в домене `jupiter.example.com`.

  Все пользователи LDAP, не удовлетворяющие указанным условиям, будут удалены из списка участников организации «Jupiter».

  В список администраторов организации «Saturn» будут добавлены все пользователи LDAP, которые удовлетворяют следующим условиям:

  - состоят в группе LDAP `admins`;
  - имеют учетную запись в домене `saturn.example.ru`.

  Все пользователи LDAP, не удовлетворяющие указанным условиям, будут удалены из списка администраторов организации «Saturn».

  Участниками организации «Saturn» являются любые пользователи, найденные в LDAP.

  Участники организации «Saturn», не найденные в LDAP, не будут удалены из контроллера.

  Подробности об ассоциации внешних пользователей с организациями контроллера см. в секции *Ассоциация с организациями*.

- **Ассоциация группы с LDAP** (LDAP Team Map) – параметры ассоциации групп LDAP с *командами* контроллера.

Пример

Пусть настройки ассоциации пользователей LDAP с командами контроллера заданы следующим образом:

```
{
  "DevOps Team": {
    "organization": "Jupiter",
    "users": "CN=devops,OU=groups,DC=jupiter,DC=example,DC=com",
    "remove": true
  }
}
```

Если организация «Jupiter» не существует, она будет создана.

Если в организации «Jupiter» не существует команда «DevOps Team», она будет создана.

В список участников команды «DevOps Team» будут добавлены все пользователи LDAP, которые удовлетворяют следующим условиям:

- входят в группу LDAP devops;
- имеют учетную запись в домене `jupiter.example.com`.

Все пользователи LDAP, не удовлетворяющие указанным условиям, будут удалены из списка участников команды «DevOps Team».

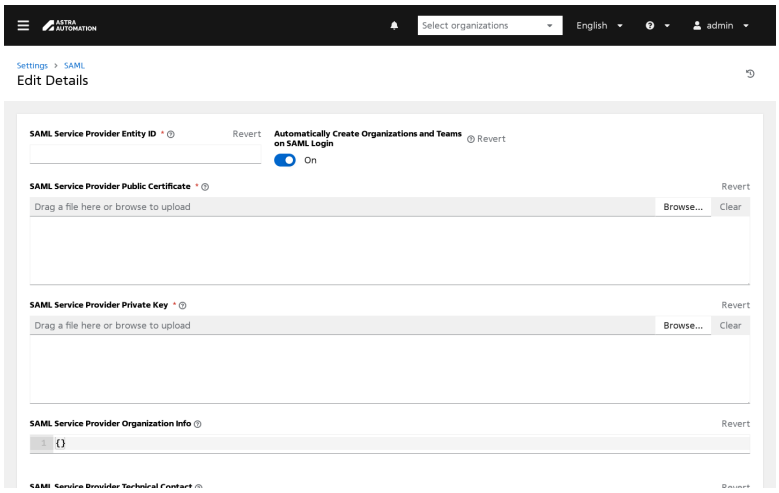
Подробности об ассоциации внешних пользователей с командами контроллера см. в секции *Ассоциация с командами*.

RADIUS

Важно: Эта часть документации находится в стадии разработки.

SAML

Окно **SAML** используется для настройки аутентификации с помощью *поставщиков услуг аутентификации SAML*.



Форма **Редактировать** (Edit) содержит следующие поля:

- **Уникальный идентификатор объекта поставщика услуг SAML** (SAML Service Provider Entity ID) – как правило, в качестве идентификатора объекта поставщика услуг

аутентификации SAML используется базовый URL контроллера. Его значение указывается в *общих системных настройках контроллера*.

Если для доступа к контроллеру используется балансировщик нагрузки, укажите его URL.

- **Автоматически создавать группы и организации при подключении к SAML** (Automatically Create Organizations and Teams on SAML Login) – при успешной аутентификации пользователей SAML необходимые *организации* и *команды* создаются автоматически.
- **Открытый сертификат поставщика услуг SAML** (SAML Service Provider Public Certificate) – открытый сертификат поставщика услуг SAML, например:

```
-----BEGIN CERTIFICATE----
AAAA231jfdsalfkjkl........
----END CERTIFICATE-----
```

- **Закрытый ключ поставщика услуг SAML** (SAML Service Provider Private Key) – закрытый ключ поставщика услуг SAML, например:

```
-----BEGIN PRIVATE KEY-----
PPPPP2183910das0921........
-----END PRIVATE KEY-----
```

- **Информация об организации поставщика услуг SAML** (SAML Service Provider Organization Info) – словарь с информацией об организации поставщика услуг SAML, например:

```
{
  "en-US": {
    "url": "https://ac.example.com",
    "displayname": "Automation Controller cluster",
    "name": "ac-cluster"
  },
  "ru-RU": {
    "url": "https://ac.example.com",
    "displayname": "Кластер Automation Controller",
    "name": "ac-cluster"
  }
}
```

- **Контакт инженера поставщика SAML** (SAML Service Provider Technical Contact) – словарь с информацией о службе технической поддержки поставщика услуг SAML, например:

```
{
  "givenName": "SAML Technical Support",
  "emailAddress": "tech-support@saml.example.com"
}
```

- **Контакт службы поддержки поставщика SAML** (SAML Service Provider Support Contact) – словарь с информацией о службе поддержки пользователей поставщика услуг SAML, например:

```
{
  "givenName": "SAML User Support",
  "emailAddress": "user-support@saml.example.com"
}
```

- **Разрешить использование провайдеров контроля идентичности через SAML** (SAML Enabled Identity Providers) – словарь с настройками отдельных поставщиков услуг SAML.

Пример

Пусть словарь настроек содержит следующие данные о поставщиках услуг SAML:

```
{
  "keycloak": {
    "entity_id": "https://saml.example.com/keycloak/b2912kdsjkl1k",
    "url": "https://saml.example.com/keycloak/sso/",
    "x509cert": "MIID0DCCAj12309jklwqW...3123jklfdsajkl213",
    "attr_user_permanent_id": "user_id",
    "attr_username": "Email",
    "attr_email": "Email",
    "attr_first_name": "FirstName",
    "attr_last_name": "LastName"
  }
}
```

Здесь:

- `keycloak` – идентификатор поставщика услуг аутентификации SAML.
- `entity_id` – идентификатор профиля, предоставленный поставщиком услуг аутентификации SAML.
- `url` – URL, на который Automation Controller должен перенаправлять пользователей для аутентификации через SAML.
- `x509cert` – сертификат провайдера идентификации (IdP) SAML.

  При заполнении этого поля необходимо убрать из сертификата маркеры начала (`-----BEGIN CERTIFICATE-----`) и конца (`-----END CERITIFICATE-----`), а оставшиеся строки объединить в одну.
- `attr_user_permanent_id` – название атрибута пользователя SAML, в котором хранится его уникальный идентификатор.
- `attr_username` – название атрибута SAML, в котором хранится название учетной записи пользователя. В данном случае в качестве названия учетной записи пользователя Automation Controller используется адрес электронной почты пользователя SAML, поэтому в значении ключа `attr_username` указано значение `Email`.
- `attr_email` – название атрибута SAML, в котором хранится адрес электронной почты пользователя.
- `attr_first_name` – название атрибута пользователя SAML, в котором хранится имя пользователя.
- `attr_last_name` – название атрибута пользователя SAML, в котором хранится фамилия пользователя.

- **Ассоциация организаций с помощью SAML** (SAML Organization Map) – правила назначения пользователям SAML ролей «Администратор» и «Участник» на уровне организаций.

Пример

Пусть настройки ассоциации пользователей SAML с организациями контроллера заданы следующим образом:

```
{
  "Jupiter": {
    "admins": "admin@jupiter.example.com",
    "users": [
      "/.*?@users\.example\.com$/i",
      "/.*?devops\.example\.com$/i"
    ],
    "remove_admins": true,
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    "remove_users": true
  }
}
```

Если организация «Jupiter» не существует, она будет создана.

В список администраторов организации «Jupiter» будет добавлен пользователь SAML с названием учетной записи `admin@jupiter.example.com`. Все пользователи SAML, не удовлетворяющие указанному условию, будут удалены из списка администраторов организации «Jupiter».

В список участников организации «Jupiter» будут добавлены все пользователи SAML, названия учетных записей которых заканчиваются на `@users.example.com` и `@devops.example.com` (без учета регистра). Все пользователи SAML, не удовлетворяющие указанным условиям, будут удалены из списка участников организации «Jupiter».

Подробности см. в секции *Ассоциация с организациями*.

- **Ассоциация атрибутов организаций с помощью SAML** (SAML Organization Attribute Mapping) – словарь с названиями атрибутов, которыми должен обладать пользователь SAML для назначения ему ролей «Администратор» и «Участник» на уровне организаций.

Пример

Пусть настройки ассоциации атрибутов организаций заданы следующим образом:

```
{
  "saml_attr": "is-member",
  "saml_admin_attr": "is-admin",
  "remove": true,
  "remove_admins": true
}
```

Если в данных пользователя SAML есть атрибут `is-member`, ему назначается роль «Участник» на уровне соответствующих организаций контроллера.

Если в данных пользователя SAML есть атрибут `is-admin`, ему назначается роль «Администратор» на уровне соответствующих организаций контроллера.

У пользователей SAML, не имеющих указанных атрибутов, роли «Администратор» и «Участник» на уровне организаций отзываются автоматически.

Подробности см. в секции *Ассоциация атрибутов организаций*.

- **Ассоциация команд с помощью SAML** (SAML Team Map) – правила назначения пользователям SAML ролей «Администратор» и «Участник» на уровне команд.

Пример

Пусть настройки ассоциации пользователей SAML с командами контроллера заданы следующим образом:

```
{
  "DevOps Team": {
    "organization": "Jupiter",
    "users": "/.*?@jupiter\.example\.com$/",
    "remove": true
  }
}
```

Если организация «Jupiter» не существует, она будет создана.

Если в организации «Jupiter» не существует команда «DevOps Team», она будет создана.

Пользователям SAML, название учетной записи которых заканчивается на `@jupiter.example.com`, автоматически назначается роль «Участник» на уровне команды «DevOps Team». Пользователи SAML, не удовлетворяющие этому условию, автоматически удаляются из списка участников команды «DevOps Team».

Подробности см. в секции *Ассоциация с командами*.

- **Ассоциация атрибутов команд с помощью SAML** (SAML Team Attribute Mapping) – словарь с названиями атрибутов, которыми должен обладать пользователь SAML для назначения ему ролей «Администратор» и «Участник» на уровне команд.

Пример

Пусть настройки ассоциации атрибутов команд заданы следующим образом:

```
{
  "saml_attr": "team",
  "remove": true,
  "team_org_map": [{
    "team": "data:team:devOps",
    "team_alias": "DevOps Team",
    "organization": "org.jptr"
  }, {
    "team": "data:team:auditors",
    "team_alias": "Auditors Team",
    "organization": "org.jptr"
  }]
}
```

Если в контроллере не существует организация `org.jptr`, она будет создана.

Если в организации `org.jptr` не существуют команды «DevOps Team» и «Auditors Team», они будут созданы.

Пользователям SAML, имеющим атрибут `team`, значение которого равно `data:team:devOps`, назначается роль «Участник» в команде «DevOps Team». Из списка участников команды «DevOps Team» автоматически удаляются пользователи, которые:

- не имеют атрибута `team`, либо
- имеют атрибут `team`, но его значение не равно `data:team:devOps`.

Пользователям SAML, имеющим атрибут `team`, значение которого равно `data:team:auditors`, назначается роль «Участник» в команде «Auditors Team». Из списка участников команды «Auditors Team» автоматически удаляются пользователи, которые:

- не имеют атрибута `team`, либо
- имеют атрибут `team`, но его значение не равно `data:team:auditors`.

Подробности см. в секции *Ассоциация атрибутов команд*.

- **Ассоциация атрибутов пользователей SAML** (SAML User Flags Attribute Mapping) – настройки назначения пользователям SAML ролей «Системный администратор» и «Системный аудитор».

Пример

Пусть настройки ассоциации атрибутов пользователей SAML заданы следующим образом:

```
{
  "is_superuser_role": [
    "admin",
    "root"
  ],
  "is_superuser_attr": "groups",
  "is_superuser_value": [
    "admin",
    "root",
    "astra-admin"
  ],
  "is_system_auditor_role": "auditor",
  "is_system_auditor_attr": "groups",
  "is_system_auditor_value": ["auditor"]
}
```

Роль «Системный администратор» назначается пользователям SAML, удовлетворяющим хотя бы одному из указанных условий:

- наличие атрибута `groups`, хотя бы одно из значений которого равно `admin`, `root` или `astra-admin`;
- наличие роли `admin` или `root`.

У пользователей SAML, не удовлетворяющих указанным условиям, роль «Системный администратор» автоматически отзывается.

Роль «Системный аудитор» назначается пользователям SAML, удовлетворяющим хотя бы одному из указанных условий:

- наличие атрибута `groups`, хотя бы одно из значений которого равно `auditor`;
- наличие роли `auditor`.

У пользователей SAML, не удовлетворяющим указанным условиям, роль «Системный аудитор» автоматически отзывается.

Подробности см. в секции *Атрибуты пользователей*.

- **Конфигурация безопасности SAML** (SAML Security Config) – настройки безопасности поставщика услуг аутентификации SAML.

  Значение по умолчанию:

```
{
  "requestedAuthnContext": false
}
```

  Подробное описание доступных настроек см. в документации библиотеки python-saml[150].

- **Дополнительные данные настройки сервиса идентификации SAML** (SAML Service Provider extra configuration data) – дополнительные настройки поставщика услуг аутентификации SAML.

  Подробное описание доступных настроек см. в документации библиотеки python-saml[151].

- **Ассоциация SAML IDP с атрибутами extra_data** (SAML IDP to extra_data attribute mapping) – список кортежей с названиями атрибутов поставщика услуг идентификации (IdP), преобразуемых в атрибуты `extra_data`, например:

[150] https://github.com/SAML-Toolkits/python-saml?tab=readme-ov-file#settings
[151] https://github.com/SAML-Toolkits/python-saml?tab=readme-ov-file#settings

```
[
  ('attribute_name', 'extra_data_name_for_attribute'),
  ('department', 'department'),
  ('user_first_name', 'first_name')
]
```

Подробное описание доступных настроек см. в документации библиотеки python-social-auth[152].

TACACS+

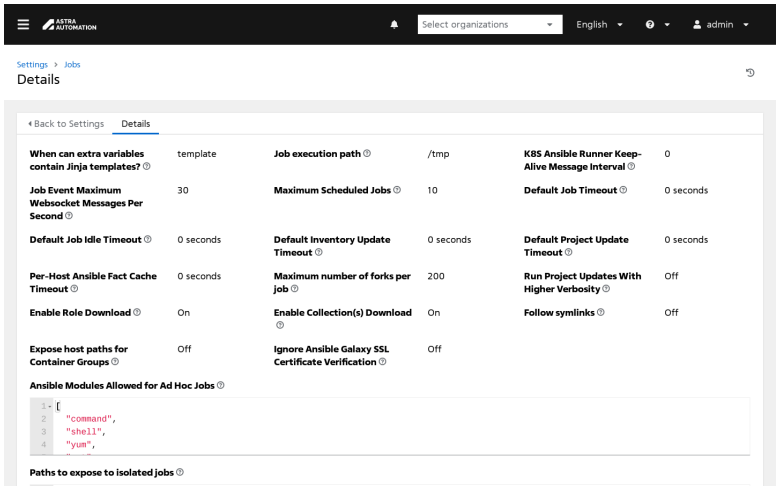
Важно: Эта часть документации находится в стадии разработки.

Общий OIDC

Важно: Эта часть документации находится в стадии разработки.

Задания

Окно **Задания** (Jobs) содержит настройки, которые управляют запуском и исполнением всех заданий в Automation Controller.



Форма **Редактировать** (Edit) содержит следующие поля:

- **Путь выполнения заданий** (Job execution path)

  Путь к каталогу, в котором контроллер должен создать временный подкаталог для хранения данных, связанных с заданием, например, файлы полномочий.

  Значение по умолчанию – `/tmp`.

- **Максимальное количество запланированных заданий** (Maximum Scheduled Jobs)

  Количество заданий на основе одного и того же шаблона, которые можно поставить в очередь выполнения. Если указанное значение будет достигнуто, создавать новые задания будет нельзя.

  Значение по умолчанию – `10`.

[152] https://python-social-auth.readthedocs.io/en/latest/backends/saml.html#advanced-settings

- **K8S Ansible Runner Keep-Alive Message Interval**

  Интервал в секундах, через который узлам из группы контейнеров отправляется сообщение для поддержания соединения. При значении `0` сообщения не отправляются.

  Значение по умолчанию – `0`.

- **Тайм-аут задания по умолчанию** (Default Job Timeout)

  Время в секундах, отведенное на исполнение каждого задания. Значение этой настройки может быть переопределено в настройках шаблона задания. При значении `0` задания выполняются без ограничений по времени.

  Значение по умолчанию – `0`.

- **Default Job Idle Timeout** (Default Job Idle Timeout)

  Если за указанный период времени в секундах не происходит вывода Ansible, исполнение задания прерывается. При значении `0` задания могут выполняться без вывода неограниченно долго.

  Значение по умолчанию – `0`.

- **Тайм-аут обновления инвентаря по умолчанию** (Default Inventory Update Timeout)

  Время в секундах, отведенное на выполнение задачи обновления инвентаря из внешнего источника. Значение этой настройки может быть переопределено в настройках инвентаря. При значении `0` задания обновления инвентаря выполняются без ограничений по времени.

  Значение по умолчанию – `0`.

- **Тайм-аут обновления проекта по умолчанию** (Default Project Update Timeout)

  Время в секундах, отведенное на выполнение задачи обновления проекта из внешнего источника. Значение этой настройки может быть переопределено в настройках проекта. При значении `0` задания обновления проекта выполняются без ограничений по времени.

  Значение по умолчанию – `0`.

- **Тайм-аут кэширования ansible фактов для каждого узла** (Per-Host Ansible Fact Cache Timeout)

  Период времени в секундах, в течение которого факты Ansible считаются действительными с момента их последнего изменения. В playbook будут доступны только действительные, не устаревшие факты. При значении `0` факты не устаревают.

  Важно: Эта настройка не влияет на удаление фактов в базе данных Automation Controller.

  Значение по умолчанию – `0`.

- **Максимальное количество ответвлений задания** (Maximum number of forks per job)

  Максимальное количество ответвлений, которое можно задать в настройках шаблона задания. Попытка сохранить шаблон задания со значением больше указанного приведет к ошибке.

  Значение по умолчанию – `200`.

- **В каких случаях дополнительные переменные могут содержать шаблоны Jinja?** (When can extra variables contain Jinja templates?)

  Разрешение на использование шаблонов Jinja2 в дополнительных переменных. Поддерживаются следующие значения:

  - `Всегда` (Always);
  - `Никогда` (Never);
  - `В шаблонах` (Template).

> **Предупреждение:** Для этой настройки рекомендуется использовать значение Никогда или В шаблонах. Это связано с тем, что шаблоны Jinja2 потенциально могут быть использованы для запуска произвольного кода на языке Python, что создает угрозу безопасности.

Значение по умолчанию – В шаблонах (Template).

- **Выполнить обновление проекта с большей детализацией** (Run Project Updates With Higher Verbosity)

  Если эта настройка включена, используется более детализированный вывод при выполнении заданий обновления проектов. Использование этой настройки эквивалентно выполнению команды `ansible-playbook project_update.yml` с параметром `-vvv`.

  Значение по умолчанию – выключено.

- **Игнорировать проверку SSL-сертификата Ansible Galaxy** (Ingore Ansible Galaxy SSL Certificate Verification)

  Если эта настройка включена, при подключении к реестрам коллекций Ansible не выполняется проверка сертификата TLS.

  Значение по умолчанию – выключено (проверка выполняется).

- **Включить загрузку ролей** (Enable Role Download)

  Если эта настройка включена и источником кода проекта является система контроля версий, разрешена загрузка ролей, перечисленных в файле проекта `roles/requirements.yml`.

  Значение по умолчанию – включено.

- **Разрешить загрузку коллекции(й)** (Enable Collection(s) Download)

  Если эта настройка включена и источником кода проекта является система контроля версий, разрешена загрузка коллекций, перечисленных в файле проекта `collections/requirements.yml`.

  Значение по умолчанию – включено.

- **Переходы по символическим ссылкам** (Follow symlinks)

  Если эта настройка включена, при обработке playbook разрешен переход по символическим ссылкам.

  > **Предупреждение:** Переход по ссылкам может привести к бесконечной рекурсии, если ссылка указывает на свой родительский каталог.

  Значение по умолчанию – выключено.

- **Expose host paths for Container Groups**

  Если эта настройка включена, разрешено использование `hostPath` для подов, созданных в группе контейнеров.

  > **Предупреждение:** Использование томов `hostPath` содержит потенциальные риски безопасности. Рекомендуется избегать использования `hostPath`, где это возможно.

  Значение по умолчанию – выключено.

- **Модули Ansible, разрешенные для выполнения специальных заданий** (Ansible Modules Allowed for Ad Hoc Jobs)

  Список модулей Ansible, которые разрешено использовать при выполнении специальных (ad-hoc) команд.

Значение по умолчанию:

```
[
  "command",
  "shell",
  "yum",
  "apt",
  "apt_key",
  "apt_repository",
  "apt_rpm",
  "service",
  "group",
  "user",
  "mount",
  "ping",
  "selinux",
  "setup",
  "win_ping",
  "win_service",
  "win_updates",
  "win_group",
  "win_user"
]
```

- **Расширения Ansible Callback** (Ansible Callback Plugins)

  Список путей для поиска расширений обратного вызова Ansible, по одному на строку. При значении `[]` расширения обратного вызова не используются.

  Значение по умолчанию – `[]`.

- **Пути для доступа к изолированным заданиям** (Path to expose to isolated jobs)

  Список путей, монтируемых на исполняющих узлах в контейнеры как отдельные тома, по одному на строку. Пути, не указанные в этом списке, недоступны в контейнерах и не могут быть использованы при выполнении заданий.

  Формат записей:

```
<HOST_DIR>[:<CONTAINER_DIR>[:<OPTIONS>]]
```

  Здесь:

  - <HOST_DIR> – путь к каталогу в файловой системе исполняющего узла;
  - <CONTAINER_DIR> – точка монтирования каталога в файловой системе контейнера;
  - <OPTIONS> – опциональные параметры монтирования.

  Значение по умолчанию:

```
[
   "/etc/pki/ca-trust:/etc/pki/ca-trust:O",
   "/usr/share/pki:/usr/share/pki:O"
]
```

- **Environment Variables for Galaxy Commands**

  Дополнительные переменные окружения, используемые при вызове команды `ansible-galaxy` для обновления проекта. Использование этих переменных может быть полезно, если команда `ansible-galaxy` работает через прокси-сервер, но не Git.

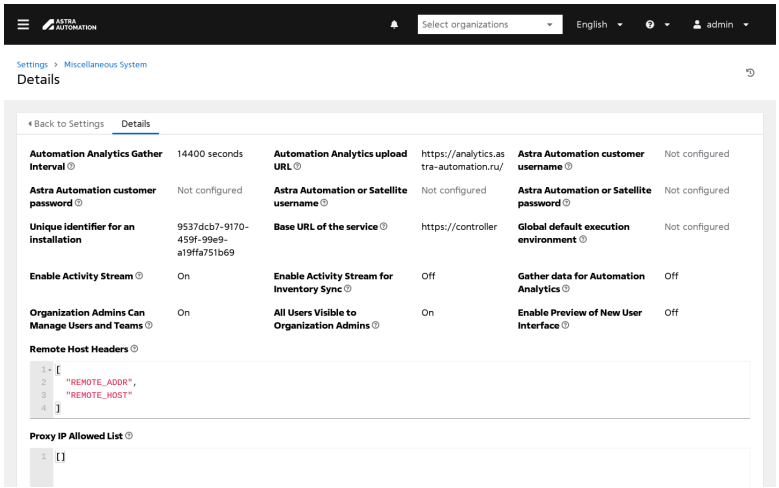
  Значение по умолчанию:

```
{
  "ANSIBLE_FORCE_COLOR": "false",
  "GIT_SSH_COMMAND": "ssh -o StrictHostKeyChecking=no"
}
```

Система

Панель **Система** (System) содержит ссылки на разделы, используемые для управления общими параметрами системы, параметрами встроенного провайдера аутентификации и журналированием.

Общие системные настройки



Окно **Общие системные настройки** (Miscellaneous System settings) содержит следующие настройки:

- **Включить ленту активности** (Enable Activity Stream)

  Если эта настройка включена, ведется журнал активности (*Режимы просмотра* ▸ *Лента активности*).

  Значение по умолчанию – включено.

- **Включить ленту активности для синхронизации инвентаря** (Enable Acitivity Stream for Inventory Sync)

  Если эта настройка включена, в ленте активности выводятся сведения о событиях синхронизации инвентаря.

  Значение по умолчанию – выключено.

- **Глобальная среда исполнения по умолчанию** (Global default execution environment)

  Среда исполнения, используемая по умолчанию на уровне контроллера.

- **Базовый URL сервиса** (Base URL of the service)

  URL, используемый при создании ссылок на страницу контроллера, в том числе при настройке аутентификации SAML. Укажите в этом поле доменное имя, используемое для доступа к контроллеру.

  При использовании балансировщика нагрузки укажите его URL.

  Значение по умолчанию – `https://controller/`.

- **Пользователи, доступные администраторам организации** (All Users Visible to Organization Admins)

  Если эта настройка включена, администраторы организаций имеют возможность управлять пользователями других организаций.

  Значение по умолчанию – включено.

- **Администраторы организации могут управлять пользователями и командами** (Organization Admins Can Manage Users and Teams)

Если эта настройка включена, администраторы организаций могут управлять своими пользователями и командами.

Значение по умолчанию – выключено.

- **Gather data for Automation Analytics**

  Сбор данных для сервиса Automation Analytics.

  Значение по умолчанию – выключено.

- **Astra Automation customer username** – название учетной записи пользователя, используемой для подключения к серверу Automation Analytics.
- **Astra Automation customer password**

  Пароль пользователя, используемый для подключения к серверу Automation Analytics.

- **Astra Automation or Satellite username**
- **Astra Automation or Satellite password**
- **Automation Analytics upload URL**

  URL для передачи информации, собранной службой Automation Analytics.

- **Automation Analytics Gather Inverval**

  Период (в секундах) отправки информации, собранной службой Automation Analytics.

  Значение по умолчанию – `14400` (4 часа).

- **Enable Preview of New User Interface**

  Если эта настройка включена, используется экспериментальный пользовательский интерфейс.

  Значение по умолчанию – выключено.

- **Last gathered entries from the data collection service of Automation Analytics**
- **Заголовки (headers) удаленного узла** (Remote Host Headers)

  Заголовки HTTP и мета-ключи, используемые для определения имени удаленного хоста или его IP-адреса. Если доступ к контроллеру осуществляется через обратный прокси-сервер, добавьте в список дополнительные заголовки, например, `HTTP_X_FORWARDED_FOR`.

  Значение по умолчанию:

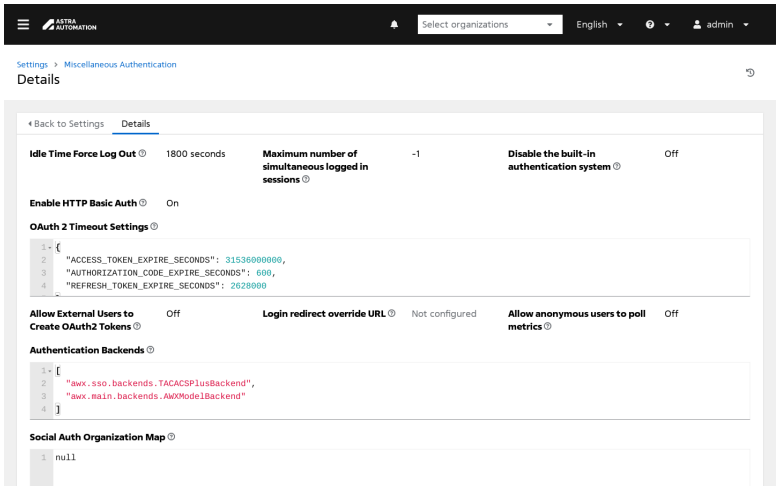
```
[
    "REMOTE_ADDR",
    "REMOTE_HOST"
]
```

- **IP-адреса разрешенных прокси-серверов** (Proxy IP Allowed List)

  Список IP-адресов разрешенных прокси-серверов или балансировщика нагрузки. При значении `[]` запрещено использование прокси-серверов или балансировщика нагрузки.

  Значение по умолчанию – `[]`.

Общие параметры аутентификации



Окно **Общие параметры аутентификации** (Miscellaneous Authentication settings) содержит настройки, которые управляют встроенным провайдером аутентификации:

- **Отключить встроенную систему аутентификации** (Disable the built-in authentication system)

  Если эта настройка выключена, в контроллере используется встроенная система аутентификации. Если для аутентификации в контроллере используется внешний провайдер аутентификации, например, LDAP или SAML, эту настройку необходимо включить.

  Значение по умолчанию – выключено.

- **Время бездействия до принудительного выхода** (Idle Time Force Log Out)

  Время в секундах, по истечение которого сессии неактивных пользователей автоматически завершаются.

  Значение по умолчанию – `1800` (30 минут).

- **Максимальное количество одновременных сеансов входа в систему** (Maxumum number of simultaneous logged in sessions)

  Максимальное количество одновременных сеансов пользователя. При значении `-1` количество сеансов не ограничено.

  Значение по умолчанию – `-1`.

- **Включить HTTP Basic Auth** (Enable HTTP Basic Auth)

  Если эта настройка включена, разрешено использование браузерного API «HTTP Basic Auth».

  Примечание: Подробности об API «HTTP Basic Auth» см. в **RFC 7617**[153].

  Значение по умолчанию – включено.

- **Разрешить внешним пользователям создавать токены OAuth2** (Allow External Users to Create OAuth2 Tokens)

  Если эта настройка включена, пользователи от внешних провайдеров аутентификации (LDAP, SAML и другие) получают возможность создавать токены OAuth2.

  Значение по умолчанию – выключено.

- **Перенаправление URL при входе** (Login redirect override URL)

[153] https://datatracker.ietf.org/doc/html/rfc7617.html

URL, по которому для входа в систему будут перенаправлены неавторизованные пользователи. Если значение не указано, используется перенаправление на страницу входа.

Значение по умолчанию – пустая строка.

- **Истечение срока действия токена доступа** (Access Token Expiration)

  Время в секундах, в течение которого действует токен доступа (access token).

  Значение по умолчанию – `31536000000` (~1000 лет).

- **Срок действия токена обновления истек** (Refresh Token Expiration)

  Время в секундах, в течение которого действует токен обновления (refresh token) после истечения токена доступа.

  Значение по умолчанию – `2628000` (~30 дней).

- **Истечение срока действия кода авторизации** (Authorization Code Expiration)

  Время в секундах, в течение которого действует код авторизации.

  Значение по умолчанию – `600` (10 минут).

- **Ассоциация организаций с публичными системами аутентификации** (Social Auth Organization Map)
- **Ассоциация команд с публичными системами** (Social Auth Team Map)
- **Поля для идентификации пользователя через публичные сервисы аутентификации** (Social Auth User Fields)
- **Use Email address for usernames**

  Если эта настройка включена, для аутентификации в качестве имени пользователя используется его адрес электронной почты.

  Значение по умолчанию – выключено.

- **Minimum number of characters in local password**

  Минимальное количество буквенных символов в паролях, используемых для аутентификации через встроенную систему.

  Значение по умолчанию – `0`.

- **Minimum number of digit characters in local password**

  Минимальное количество цифр в паролях, используемых для аутентификации через встроенную систему.

  Значение по умолчанию – `0`.

- **Minimum number of uppercase characters in local password**

  Минимальное количество буквенных символов верхнего регистра в паролях, используемых для аутентификации через встроенную систему.

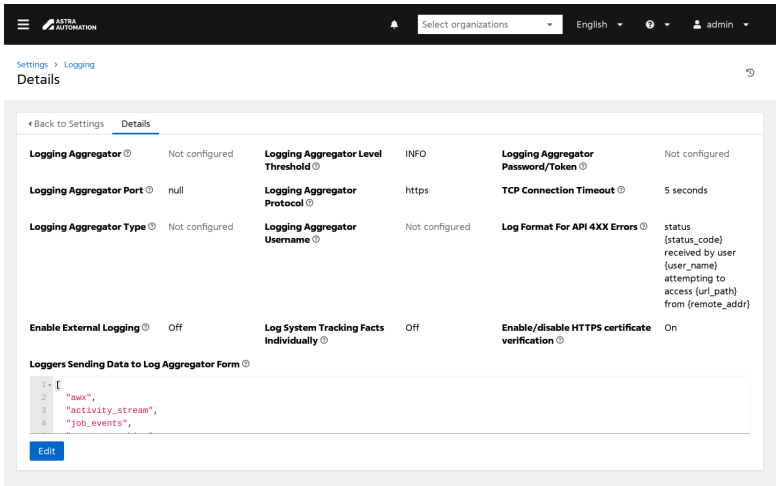
  Значение по умолчанию – `0`.

- **Minimum number of special characters in local password**

  Минимальное количество специальных символов в паролях, используемых для аутентификации через встроенную систему.

  Значение по умолчанию – `0`.

Настройки журналирования



Окно **Настройки журналирования** (Logging settings) содержит настройки, управляющие ведением журнала контроллера:

- **Включить внешнее ведение журнала** (Enable External Logging)

  Если эта настройка включена, для хранения журналов используется внешняя система журналирования.

  Примечание: Для включения этой настройки необходимо задать значения настроек **Агрегатор протоколирования** и **Тип агрегатора ведения журнала**.

- **Агрегатор протоколирования** (Logging Aggregator)

  FQDN или IP-адрес сервера, на котором размещена внешняя система журналирования.

- **Порт агрегатора ведения журнала** (Logging Aggregator Port)

  Номер порта, через который осуществляется подключение к внешней системе журналирования.

- **Тип агрегатора ведения журнала** (Logging Aggregator Type)

  Тип внешней системы журналирования. Поддерживаются следующие значения:

  - `---------` – внутренняя система журналирования;
  - `logstash` – Elastic Logstash;
  - `splunk` – Splunk;
  - `loggly` – Solarwinds Loggly;
  - `sumologic` – Sumo Logic;
  - `other` – внешняя система журналирования с настройками, заданными вручную.

- **Имя пользователя агрегатора ведения журнала** (Logging Aggregator Username) – название учетной записи, используемой для доступа к внешней системе журналирования.

- **Пароль/токен агрегатора ведения журнала** (Logging Aggregator Password/Token)

  Пароль или токен для доступа к внешней системе журналирования.

- **Вести индивидуальный журнал по фактам, собранным системой** (Log System Tracking Facts Individually)

  Если этот параметр включен, данные системного отслеживания фактов будут отправляться для каждого пакета, сервиса или другой сущности, найденной при сканировании, что обеспечивает большую детализацию поискового запроса. Если этот параметр

выключен, факты будут отправляться в виде единого словаря, что обеспечивает большую эффективность обработки фактов.

Значение по умолчанию – выключено.

- **Протокол ведения журналов** (Logging Aggregator Protocol)

  Протокол соединения с внешней системов журналирования. Поддерживаются следующие значения:

  - `HTTPS/HTTP`.

    Примечание: При выборе этого значения используется протокол HTTPS, если только протокол HTTP не указан явно в значении настройки **Агрегатор протоколирования** (Logging Aggregator).

  - `TCP`.
  - `UDP`.

  Значение по умолчанию – `HTTPS/HTTP`.

- **Порог уровня агрегатора ведения журнала** (Logging Aggregator Level Threshold)

  Минимальный уровень сообщения, необходимый для его записи в журнал. Уровни сообщений, от наименьшего к наибольшему:

  1. `DEBUG`;
  2. `INFO`;
  3. `WARNING`;
  4. `ERROR`;
  5. `CRITICAL`.

  Сообщения с уровнем ниже указанного будут пропущены обработчиком журнала.

  Примечание: Действие этого параметра не распространяется на сообщения из категории `awx.analytics`.

  Значение по умолчанию – `INFO`.

- **Тайм-аут TCP-соединения** (TCP Connection Timeout)

  Время в секундах, по истечении которого соединение с внешней системой журналирования разрывается. Значение этой настройки используется только если для настройки **Протокол ведения журналов** задано значение `HTTPS/HTTP` или `TCP`.

  Значение по умолчанию – 5.

- **Включить/отключить проверку сертификата HTTPS** (Enable/disable HTTPS certificate verification)

  Если эта настройка включена, при соединении с внешней системой журналирования по протоколу HTTPS выполняется проверка сертификата.

  Значение по умолчанию – включено.

- **Регистраторы, отправляющие данные в форму агрегатора журналов** (Loggers Sending Data to Log Aggregator Form)

  Список типов регистраторов, отправляющих данные в агрегатор журналов.

  Поддерживаются следующие значения:

  - `awx` – внутренние события контроллера;
  - `activity_stream` – изменение ресурсов;
  - `job_events` – события, связанные с заданиями;

- `system_tracking` – факты, собранные при выполнении заданий.

  Примечание: Для сбора фактов необходимо включить настройку **Включить хранилище фактов** (Enable Fact Storage) в свойствах шаблона задания.

- `broadcast_websocket` – ошибки широковещательных рассылок веб-сокетов.

Значение по умолчанию:

```
[
    "awx",
    "activity_strem",
    "job_events",
    "system_tracking",
    "broadcast_websocket"
]
```

- **Log Format For API 4XX Errors**

  Шаблон записей о событиях, связанных с кодами HTTP 4XX при обращении к конечным точкам API.

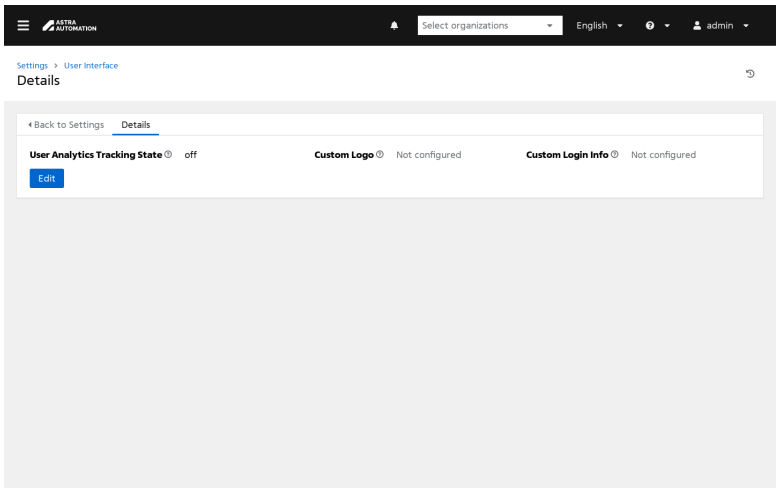
  В шаблоне поддерживается использование следующих переменных:

  - `status_code` – код статуса HTTP;
  - `user_name` – название учетной записи пользователя, выполнившего запрос к конечной точке API;
  - `url_path` – URL конечной точки API, обращение к которой привело к ошибке;
  - `remote_addr` – удаленный адрес, отображаемый для ошибки пользователя;
  - `error` – сообщение об ошибке, возвращенное конечной точкой API.

  Если конечная точка API не вернула текст сообщения об ошибке, используется строковое название статуса HTTP.

Пользовательский интерфейс

Окно **Пользовательский интерфейс** (User Interface) содержит настройки, управляющие внешним видом страницы аутентификации.



Форма **Редактировать** (Edit) содержит следующие поля:

- **Пользовательская информация для входа в систему** (Custom Login Info) – текст, выводимый на странице аутентификации. Поддерживаются форматы Plain Text и HTML.

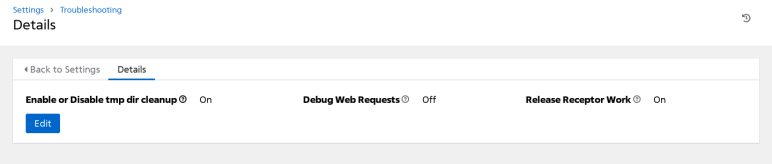
- **Пользовательский логотип** (Custom Logo) – изображение, отображаемое на странице аутентификации.

  Поддерживаются следующие форматы:

  - GIF;
  - JPEG;
  - PNG (рекомендуется).

Диагностика проблем

Окно **Диагностика проблем** (Troubleshooting) содержит настройки, управляющие отладкой.



Форма **Редактировать** (Edit) содержит следующие настройки:

- **Включить или отключить очистку каталога tmp** (Enable or Disable tmp dir cleanup)

  Если эта настройка включена, происходит автоматическая очистка временных каталогов, используемых контроллером. Это предотвращает накопление временных файлов, которые могут занимать значительное дисковое пространство. Отключение настройки может быть полезно для отладки и анализа временных файлов, но требует ручного управления их удалением.

  Значение по умолчанию – включено.

- **Debug Web Requests**

  Если эта настройка включена, происходит запись в журнал сообщений о всех веб-запросах. Это полезно при диагностике и анализе проблем, связанных с веб-запросами. Отключение настройки снижает нагрузку на систему.

  Значение по умолчанию – выключено.

- **Release Receptor Work**

  Если эта настройка включена, служба рецепторов освобождает ресурсы по мере завершения заданий.

  Значение по умолчанию – включено.

7.4 Программный интерфейс (API)

В этом разделе приводятся инструкции по управлению Automation Controller с помощью программного интерфейса – API (Application Programming Interface).

Для взаимодействия с API достаточно отправлять HTTP-запросы к определенным точкам доступа API (API endpoint), которые предоставляют доступ к нужным данным или выполняют определенные действия.

API используется для интеграции между различными приложениями, создания расширений, автоматизации задач, доступа к сторонним сервисам и многих других целей. API позволяет различным системам и приложениям обмениваться данными и ресурсами, чтобы улучшить их функциональность и эффективность.

Преимущества использования API:

- Масштабируемость – поддерживает горизонтальное масштабирование, что позволяет расширять его для обработки больших объемов запросов и данных.

- Независимость от языка и платформы – может быть использован с любым языком программирования и на любой платформе, что делает его универсальным и легко интегрируемым.
- Безопасность – использует стандартные методы аутентификации, такие как OAuth, что обеспечивает безопасность передачи данных.
- Открытость – имеет открытую структуру, что позволяет разработчикам легко получать доступ к различным данным и ресурсам.

7.4.1 Общий синтаксис

В этом документе рассматривается синтаксис запросов к API и получаемых ответов.

Структура запроса

Запрос API состоит из следующих элементов:

```
<method> <URI>[<query>]
   [<headers>]
   [{<body>}]
```

- <method> – указывает, какого рода действие нужно выполнить с ресурсом:
  - GET – получение данных об объекте (ресурсе) или данных в виде списка объектов по указанному URI.
  - POST – создание нового объекта или выполнение команды контроллера.
  - PUT и PATCH – обновление указанного объекта.
  - DELETE – удаление объекта.
- <URI> – указывает на конкретный ресурс, с которым необходимо выполнить действие. При необходимости к URI добавляют параметры запроса (query string).
- <headers> – заголовки HTTP содержат дополнительные метаданные запроса, такие как:
  - тип контента;
  - параметры авторизации;
  - язык;
  - формат.
- <body> – тело запроса содержит данные, которые нужно передать на сервер.

Например, в результате следующего запроса будет создан пользователь с именем <your_username> и паролем <your_password>:

```
POST https://<address>/api/v2/users/ HTTP/1.1
Content-Type: application/json
Authorization: Bearer <your_token>

   {
      "username": "<your_username>",
      "password": "<your_password>"
   }
```

Здесь:

- POST – указывает на то, что новый объект должен быть создан на сервере.
- `https://<address>/api/v2/users/` – указывает на конкретный ресурс, к которому обращается запрос. В данном случае запрос отправляется к ресурсу `users` в рамках API.

- `HTTP/1.1` – название и версия протокола HTTP, используемые для связи между клиентом и сервером.
- `Content-Type: application/json` – заголовок, указывающий на то, что данные отправляются в формате JSON.
- `Authorization: Bearer <your_token>` – заголовок, указывающий на использование bearer token для авторизации и содержащий сам токен.

Структура ответа

Ответ на HTTP-запрос состоит из следующих частей:

```
<status_line>
   [<headers>]
   [{<body>}]
```

- <status_line> – статусная строка. Содержит название и номер версии используемого протокола, код статуса и его краткое описание. Самыми распространенными кодами статуса являются:
  - 200 – запрос успешно выполнен.
  - 201 – в результате запроса был создан новый объект.
  - 202 – запрос получен, но еще не обработан.
  - 301 – URL запрошенного ресурса был изменен навсегда. Новый URL будет указан в ответе.
  - 302 – URL запрошенного ресурса был временно изменен.
  - 400 – запрос был сформирован с ошибками.
  - 403 – нет прав доступа к запрошенному ресурсу.
  - 404 – запрошенный ресурс не существует.
  - 500 – на сервере произошла ошибка, в результате которой он не может успешно обработать запрос.
- <headers> – заголовки, содержащие сведения об ответе. Могут включать информацию о типе содержимого, кешировании, дате и времени, сервере, длине тела ответа и другие сведения.
- <body> – тело ответа, содержащее данные, отправленные от сервера к клиенту в ответ на запрос. Может содержать данные различных типов, например, текст, JSON и так далее, в зависимости от запроса и целей обмена информацией.

Пример ответа на запрос о создании пользователя:

```
HTTP/1.1 201 Created
Content-Type: application/json

   {
   "id": 7,
   "type": "user",
   "url": "/api/v2/users/7/",
   "related": {
      "named_url": "/api/v2/users/<your_username>/",
      "teams": "/api/v2/users/7/teams/",
      "organizations": "/api/v2/users/7/organizations/",
      "admin_of_organizations": "/api/v2/users/7/admin_of_organizations/",
      "projects": "/api/v2/users/7/projects/",
      "credentials": "/api/v2/users/7/credentials/",
      "roles": "/api/v2/users/7/roles/",
      "activity_stream": "/api/v2/users/7/activity_stream/",
      "access_list": "/api/v2/users/7/access_list/",
```

(continues on next page)

(продолжение с предыдущей страницы)

```
        "tokens": "/api/v2/users/7/tokens/",
        "authorized_tokens": "/api/v2/users/7/authorized_tokens/",
        "personal_tokens": "/api/v2/users/7/personal_tokens/"
    },
    "summary_fields": {
        "user_capabilities": {
              "edit": true,
              "delete": true
        }
    },
    "created": "2024-03-28T22:53:18.939698Z",
    "modified": null,
    "username": "<your_username>",
    "first_name": "",
    "last_name": "",
    "email": "",
    "is_superuser": false,
    "is_system_auditor": false,
    "ldap_dn": "",
    "last_login": null,
    "external_account": null,
    "auth": []
    }
```

Здесь:

- `id` – идентификатор созданного пользователя;
- `type` – тип пользователя;
- `url` – идентификатор ресурса;
- `related` – связанные ссылки на другие ресурсы, такие как команды, организации, проекты, роли и другие;
- `summaryfields` – сводные поля с информацией о возможностях пользователя, в данном случае: редактирование, удаление;
- `created` – дата и время создания пользователя;
- `modified` – дата и время последней модификации пользователя;
- `username` – название учетной записи пользователя;
- `firstname` – имя пользователя;
- `lastname` – фамилия пользователя;
- `email` – электронная почта пользователя;
- `issuperuser` – сведения о том, является ли пользователь системным администратором;
- `issystemauditor` – сведения о том, является ли пользователь системным аудитором;
- `ldapdn` – уникальное имя пользователя в системе LDAP;
- `lastlogin` – дата последнего входа пользователя;
- `externalaccount` – внешний аккаунт;
- `auth` – данные аутентификации пользователя.

7.4.2 Методы авторизации

API Automation Controller поддерживает авторизацию следующими способами:

- авторизация сеанса (session authorization);
- базовая авторизация (basic authorization);
- авторизация через токен OAuth 2 (OAuth 2 token authorization).

Примечание: Для доступа к API рекомендуется использовать авторизацию через токен OAuth 2.

Авторизация сеанса

Авторизация сеанса используется при входе в API или пользовательский интерфейс Automation Controller. С помощью утилиты `curl` можно увидеть активность, которая происходит при входе в Automation Controller:

1. Получите X-CSRFToken с помощью команды:

```
curl -k -c - https://<address>/api/login/
```

Пример вывода:

```
192.168.56.11   FALSE   /       TRUE    0       csrftoken       n97Hk5MjncJ45qHYmPFoqx8dLy8MPQMtfOafEZZlO9oPTJIztoD8qXRVuJY1OVsQ
```

Здесь:

- FALSE – указывает, что cookie не требует безопасного соединения (HTTPS);
- `/` – URI, к которому относится установленный cookie;
- TRUE – указывает, что cookie доступен для всех поддоменов этого домена;
- `0` – срок действия cookie (в секундах) отсутствует, поэтому cookie действителен только в рамках текущей сессии;
- `csrftoken` – название cookie;
- `n97Hk5MjncJ45qHYmPFoqx8dLy8MPQMtfOafEZZlO9oPTJIztoD8qXRVuJY1OVsQ` – CSRF-токен.

2. Выполните запрос авторизации:

```
curl -X POST -H 'Content-Type: application/x-www-form-urlencoded' \
     --referer https://<address>/api/login/ \
     -H 'X-CSRFToken: <your_CSRFToken>' \
     --data 'username=<username>&password=<password>' \
     --cookie 'csrftoken=<your_CSRFToken>' \
     https://<address>/api/login/ -k -D - -o /dev/null
```

Здесь:

- <your_CSRFToken> – CSRF-токен, полученный ранее;
- <username> – название учетной записи пользователя;
- <password> – пароль.

Если пользователь успешно авторизован, сервер возвращает ответ, в заголовке которого в поле `X-API-Session-Cookie-Name` указывает название ключа, определяющего идентификатор сеанса связи, в составе cookie. По умолчанию значение этого параметра равно `awx_sessionid`.

Пример ответа:

```
Server: nginx
Date: Mon, 21 Mar 2024 18:19:04 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 0
Connection: keep-alive
Location: /accounts/profile/
X-API-Session-Cookie-Name: awx_sessionid
Expires: Mon, 21 Mar 2024 18:19:04 GMT
Cache-Control: max-age=0, no-cache, no-store, must-revalidate, private
Vary: Cookie, Accept-Language, Origin
Session-Timeout: 1800
Content-Language: en
X-API-Total-Time: 0.888s
X-API-Request-Id: 0d9f18d32e6d40efb6e0427353c073d6
Access-Control-Expose-Headers: X-API-Request-Id
Set-Cookie: userLoggedIn=true; Path=/
Set-Cookie:␣
↪csrftoken=ZG2vbmRfTDksShZVBoHltV7pRtCpZcK9G2WBir1hsgOq6vpsM7IRvH9QmjSSBDBm; Path=/
↪; SameSite=Lax; Secure
Set-Cookie: awx_sessionid=89174mqdxcd2etvhp5umdysjxpr5fxdu; expires=Mon, 21 Mar␣
↪2024 18:49:04 GMT; HttpOnly; Max-Age=1800; Path=/; SameSite=Lax; Secure
Strict-Transport-Security: max-age=63072000
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
Cache-Control: no-cache, no-store, must-revalidate
Expires: 0
Pragma: no-cache
```

Базовая авторизация

При использовании базовой авторизации состояние не сохраняется. Поэтому название учетной записи пользователя (username) и пароль (password) в формате Base64 должны отправляться вместе с каждым запросом через заголовок авторизации. Это применимо для локальных записей и для учетных записей LDAP.

Пример запроса:

```
curl -X GET --user '<username>:<password>' https://<address>/api/v2/credentials -k -L
```

В целях безопасности вы можете отключить базовую авторизацию в графическом интерфейсе. Для этого выполните следующие действия:

1. Перейдите в раздел **Настройки** (Settings).
2. На панели **Система** (System) нажмите на ссылку *Общие параметры аутентификации* (Miscellaneous Authentication settings).
3. Нажмите кнопку *Редактировать* (Edit).
4. Выключите флаг **Включить HTTP Basic Auth** (Enable HTTP Basic Auth).
5. Нажмите кнопку *Сохранить* (Save).

Авторизация через токен OAuth 2

Авторизация через токен OAuth 2 обычно используется при программном взаимодействии с API Automation Controller. Как и при базовой авторизации, токен OAuth 2 предоставляется с каждым запросом API через заголовок авторизации. Токены имеют настраиваемый срок действия и при необходимости могут быть отозваны администратором для одного пользователя или для всей системы. По умолчанию внешним пользователям, например, созданным с помощью единого входа, не разрешено создавать токены OAuth 2 в целях безопасности.

Чтобы разрешить внешним пользователям создавать токены OAuth 2, выполните следующие действия в графическом интерфейсе контроллера:

1. Перейдите в раздел **Настройки** (Settings).
2. На панели **Система** (System) нажмите на ссылку *Общие параметры аутентификации* (Miscellaneous Authentication settings).
3. Нажмите кнопку *Редактировать* (Edit).
4. Включите флаг **Разрешить внешним пользователям создавать токены OAuth2** (Allow External Users to Create OAuth2 Tokens).
5. Нажмите кнопку *Сохранить* (Save).

Создать токен доступа OAuth 2 можно с помощью следующих средств:

- графический интерфейс;
- запрос API.

Создание токена в графическом интерфейсе

Пошаговые инструкции по управлению токенами через графический интерфейс доступны в секции *Управление токенами*.

Создание токена с помощью запроса

Создайте токен с помощью команды:

```
curl -u <username>:<password> -k -X POST https://<address>/api/v2/tokens/
```

Пример вывода команды:

```
{
   "id": 6,
   "type": "o_auth2_access_token",
   "url": "/api/v2/tokens/6/",
   "related": {
      "user": "/api/v2/users/1/",
      "activity_stream": "/api/v2/tokens/6/activity_stream/"
   },
   "summary_fields": {
      "user": {
         "id": 1,
         "username": "admin",
         "first_name": "",
         "last_name": ""
      }
   },
   "created": "2024-03-21T20:53:09.035133Z",
   "modified": "2024-03-21T20:53:09.041729Z",
   "description": "",
   "user": 1,
```

(continues on next page)

(продолжение с предыдущей страницы)

```
   "token": "<your_token>",
   "refresh_token": null,
   "application": null,
   "expires": "3023-07-23T20:53:09.030794Z",
   "scope": "write"
}
```

Здесь <your_token> – токен авторизации, который теперь можно использовать для выполнения запроса, например:

```
curl -k -X POST \
   https://<address>/api/v2/hosts/ \
   -H “Content-Type: application/json” \
   -H “Authorization: Bearer <your_token>`
```

7.4.3 Клиенты API

Для взаимодействия с контроллером по API можно использовать следующие типы клиентов:

- браузер;
- утилиту curl[154];
- специальные приложения, часто называемые REST API clients[155];
- расширения для браузера (например, Postman[156]);
- программу на любом языке программирования (например, Python[157]).

Браузер

Примечание: Доступ к API с помощью браузера возможен только после прохождения авторизации в графическом интерфейсе контроллера.

В адресной строке браузера введите `https://<address>/api/v2/`, где <address> – *FQDN* или IP-адрес контроллера. При выборе любой точки доступа API (API endpoint) отображается страница с результатом GET-запроса к ней. Например, по адресу `https://<address>/api/v2/` вы увидите результат запроса на получение списка всех доступных точек доступа. А по адресу `https://<address>/api/v2/ping/` вы увидите результат выполнения команды `GET /api/v2/ping/`, которая используется для проверки доступности сервера через API.

Утилита curl

Чтобы улучшить формат вывода JSON, в дополнение к утилите `curl` установите утилиту `jq`, с помощью команды:

```
sudo apt install jq
```

Для составления curl-запроса используйте шаблон:

```
curl -X GET --user <username>:<password> https://<address>/api/v2/<api_endpoint> -k -s |␣
↪jq .
```

Здесь:

[154] https://curl.se/
[155] https://hevodata.com/learn/rest-clients/
[156] https://www.postman.com/
[157] https://www.python.org/

- `-X GET` – метод HTTP, который вы хотите использовать в своем запросе. GET используется для запроса данных с сервера.
- `--user <username>:<password>` – ключ, в котором необходимо указать название учетной записи пользователя и пароль для аутентификации на сервере.
- `https://<address>/api/v2/<api_endpoint>` – URI, на который отправляется запрос.
- <address> – FQDN или IP-адрес контроллера.
- <api_endpoint> – точка доступа API (URI), к которой делается запрос.
- `-k` или `--insecure` – флаг, позволяющий утилите `curl` игнорировать проверки сертификата SSL. Это может быть полезно для тестов или в случаях, когда используются самоподписанные сертификаты.
- `-s` или `--silent` – флаг, отключающий индикаторы прогресса и сообщения об ошибках.
- `jq` – утилита для фильтрации и преобразования данных JSON.

Примеры использования curl

Обращение к корневой точке доступа API:

```
curl -X GET -k https://<address>/api/ | jq .
```

Пример вывода команды:

```
{
   "description": "AWX REST API",
   "current_version": "/api/v2/",
   "available_versions": {
      "v2": "/api/v2/"
   },
   "oauth2": "/api/o/",
   "custom_logo": "",
   "custom_login_info": "",
   "login_redirect_override": ""
}
```

Получение списка всех точек доступа:

```
curl -X GET -k https://<address>/api/v2/ | jq .
```

Пример вывода команды:

```
{
  "ping": "/api/v2/ping/",
  "instances": "/api/v2/instances/",
  "instance_groups": "/api/v2/instance_groups/",
  "config": "/api/v2/config/",
  "settings": "/api/v2/settings/",
  "me": "/api/v2/me/",
  "dashboard": "/api/v2/dashboard/",
  "organizations": "/api/v2/organizations/",
  "users": "/api/v2/users/",
  "execution_environments": "/api/v2/execution_environments/",
  "projects": "/api/v2/projects/",
  "project_updates": "/api/v2/project_updates/",
  "teams": "/api/v2/teams/",
  "credentials": "/api/v2/credentials/",
  "credential_types": "/api/v2/credential_types/",
  "credential_input_sources": "/api/v2/credential_input_sources/",
  "applications": "/api/v2/applications/",
```

(continues on next page)

(продолжение с предыдущей страницы)

```
  "tokens": "/api/v2/tokens/",
  "metrics": "/api/v2/metrics/",
  "inventory": "/api/v2/inventories/",
  "constructed_inventory": "/api/v2/constructed_inventories/",
  "inventory_sources": "/api/v2/inventory_sources/",
  "inventory_updates": "/api/v2/inventory_updates/",
  "groups": "/api/v2/groups/",
  "hosts": "/api/v2/hosts/",
  "host_metrics": "/api/v2/host_metrics/",
  "job_templates": "/api/v2/job_templates/",
  "jobs": "/api/v2/jobs/",
  "ad_hoc_commands": "/api/v2/ad_hoc_commands/",
  "system_job_templates": "/api/v2/system_job_templates/",
  "system_jobs": "/api/v2/system_jobs/",
  "schedules": "/api/v2/schedules/",
  "roles": "/api/v2/roles/",
  "notification_templates": "/api/v2/notification_templates/",
  "notifications": "/api/v2/notifications/",
  "labels": "/api/v2/labels/",
  "unified_job_templates": "/api/v2/unified_job_templates/",
  "unified_jobs": "/api/v2/unified_jobs/",
  "activity_stream": "/api/v2/activity_stream/",
  "workflow_job_templates": "/api/v2/workflow_job_templates/",
  "workflow_jobs": "/api/v2/workflow_jobs/",
  "workflow_approvals": "/api/v2/workflow_approvals/",
  "workflow_job_template_nodes": "/api/v2/workflow_job_template_nodes/",
  "workflow_job_nodes": "/api/v2/workflow_job_nodes/",
  "mesh_visualizer": "/api/v2/mesh_visualizer/",
  "bulk": "/api/v2/bulk/",
  "analytics": "/api/v2/analytics/"
}
```

Получение списка шаблонов заданий:

```
curl -X GET --user <admin>:<password> -k https://<address>/api/v2/job_templates/ | jq .
```

Пример вывода команды:

```
{
  "count": 2,
  "next": null,
  "previous": null,
  "results": [
    {
      "id": 7,
      "type": "job_template",
      "url": "/api/v2/job_templates/7/",
      "related": {
        "created_by": "/api/v2/users/1/",
        "modified_by": "/api/v2/users/1/",
        "labels": "/api/v2/job_templates/7/labels/",
        "inventory": "/api/v2/inventories/1/",
        "project": "/api/v2/projects/6/",
        "organization": "/api/v2/organizations/1/",
        "credentials": "/api/v2/job_templates/7/credentials/",
        "last_job": "/api/v2/jobs/18/",
        "jobs": "/api/v2/job_templates/7/jobs/",
        "schedules": "/api/v2/job_templates/7/schedules/",
        "activity_stream": "/api/v2/job_templates/7/activity_stream/",
        "launch": "/api/v2/job_templates/7/launch/",
        "webhook_key": "/api/v2/job_templates/7/webhook_key/",
        "webhook_receiver": "",
```

(continues on next page)

(продолжение с предыдущей страницы)


```
        "notification_templates_started": "/api/v2/job_templates/7/notification_
↪templates_started/",
        "notification_templates_success": "/api/v2/job_templates/7/notification_
↪templates_success/",
        "notification_templates_error": "/api/v2/job_templates/7/notification_templates_
↪error/",
        "access_list": "/api/v2/job_templates/7/access_list/",
        "survey_spec": "/api/v2/job_templates/7/survey_spec/",
        "object_roles": "/api/v2/job_templates/7/object_roles/",
        "instance_groups": "/api/v2/job_templates/7/instance_groups/",
        "slice_workflow_jobs": "/api/v2/job_templates/7/slice_workflow_jobs/",
        "copy": "/api/v2/job_templates/7/copy/"
      },
      "summary_fields": {
        "organization": {
          "id": 1,
          "name": "Default",
          "description": ""
        },
        "inventory": {
          "id": 1,
          "name": "Demo Inventory",
          "description": "",
          "has_active_failures": false,
          "total_hosts": 1,
          "hosts_with_active_failures": 0,
          "total_groups": 0,
          "has_inventory_sources": false,
          "total_inventory_sources": 0,
          "inventory_sources_with_failures": 0,
          "organization_id": 1,
          "kind": ""
        },
        "project": {
          "id": 6,
          "name": "Demo Project",
          "description": "",
          "status": "successful",
          "scm_type": "git",
          "allow_override": false
        },
        "last_job": {
          "id": 18,
          "name": "Demo Job Template",
          "description": "",
          "finished": "2024-03-15T12:21:31.648339Z",
          "status": "successful",
          "failed": false
        },
        "last_update": {
          "id": 18,
          "name": "Demo Job Template",
          "description": "",
          "status": "successful",
          "failed": false
        },
        "created_by": {
          "id": 1,
          "username": "admin",
          "first_name": "",
          "last_name": ""
        },
        "modified_by": {
          "id": 1,
```


(continues on next page)

(продолжение с предыдущей страницы)


```
          "username": "admin",
          "first_name": "",
          "last_name": ""
        },
        "object_roles": {
          "admin_role": {
            "description": "Can manage all aspects of the job template",
            "name": "Admin",
            "id": 37
          },
          "execute_role": {
            "description": "May run the job template",
            "name": "Execute",
            "id": 38
          },
          "read_role": {
            "description": "May view settings for the job template",
            "name": "Read",
            "id": 39
          }
        },
        "user_capabilities": {
          "edit": true,
          "delete": true,
          "start": true,
          "schedule": true,
          "copy": true
        },
        "labels": {
          "count": 0,
          "results": []
        },
        "recent_jobs": [
          {
            "id": 18,
            "status": "successful",
            "finished": "2024-03-15T12:21:31.648339Z",
            "canceled_on": null,
            "type": "job"
          },
          {
            "id": 1,
            "status": "successful",
            "finished": "2024-02-16T00:15:25.600100Z",
            "canceled_on": null,
            "type": "job"
          }
        ],
        "credentials": [
          {
            "id": 1,
            "name": "Demo Credential",
            "description": "",
            "kind": "ssh",
            "cloud": false
          }
        ]
      },
      "created": "2024-02-16T00:06:03.362775Z",
      "modified": "2024-02-16T00:06:03.362794Z",
      "name": "Demo Job Template",
      "description": "",
      "job_type": "run",
      "inventory": 1,
```


(continues on next page)

(продолжение с предыдущей страницы)


```
      "project": 6,
      "playbook": "hello_world.yml",
      "scm_branch": "",
      "forks": 0,
      "limit": "",
      "verbosity": 0,
      "extra_vars": "",
      "job_tags": "",
      "force_handlers": false,
      "skip_tags": "",
      "start_at_task": "",
      "timeout": 0,
      "use_fact_cache": false,
      "organization": 1,
      "last_job_run": "2024-03-15T12:21:31.648339Z",
      "last_job_failed": false,
      "next_job_run": null,
      "status": "successful",
      "execution_environment": null,
      "host_config_key": "",
      "ask_scm_branch_on_launch": false,
      "ask_diff_mode_on_launch": false,
      "ask_variables_on_launch": false,
      "ask_limit_on_launch": false,
      "ask_tags_on_launch": false,
      "ask_skip_tags_on_launch": false,
      "ask_job_type_on_launch": false,
      "ask_verbosity_on_launch": false,
      "ask_inventory_on_launch": false,
      "ask_credential_on_launch": false,
      "ask_execution_environment_on_launch": false,
      "ask_labels_on_launch": false,
      "ask_forks_on_launch": false,
      "ask_job_slice_count_on_launch": false,
      "ask_timeout_on_launch": false,
      "ask_instance_groups_on_launch": false,
      "survey_enabled": false,
      "become_enabled": false,
      "diff_mode": false,
      "allow_simultaneous": false,
      "custom_virtualenv": null,
      "job_slice_count": 1,
      "webhook_service": "",
      "webhook_credential": null,
      "prevent_instance_group_fallback": false
    },
    {
      "id": 9,
      "type": "job_template",
      "url": "/api/v2/job_templates/9/",
      "related": {
        "created_by": "/api/v2/users/1/",
        "modified_by": "/api/v2/users/1/",
        "labels": "/api/v2/job_templates/9/labels/",
        "inventory": "/api/v2/inventories/2/",
        "project": "/api/v2/projects/8/",
        "organization": "/api/v2/organizations/1/",
        "credentials": "/api/v2/job_templates/9/credentials/",
        "last_job": "/api/v2/jobs/17/",
        "jobs": "/api/v2/job_templates/9/jobs/",
        "schedules": "/api/v2/job_templates/9/schedules/",
        "activity_stream": "/api/v2/job_templates/9/activity_stream/",
        "launch": "/api/v2/job_templates/9/launch/",
        "webhook_key": "/api/v2/job_templates/9/webhook_key/",
```


(continues on next page)

(продолжение с предыдущей страницы)


```
        "webhook_receiver": "",
        "notification_templates_started": "/api/v2/job_templates/9/notification_
↪templates_started/",
        "notification_templates_success": "/api/v2/job_templates/9/notification_
↪templates_success/",
        "notification_templates_error": "/api/v2/job_templates/9/notification_templates_
↪error/",
        "access_list": "/api/v2/job_templates/9/access_list/",
        "survey_spec": "/api/v2/job_templates/9/survey_spec/",
        "object_roles": "/api/v2/job_templates/9/object_roles/",
        "instance_groups": "/api/v2/job_templates/9/instance_groups/",
        "slice_workflow_jobs": "/api/v2/job_templates/9/slice_workflow_jobs/",
        "copy": "/api/v2/job_templates/9/copy/"
      },
      "summary_fields": {
        "organization": {
          "id": 1,
          "name": "Default",
          "description": ""
        },
        "inventory": {
          "id": 2,
          "name": "NGINX inventory",
          "description": "",
          "has_active_failures": true,
          "total_hosts": 1,
          "hosts_with_active_failures": 1,
          "total_groups": 0,
          "has_inventory_sources": false,
          "total_inventory_sources": 0,
          "inventory_sources_with_failures": 0,
          "organization_id": 1,
          "kind": ""
        },
        "project": {
          "id": 8,
          "name": "Развертывание NGINX",
          "description": "",
          "status": "successful",
          "scm_type": "git",
          "allow_override": false
        },
        "last_job": {
          "id": 17,
          "name": "Развертывание NGINX",
          "description": "",
          "finished": "2024-03-15T12:11:55.017427Z",
          "status": "failed",
          "failed": true
        },
        "last_update": {
          "id": 17,
          "name": "Развертывание NGINX",
          "description": "",
          "status": "failed",
          "failed": true
        },
        "created_by": {
          "id": 1,
          "username": "admin",
          "first_name": "",
          "last_name": ""
        },
        "modified_by": {
```


(continues on next page)

(продолжение с предыдущей страницы)


```
          "id": 1,
          "username": "admin",
          "first_name": "",
          "last_name": ""
        },
        "object_roles": {
          "admin_role": {
            "description": "Can manage all aspects of the job template",
            "name": "Admin",
            "id": 64
          },
          "execute_role": {
            "description": "May run the job template",
            "name": "Execute",
            "id": 65
          },
          "read_role": {
            "description": "May view settings for the job template",
            "name": "Read",
            "id": 66
          }
        },
        "user_capabilities": {
          "edit": true,
          "delete": true,
          "start": true,
          "schedule": true,
          "copy": true
        },
        "labels": {
          "count": 0,
          "results": []
        },
        "recent_jobs": [
          {
            "id": 17,
            "status": "failed",
            "finished": "2024-03-15T12:11:55.017427Z",
            "canceled_on": null,
            "type": "job"
          }
        ],
        "credentials": []
      },
      "created": "2024-03-15T12:11:27.168941Z",
      "modified": "2024-03-15T12:11:27.168956Z",
      "name": "Развертывание NGINX",
      "description": "",
      "job_type": "run",
      "inventory": 2,
      "project": 8,
      "playbook": "nginx/site.yml",
      "scm_branch": "",
      "forks": 0,
      "limit": "",
      "verbosity": 0,
      "extra_vars": "---",
      "job_tags": "",
      "force_handlers": false,
      "skip_tags": "",
      "start_at_task": "",
      "timeout": 0,
      "use_fact_cache": false,
      "organization": 1,
```


(continues on next page)

(продолжение с предыдущей страницы)

```
      "last_job_run": "2024-03-15T12:11:55.017427Z",
      "last_job_failed": true,
      "next_job_run": null,
      "status": "failed",
      "execution_environment": null,
      "host_config_key": "",
      "ask_scm_branch_on_launch": false,
      "ask_diff_mode_on_launch": false,
      "ask_variables_on_launch": false,
      "ask_limit_on_launch": false,
      "ask_tags_on_launch": false,
      "ask_skip_tags_on_launch": false,
      "ask_job_type_on_launch": false,
      "ask_verbosity_on_launch": false,
      "ask_inventory_on_launch": false,
      "ask_credential_on_launch": false,
      "ask_execution_environment_on_launch": false,
      "ask_labels_on_launch": false,
      "ask_forks_on_launch": false,
      "ask_job_slice_count_on_launch": false,
      "ask_timeout_on_launch": false,
      "ask_instance_groups_on_launch": false,
      "survey_enabled": false,
      "become_enabled": false,
      "diff_mode": false,
      "allow_simultaneous": false,
      "custom_virtualenv": null,
      "job_slice_count": 1,
      "webhook_service": "",
      "webhook_credential": null,
      "prevent_instance_group_fallback": false
    }
  ]
}
```

Запуск задания из шаблона:

```
curl -X POST --user <admin>:<password> -k https://<address>/api/v2/job_templates/7/
↪launch/ | jq .
```

Пример вывода команды:

```
{
  "job": 26,
  "ignored_fields": {},
  "id": 26,
  "type": "job",
  "url": "/api/v2/jobs/26/",
  "related": {
    "created_by": "/api/v2/users/1/",
    "modified_by": "/api/v2/users/1/",
    "labels": "/api/v2/jobs/26/labels/",
    "inventory": "/api/v2/inventories/1/",
    "project": "/api/v2/projects/6/",
    "organization": "/api/v2/organizations/1/",
    "credentials": "/api/v2/jobs/26/credentials/",
    "unified_job_template": "/api/v2/job_templates/7/",
    "stdout": "/api/v2/jobs/26/stdout/",
    "job_events": "/api/v2/jobs/26/job_events/",
    "job_host_summaries": "/api/v2/jobs/26/job_host_summaries/",
    "activity_stream": "/api/v2/jobs/26/activity_stream/",
    "notifications": "/api/v2/jobs/26/notifications/",
```

(continues on next page)

(продолжение с предыдущей страницы)


```
    "create_schedule": "/api/v2/jobs/26/create_schedule/",
    "job_template": "/api/v2/job_templates/7/",
    "cancel": "/api/v2/jobs/26/cancel/",
    "relaunch": "/api/v2/jobs/26/relaunch/"
  },
  "summary_fields": {
    "organization": {
      "id": 1,
      "name": "Default",
      "description": ""
    },
    "inventory": {
      "id": 1,
      "name": "Demo Inventory",
      "description": "",
      "has_active_failures": false,
      "total_hosts": 1,
      "hosts_with_active_failures": 0,
      "total_groups": 0,
      "has_inventory_sources": false,
      "total_inventory_sources": 0,
      "inventory_sources_with_failures": 0,
      "organization_id": 1,
      "kind": ""
    },
    "project": {
      "id": 6,
      "name": "Demo Project",
      "description": "",
      "status": "successful",
      "scm_type": "git",
      "allow_override": false
    },
    "job_template": {
      "id": 7,
      "name": "Demo Job Template",
      "description": ""
    },
    "unified_job_template": {
      "id": 7,
      "name": "Demo Job Template",
      "description": "",
      "unified_job_type": "job"
    },
    "created_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "modified_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "user_capabilities": {
      "delete": true,
      "start": true
    },
    "labels": {
      "count": 0,
      "results": []
    },
```


(continues on next page)

(продолжение с предыдущей страницы)


```
    "credentials": [
      {
        "id": 1,
        "name": "Demo Credential",
        "description": "",
        "kind": "ssh",
        "cloud": false
      }
    ]
  },
  "created": "2024-03-19T06:41:24.347473Z",
  "modified": "2024-03-19T06:41:24.389125Z",
  "name": "Demo Job Template",
  "description": "",
  "job_type": "run",
  "inventory": 1,
  "project": 6,
  "playbook": "hello_world.yml",
  "scm_branch": "",
  "forks": 0,
  "limit": "",
  "verbosity": 0,
  "extra_vars": "{}",
  "job_tags": "",
  "force_handlers": false,
  "skip_tags": "",
  "start_at_task": "",
  "timeout": 0,
  "use_fact_cache": false,
  "organization": 1,
  "unified_job_template": 7,
  "launch_type": "manual",
  "status": "pending",
  "execution_environment": null,
  "failed": false,
  "started": null,
  "finished": null,
  "canceled_on": null,
  "elapsed": 0,
  "job_args": "",
  "job_cwd": "",
  "job_env": {},
  "job_explanation": "",
  "execution_node": "",
  "controller_node": "",
  "result_traceback": "",
  "event_processing_finished": false,
  "launched_by": {
    "id": 1,
    "name": "admin",
    "type": "user",
    "url": "/api/v2/users/1/"
  },
  "work_unit_id": null,
  "job_template": 7,
  "passwords_needed_to_start": [],
  "allow_simultaneous": false,
  "artifacts": {},
  "scm_revision": "",
  "instance_group": null,
  "diff_mode": false,
  "job_slice_number": 0,
  "job_slice_count": 1,
  "webhook_service": "",
```


(continues on next page)

(продолжение с предыдущей страницы)

```
  "webhook_credential": null,
  "webhook_guid": ""
}
```

Получение токена авторизации:

```
curl -X POST \
https://<address>/api/v2/tokens/ \
-H 'Content-Type: application/json' \
-d '{
      "description": "Token description",
      "application": null,
      "user": "admin",
      "scope": "write"
   }' \
-u '<admin>:<password>' | jq .
```

Пример вывода команды:

```
{
  "id": 2,
  "type": "o_auth2_access_token",
  "url": "/api/v2/tokens/2/",
  "related": {
    "user": "/api/v2/users/1/",
    "activity_stream": "/api/v2/tokens/2/activity_stream/"
  },
  "summary_fields": {
    "user": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    }
  },
  "created": "2024-03-19T06:55:01.493262Z",
  "modified": "2024-03-19T06:55:01.506786Z",
  "description": "Token description",
  "user": 1,
  "token": "<your_token>",
  "refresh_token": null,
  "application": null,
  "expires": "3023-07-27T06:55:01.487431Z",
  "scope": "write"
}
```

Здесь <your_token> – токен авторизации.

Проверка статуса задания, запущенного ранее:

```
curl -X GET \
https://<address>/api/v2/jobs/<jobs_id>/ \
-H 'Authorization: Bearer <your_token>' \
-k -s | jq .
```

Здесь <jobs_id> – идентификатор задания, запущенного ранее.

Пример вывода команды:


```
{
  "id": 26,
  "type": "job",
  "url": "/api/v2/jobs/26/",
  "related": {
    "created_by": "/api/v2/users/1/",
    "labels": "/api/v2/jobs/26/labels/",
    "inventory": "/api/v2/inventories/1/",
    "project": "/api/v2/projects/6/",
    "organization": "/api/v2/organizations/1/",
    "credentials": "/api/v2/jobs/26/credentials/",
    "unified_job_template": "/api/v2/job_templates/7/",
    "stdout": "/api/v2/jobs/26/stdout/",
    "execution_environment": "/api/v2/execution_environments/1/",
    "job_events": "/api/v2/jobs/26/job_events/",
    "job_host_summaries": "/api/v2/jobs/26/job_host_summaries/",
    "activity_stream": "/api/v2/jobs/26/activity_stream/",
    "notifications": "/api/v2/jobs/26/notifications/",
    "create_schedule": "/api/v2/jobs/26/create_schedule/",
    "job_template": "/api/v2/job_templates/7/",
    "cancel": "/api/v2/jobs/26/cancel/",
    "relaunch": "/api/v2/jobs/26/relaunch/"
  },
  "summary_fields": {
    "organization": {
      "id": 1,
      "name": "Default",
      "description": ""
    },
    "inventory": {
      "id": 1,
      "name": "Demo Inventory",
      "description": "",
      "has_active_failures": false,
      "total_hosts": 1,
      "hosts_with_active_failures": 0,
      "total_groups": 0,
      "has_inventory_sources": false,
      "total_inventory_sources": 0,
      "inventory_sources_with_failures": 0,
      "organization_id": 1,
      "kind": ""
    },
    "execution_environment": {
      "id": 1,
      "name": "Default execution environment",
      "description": "",
      "image": "registry.astralinux.ru/aa/aa-base-ee:0.2.1"
    },
    "project": {
      "id": 6,
      "name": "Demo Project",
      "description": "",
      "status": "successful",
      "scm_type": "git",
      "allow_override": false
    },
    "job_template": {
      "id": 7,
      "name": "Demo Job Template",
      "description": ""
    },
    "unified_job_template": {
```


(continues on next page)

(продолжение с предыдущей страницы)


```
      "id": 7,
      "name": "Demo Job Template",
      "description": "",
      "unified_job_type": "job"
    },
    "instance_group": {
      "id": 1,
      "name": "controlplane",
      "is_container_group": false
    },
    "created_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "user_capabilities": {
      "delete": true,
      "start": true
    },
    "labels": {
      "count": 0,
      "results": []
    },
    "credentials": [
      {
        "id": 1,
        "name": "Demo Credential",
        "description": "",
        "kind": "ssh",
        "cloud": false
      }
    ]
  },
  "created": "2024-03-19T06:41:24.347473Z",
  "modified": "2024-03-19T06:41:24.596205Z",
  "name": "Demo Job Template",
  "description": "",
  "job_type": "run",
  "inventory": 1,
  "project": 6,
  "playbook": "hello_world.yml",
  "scm_branch": "",
  "forks": 0,
  "limit": "",
  "verbosity": 0,
  "extra_vars": "{}",
  "job_tags": "",
  "force_handlers": false,
  "skip_tags": "",
  "start_at_task": "",
  "timeout": 0,
  "use_fact_cache": false,
  "organization": 1,
  "unified_job_template": 7,
  "launch_type": "manual",
  "status": "successful",
  "execution_environment": 1,
  "failed": false,
  "started": "2024-03-19T06:41:24.694396Z",
  "finished": "2024-03-19T06:41:28.962513Z",
  "canceled_on": null,
  "elapsed": 4.268,
  "job_args": "[\"podman\", \"run\", \"--rm\", \"--tty\", \"--interactive\", \"--workdir\
```


(continues on next page)

(продолжение с предыдущей страницы)


```
↪", \"/runner/project\", \"-v\", \"/tmp/awx_26_b59w6psk/:/runner/:Z\", \"--env-file\", \
↪"/tmp/awx_26_b59w6psk/artifacts/26/env.list\", \"--quiet\", \"--name\", \"ansible_
↪runner_26\", \"--user=root\", \"--network\", \"slirp4netns:enable_ipv6=true\", \
↪"registry.astralinux.ru/aa/aa-base-ee:0.2.1\", \"ansible-playbook\", \"-u\", \"admin\",
↪ \"-i\", \"/runner/inventory/hosts\", \"-e\", \"@/runner/env/extravars\", \"hello_
↪world.yml\"]",
  "job_cwd": "/runner/project",
  "job_env": {
    "ANSIBLE_UNSAFE_WRITES": "1",
    "AWX_ISOLATED_DATA_DIR": "/runner/artifacts/26",
    "ANSIBLE_FORCE_COLOR": "True",
    "ANSIBLE_HOST_KEY_CHECKING": "False",
    "ANSIBLE_INVENTORY_UNPARSED_FAILED": "True",
    "ANSIBLE_PARAMIKO_RECORD_HOST_KEYS": "False",
    "HOME": "/var/lib/awx",
    "AWX_PRIVATE_DATA_DIR": "/tmp/awx_26_b59w6psk",
    "JOB_ID": "26",
    "INVENTORY_ID": "1",
    "PROJECT_REVISION": "347e44fea036c94d5f60e544de006453ee5c71ad",
    "ANSIBLE_RETRY_FILES_ENABLED": "False",
    "MAX_EVENT_RES": "700000",
    "AWX_HOST": "https://controller",
    "ANSIBLE_SSH_CONTROL_PATH_DIR": "/runner/cp",
    "ANSIBLE_COLLECTIONS_PATHS": "/runner/requirements_collections:~/.ansible/
↪collections:/usr/share/ansible/collections",
    "ANSIBLE_ROLES_PATH": "/runner/requirements_roles:~/.ansible/roles:/usr/share/
↪ansible/roles:/etc/ansible/roles",
    "ANSIBLE_CALLBACK_PLUGINS": "/runner/artifacts/26/callback",
    "ANSIBLE_STDOUT_CALLBACK": "awx_display",
    "RUNNER_OMIT_EVENTS": "False",
    "RUNNER_ONLY_FAILED_EVENTS": "False"
  },
  "job_explanation": "",
  "execution_node": "192.168.56.11",
  "controller_node": "192.168.56.11",
  "result_traceback": "",
  "event_processing_finished": true,
  "launched_by": {
    "id": 1,
    "name": "admin",
    "type": "user",
    "url": "/api/v2/users/1/"
  },
  "work_unit_id": "Vt4HYQqA",
  "job_template": 7,
  "passwords_needed_to_start": [],
  "allow_simultaneous": false,
  "artifacts": {},
  "scm_revision": "347e44fea036c94d5f60e544de006453ee5c71ad",
  "instance_group": 1,
  "diff_mode": false,
  "job_slice_number": 0,
  "job_slice_count": 1,
  "webhook_service": "",
  "webhook_credential": null,
  "webhook_guid": "",
  "host_status_counts": {
    "ok": 1
  },
  "playbook_counts": {
    "play_count": 1,
    "task_count": 2
  },
  "custom_virtualenv": null
```


(continues on next page)

(продолжение с предыдущей страницы)

```
}
```

Создание проекта:

```
curl -X POST \
https://<address>/api/v2/projects/ \
-H 'Authorization: Bearer <your_token>' \
-H 'Content-Type: application/json' \
-d '{
   "name": "New Project",
   "organization": 1,
   "scm_type": "git",
   "scm_url": "https://github.com/ansible/ansible-tower-samples"
}' \
-k -s | jq .
```

Пример вывода команды:

```
{
  "id": 13,
  "type": "project",
  "url": "/api/v2/projects/13/",
  "related": {
    "named_url": "/api/v2/projects/New Project++Default/",
    "created_by": "/api/v2/users/1/",
    "modified_by": "/api/v2/users/1/",
    "current_job": "/api/v2/project_updates/29/",
    "teams": "/api/v2/projects/13/teams/",
    "playbooks": "/api/v2/projects/13/playbooks/",
    "inventory_files": "/api/v2/projects/13/inventories/",
    "update": "/api/v2/projects/13/update/",
    "project_updates": "/api/v2/projects/13/project_updates/",
    "scm_inventory_sources": "/api/v2/projects/13/scm_inventory_sources/",
    "schedules": "/api/v2/projects/13/schedules/",
    "activity_stream": "/api/v2/projects/13/activity_stream/",
    "notification_templates_started": "/api/v2/projects/13/notification_templates_
↪started/",
    "notification_templates_success": "/api/v2/projects/13/notification_templates_
↪success/",
    "notification_templates_error": "/api/v2/projects/13/notification_templates_error/",
    "access_list": "/api/v2/projects/13/access_list/",
    "object_roles": "/api/v2/projects/13/object_roles/",
    "copy": "/api/v2/projects/13/copy/",
    "organization": "/api/v2/organizations/1/",
    "current_update": "/api/v2/project_updates/29/"
  },
  "summary_fields": {
    "organization": {
      "id": 1,
      "name": "Default",
      "description": ""
    },
    "current_update": {
      "id": 29,
      "name": "New Project",
      "description": "",
      "status": "pending",
      "failed": false
    },
    "current_job": {
      "id": 29,
      "name": "New Project",
```

(continues on next page)

(продолжение с предыдущей страницы)


```
      "description": "",
      "status": "pending",
      "failed": false
    },
    "created_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "modified_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "object_roles": {
      "admin_role": {
        "description": "Can manage all aspects of the project",
        "name": "Admin",
        "id": 79
      },
      "use_role": {
        "description": "Can use the project in a job template",
        "name": "Use",
        "id": 80
      },
      "update_role": {
        "description": "May update the project",
        "name": "Update",
        "id": 81
      },
      "read_role": {
        "description": "May view settings for the project",
        "name": "Read",
        "id": 82
      }
    },
    "user_capabilities": {
      "edit": true,
      "delete": true,
      "start": true,
      "schedule": true,
      "copy": true
    }
  },
  "created": "2024-03-19T14:26:16.980872Z",
  "modified": "2024-03-19T14:26:16.980886Z",
  "name": "New Project",
  "description": "",
  "local_path": "_13__new_project",
  "scm_type": "git",
  "scm_url": "https://github.com/ansible/ansible-tower-samples",
  "scm_branch": "",
  "scm_refspec": "",
  "scm_clean": false,
  "scm_track_submodules": false,
  "scm_delete_on_update": false,
  "credential": null,
  "timeout": 0,
  "scm_revision": "",
  "last_job_run": null,
  "last_job_failed": false,
  "next_job_run": null,
```


(continues on next page)

(продолжение с предыдущей страницы)

```
  "status": "pending",
  "organization": 1,
  "scm_update_on_launch": false,
  "scm_update_cache_timeout": 0,
  "allow_override": false,
  "custom_virtualenv": null,
  "default_environment": null,
  "signature_validation_credential": null,
  "last_update_failed": false,
  "last_updated": null
}
```

Изменение проекта, созданного ранее:

```
curl -X PATCH \
https://<address>/api/v2/projects/<project_id>/ \
-H 'Authorization: Bearer <your_token>' \
-H 'Content-Type: application/json' \
-d '{
   "name": "Updated Project",
   "scm_url": "https://github.com/ansible/ansible-examples"
}' \
-k -s | jq .
```

Здесь <project_id> – идентификатор проекта, созданного ранее.

Пример вывода команды:

```
{
  "id": 13,
  "type": "project",
  "url": "/api/v2/projects/13/",
  "related": {
    "named_url": "/api/v2/projects/Updated Project++Default/",
    "created_by": "/api/v2/users/1/",
    "modified_by": "/api/v2/users/1/",
    "current_job": "/api/v2/project_updates/30/",
    "last_job": "/api/v2/project_updates/29/",
    "teams": "/api/v2/projects/13/teams/",
    "playbooks": "/api/v2/projects/13/playbooks/",
    "inventory_files": "/api/v2/projects/13/inventories/",
    "update": "/api/v2/projects/13/update/",
    "project_updates": "/api/v2/projects/13/project_updates/",
    "scm_inventory_sources": "/api/v2/projects/13/scm_inventory_sources/",
    "schedules": "/api/v2/projects/13/schedules/",
    "activity_stream": "/api/v2/projects/13/activity_stream/",
    "notification_templates_started": "/api/v2/projects/13/notification_templates_
↪started/",
    "notification_templates_success": "/api/v2/projects/13/notification_templates_
↪success/",
    "notification_templates_error": "/api/v2/projects/13/notification_templates_error/",
    "access_list": "/api/v2/projects/13/access_list/",
    "object_roles": "/api/v2/projects/13/object_roles/",
    "copy": "/api/v2/projects/13/copy/",
    "organization": "/api/v2/organizations/1/",
    "current_update": "/api/v2/project_updates/30/",
    "last_update": "/api/v2/project_updates/29/"
  },
  "summary_fields": {
    "organization": {
      "id": 1,
      "name": "Default",
```

(continues on next page)

(продолжение с предыдущей страницы)


```
      "description": ""
    },
    "last_job": {
      "id": 29,
      "name": "New Project",
      "description": "",
      "finished": "2024-03-19T14:26:21.930637Z",
      "status": "successful",
      "failed": false
    },
    "last_update": {
      "id": 29,
      "name": "New Project",
      "description": "",
      "status": "successful",
      "failed": false
    },
    "current_update": {
      "id": 30,
      "name": "Updated Project",
      "description": "",
      "status": "pending",
      "failed": false
    },
    "current_job": {
      "id": 30,
      "name": "Updated Project",
      "description": "",
      "status": "pending",
      "failed": false
    },
    "created_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "modified_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "object_roles": {
      "admin_role": {
        "description": "Can manage all aspects of the project",
        "name": "Admin",
        "id": 79
      },
      "use_role": {
        "description": "Can use the project in a job template",
        "name": "Use",
        "id": 80
      },
      "update_role": {
        "description": "May update the project",
        "name": "Update",
        "id": 81
      },
      "read_role": {
        "description": "May view settings for the project",
        "name": "Read",
        "id": 82
      }
```


(continues on next page)

(продолжение с предыдущей страницы)


```
    },
    "user_capabilities": {
      "edit": true,
      "delete": true,
      "start": true,
      "schedule": true,
      "copy": true
    },
    "resolved_environment": {
      "id": 1,
      "name": "Default execution environment",
      "description": "",
      "image": "registry.astralinux.ru/aa/aa-base-ee:0.2.1"
    }
  },
  "created": "2024-03-19T14:26:16.980872Z",
  "modified": "2024-03-19T14:28:32.951463Z",
  "name": "Updated Project",
  "description": "",
  "local_path": "_13__new_project",
  "scm_type": "git",
  "scm_url": "https://github.com/ansible/ansible-examples",
  "scm_branch": "",
  "scm_refspec": "",
  "scm_clean": false,
  "scm_track_submodules": false,
  "scm_delete_on_update": false,
  "credential": null,
  "timeout": 0,
  "scm_revision": "347e44fea036c94d5f60e544de006453ee5c71ad",
  "last_job_run": "2024-03-19T14:26:21.930637Z",
  "last_job_failed": false,
  "next_job_run": null,
  "status": "pending",
  "organization": 1,
  "scm_update_on_launch": false,
  "scm_update_cache_timeout": 0,
  "allow_override": false,
  "custom_virtualenv": null,
  "default_environment": null,
  "signature_validation_credential": null,
  "last_update_failed": false,
  "last_updated": "2024-03-19T14:26:21.930637Z"
}
```


Удаление проекта:

```
curl -X DELETE \
https://<address>/api/v2/projects/<project_id>/ \
-H 'Authorization: Bearer <your-token>'
```

Расширения для браузера

Применение расширения для браузера Google Chrome выглядит следующим образом:

1. Откройте Chrome Web Store и установите расширение Postman на ваш браузер.
2. Запустите Postman из списка расширений в вашем браузере.
3. Создайте новый запрос.
4. Введите URI объекта, который вы хотите запросить, и выберите нужный метод HTTP (GET, POST, PUT, PATCH, DELETE).
5. Отправьте запрос.

6. После отправки запроса вы увидите ответ от сервера в формате JSON, XML или другом формате. Вы также можете просмотреть статус-код ответа и другие детали.

Программа на языке программирования

Рассмотрим в качестве примера скрипт Python, который выполняет следующие действия в контроллере:

1. Создает новый проект.
2. Получает информацию о всех проектах.
3. Получает информацию о созданном проекте.
4. Обновляет проект из источника.
5. Проверяет статус обновления проекта.
6. Изменяет проект.
7. Получает информацию о созданом проекте после изменения.
8. Удаляет проект.
9. Получает информацию о всех проектах.
10. Получает конфигурацию Automation Controller.

После команд, запускающих какое-либо действие на сервере, добавлена тридцатисекундная пауза для наглядности выполнения команд.

Для запуска скрипта необходимо:

- Использовать Python не ниже версии 3.8.
- Установить пакет `requests`.

```
"""Demo scenario"""

import json
import time
import requests

# Define base URL and headers
base_url = "https://<address>/api/v2/"
headers = {"Authorization": "Bearer <your_token>", "Content-type": "application/json"}


def button_press():
    """Prompt the user to press Enter to continue."""
    input("\nНажмите Enter чтобы продолжить...\n")


def print_json(response):
    """
    Print the response JSON returned by a REST API request.

    Args:
        response: The response object returned by the REST API request
    """
    if response.text:
        print(json.dumps(response.json(), indent=4))
    else:
        print("Status code:", response.status_code)


# Define the project data
project_data = {
    "name": "New Project",
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    "organization": 1,
    "scm_type": "git",
    "scm_url": "https://github.com/ansible/ansible-tower-samples",
}

# Post new project
response_post_new_project = requests.post(
    f"{base_url}projects/",
    headers=headers,
    data=json.dumps(project_data),
    verify=False,
    timeout=10,
)
print("\n--- Создание проекта ---")
print_json(response_post_new_project)
project_id = response_post_new_project.json()["id"]
time.sleep(30)
print("\n --- Проект создан ---")
button_press()

# Get all projects
response_get_all_projects = requests.get(
    f"{base_url}projects/", headers=headers, verify=False, timeout=10
)
print("\n--- Информация о проектах ---")
print_json(response_get_all_projects)
print("\n --- Выведена информация по всем проектам ---")
button_press()

# Get project details
response_get_project_detail = requests.get(
    f"{base_url}projects/{project_id}/", headers=headers, verify=False, timeout=10
)
print("\n--- Получение деталей проекта ---")
print_json(response_get_project_detail)
print("\n --- Выведена информация о созданном проекте ---")
button_press()

# Update project from the version control source
response_update_project = requests.post(
    f"{base_url}projects/{project_id}/update/",
    headers=headers,
    verify=False,
    timeout=10,
)
print("\n--- Обновление проекта из источника контроля версий ---")
print_json(response_update_project)
time.sleep(30)
print("\n --- Проект обновлен ---")
button_press()

# Check update status information
response_project_update_status = requests.get(
    f"{base_url}projects/{project_id}/project_updates/",
    headers=headers,
    verify=False,
    timeout=10,
)
print("\n--- Проверка информации о статусе обновления ---")
print_json(response_project_update_status)
print("\n --- Выведена информация о статусе обновления ---")
button_press()

# Change the project
```

(continues on next page)

(продолжение с предыдущей страницы)

```
patch_data = {
    "name": "Updated Project",
    "scm_url": "https://github.com/ansible/ansible-examples",
}
response_patch_project = requests.patch(
    f"{base_url}projects/{project_id}/",
    headers=headers,
    data=json.dumps(patch_data),
    verify=False,
    timeout=10,
)
print("\n--- Изменение проекта ---")
print_json(response_patch_project)
time.sleep(30)
print("\n --- Проект изменён ---")
button_press()

# Get project details after the change
response_get_project_detail_after_change = requests.get(
    f"{base_url}projects/{project_id}/", headers=headers, verify=False, timeout=10
)
print("\n--- Получение деталей проекта (после обновления) ---")
print_json(response_get_project_detail_after_change)
print("\n --- Выведена информация о созданном проекте после изменения ---")
button_press()

# Delete the project
response_delete_project = requests.delete(
    f"{base_url}projects/{project_id}/", headers=headers, verify=False, timeout=10
)
print("\n--- Удаление проекта ---")
print_json(response_delete_project)
print("\n --- Проект удалён ---")
button_press()

# List all projects after deletion
response_get_all_projects_after_deletion = requests.get(
    f"{base_url}projects/", headers=headers, verify=False, timeout=10
)
print("\n--- Список проектов (после удаления) ---")
print_json(response_get_all_projects_after_deletion)
print("\n --- Выведена информация по всем проектам после удаления ---")
button_press()

# Get Automation Controller Configuration
response_get_configuration = requests.get(
    f"{base_url}config/", headers=headers, verify=False, timeout=10
)
print("\n--- Получение конфигурации Automation Controller ---")
print_json(response_get_configuration)
print("\n --- Выведена информация по серверу Automation Controller ---")
```

В демонстрационном скрипте замените значения <address> и <your_token> на свои.

Пример вывода:

Создание проекта:


```
{
  "id": 15,
  "type": "project",
  "url": "/api/v2/projects/15/",
  "related": {
    "named_url": "/api/v2/projects/New Project++Default/",
    "created_by": "/api/v2/users/1/",
    "modified_by": "/api/v2/users/1/",
    "current_job": "/api/v2/project_updates/35/",
    "teams": "/api/v2/projects/15/teams/",
    "playbooks": "/api/v2/projects/15/playbooks/",
    "inventory_files": "/api/v2/projects/15/inventories/",
    "update": "/api/v2/projects/15/update/",
    "project_updates": "/api/v2/projects/15/project_updates/",
    "scm_inventory_sources": "/api/v2/projects/15/scm_inventory_sources/",
    "schedules": "/api/v2/projects/15/schedules/",
    "activity_stream": "/api/v2/projects/15/activity_stream/",
    "notification_templates_started": "/api/v2/projects/15/notification_templates_
↪started/",
    "notification_templates_success": "/api/v2/projects/15/notification_templates_
↪success/",
    "notification_templates_error": "/api/v2/projects/15/notification_templates_error/",
    "access_list": "/api/v2/projects/15/access_list/",
    "object_roles": "/api/v2/projects/15/object_roles/",
    "copy": "/api/v2/projects/15/copy/",
    "organization": "/api/v2/organizations/1/",
    "current_update": "/api/v2/project_updates/35/"
  },
  "summary_fields": {
    "organization": {
      "id": 1,
      "name": "Default",
      "description": ""
    },
    "current_update": {
      "id": 35,
      "name": "New Project",
      "description": "",
      "status": "pending",
      "failed": false
    },
    "current_job": {
      "id": 35,
      "name": "New Project",
      "description": "",
      "status": "pending",
      "failed": false
    },
    "created_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "modified_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "object_roles": {
```


(continues on next page)

(продолжение с предыдущей страницы)


```
      "admin_role": {
        "description": "Can manage all aspects of the project",
        "name": "Admin",
        "id": 100
      },
      "use_role": {
        "description": "Can use the project in a job template",
        "name": "Use",
        "id": 101
      },
      "update_role": {
        "description": "May update the project",
        "name": "Update",
        "id": 102
      },
      "read_role": {
        "description": "May view settings for the project",
        "name": "Read",
        "id": 103
      }
    },
    "user_capabilities": {
      "edit": true,
      "delete": true,
      "start": true,
      "schedule": true,
      "copy": true
    }
  },
  "created": "2024-03-29T07:08:16.321624Z",
  "modified": "2024-03-29T07:08:16.321638Z",
  "name": "New Project",
  "description": "",
  "local_path": "_15__new_project",
  "scm_type": "git",
  "scm_url": "https://github.com/ansible/ansible-tower-samples",
  "scm_branch": "",
  "scm_refspec": "",
  "scm_clean": false,
  "scm_track_submodules": false,
  "scm_delete_on_update": false,
  "credential": null,
  "timeout": 0,
  "scm_revision": "",
  "last_job_run": null,
  "last_job_failed": false,
  "next_job_run": null,
  "status": "pending",
  "organization": 1,
  "scm_update_on_launch": false,
  "scm_update_cache_timeout": 0,
  "allow_override": false,
  "custom_virtualenv": null,
  "default_environment": null,
  "signature_validation_credential": null,
  "last_update_failed": false,
  "last_updated": null
}
```


Получение информации о проектах:


```
{
  "count": 3,
  "next": null,
```


(continues on next page)

(продолжение с предыдущей страницы)


```
  "previous": null,
  "results": [
    {
      "id": 6,
      "type": "project",
      "url": "/api/v2/projects/6/",
      "related": {
        "created_by": "/api/v2/users/1/",
        "modified_by": "/api/v2/users/1/",
        "teams": "/api/v2/projects/6/teams/",
        "playbooks": "/api/v2/projects/6/playbooks/",
        "inventory_files": "/api/v2/projects/6/inventories/",
        "update": "/api/v2/projects/6/update/",
        "project_updates": "/api/v2/projects/6/project_updates/",
        "scm_inventory_sources": "/api/v2/projects/6/scm_inventory_sources/",
        "schedules": "/api/v2/projects/6/schedules/",
        "activity_stream": "/api/v2/projects/6/activity_stream/",
        "notification_templates_started": "/api/v2/projects/6/notification_templates_
↪started/",
        "notification_templates_success": "/api/v2/projects/6/notification_templates_
↪success/",
        "notification_templates_error": "/api/v2/projects/6/notification_templates_error/
↪",
        "access_list": "/api/v2/projects/6/access_list/",
        "object_roles": "/api/v2/projects/6/object_roles/",
        "copy": "/api/v2/projects/6/copy/",
        "organization": "/api/v2/organizations/1/"
      },
      "summary_fields": {
        "organization": {
          "id": 1,
          "name": "Default",
          "description": ""
        },
        "created_by": {
          "id": 1,
          "username": "admin",
          "first_name": "",
          "last_name": ""
        },
        "modified_by": {
          "id": 1,
          "username": "admin",
          "first_name": "",
          "last_name": ""
        },
        "object_roles": {
          "admin_role": {
            "description": "Can manage all aspects of the project",
            "name": "Admin",
            "id": 22
          },
          "use_role": {
            "description": "Can use the project in a job template",
            "name": "Use",
            "id": 23
          },
          "update_role": {
            "description": "May update the project",
            "name": "Update",
            "id": 24
          },
          "read_role": {
            "description": "May view settings for the project",
```


(continues on next page)

(продолжение с предыдущей страницы)


```
            "name": "Read",
            "id": 25
          }
        },
        "user_capabilities": {
          "edit": true,
          "delete": true,
          "start": true,
          "schedule": true,
          "copy": true
        }
      },
      "created": "2024-02-16T00:06:02.670809Z",
      "modified": "2024-02-16T00:06:02.670834Z",
      "name": "Demo Project",
      "description": "",
      "local_path": "_6__demo_project",
      "scm_type": "git",
      "scm_url": "https://github.com/ansible/ansible-tower-samples",
      "scm_branch": "",
      "scm_refspec": "",
      "scm_clean": false,
      "scm_track_submodules": false,
      "scm_delete_on_update": false,
      "credential": null,
      "timeout": 0,
      "scm_revision": "347e44fea036c94d5f60e544de006453ee5c71ad",
      "last_job_run": null,
      "last_job_failed": false,
      "next_job_run": null,
      "status": "successful",
      "organization": 1,
      "scm_update_on_launch": false,
      "scm_update_cache_timeout": 0,
      "allow_override": false,
      "custom_virtualenv": null,
      "default_environment": null,
      "signature_validation_credential": null,
      "last_update_failed": false,
      "last_updated": null
    },
    {
      "id": 8,
      "type": "project",
      "url": "/api/v2/projects/8/",
      "related": {
        "created_by": "/api/v2/users/1/",
        "modified_by": "/api/v2/users/1/",
        "credential": "/api/v2/credentials/4/",
        "last_job": "/api/v2/project_updates/16/",
        "teams": "/api/v2/projects/8/teams/",
        "playbooks": "/api/v2/projects/8/playbooks/",
        "inventory_files": "/api/v2/projects/8/inventories/",
        "update": "/api/v2/projects/8/update/",
        "project_updates": "/api/v2/projects/8/project_updates/",
        "scm_inventory_sources": "/api/v2/projects/8/scm_inventory_sources/",
        "schedules": "/api/v2/projects/8/schedules/",
        "activity_stream": "/api/v2/projects/8/activity_stream/",
        "notification_templates_started": "/api/v2/projects/8/notification_templates_
↪started/",
        "notification_templates_success": "/api/v2/projects/8/notification_templates_
↪success/",
        "notification_templates_error": "/api/v2/projects/8/notification_templates_error/
↪",
```


(continues on next page)

(продолжение с предыдущей страницы)


```
        "access_list": "/api/v2/projects/8/access_list/",
        "object_roles": "/api/v2/projects/8/object_roles/",
        "copy": "/api/v2/projects/8/copy/",
        "organization": "/api/v2/organizations/1/",
        "last_update": "/api/v2/project_updates/16/"
      },
      "summary_fields": {
        "organization": {
          "id": 1,
          "name": "Default",
          "description": ""
        },
        "credential": {
          "id": 4,
          "name": "Automation Hub",
          "description": "",
          "kind": "scm",
          "cloud": false,
          "kubernetes": false,
          "credential_type_id": 2
        },
        "last_job": {
          "id": 16,
          "name": "\u0420\u0430\u0437\u0432\u0435\u0440\u0442\u044b\u0432\u0430\u043d\
↪u0438\u0435 NGINX",
          "description": "",
          "finished": "2024-03-15T12:09:44.270752Z",
          "status": "successful",
          "failed": false
        },
        "last_update": {
          "id": 16,
          "name": "\u0420\u0430\u0437\u0432\u0435\u0440\u0442\u044b\u0432\u0430\u043d\
↪u0438\u0435 NGINX",
          "description": "",
          "status": "successful",
          "failed": false
        },
        "created_by": {
          "id": 1,
          "username": "admin",
          "first_name": "",
          "last_name": ""
        },
        "modified_by": {
          "id": 1,
          "username": "admin",
          "first_name": "",
          "last_name": ""
        },
        "object_roles": {
          "admin_role": {
            "description": "Can manage all aspects of the project",
            "name": "Admin",
            "id": 60
          },
          "use_role": {
            "description": "Can use the project in a job template",
            "name": "Use",
            "id": 61
          },
          "update_role": {
            "description": "May update the project",
            "name": "Update",
```


(continues on next page)

(продолжение с предыдущей страницы)


```
            "id": 62
          },
          "read_role": {
            "description": "May view settings for the project",
            "name": "Read",
            "id": 63
          }
        },
        "user_capabilities": {
          "edit": true,
          "delete": true,
          "start": true,
          "schedule": true,
          "copy": true
        }
      },
      "created": "2024-03-15T11:57:40.851309Z",
      "modified": "2024-03-15T11:57:40.851323Z",
      "name": "\u0420\u0430\u0437\u0432\u0435\u0440\u0442\u044b\u0432\u0430\u043d\u0438\
↪u0435 NGINX",
      "description": "",
      "local_path": "_8__nginx",
      "scm_type": "git",
      "scm_url": "ssh://git@hub.astra-automation.ru:2222/aa-gca/AA/aac-samples.git",
      "scm_branch": "",
      "scm_refspec": "",
      "scm_clean": false,
      "scm_track_submodules": false,
      "scm_delete_on_update": false,
      "credential": 4,
      "timeout": 0,
      "scm_revision": "d7f0ff18cf8ffb44d64e8acd995402d545e2de90",
      "last_job_run": "2024-03-15T12:09:44.270752Z",
      "last_job_failed": false,
      "next_job_run": null,
      "status": "successful",
      "organization": 1,
      "scm_update_on_launch": false,
      "scm_update_cache_timeout": 0,
      "allow_override": false,
      "custom_virtualenv": null,
      "default_environment": null,
      "signature_validation_credential": null,
      "last_update_failed": false,
      "last_updated": "2024-03-15T12:09:44.270752Z"
    },
    {
      "id": 15,
      "type": "project",
      "url": "/api/v2/projects/15/",
      "related": {
        "created_by": "/api/v2/users/1/",
        "modified_by": "/api/v2/users/1/",
        "last_job": "/api/v2/project_updates/35/",
        "teams": "/api/v2/projects/15/teams/",
        "playbooks": "/api/v2/projects/15/playbooks/",
        "inventory_files": "/api/v2/projects/15/inventories/",
        "update": "/api/v2/projects/15/update/",
        "project_updates": "/api/v2/projects/15/project_updates/",
        "scm_inventory_sources": "/api/v2/projects/15/scm_inventory_sources/",
        "schedules": "/api/v2/projects/15/schedules/",
        "activity_stream": "/api/v2/projects/15/activity_stream/",
        "notification_templates_started": "/api/v2/projects/15/notification_templates_
↪started/",
```


(continues on next page)

(продолжение с предыдущей страницы)


```
        "notification_templates_success": "/api/v2/projects/15/notification_templates_
↪success/",
        "notification_templates_error": "/api/v2/projects/15/notification_templates_
↪error/",
        "access_list": "/api/v2/projects/15/access_list/",
        "object_roles": "/api/v2/projects/15/object_roles/",
        "copy": "/api/v2/projects/15/copy/",
        "organization": "/api/v2/organizations/1/",
        "last_update": "/api/v2/project_updates/35/"
      },
      "summary_fields": {
        "organization": {
          "id": 1,
          "name": "Default",
          "description": ""
        },
        "last_job": {
          "id": 35,
          "name": "New Project",
          "description": "",
          "finished": "2024-03-29T07:08:21.153077Z",
          "status": "successful",
          "failed": false
        },
        "last_update": {
          "id": 35,
          "name": "New Project",
          "description": "",
          "status": "successful",
          "failed": false
        },
        "created_by": {
          "id": 1,
          "username": "admin",
          "first_name": "",
          "last_name": ""
        },
        "modified_by": {
          "id": 1,
          "username": "admin",
          "first_name": "",
          "last_name": ""
        },
        "object_roles": {
          "admin_role": {
            "description": "Can manage all aspects of the project",
            "name": "Admin",
            "id": 100
          },
          "use_role": {
            "description": "Can use the project in a job template",
            "name": "Use",
            "id": 101
          },
          "update_role": {
            "description": "May update the project",
            "name": "Update",
            "id": 102
          },
          "read_role": {
            "description": "May view settings for the project",
            "name": "Read",
            "id": 103
          }
```


(continues on next page)

(продолжение с предыдущей страницы)


```
        },
        "user_capabilities": {
          "edit": true,
          "delete": true,
          "start": true,
          "schedule": true,
          "copy": true
        }
      },
      "created": "2024-03-29T07:08:16.321624Z",
      "modified": "2024-03-29T07:08:16.321638Z",
      "name": "New Project",
      "description": "",
      "local_path": "_15__new_project",
      "scm_type": "git",
      "scm_url": "https://github.com/ansible/ansible-tower-samples",
      "scm_branch": "",
      "scm_refspec": "",
      "scm_clean": false,
      "scm_track_submodules": false,
      "scm_delete_on_update": false,
      "credential": null,
      "timeout": 0,
      "scm_revision": "347e44fea036c94d5f60e544de006453ee5c71ad",
      "last_job_run": "2024-03-29T07:08:21.153077Z",
      "last_job_failed": false,
      "next_job_run": null,
      "status": "successful",
      "organization": 1,
      "scm_update_on_launch": false,
      "scm_update_cache_timeout": 0,
      "allow_override": false,
      "custom_virtualenv": null,
      "default_environment": null,
      "signature_validation_credential": null,
      "last_update_failed": false,
      "last_updated": "2024-03-29T07:08:21.153077Z"
    }
  ]
}
```


Получение деталей проекта:


```
{
  "id": 15,
  "type": "project",
  "url": "/api/v2/projects/15/",
  "related": {
    "named_url": "/api/v2/projects/New Project++Default/",
    "created_by": "/api/v2/users/1/",
    "modified_by": "/api/v2/users/1/",
    "last_job": "/api/v2/project_updates/35/",
    "teams": "/api/v2/projects/15/teams/",
    "playbooks": "/api/v2/projects/15/playbooks/",
    "inventory_files": "/api/v2/projects/15/inventories/",
    "update": "/api/v2/projects/15/update/",
    "project_updates": "/api/v2/projects/15/project_updates/",
    "scm_inventory_sources": "/api/v2/projects/15/scm_inventory_sources/",
    "schedules": "/api/v2/projects/15/schedules/",
    "activity_stream": "/api/v2/projects/15/activity_stream/",
    "notification_templates_started": "/api/v2/projects/15/notification_templates_
↪started/",
    "notification_templates_success": "/api/v2/projects/15/notification_templates_
↪success/",
```


(continues on next page)

(продолжение с предыдущей страницы)


```
    "notification_templates_error": "/api/v2/projects/15/notification_templates_error/",
    "access_list": "/api/v2/projects/15/access_list/",
    "object_roles": "/api/v2/projects/15/object_roles/",
    "copy": "/api/v2/projects/15/copy/",
    "organization": "/api/v2/organizations/1/",
    "last_update": "/api/v2/project_updates/35/"
  },
  "summary_fields": {
    "organization": {
      "id": 1,
      "name": "Default",
      "description": ""
    },
    "last_job": {
      "id": 35,
      "name": "New Project",
      "description": "",
      "finished": "2024-03-29T07:08:21.153077Z",
      "status": "successful",
      "failed": false
    },
    "last_update": {
      "id": 35,
      "name": "New Project",
      "description": "",
      "status": "successful",
      "failed": false
    },
    "created_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "modified_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "object_roles": {
      "admin_role": {
        "description": "Can manage all aspects of the project",
        "name": "Admin",
        "id": 100
      },
      "use_role": {
        "description": "Can use the project in a job template",
        "name": "Use",
        "id": 101
      },
      "update_role": {
        "description": "May update the project",
        "name": "Update",
        "id": 102
      },
      "read_role": {
        "description": "May view settings for the project",
        "name": "Read",
        "id": 103
      }
    },
    "user_capabilities": {
      "edit": true,
```


(continues on next page)

(продолжение с предыдущей страницы)


```
      "delete": true,
      "start": true,
      "schedule": true,
      "copy": true
    },
    "resolved_environment": {
      "id": 1,
      "name": "Default execution environment",
      "description": "",
      "image": "registry.astralinux.ru/aa/aa-base-ee:0.2.1"
    }
  },
  "created": "2024-03-29T07:08:16.321624Z",
  "modified": "2024-03-29T07:08:16.321638Z",
  "name": "New Project",
  "description": "",
  "local_path": "_15__new_project",
  "scm_type": "git",
  "scm_url": "https://github.com/ansible/ansible-tower-samples",
  "scm_branch": "",
  "scm_refspec": "",
  "scm_clean": false,
  "scm_track_submodules": false,
  "scm_delete_on_update": false,
  "credential": null,
  "timeout": 0,
  "scm_revision": "347e44fea036c94d5f60e544de006453ee5c71ad",
  "last_job_run": "2024-03-29T07:08:21.153077Z",
  "last_job_failed": false,
  "next_job_run": null,
  "status": "successful",
  "organization": 1,
  "scm_update_on_launch": false,
  "scm_update_cache_timeout": 0,
  "allow_override": false,
  "custom_virtualenv": null,
  "default_environment": null,
  "signature_validation_credential": null,
  "last_update_failed": false,
  "last_updated": "2024-03-29T07:08:21.153077Z"
}
```


Обновление проекта из источника контроля версий:


```
{
  "project_update": 36,
  "id": 36,
  "type": "project_update",
  "url": "/api/v2/project_updates/36/",
  "related": {
    "created_by": "/api/v2/users/1/",
    "modified_by": "/api/v2/users/1/",
    "unified_job_template": "/api/v2/projects/15/",
    "stdout": "/api/v2/project_updates/36/stdout/",
    "project": "/api/v2/projects/15/",
    "cancel": "/api/v2/project_updates/36/cancel/",
    "scm_inventory_updates": "/api/v2/project_updates/36/scm_inventory_updates/",
    "notifications": "/api/v2/project_updates/36/notifications/",
    "events": "/api/v2/project_updates/36/events/"
  },
  "summary_fields": {
    "organization": {
      "id": 1,
      "name": "Default",
```


(continues on next page)

(продолжение с предыдущей страницы)


```
      "description": ""
    },
    "project": {
      "id": 15,
      "name": "New Project",
      "description": "",
      "status": "pending",
      "scm_type": "git",
      "allow_override": false
    },
    "unified_job_template": {
      "id": 15,
      "name": "New Project",
      "description": "",
      "unified_job_type": "project_update"
    },
    "created_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "modified_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "user_capabilities": {
      "delete": true,
      "start": true
    }
  },
  "created": "2024-03-29T07:10:02.005597Z",
  "modified": "2024-03-29T07:10:02.013457Z",
  "name": "New Project",
  "description": "",
  "local_path": "_15__new_project",
  "scm_type": "git",
  "scm_url": "https://github.com/ansible/ansible-tower-samples",
  "scm_branch": "",
  "scm_refspec": "",
  "scm_clean": false,
  "scm_track_submodules": false,
  "scm_delete_on_update": false,
  "credential": null,
  "timeout": 0,
  "scm_revision": "",
  "unified_job_template": 15,
  "launch_type": "manual",
  "status": "pending",
  "execution_environment": null,
  "failed": false,
  "started": null,
  "finished": null,
  "canceled_on": null,
  "elapsed": 0.0,
  "job_args": "",
  "job_cwd": "",
  "job_env": {},
  "job_explanation": "",
  "execution_node": "",
  "result_traceback": "",
  "event_processing_finished": false,
```


(continues on next page)

(продолжение с предыдущей страницы)


```
  "launched_by": {
    "id": 1,
    "name": "admin",
    "type": "user",
    "url": "/api/v2/users/1/"
  },
  "work_unit_id": null,
  "project": 15,
  "job_type": "check",
  "job_tags": "update_git,install_roles,install_collections"
}
```


Проверка информации о статусе обновления:


```
{
  "count": 2,
  "next": null,
  "previous": null,
  "results": [
    {
      "id": 35,
      "type": "project_update",
      "url": "/api/v2/project_updates/35/",
      "related": {
        "created_by": "/api/v2/users/1/",
        "unified_job_template": "/api/v2/projects/15/",
        "stdout": "/api/v2/project_updates/35/stdout/",
        "execution_environment": "/api/v2/execution_environments/2/",
        "project": "/api/v2/projects/15/",
        "cancel": "/api/v2/project_updates/35/cancel/",
        "scm_inventory_updates": "/api/v2/project_updates/35/scm_inventory_updates/",
        "notifications": "/api/v2/project_updates/35/notifications/",
        "events": "/api/v2/project_updates/35/events/"
      },
      "summary_fields": {
        "organization": {
          "id": 1,
          "name": "Default",
          "description": ""
        },
        "execution_environment": {
          "id": 2,
          "name": "Control Plane Execution Environment",
          "description": "",
          "image": "registry.astralinux.ru/aa/aa-base-ee:0.2.1"
        },
        "project": {
          "id": 15,
          "name": "New Project",
          "description": "",
          "status": "successful",
          "scm_type": "git",
          "allow_override": false
        },
        "unified_job_template": {
          "id": 15,
          "name": "New Project",
          "description": "",
          "unified_job_type": "project_update"
        },
        "instance_group": {
          "id": 1,
          "name": "controlplane",
          "is_container_group": false
```


(continues on next page)

(продолжение с предыдущей страницы)


```
        },
        "created_by": {
          "id": 1,
          "username": "admin",
          "first_name": "",
          "last_name": ""
        },
        "user_capabilities": {
          "delete": true,
          "start": true
        }
      },
      "created": "2024-03-29T07:08:16.402906Z",
      "modified": "2024-03-29T07:08:16.813818Z",
      "name": "New Project",
      "description": "",
      "unified_job_template": 15,
      "launch_type": "manual",
      "status": "successful",
      "execution_environment": 2,
      "failed": false,
      "started": "2024-03-29T07:08:16.878716Z",
      "finished": "2024-03-29T07:08:21.153077Z",
      "canceled_on": null,
      "elapsed": 4.274,
      "job_explanation": "",
      "execution_node": "192.168.56.11",
      "launched_by": {
        "id": 1,
        "name": "admin",
        "type": "user",
        "url": "/api/v2/users/1/"
      },
      "work_unit_id": "kROVdZEb",
      "local_path": "_15__new_project",
      "scm_type": "git",
      "scm_url": "https://github.com/ansible/ansible-tower-samples",
      "scm_branch": "",
      "scm_refspec": "",
      "scm_clean": false,
      "scm_track_submodules": false,
      "scm_delete_on_update": false,
      "credential": null,
      "timeout": 0,
      "scm_revision": "347e44fea036c94d5f60e544de006453ee5c71ad",
      "project": 15,
      "job_type": "check",
      "job_tags": "update_git,install_roles,install_collections"
    },
    {
      "id": 36,
      "type": "project_update",
      "url": "/api/v2/project_updates/36/",
      "related": {
        "created_by": "/api/v2/users/1/",
        "unified_job_template": "/api/v2/projects/15/",
        "stdout": "/api/v2/project_updates/36/stdout/",
        "execution_environment": "/api/v2/execution_environments/2/",
        "project": "/api/v2/projects/15/",
        "cancel": "/api/v2/project_updates/36/cancel/",
        "scm_inventory_updates": "/api/v2/project_updates/36/scm_inventory_updates/",
        "notifications": "/api/v2/project_updates/36/notifications/",
        "events": "/api/v2/project_updates/36/events/"
      },
```


(continues on next page)

(продолжение с предыдущей страницы)


```
    "summary_fields": {
      "organization": {
        "id": 1,
        "name": "Default",
        "description": ""
      },
      "execution_environment": {
        "id": 2,
        "name": "Control Plane Execution Environment",
        "description": "",
        "image": "registry.astralinux.ru/aa/aa-base-ee:0.2.1"
      },
      "project": {
        "id": 15,
        "name": "New Project",
        "description": "",
        "status": "successful",
        "scm_type": "git",
        "allow_override": false
      },
      "unified_job_template": {
        "id": 15,
        "name": "New Project",
        "description": "",
        "unified_job_type": "project_update"
      },
      "instance_group": {
        "id": 1,
        "name": "controlplane",
        "is_container_group": false
      },
      "created_by": {
        "id": 1,
        "username": "admin",
        "first_name": "",
        "last_name": ""
      },
      "user_capabilities": {
        "delete": true,
        "start": true
      }
    },
    "created": "2024-03-29T07:10:02.005597Z",
    "modified": "2024-03-29T07:10:02.188768Z",
    "name": "New Project",
    "description": "",
    "unified_job_template": 15,
    "launch_type": "manual",
    "status": "successful",
    "execution_environment": 2,
    "failed": false,
    "started": "2024-03-29T07:10:02.256356Z",
    "finished": "2024-03-29T07:10:06.368318Z",
    "canceled_on": null,
    "elapsed": 4.112,
    "job_explanation": "",
    "execution_node": "192.168.56.11",
    "launched_by": {
      "id": 1,
      "name": "admin",
      "type": "user",
      "url": "/api/v2/users/1/"
    },
    "work_unit_id": "5bmmFdvg",
```


(continues on next page)

(продолжение с предыдущей страницы)


```
      "local_path": "_15__new_project",
      "scm_type": "git",
      "scm_url": "https://github.com/ansible/ansible-tower-samples",
      "scm_branch": "",
      "scm_refspec": "",
      "scm_clean": false,
      "scm_track_submodules": false,
      "scm_delete_on_update": false,
      "credential": null,
      "timeout": 0,
      "scm_revision": "347e44fea036c94d5f60e544de006453ee5c71ad",
      "project": 15,
      "job_type": "check",
      "job_tags": "update_git,install_roles,install_collections"
    }
  ]
}
```


Изменение проекта:


```
{
  "id": 15,
  "type": "project",
  "url": "/api/v2/projects/15/",
  "related": {
    "named_url": "/api/v2/projects/Updated Project++Default/",
    "created_by": "/api/v2/users/1/",
    "modified_by": "/api/v2/users/1/",
    "current_job": "/api/v2/project_updates/37/",
    "last_job": "/api/v2/project_updates/36/",
    "teams": "/api/v2/projects/15/teams/",
    "playbooks": "/api/v2/projects/15/playbooks/",
    "inventory_files": "/api/v2/projects/15/inventories/",
    "update": "/api/v2/projects/15/update/",
    "project_updates": "/api/v2/projects/15/project_updates/",
    "scm_inventory_sources": "/api/v2/projects/15/scm_inventory_sources/",
    "schedules": "/api/v2/projects/15/schedules/",
    "activity_stream": "/api/v2/projects/15/activity_stream/",
    "notification_templates_started": "/api/v2/projects/15/notification_templates_
↪started/",
    "notification_templates_success": "/api/v2/projects/15/notification_templates_
↪success/",
    "notification_templates_error": "/api/v2/projects/15/notification_templates_error/",
    "access_list": "/api/v2/projects/15/access_list/",
    "object_roles": "/api/v2/projects/15/object_roles/",
    "copy": "/api/v2/projects/15/copy/",
    "organization": "/api/v2/organizations/1/",
    "current_update": "/api/v2/project_updates/37/",
    "last_update": "/api/v2/project_updates/36/"
  },
  "summary_fields": {
    "organization": {
      "id": 1,
      "name": "Default",
      "description": ""
    },
    "last_job": {
      "id": 36,
      "name": "New Project",
      "description": "",
      "finished": "2024-03-29T07:10:06.368318Z",
      "status": "successful",
      "failed": false
    },
```


(continues on next page)

(продолжение с предыдущей страницы)


```
    "last_update": {
      "id": 36,
      "name": "New Project",
      "description": "",
      "status": "successful",
      "failed": false
    },
    "current_update": {
      "id": 37,
      "name": "Updated Project",
      "description": "",
      "status": "pending",
      "failed": false
    },
    "current_job": {
      "id": 37,
      "name": "Updated Project",
      "description": "",
      "status": "pending",
      "failed": false
    },
    "created_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "modified_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "object_roles": {
      "admin_role": {
        "description": "Can manage all aspects of the project",
        "name": "Admin",
        "id": 100
      },
      "use_role": {
        "description": "Can use the project in a job template",
        "name": "Use",
        "id": 101
      },
      "update_role": {
        "description": "May update the project",
        "name": "Update",
        "id": 102
      },
      "read_role": {
        "description": "May view settings for the project",
        "name": "Read",
        "id": 103
      }
    },
    "user_capabilities": {
      "edit": true,
      "delete": true,
      "start": true,
      "schedule": true,
      "copy": true
    },
    "resolved_environment": {
      "id": 1,
```


(continues on next page)

(продолжение с предыдущей страницы)


```
      "name": "Default execution environment",
      "description": "",
      "image": "registry.astralinux.ru/aa/aa-base-ee:0.2.1"
    }
  },
  "created": "2024-03-29T07:08:16.321624Z",
  "modified": "2024-03-29T07:11:25.337583Z",
  "name": "Updated Project",
  "description": "",
  "local_path": "_15__new_project",
  "scm_type": "git",
  "scm_url": "https://github.com/ansible/ansible-examples",
  "scm_branch": "",
  "scm_refspec": "",
  "scm_clean": false,
  "scm_track_submodules": false,
  "scm_delete_on_update": false,
  "credential": null,
  "timeout": 0,
  "scm_revision": "347e44fea036c94d5f60e544de006453ee5c71ad",
  "last_job_run": "2024-03-29T07:10:06.368318Z",
  "last_job_failed": false,
  "next_job_run": null,
  "status": "pending",
  "organization": 1,
  "scm_update_on_launch": false,
  "scm_update_cache_timeout": 0,
  "allow_override": false,
  "custom_virtualenv": null,
  "default_environment": null,
  "signature_validation_credential": null,
  "last_update_failed": false,
  "last_updated": "2024-03-29T07:10:06.368318Z"
}
```


Получение деталей проекта (после обновления):


```
{
  "id": 15,
  "type": "project",
  "url": "/api/v2/projects/15/",
  "related": {
    "named_url": "/api/v2/projects/Updated Project++Default/",
    "created_by": "/api/v2/users/1/",
    "modified_by": "/api/v2/users/1/",
    "last_job": "/api/v2/project_updates/37/",
    "teams": "/api/v2/projects/15/teams/",
    "playbooks": "/api/v2/projects/15/playbooks/",
    "inventory_files": "/api/v2/projects/15/inventories/",
    "update": "/api/v2/projects/15/update/",
    "project_updates": "/api/v2/projects/15/project_updates/",
    "scm_inventory_sources": "/api/v2/projects/15/scm_inventory_sources/",
    "schedules": "/api/v2/projects/15/schedules/",
    "activity_stream": "/api/v2/projects/15/activity_stream/",
    "notification_templates_started": "/api/v2/projects/15/notification_templates_
↪started/",
    "notification_templates_success": "/api/v2/projects/15/notification_templates_
↪success/",
    "notification_templates_error": "/api/v2/projects/15/notification_templates_error/",
    "access_list": "/api/v2/projects/15/access_list/",
    "object_roles": "/api/v2/projects/15/object_roles/",
    "copy": "/api/v2/projects/15/copy/",
    "organization": "/api/v2/organizations/1/",
    "last_update": "/api/v2/project_updates/37/"
```


(continues on next page)

(продолжение с предыдущей страницы)


```
  },
  "summary_fields": {
    "organization": {
      "id": 1,
      "name": "Default",
      "description": ""
    },
    "last_job": {
      "id": 37,
      "name": "Updated Project",
      "description": "",
      "finished": "2024-03-29T07:11:51.297476Z",
      "status": "successful",
      "failed": false
    },
    "last_update": {
      "id": 37,
      "name": "Updated Project",
      "description": "",
      "status": "successful",
      "failed": false
    },
    "created_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "modified_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "object_roles": {
      "admin_role": {
        "description": "Can manage all aspects of the project",
        "name": "Admin",
        "id": 100
      },
      "use_role": {
        "description": "Can use the project in a job template",
        "name": "Use",
        "id": 101
      },
      "update_role": {
        "description": "May update the project",
        "name": "Update",
        "id": 102
      },
      "read_role": {
        "description": "May view settings for the project",
        "name": "Read",
        "id": 103
      }
    },
    "user_capabilities": {
      "edit": true,
      "delete": true,
      "start": true,
      "schedule": true,
      "copy": true
    },
    "resolved_environment": {
```


(continues on next page)

(продолжение с предыдущей страницы)

```
      "id": 1,
      "name": "Default execution environment",
      "description": "",
      "image": "registry.astralinux.ru/aa/aa-base-ee:0.2.1"
    }
  },
  "created": "2024-03-29T07:08:16.321624Z",
  "modified": "2024-03-29T07:11:25.337583Z",
  "name": "Updated Project",
  "description": "",
  "local_path": "_15__new_project",
  "scm_type": "git",
  "scm_url": "https://github.com/ansible/ansible-examples",
  "scm_branch": "",
  "scm_refspec": "",
  "scm_clean": false,
  "scm_track_submodules": false,
  "scm_delete_on_update": false,
  "credential": null,
  "timeout": 0,
  "scm_revision": "b50586543c6c4be907fdc88f9f78a2b35d2a895f",
  "last_job_run": "2024-03-29T07:11:51.297476Z",
  "last_job_failed": false,
  "next_job_run": null,
  "status": "successful",
  "organization": 1,
  "scm_update_on_launch": false,
  "scm_update_cache_timeout": 0,
  "allow_override": false,
  "custom_virtualenv": null,
  "default_environment": null,
  "signature_validation_credential": null,
  "last_update_failed": false,
  "last_updated": "2024-03-29T07:11:51.297476Z"
}
```

Удаление проекта:

```
Status code: 204
```

Получение списка проектов (после удаления):

```
{
  "count": 2,
  "next": null,
  "previous": null,
  "results": [
    {
      "id": 6,
      "type": "project",
      "url": "/api/v2/projects/6/",
      "related": {
        "created_by": "/api/v2/users/1/",
        "modified_by": "/api/v2/users/1/",
        "teams": "/api/v2/projects/6/teams/",
        "playbooks": "/api/v2/projects/6/playbooks/",
        "inventory_files": "/api/v2/projects/6/inventories/",
        "update": "/api/v2/projects/6/update/",
        "project_updates": "/api/v2/projects/6/project_updates/",
        "scm_inventory_sources": "/api/v2/projects/6/scm_inventory_sources/",
        "schedules": "/api/v2/projects/6/schedules/",
        "activity_stream": "/api/v2/projects/6/activity_stream/",
        "notification_templates_started": "/api/v2/projects/6/notification_templates_
↪started/",
```

(continues on next page)

(продолжение с предыдущей страницы)


```
        "notification_templates_success": "/api/v2/projects/6/notification_templates_
↪success/",
        "notification_templates_error": "/api/v2/projects/6/notification_templates_error/
↪",
        "access_list": "/api/v2/projects/6/access_list/",
        "object_roles": "/api/v2/projects/6/object_roles/",
        "copy": "/api/v2/projects/6/copy/",
        "organization": "/api/v2/organizations/1/"
      },
      "summary_fields": {
        "organization": {
          "id": 1,
          "name": "Default",
          "description": ""
        },
        "created_by": {
          "id": 1,
          "username": "admin",
          "first_name": "",
          "last_name": ""
        },
        "modified_by": {
          "id": 1,
          "username": "admin",
          "first_name": "",
          "last_name": ""
        },
        "object_roles": {
          "admin_role": {
            "description": "Can manage all aspects of the project",
            "name": "Admin",
            "id": 22
          },
          "use_role": {
            "description": "Can use the project in a job template",
            "name": "Use",
            "id": 23
          },
          "update_role": {
            "description": "May update the project",
            "name": "Update",
            "id": 24
          },
          "read_role": {
            "description": "May view settings for the project",
            "name": "Read",
            "id": 25
          }
        },
        "user_capabilities": {
          "edit": true,
          "delete": true,
          "start": true,
          "schedule": true,
          "copy": true
        }
      },
      "created": "2024-02-16T00:06:02.670809Z",
      "modified": "2024-02-16T00:06:02.670834Z",
      "name": "Demo Project",
      "description": "",
      "local_path": "_6__demo_project",
      "scm_type": "git",
      "scm_url": "https://github.com/ansible/ansible-tower-samples",
```


(continues on next page)

(продолжение с предыдущей страницы)


```
      "scm_branch": "",
      "scm_refspec": "",
      "scm_clean": false,
      "scm_track_submodules": false,
      "scm_delete_on_update": false,
      "credential": null,
      "timeout": 0,
      "scm_revision": "347e44fea036c94d5f60e544de006453ee5c71ad",
      "last_job_run": null,
      "last_job_failed": false,
      "next_job_run": null,
      "status": "successful",
      "organization": 1,
      "scm_update_on_launch": false,
      "scm_update_cache_timeout": 0,
      "allow_override": false,
      "custom_virtualenv": null,
      "default_environment": null,
      "signature_validation_credential": null,
      "last_update_failed": false,
      "last_updated": null
    },
    {
      "id": 8,
      "type": "project",
      "url": "/api/v2/projects/8/",
      "related": {
        "created_by": "/api/v2/users/1/",
        "modified_by": "/api/v2/users/1/",
        "credential": "/api/v2/credentials/4/",
        "last_job": "/api/v2/project_updates/16/",
        "teams": "/api/v2/projects/8/teams/",
        "playbooks": "/api/v2/projects/8/playbooks/",
        "inventory_files": "/api/v2/projects/8/inventories/",
        "update": "/api/v2/projects/8/update/",
        "project_updates": "/api/v2/projects/8/project_updates/",
        "scm_inventory_sources": "/api/v2/projects/8/scm_inventory_sources/",
        "schedules": "/api/v2/projects/8/schedules/",
        "activity_stream": "/api/v2/projects/8/activity_stream/",
        "notification_templates_started": "/api/v2/projects/8/notification_templates_
↪started/",
        "notification_templates_success": "/api/v2/projects/8/notification_templates_
↪success/",
        "notification_templates_error": "/api/v2/projects/8/notification_templates_error/
↪",
        "access_list": "/api/v2/projects/8/access_list/",
        "object_roles": "/api/v2/projects/8/object_roles/",
        "copy": "/api/v2/projects/8/copy/",
        "organization": "/api/v2/organizations/1/",
        "last_update": "/api/v2/project_updates/16/"
      },
      "summary_fields": {
        "organization": {
          "id": 1,
          "name": "Default",
          "description": ""
        },
        "credential": {
          "id": 4,
          "name": "Automation Hub",
          "description": "",
          "kind": "scm",
          "cloud": false,
          "kubernetes": false,
```


(continues on next page)

(продолжение с предыдущей страницы)


```
          "credential_type_id": 2
        },
        "last_job": {
          "id": 16,
          "name": "\u0420\u0430\u0437\u0432\u0435\u0440\u0442\u044b\u0432\u0430\u043d\
↪u0438\u0435 NGINX",
          "description": "",
          "finished": "2024-03-15T12:09:44.270752Z",
          "status": "successful",
          "failed": false
        },
        "last_update": {
          "id": 16,
          "name": "\u0420\u0430\u0437\u0432\u0435\u0440\u0442\u044b\u0432\u0430\u043d\
↪u0438\u0435 NGINX",
          "description": "",
          "status": "successful",
          "failed": false
        },
        "created_by": {
          "id": 1,
          "username": "admin",
          "first_name": "",
          "last_name": ""
        },
        "modified_by": {
          "id": 1,
          "username": "admin",
          "first_name": "",
          "last_name": ""
        },
        "object_roles": {
          "admin_role": {
            "description": "Can manage all aspects of the project",
            "name": "Admin",
            "id": 60
          },
          "use_role": {
            "description": "Can use the project in a job template",
            "name": "Use",
            "id": 61
          },
          "update_role": {
            "description": "May update the project",
            "name": "Update",
            "id": 62
          },
          "read_role": {
            "description": "May view settings for the project",
            "name": "Read",
            "id": 63
          }
        },
        "user_capabilities": {
          "edit": true,
          "delete": true,
          "start": true,
          "schedule": true,
          "copy": true
        }
      },
      "created": "2024-03-15T11:57:40.851309Z",
      "modified": "2024-03-15T11:57:40.851323Z",
      "name": "\u0420\u0430\u0437\u0432\u0435\u0440\u0442\u044b\u0432\u0430\u043d\u0438\
```


(continues on next page)

(продолжение с предыдущей страницы)

```
↪u0435 NGINX",
      "description": "",
      "local_path": "_8__nginx",
      "scm_type": "git",
      "scm_url": "ssh://git@hub.astra-automation.ru:2222/aa-gca/AA/aac-samples.git",
      "scm_branch": "",
      "scm_refspec": "",
      "scm_clean": false,
      "scm_track_submodules": false,
      "scm_delete_on_update": false,
      "credential": 4,
      "timeout": 0,
      "scm_revision": "d7f0ff18cf8ffb44d64e8acd995402d545e2de90",
      "last_job_run": "2024-03-15T12:09:44.270752Z",
      "last_job_failed": false,
      "next_job_run": null,
      "status": "successful",
      "organization": 1,
      "scm_update_on_launch": false,
      "scm_update_cache_timeout": 0,
      "allow_override": false,
      "custom_virtualenv": null,
      "default_environment": null,
      "signature_validation_credential": null,
      "last_update_failed": false,
      "last_updated": "2024-03-15T12:09:44.270752Z"
    }
  ]
}
```

Получение конфигурации Automation Controller:

```
{
  "time_zone": "UTC",
  "license_info": {
    "license_type": "open",
    "valid_key": true,
    "subscription_name": "OPEN",
    "product_name": "AWX"
  },
  "version": "1.0.0+22.1.0",
  "eula": "",
  "analytics_status": "off",
  "analytics_collectors": {
    "config": {
      "name": "config",
      "version": "1.6",
      "description": "General platform configuration."
    },
    "counts": {
      "name": "counts",
      "version": "1.2",
      "description": "Counts of objects such as organizations, inventories, and projects"
    },
    "cred_type_counts": {
      "name": "cred_type_counts",
      "version": "1.0",
      "description": "Counts of credentials by credential type"
    },
    "events_table": {
      "name": "events_table",
      "version": "1.5",
      "description": "Automation task records"
    },
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    "host_metric_table": {
      "name": "host_metric_table",
      "version": "1.0",
      "description": "Host Metric data, incremental/full sync"
    },
    "instance_info": {
      "name": "instance_info",
      "version": "1.3",
      "description": "Cluster topology and capacity"
    },
    "inventory_counts": {
      "name": "inventory_counts",
      "version": "1.2",
      "description": "Inventories, their inventory sources, and host counts"
    },
    "org_counts": {
      "name": "org_counts",
      "version": "1.0",
      "description": "Counts of users and teams by organization"
    },
    "projects_by_scm_type": {
      "name": "projects_by_scm_type",
      "version": "1.0",
      "description": "Counts of projects by source control type"
    },
    "query_info": {
      "name": "query_info",
      "version": "1.0",
      "description": "Metadata about the analytics collected"
    },
    "unified_job_template_table": {
      "name": "unified_job_template_table",
      "version": "1.1",
      "description": "Data on job templates"
    },
    "unified_jobs_table": {
      "name": "unified_jobs_table",
      "version": "1.4",
      "description": "Data on jobs run"
    },
    "workflow_job_node_table": {
      "name": "workflow_job_node_table",
      "version": "1.0",
      "description": "Data on workflow runs"
    },
    "workflow_job_template_node_table": {
      "name": "workflow_job_template_node_table",
      "version": "1.0",
      "description": "Data on workflows"
    }
  },
  "become_methods": [
    ["sudo", "Sudo"],
    ["su", "Su"],
    ["pbrun", "Pbrun"],
    ["pfexec", "Pfexec"],
    ["dzdo", "DZDO"],
    ["pmrun", "Pmrun"],
    ["runas", "Runas"],
    ["enable", "Enable"],
    ["doas", "Doas"],
    ["ksu", "Ksu"],
    ["machinectl", "Machinectl"],
    ["sesu", "Sesu"]
```

(continues on next page)

(продолжение с предыдущей страницы)

```
  ],
  "ui_next": false,
  "project_base_dir": "/var/lib/awx/projects",
  "project_local_paths": [],
  "custom_virtualenvs": []
}
```

7.4.4 Базовый сценарий

В этом документе приведен базовый сценарий использования API Automation Controller. В ходе выполнения сценария производятся следующие действия:

1. Получение токена авторизации.
2. Создание организации.
3. Добавление полномочий Ansible Galaxy для созданной организации.
4. Создание инвентаря.
5. Добавление управляемого узла в инвентарь.
6. Создание проекта.
7. Создание шаблона задания для установки NGINX.
8. Запуск шаблона задания.
9. Проверка статуса выполнения задания.

Предварительные требования

Перед выполнением сценария произведите следующие действия в графическом интерфейсе контроллера:

1. Создайте полномочия для доступа к следующим ресурсам:
   - управляемый узел;
   - Ansible Galaxy;
   - Private Automation Hub.

   Примечание: С подробной информацией о работе с полномочиями можно ознакомиться в документе *Полномочия*.

2. Получите идентификаторы созданных полномочий:
   a. На панели навигации выберите *Ресурсы* ▸ *Полномочия* (*Resources* ▸ *Credentials*).
   b. Нажмите на ссылку с названием нужного полномочия, например, Ansible Galaxy.
   c. Скопируйте <credentials_id> в адресной строке браузера, которая имеет следующий вид `https://<address>/#/credentials/<credentials_id>/details`.

Подготовка организационной структуры

Для подготовки организационной структуры выполните следующие действия:

1. Получите токен авторизации:

```
curl -X POST \
   https://<address>/api/v2/tokens/ \
   -H 'Content-Type: application/json' \
   -d '{
            "description": "Token description",
            "application": null,
            "user": "<username>",
            "scope": "write"
       }' \
   -u '<username>:<password>' | jq .
```

   Подставьте собственные значения параметров вместо употребленных в примере:

   - <address> – FQDN или IP-адрес контроллера, например, `192.168.56.11`;
   - <username> – название учетной записи пользователя, используемой для аутентификации в контроллере, например, `admin`;
   - <password> – пароль, например, `awx`.

   Эта команда возвращает JSON следующего вида:

```
{
   "id": 13,
   "type": "o_auth2_access_token",
   "url": "/api/v2/tokens/13/",
   "related": {
      "user": "/api/v2/users/1/",
      "activity_stream": "/api/v2/tokens/13/activity_stream/"
   },
   "summary_fields": {
      "user": {
         "id": 1,
         "username": "admin",
         "first_name": "",
         "last_name": ""
      }
   },
   "created": "2024-03-21T08:38:43.412155Z",
   "modified": "2024-03-21T08:38:43.418814Z",
   "description": "Token description",
   "user": 1,
   "token": "<your_token>",
   "refresh_token": null,
   "application": null,
   "expires": "3023-07-23T08:38:43.406429Z",
   "scope": "write"
}
```

   Здесь `token` – значение созданного токена. Сохраните его в надежном месте, оно понадобится позже.

2. Получите идентификатор среды исполнения в графическом интерфейсе контроллера:

   a. На панели навигации выберите *Администрирование* ▸ *Среды исполнения* (*Administration* ▸ *Execution Environments*).

   b. Нажмите на ссылку с названием нужной среды исполнения, например, Default execution environment.

   c. Скопируйте <execution_environment_id> в адресной строке браузера, которая имеет следующий вид `https://<address>/#/execution_environments/<execution_environments_id>/details`.

3. Создайте организацию:

```
curl -X POST \
   -H 'Content-Type: application/json' \
   -H 'Authorization: Bearer <your_token>' \
   -d '{
            "name": "<your_organization_name>",
            "default_environment": "<execution_environment_id>"
       }' \
   https://<address>/api/v2/organizations/ | jq .
```

Подставьте собственные значения параметров вместо употребленных в примере:

- <your_token> – токен, полученный на предыдущем шаге, например, 3wijiG7********VtL2gseJESzjJsD;
- <your_organization_name> – название организации, например, Jupiter;
- <execution_environment_id> – идентификатор среды исполнения, например, 1.

Organization

```
{
  "id": 4,
  "type": "organization",
  "url": "/api/v2/organizations/4/",
  "related": {
    "named_url": "/api/v2/organizations/Jupiter/",
    "created_by": "/api/v2/users/1/",
    "modified_by": "/api/v2/users/1/",
    "execution_environments": "/api/v2/organizations/4/execution_environments/",
    "projects": "/api/v2/organizations/4/projects/",
    "inventories": "/api/v2/organizations/4/inventories/",
    "job_templates": "/api/v2/organizations/4/job_templates/",
    "workflow_job_templates": "/api/v2/organizations/4/workflow_job_templates/",
    "users": "/api/v2/organizations/4/users/",
    "admins": "/api/v2/organizations/4/admins/",
    "teams": "/api/v2/organizations/4/teams/",
    "credentials": "/api/v2/organizations/4/credentials/",
    "applications": "/api/v2/organizations/4/applications/",
    "activity_stream": "/api/v2/organizations/4/activity_stream/",
    "notification_templates": "/api/v2/organizations/4/notification_templates/",
    "notification_templates_started": "/api/v2/organizations/4/notification_
↪templates_started/",
    "notification_templates_success": "/api/v2/organizations/4/notification_
↪templates_success/",
    "notification_templates_error": "/api/v2/organizations/4/notification_templates_
↪error/",
    "notification_templates_approvals": "/api/v2/organizations/4/notification_
↪templates_approvals/",
    "object_roles": "/api/v2/organizations/4/object_roles/",
    "access_list": "/api/v2/organizations/4/access_list/",
    "instance_groups": "/api/v2/organizations/4/instance_groups/",
    "galaxy_credentials": "/api/v2/organizations/4/galaxy_credentials/",
    "default_environment": "/api/v2/execution_environments/1/"
  },
  "summary_fields": {
    "default_environment": {
      "id": 1,
      "name": "Default execution environment",
      "description": "",
      "image": "registry.astralinux.ru/aa/aa-base-ee:0.2.1"
    },
    "created_by": {
      "id": 1,
```

(continues on next page)

(продолжение с предыдущей страницы)


```
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "modified_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "object_roles": {
      "admin_role": {
        "description": "Can manage all aspects of the organization",
        "name": "Admin",
        "id": 103,
        "user_only": true
      },
      "execute_role": {
        "description": "May run any executable resources in the organization",
        "name": "Execute",
        "id": 104
      },
      "project_admin_role": {
        "description": "Can manage all projects of the organization",
        "name": "Project Admin",
        "id": 105
      },
      "inventory_admin_role": {
        "description": "Can manage all inventories of the organization",
        "name": "Inventory Admin",
        "id": 106
      },
      "credential_admin_role": {
        "description": "Can manage all credentials of the organization",
        "name": "Credential Admin",
        "id": 107
      },
      "workflow_admin_role": {
        "description": "Can manage all workflows of the organization",
        "name": "Workflow Admin",
        "id": 108
      },
      "notification_admin_role": {
        "description": "Can manage all notifications of the organization",
        "name": "Notification Admin",
        "id": 109
      },
      "job_template_admin_role": {
        "description": "Can manage all job templates of the organization",
        "name": "Job Template Admin",
        "id": 110
      },
      "execution_environment_admin_role": {
        "description": "Can manage all execution environments of the organization",
        "name": "Execution Environment Admin",
        "id": 111
      },
      "auditor_role": {
        "description": "Can view all aspects of the organization",
        "name": "Auditor",
        "id": 112
      },
      "member_role": {
        "description": "User is a member of the organization",
```


(continues on next page)

(продолжение с предыдущей страницы)


```
        "name": "Member",
        "id": 113,
        "user_only": true
      },
      "read_role": {
        "description": "May view settings for the organization",
        "name": "Read",
        "id": 114
      },
      "approval_role": {
        "description": "Can approve or deny a workflow approval node",
        "name": "Approve",
        "id": 115
      }
    },
    "user_capabilities": {
      "edit": true,
      "delete": true
    },
    "related_field_counts": {
      "inventories": 0,
      "teams": 0,
      "users": 0,
      "job_templates": 0,
      "admins": 0,
      "projects": 0
    }
  },
  "created": "2024-03-21T08:48:05.919683Z",
  "modified": "2024-03-21T08:48:05.919698Z",
  "name": "Jupiter",
  "max_hosts": 0,
  "custom_virtualenv": null,
  "default_environment": 1
}
```


4. Добавьте полномочия Ansible Galaxy для созданной организации:

```
curl -X POST \
   -H 'Content-Type: application/json' \
   -H 'Authorization: Bearer <your_token>' \
   -d '{
            "id":<credentials_id>
       }' \
   https://<address>/api/v2/organizations/<organization_id>/galaxy_credentials/ |␣
↪jq .
```

Подставьте собственные значения параметров вместо употребленных в примере:

- <credentials_id> – идентификатор полномочий для доступа к Ansible Galaxy, например, 2;
- <organization_id> – идентификатор организации, созданной ранее, например, 4.

Подготовка инвентаря

Для подготовки инвентаря выполните следующие действия:

1. Создайте инвентарь:

```
curl -X POST \
   -H 'Content-Type: application/json' \
   -H 'Authorization: Bearer <address>' \
   -d '{
            "name": "<your_inventory_name>",
            "organization": <organization_id>
       }' \
   https://<address>/api/v2/inventories/ | jq .
```

Подставьте собственные значения параметров вместо употребленных в примере. Здесь <your_inventory_name> – название инвентаря, например, `Demo Inventory`.

Inventory


```
{
  "id": 4,
  "type": "inventory",
  "url": "/api/v2/inventories/4/",
  "related": {
    "named_url": "/api/v2/inventories/Demo Inventory++Jupiter/",
    "created_by": "/api/v2/users/1/",
    "modified_by": "/api/v2/users/1/",
    "hosts": "/api/v2/inventories/4/hosts/",
    "variable_data": "/api/v2/inventories/4/variable_data/",
    "script": "/api/v2/inventories/4/script/",
    "activity_stream": "/api/v2/inventories/4/activity_stream/",
    "job_templates": "/api/v2/inventories/4/job_templates/",
    "ad_hoc_commands": "/api/v2/inventories/4/ad_hoc_commands/",
    "access_list": "/api/v2/inventories/4/access_list/",
    "object_roles": "/api/v2/inventories/4/object_roles/",
    "instance_groups": "/api/v2/inventories/4/instance_groups/",
    "copy": "/api/v2/inventories/4/copy/",
    "labels": "/api/v2/inventories/4/labels/",
    "groups": "/api/v2/inventories/4/groups/",
    "root_groups": "/api/v2/inventories/4/root_groups/",
    "update_inventory_sources": "/api/v2/inventories/4/update_inventory_sources/",
    "inventory_sources": "/api/v2/inventories/4/inventory_sources/",
    "tree": "/api/v2/inventories/4/tree/",
    "organization": "/api/v2/organizations/4/"
  },
  "summary_fields": {
    "organization": {
      "id": 4,
      "name": "Jupiter"
    },
    "created_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "modified_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "object_roles": {
```


(continues on next page)

(продолжение с предыдущей страницы)

```
      "admin_role": {
        "description": "Can manage all aspects of the inventory",
        "name": "Admin",
        "id": 116
      },
      "update_role": {
        "description": "May update the inventory",
        "name": "Update",
        "id": 117
      },
      "adhoc_role": {
        "description": "May run ad hoc commands on the inventory",
        "name": "Ad Hoc",
        "id": 118
      },
      "use_role": {
        "description": "Can use the inventory in a job template",
        "name": "Use",
        "id": 119
      },
      "read_role": {
        "description": "May view settings for the inventory",
        "name": "Read",
        "id": 120
      }
    },
    "user_capabilities": {
      "edit": true,
      "delete": true,
      "copy": true,
      "adhoc": true
    },
    "labels": {
      "count": 0,
      "results": []
    }
  },
  "created": "2024-03-21T08:53:27.613062Z",
  "modified": "2024-03-21T08:53:27.613074Z",
  "name": "Jupiter",
  "organization": 4,
  "kind": "",
  "host_filter": null,
  "variables": "",
  "has_active_failures": false,
  "total_hosts": 0,
  "hosts_with_active_failures": 0,
  "total_groups": 0,
  "has_inventory_sources": false,
  "total_inventory_sources": 0,
  "inventory_sources_with_failures": 0,
  "pending_deletion": false,
  "prevent_instance_group_fallback": false
}
```

2. Добавьте управляемый узел в инвентарь:

```
curl -X POST \
   -H 'Content-Type: application/json' \
   -H 'Authorization: Bearer <your_token>' \
   -d '{
            "name": "<node_ip>",
            "description": "<node_name>",
            "inventory": <inventory_id>,
```

(continues on next page)

(продолжение с предыдущей страницы)

```
            "enabled": true
        }' \
    https://<address>/api/v2/hosts/ | jq .
```

Подставьте собственные значения параметров вместо употребленных в примере:

- <node_ip> – IP-адрес управляемого узла например, 10.2.0.6;
- <node_name> – название управляемого узла, например, node01;
- <inventory_id> – идентификатор инвентаря, например, 4.

Host


```
{
  "id": 7,
  "type": "host",
  "url": "/api/v2/hosts/7/",
  "related": {
    "named_url": "/api/v2/hosts/10.2.0.6++Demo Inventory++Jupiter/",
    "created_by": "/api/v2/users/1/",
    "modified_by": "/api/v2/users/1/",
    "variable_data": "/api/v2/hosts/7/variable_data/",
    "groups": "/api/v2/hosts/7/groups/",
    "all_groups": "/api/v2/hosts/7/all_groups/",
    "job_events": "/api/v2/hosts/7/job_events/",
    "job_host_summaries": "/api/v2/hosts/7/job_host_summaries/",
    "activity_stream": "/api/v2/hosts/7/activity_stream/",
    "inventory_sources": "/api/v2/hosts/7/inventory_sources/",
    "smart_inventories": "/api/v2/hosts/7/smart_inventories/",
    "ad_hoc_commands": "/api/v2/hosts/7/ad_hoc_commands/",
    "ad_hoc_command_events": "/api/v2/hosts/7/ad_hoc_command_events/",
    "ansible_facts": "/api/v2/hosts/7/ansible_facts/",
    "inventory": "/api/v2/inventories/4/"
  },
  "summary_fields": {
    "inventory": {
      "id": 4,
      "name": "Demo Inventory",
      "has_active_failures": false,
      "total_hosts": 0,
      "hosts_with_active_failures": 0,
      "total_groups": 0,
      "has_inventory_sources": false,
      "total_inventory_sources": 0,
      "inventory_sources_with_failures": 0,
      "organization_id": 4,
      "kind": ""
    },
    "created_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "modified_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "user_capabilities": {
      "edit": true,
```


(continues on next page)

(продолжение с предыдущей страницы)

```
      "delete": true
    },
    "groups": {
      "count": 0,
      "results": []
    },
    "recent_jobs": []
  },
  "created": "2024-03-21T10:03:33.967091Z",
  "modified": "2024-03-21T10:03:33.967104Z",
  "name": "10.2.0.6",
  "description": "node01",
  "inventory": 4,
  "enabled": true,
  "instance_id": "",
  "variables": "",
  "has_active_failures": false,
  "has_inventory_sources": false,
  "last_job": null,
  "last_job_host_summary": null,
  "ansible_facts_modified": null
}
```

Подготовка задания

Для подготовки задания выполните следующие действия:

1. Создайте проект:

Примечание: В качестве источника при создании проекта используется репозиторий с примерами[158].

```
curl -X POST \
   -H 'Content-Type: application/json' \
   -H 'Authorization: Bearer <your_token>' \
   -d '{
            "name": "<your_project_name>",
            "scm_type": "git",
            "scm_url": "ssh://git@source.astragroup.ru:2222/aa-gca/AA/aac-samples.
↪git",
            "scm_branch": "master",
            "default_environment": <environment_id>,
            "credential": <credential_id>,
            "organization": <organization_id>
       }' \
   https://<address>/api/v2/projects/ | jq .
```

Подставьте собственные значения параметров вместо употребленных в примере:

- <your_project_name> – название проекта, например, `Demo NGINX Project`;
- <credentials_id> – идентификатор полномочий для доступа к Private Automation Hub, например, 3.

[158] https://source.astragroup.ru/

Project


```
{
  "id": 14,
  "type": "project",
  "url": "/api/v2/projects/14/",
  "related": {
    "named_url": "/api/v2/projects/Demo NGINX Project++Jupiter/",
    "created_by": "/api/v2/users/1/",
    "modified_by": "/api/v2/users/1/",
    "credential": "/api/v2/credentials/3/",
    "current_job": "/api/v2/project_updates/9/",
    "teams": "/api/v2/projects/14/teams/",
    "playbooks": "/api/v2/projects/14/playbooks/",
    "inventory_files": "/api/v2/projects/14/inventories/",
    "update": "/api/v2/projects/14/update/",
    "project_updates": "/api/v2/projects/14/project_updates/",
    "scm_inventory_sources": "/api/v2/projects/14/scm_inventory_sources/",
    "schedules": "/api/v2/projects/14/schedules/",
    "activity_stream": "/api/v2/projects/14/activity_stream/",
    "notification_templates_started": "/api/v2/projects/14/notification_templates_
↪started/",
    "notification_templates_success": "/api/v2/projects/14/notification_templates_
↪success/",
    "notification_templates_error": "/api/v2/projects/14/notification_templates_
↪error/",
    "access_list": "/api/v2/projects/14/access_list/",
    "object_roles": "/api/v2/projects/14/object_roles/",
    "copy": "/api/v2/projects/14/copy/",
    "organization": "/api/v2/organizations/4/",
    "default_environment": "/api/v2/execution_environments/1/",
    "current_update": "/api/v2/project_updates/9/"
  },
  "summary_fields": {
    "organization": {
      "id": 4,
      "name": "Jupiter"
    },
    "default_environment": {
      "id": 1,
      "name": "Default execution environment",
      "description": "",
      "image": "registry.astralinux.ru/aa/aa-base-ee:0.2.1"
    },
    "credential": {
      "id": 3,
      "name": "Credentials for AA Hub",
      "description": "",
      "kind": "scm",
      "cloud": false,
      "kubernetes": false,
      "credential_type_id": 2
    },
    "current_update": {
      "id": 9,
      "name": "Demo NGINX Project",
      "status": "pending",
      "failed": false
    },
    "current_job": {
      "id": 9,
      "name": "Demo NGINX Project",
      "status": "pending",
      "failed": false
    },
```


(continues on next page)

(продолжение с предыдущей страницы)


```
    "created_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "modified_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "object_roles": {
      "admin_role": {
        "description": "Can manage all aspects of the project",
        "name": "Admin",
        "id": 134
      },
      "use_role": {
        "description": "Can use the project in a job template",
        "name": "Use",
        "id": 135
      },
      "update_role": {
        "description": "May update the project",
        "name": "Update",
        "id": 136
      },
      "read_role": {
        "description": "May view settings for the project",
        "name": "Read",
        "id": 137
      }
    },
    "user_capabilities": {
      "edit": true,
      "delete": true,
      "start": true,
      "schedule": true,
      "copy": true
    }
  },
  "created": "2024-03-21T09:44:26.824017Z",
  "modified": "2024-03-21T09:44:26.824031Z",
  "name": "Demo NGINX Project",
  "local_path": "_14__demo_nginx_project",
  "scm_type": "git",
  "scm_url": "ssh://git@hub.astra-automation.ru:2222/aa-gca/AA/aac-samples.git",
  "scm_branch": "master",
  "scm_refspec": "",
  "scm_clean": false,
  "scm_track_submodules": false,
  "scm_delete_on_update": false,
  "credential": 3,
  "timeout": 0,
  "scm_revision": "",
  "last_job_run": null,
  "last_job_failed": false,
  "next_job_run": null,
  "status": "pending",
  "organization": 4,
  "scm_update_on_launch": false,
  "scm_update_cache_timeout": 0,
  "allow_override": false,
```


(continues on next page)

(продолжение с предыдущей страницы)

```
  "custom_virtualenv": null,
  "default_environment": 1,
  "signature_validation_credential": null,
  "last_update_failed": false,
  "last_updated": null
}
```

2. Создайте шаблон задания для установки NGINX:

```
curl -X POST \
   -H 'Content-Type: application/json' \
   -H 'Authorization: Bearer <your_token>' \
   -d '{
            "name": "NGINX Deployment",
            "job_type": "run",
            "inventory": <inventory_id>,
            "project": <project_id>,
            "playbook": "nginx/site.yml"
       }' \
   https://<address>/api/v2/job_templates/ | jq .
```

Подставьте собственные значения параметров вместо употребленных в примере. Здесь <project_id> – идентификатор проекта, созданного ранее, например, 14.

Job template

```
{
  "id": 15,
  "type": "job_template",
  "url": "/api/v2/job_templates/15/",
  "related": {
    "named_url": "/api/v2/job_templates/NGINX Deployment++Jupiter/",
    "created_by": "/api/v2/users/1/",
    "modified_by": "/api/v2/users/1/",
    "labels": "/api/v2/job_templates/15/labels/",
    "inventory": "/api/v2/inventories/4/",
    "project": "/api/v2/projects/14/",
    "organization": "/api/v2/organizations/4/",
    "credentials": "/api/v2/job_templates/15/credentials/",
    "jobs": "/api/v2/job_templates/15/jobs/",
    "schedules": "/api/v2/job_templates/15/schedules/",
    "activity_stream": "/api/v2/job_templates/15/activity_stream/",
    "launch": "/api/v2/job_templates/15/launch/",
    "webhook_key": "/api/v2/job_templates/15/webhook_key/",
    "webhook_receiver": "",
    "notification_templates_started": "/api/v2/job_templates/15/notification_
↪templates_started/",
    "notification_templates_success": "/api/v2/job_templates/15/notification_
↪templates_success/",
    "notification_templates_error": "/api/v2/job_templates/15/notification_
↪templates_error/",
    "access_list": "/api/v2/job_templates/15/access_list/",
    "survey_spec": "/api/v2/job_templates/15/survey_spec/",
    "object_roles": "/api/v2/job_templates/15/object_roles/",
    "instance_groups": "/api/v2/job_templates/15/instance_groups/",
    "slice_workflow_jobs": "/api/v2/job_templates/15/slice_workflow_jobs/",
    "copy": "/api/v2/job_templates/15/copy/"
  },
  "summary_fields": {
    "organization": {
      "id": 4,
      "name": "Jupiter"
```

(continues on next page)

(продолжение с предыдущей страницы)


```
    },
    "inventory": {
      "id": 4,
      "name": "Demo Inventory",
      "has_active_failures": false,
      "total_hosts": 0,
      "hosts_with_active_failures": 0,
      "total_groups": 0,
      "has_inventory_sources": false,
      "total_inventory_sources": 0,
      "inventory_sources_with_failures": 0,
      "organization_id": 4,
      "kind": ""
    },
    "project": {
      "id": 14,
      "name": "Demo NGINX Project",
      "status": "successful",
      "scm_type": "git",
      "allow_override": false
    },
    "created_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "modified_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "object_roles": {
      "admin_role": {
        "description": "Can manage all aspects of the job template",
        "name": "Admin",
        "id": 138
      },
      "execute_role": {
        "description": "May run the job template",
        "name": "Execute",
        "id": 139
      },
      "read_role": {
        "description": "May view settings for the job template",
        "name": "Read",
        "id": 140
      }
    },
    "user_capabilities": {
      "edit": true,
      "delete": true,
      "start": true,
      "schedule": true,
      "copy": true
    },
    "labels": {
      "count": 0,
      "results": []
    },
    "recent_jobs": [],
    "credentials": []
  },
```


(continues on next page)

(продолжение с предыдущей страницы)

```
  "created": "2024-03-21T09:52:41.673531Z",
  "modified": "2024-03-21T09:52:41.673545Z",
  "name": "NGINX Deployment",
  "description": "",
  "job_type": "run",
  "inventory": 4,
  "project": 14,
  "playbook": "nginx/site.yml",
  "scm_branch": "",
  "forks": 0,
  "limit": "",
  "verbosity": 0,
  "extra_vars": "",
  "job_tags": "",
  "force_handlers": false,
  "skip_tags": "",
  "start_at_task": "",
  "timeout": 0,
  "use_fact_cache": false,
  "organization": 4,
  "last_job_run": null,
  "last_job_failed": false,
  "next_job_run": null,
  "status": "never updated",
  "execution_environment": null,
  "host_config_key": "",
  "ask_scm_branch_on_launch": false,
  "ask_diff_mode_on_launch": false,
  "ask_variables_on_launch": false,
  "ask_limit_on_launch": false,
  "ask_tags_on_launch": false,
  "ask_skip_tags_on_launch": false,
  "ask_job_type_on_launch": false,
  "ask_verbosity_on_launch": false,
  "ask_inventory_on_launch": false,
  "ask_credential_on_launch": false,
  "ask_execution_environment_on_launch": false,
  "ask_labels_on_launch": false,
  "ask_forks_on_launch": false,
  "ask_job_slice_count_on_launch": false,
  "ask_timeout_on_launch": false,
  "ask_instance_groups_on_launch": false,
  "survey_enabled": false,
  "become_enabled": false,
  "diff_mode": false,
  "allow_simultaneous": false,
  "custom_virtualenv": null,
  "job_slice_count": 1,
  "webhook_service": "",
  "webhook_credential": null,
  "prevent_instance_group_fallback": false
}
```

3. Добавьте полномочия для шаблона задания:

```
curl -X POST \
   -H 'Content-Type: application/json' \
   -H 'Authorization: Bearer <your_token>' \
   -d '{
            "id":<credential_id>
       }' \
   https://<address>/api/v2/job_templates/<job_template_id>/credentials/ | jq .
```

Подставьте собственные значения параметров вместо употребленных в примере:

- <credentials_id> – идентификатор полномочий для доступа к управляемому узлу, например, 1;
- <job_template_id> – идентификатор созданного ранее шаблона задания, например, 15.

Выполнение и проверка задания

Для запуска и проверки статуса задания выполните следующие действия:

1. Запустите задание, используя созданный шаблон:

```
curl -X POST \
   -H 'Authorization: Bearer <your_token>' \
   https://<address>/api/v2/job_templates/<job_template_id>/launch/ | jq .
```

Job template


```
{
  "job": 10,
  "ignored_fields": {},
  "id": 10,
  "type": "job",
  "url": "/api/v2/jobs/10/",
  "related": {
    "created_by": "/api/v2/users/1/",
    "modified_by": "/api/v2/users/1/",
    "labels": "/api/v2/jobs/10/labels/",
    "inventory": "/api/v2/inventories/4/",
    "project": "/api/v2/projects/14/",
    "organization": "/api/v2/organizations/4/",
    "credentials": "/api/v2/jobs/10/credentials/",
    "unified_job_template": "/api/v2/job_templates/15/",
    "stdout": "/api/v2/jobs/10/stdout/",
    "job_events": "/api/v2/jobs/10/job_events/",
    "job_host_summaries": "/api/v2/jobs/10/job_host_summaries/",
    "activity_stream": "/api/v2/jobs/10/activity_stream/",
    "notifications": "/api/v2/jobs/10/notifications/",
    "create_schedule": "/api/v2/jobs/10/create_schedule/",
    "job_template": "/api/v2/job_templates/15/",
    "cancel": "/api/v2/jobs/10/cancel/",
    "relaunch": "/api/v2/jobs/10/relaunch/"
  },
  "summary_fields": {
    "organization": {
      "id": 4,
      "name": "Jupiter"
    },
    "inventory": {
      "id": 4,
      "name": "Demo Inventory",
      "has_active_failures": false,
      "total_hosts": 0,
      "hosts_with_active_failures": 0,
      "total_groups": 0,
      "has_inventory_sources": false,
      "total_inventory_sources": 0,
      "inventory_sources_with_failures": 0,
      "organization_id": 4,
      "kind": ""
    },
    "project": {
      "id": 14,
```


(continues on next page)

(продолжение с предыдущей страницы)


```
      "name": "Demo NGINX Project",
      "status": "successful",
      "scm_type": "git",
      "allow_override": false
    },
    "job_template": {
      "id": 15,
      "name": "NGINX Deployment",
      "description": ""
    },
    "unified_job_template": {
      "id": 15,
      "name": "NGINX Deployment",
      "description": "",
      "unified_job_type": "job"
    },
    "created_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "modified_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "user_capabilities": {
      "delete": true,
      "start": true
    },
    "labels": {
      "count": 0,
      "results": []
    },
    "credentials": [
      {
        "id": 4,
        "name": "NGINX Machine Credential",
        "description": "",
        "kind": "ssh",
        "cloud": false
      }
    ]
  },
  "created": "2024-03-21T10:02:14.206401Z",
  "modified": "2024-03-21T10:02:14.230669Z",
  "name": "NGINX Deployment",
  "description": "",
  "job_type": "run",
  "inventory": 4,
  "project": 14,
  "playbook": "nginx/site.yml",
  "scm_branch": "",
  "forks": 0,
  "limit": "",
  "verbosity": 0,
  "extra_vars": "{}",
  "job_tags": "",
  "force_handlers": false,
  "skip_tags": "",
  "start_at_task": "",
  "timeout": 0,
```


(continues on next page)

(продолжение с предыдущей страницы)

```
  "use_fact_cache": false,
  "organization": 4,
  "unified_job_template": 15,
  "launch_type": "manual",
  "status": "pending",
  "execution_environment": null,
  "failed": false,
  "started": null,
  "finished": null,
  "canceled_on": null,
  "elapsed": 0,
  "job_args": "",
  "job_cwd": "",
  "job_env": {},
  "job_explanation": "",
  "execution_node": "",
  "controller_node": "",
  "result_traceback": "",
  "event_processing_finished": false,
  "launched_by": {
    "id": 1,
    "name": "admin",
    "type": "user",
    "url": "/api/v2/users/1/"
  },
  "work_unit_id": null,
  "job_template": 15,
  "passwords_needed_to_start": [],
  "allow_simultaneous": false,
  "artifacts": {},
  "scm_revision": "",
  "instance_group": null,
  "diff_mode": false,
  "job_slice_number": 0,
  "job_slice_count": 1,
  "webhook_service": "",
  "webhook_credential": null,
  "webhook_guid": ""
}
```

2. Проверьте статус выполнения задания:

```
curl -X GET \
   https://<address>/api/v2/jobs/<id_job>/ \
   -H 'Authorization: Bearer <your_token>' \
   -k -s | jq .
```

Подставьте собственные значения параметров вместо употребленных в примере. Здесь <id_job> – идентификатор задания, запущенного ранее, например, 11.

Job

```
{
  "id": 11,
  "type": "job",
  "url": "/api/v2/jobs/11/",
  "related": {
    "created_by": "/api/v2/users/1/",
    "labels": "/api/v2/jobs/11/labels/",
    "inventory": "/api/v2/inventories/4/",
    "project": "/api/v2/projects/14/",
    "organization": "/api/v2/organizations/4/",
```

(continues on next page)

(продолжение с предыдущей страницы)


```
    "credentials": "/api/v2/jobs/11/credentials/",
    "unified_job_template": "/api/v2/job_templates/15/",
    "stdout": "/api/v2/jobs/11/stdout/",
    "execution_environment": "/api/v2/execution_environments/1/",
    "job_events": "/api/v2/jobs/11/job_events/",
    "job_host_summaries": "/api/v2/jobs/11/job_host_summaries/",
    "activity_stream": "/api/v2/jobs/11/activity_stream/",
    "notifications": "/api/v2/jobs/11/notifications/",
    "create_schedule": "/api/v2/jobs/11/create_schedule/",
    "job_template": "/api/v2/job_templates/15/",
    "cancel": "/api/v2/jobs/11/cancel/",
    "relaunch": "/api/v2/jobs/11/relaunch/"
  },
  "summary_fields": {
    "organization": {
      "id": 4,
      "name": "Jupiter"
    },
    "inventory": {
      "id": 4,
      "name": "Demo Inventory",
      "has_active_failures": false,
      "total_hosts": 1,
      "hosts_with_active_failures": 0,
      "total_groups": 0,
      "has_inventory_sources": false,
      "total_inventory_sources": 0,
      "inventory_sources_with_failures": 0,
      "organization_id": 4,
      "kind": ""
    },
    "execution_environment": {
      "id": 1,
      "name": "Default execution environment",
      "description": "",
      "image": "registry.astralinux.ru/aa/aa-base-ee:0.2.1"
    },
    "project": {
      "id": 14,
      "name": "Demo NGINX Project",
      "status": "successful",
      "scm_type": "git",
      "allow_override": false
    },
    "job_template": {
      "id": 15,
      "name": "NGINX Deployment",
      "description": ""
    },
    "unified_job_template": {
      "id": 15,
      "name": "NGINX Deployment",
      "description": "",
      "unified_job_type": "job"
    },
    "instance_group": {
      "id": 2,
      "name": "default",
      "is_container_group": false
    },
    "created_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
```


(continues on next page)

(продолжение с предыдущей страницы)


```
      "last_name": ""
    },
    "user_capabilities": {
      "delete": true,
      "start": true
    },
    "labels": {
      "count": 0,
      "results": []
    },
    "credentials": [
      {
        "id": 4,
        "name": "NGINX Machine Credentials",
        "description": "",
        "kind": "ssh",
        "cloud": false
      }
    ]
  },
  "created": "2024-03-21T10:03:39.307771Z",
  "modified": "2024-03-21T10:03:39.534635Z",
  "name": "NGINX Deployment",
  "description": "",
  "job_type": "run",
  "inventory": 4,
  "project": 14,
  "playbook": "nginx/site.yml",
  "scm_branch": "",
  "forks": 0,
  "limit": "",
  "verbosity": 0,
  "extra_vars": "{}",
  "job_tags": "",
  "force_handlers": false,
  "skip_tags": "",
  "start_at_task": "",
  "timeout": 0,
  "use_fact_cache": false,
  "organization": 4,
  "unified_job_template": 15,
  "launch_type": "manual",
  "status": "successful",
  "execution_environment": 1,
  "failed": false,
  "started": "2024-03-21T10:03:39.605438Z",
  "finished": "2024-03-21T10:03:49.709550Z",
  "canceled_on": null,
  "elapsed": 10.104,
  "job_args": "[\"podman\", \"run\", \"--rm\", \"--tty\", \"--interactive\", \"--
↪workdir\", \"/runner/project\", \"-v\", \"/tmp/awx_11_9_5zjzyv/:/runner/:Z\", \"--
↪env-file\", \"/tmp/awx_11_9_5zjzyv/artifacts/11/env.list\", \"--quiet\", \"--name\
↪", \"ansible_runner_11\", \"--user=root\", \"--network\", \"slirp4netns:enable_
↪ipv6=true\", \"registry.astralinux.ru/aa/aa-base-ee:0.2.1\", \"ssh-agent\", \"sh\
↪", \"-c\", \"trap 'rm -f /runner/artifacts/11/ssh_key_data' EXIT && ssh-add /
↪runner/artifacts/11/ssh_key_data && rm -f /runner/artifacts/11/ssh_key_data &&␣
↪ansible-playbook -u astra -i /runner/inventory/hosts -e @/runner/env/extravars␣
↪nginx/site.yml\"]",
  "job_cwd": "/runner/project",
  "job_env": {
    "ANSIBLE_UNSAFE_WRITES": "1",
    "AWX_ISOLATED_DATA_DIR": "/runner/artifacts/11",
    "ANSIBLE_FORCE_COLOR": "True",
    "ANSIBLE_HOST_KEY_CHECKING": "False",
```


(continues on next page)

(продолжение с предыдущей страницы)


```
    "ANSIBLE_INVENTORY_UNPARSED_FAILED": "True",
    "ANSIBLE_PARAMIKO_RECORD_HOST_KEYS": "False",
    "HOME": "/var/lib/awx",
    "AWX_PRIVATE_DATA_DIR": "/tmp/awx_11_9_5zjzyv",
    "JOB_ID": "11",
    "INVENTORY_ID": "4",
    "PROJECT_REVISION": "d7f0ff18cf8ffb44d64e8acd995402d545e2de90",
    "ANSIBLE_RETRY_FILES_ENABLED": "False",
    "MAX_EVENT_RES": "700000",
    "AWX_HOST": "https://controller",
    "ANSIBLE_SSH_CONTROL_PATH_DIR": "/runner/cp",
    "ANSIBLE_COLLECTIONS_PATHS": "/runner/requirements_collections:~/.ansible/
↪collections:/usr/share/ansible/collections",
    "ANSIBLE_ROLES_PATH": "/runner/requirements_roles:~/.ansible/roles:/usr/share/
↪ansible/roles:/etc/ansible/roles",
    "ANSIBLE_CALLBACK_PLUGINS": "/runner/artifacts/11/callback",
    "ANSIBLE_STDOUT_CALLBACK": "awx_display",
    "RUNNER_OMIT_EVENTS": "False",
    "RUNNER_ONLY_FAILED_EVENTS": "False"
  },
  "job_explanation": "",
  "execution_node": "10.2.0.5",
  "controller_node": "10.2.0.5",
  "result_traceback": "",
  "event_processing_finished": true,
  "launched_by": {
    "id": 1,
    "name": "admin",
    "type": "user",
    "url": "/api/v2/users/1/"
  },
  "work_unit_id": "EQMxvG0G",
  "job_template": 15,
  "passwords_needed_to_start": [],
  "allow_simultaneous": false,
  "artifacts": {},
  "scm_revision": "d7f0ff18cf8ffb44d64e8acd995402d545e2de90",
  "instance_group": 2,
  "diff_mode": false,
  "job_slice_number": 0,
  "job_slice_count": 1,
  "webhook_service": "",
  "webhook_credential": null,
  "webhook_guid": "",
  "host_status_counts": {
    "ok": 1
  },
  "playbook_counts": {
    "play_count": 1,
    "task_count": 16
  },
  "custom_virtualenv": null
}
```


Если в поле `failed` указано значение `false`, значит задание завершилось успешно.

Примечание: Подробное формальное описание HTTP API представлено в спецификации.

7.5 Интерфейс командной строки (CLI)

Удобным средством управления контроллером является пакет программ под общим название CLI (command line interface), который позволяет обращаться к программному интерфейсу контроллера с помощью командной строки.

CLI позволяет управлять ресурсами и настройками контроллера:

- создавать, изменять и удалять инвентарные списки, сведения об управляемых узлах, проектах, шаблонах и других ресурсах;
- управлять организациями, командами, пользователями и их ролями;
- создавать, запускать, отменять, планировать, прерывать задания и проверять статус их выполнения;
- и многое другое.

7.5.1 Установка

Для установки CLI контроллера на рабочую машину под управлением ОС Astra Linux Special Edition 1.7 выполните следующие действия:

1. В каталоге `/etc/apt/sources.list.d/` создайте файл `astra-automation.list` со ссылкой на репозиторий Astra Automation:

   ```
   deb https://dl.astralinux.ru/aa/aa-debs-for-alse-1.7 <version> main
   ```

   Вместо <version> необходимо подставить версию устанавливаемой платформы, например, `1.1-upd1`.

   Доступные версии продукта опубликованы в таблице *История обновлений*.

2. Обновите список доступных пакетов:

   ```
   sudo apt update
   ```

3. Установите пакеты:

   ```
   sudo apt install python3.9 python3.9-distutils automation-controller-cli --yes
   ```

7.5.2 Авторизация

Авторизацию пользователя на выполнение действия в контроллере рекомендуется производить с помощью токена OAuth2.0. Для токена могут быть настроены разрешения на чтение и запись, его можно отозвать в любой момент.

Чтобы авторизоваться с помощью OAuth2.0, выполните следующие действия:

1. Сгенерируйте персональный токен доступа на установочном узле контроллера:

   ```
   CONTROLLER_HOST=<address> CONTROLLER_USERNAME=<username> CONTROLLER_PASSWORD=
   ↪<password> awx login
   ```

   Здесь:

   - <address> – URL контроллера, например, `https://192.168.56.11`;
   - <username> – название учетной записи пользователя, например, `admin`;
   - <password> – пароль, например, `awx`.

   Пример вывода команды:

```
{
   "token": "70L1bQv6cj***EHkFaqSGrNZ2S5l4n"
}
```

Важно: Сохраните полученный токен в надежном месте – в случае утраты его нужно будет сгенерировать заново. Восстановить существующий токен невозможно.

Если необходимо создать токен только для чтения, добавьте в команду аргумент `--conf.scope read`:

```
CONTROLLER_HOST=https://<address> CONTROLLER_USERNAME=<username> CONTROLLER_
↪PASSWORD=<password> awx login --conf.scope read
```

Клиентская часть должна иметь возможность установить защищенное соединение с сервером, используя TLS-сертификаты, и провести проверку их подлинности. В случае отсутствия сертификата возможно возникновение ошибки:

```
Could not establish a secure connection.
Please add your server to your certificate authority.
You can also run this command by specifying -k or --conf.insecure
HTTPSConnectionPool(host='192.168.56.11', port=443): Max retries exceeded with url:␣
↪/api/ (Caused by SSLError(SSLCertVerificationError("hostname '192.168.56.11' doesn
↪'t match '=192.168.56.11'")))
```

Если верификация сертификата не требуется или невозможна, добавьте к параметрам вызова команды ключ `--conf.insecure` (`-k`).

2. Для использования токена в последующих командах сохраните его значение в переменную окружения `CONTROLLER_OAUTH_TOKEN`:

```
export CONTROLLER_OAUTH_TOKEN=<token>
```

7.5.3 Структура команды

Основу CLI контроллера составляет утилита awx, синтаксис вызова которой имеет следующий вид:

```
[<enviroments>] awx [<global-options>] <resource> <action> [<arguments>]
```

- <enviroments> – переменные окружения:
  - `CONTROLLER_HOST` – URL контроллера.
  - `CONTROLLER_VERIFY_SSL` – параметры проверки сертификата TLS. Возможные значения:
    - `true` – строгая проверка сертификата;
    - `false` – проверка сертификата не выполняется.
  - `CONTROLLER_USERNAME` – название учетной записи пользователя контроллера.
  - `CONTROLLER_PASSWORD` – пароль пользователя контроллера.
  - `CONTROLLER_OAUTH_TOKEN` – токен пользователя контроллера.

Примечание: Переменные окружения и глобальные опции, указывающие одни и те же значения, не могут использоваться одновременно.

- <global-options> – глобальные опции, которые могут применяться ко всем командам:

- **-h, --help**

  Информация о доступных опциях и командах.

- **-v, --verbose**

  Дополнительная информация о ходе выполнения команды.

- **-f, --conf.format**

  Формат вывода:
  * `json` – JSON;
  * `jaml` – YAML;
  * `jq` – формат, используемый утилитой jq[159];
  * `human` – вывод, удобный для чтения человеком.

- **--conf.host**

  URL контроллера.

- **-k, --conf.insecure**

  Разрешение небезопасного соединения с сервером при использовании сертификатов TLS.

- **--conf.username**

  Название учетной записи пользователя контроллера.

- **--conf.password**

  Пароль пользователя контроллера.

- **--conf.token**

  Токен пользователя контроллера.

- <resource> – тип объекта или ресурса. Для получения полного списка объектов и ресурсов выполните команду:

```
awx --help
```

- <action> – действие, которое необходимо выполнить с объектом или ресурсом. Ресурсы и объекты, как правило, имеют базовый набор действий:
  - `get` – получение сведений об отдельном экземпляре;
  - `list` – получение списка экземпляров;
  - `create` – создание экземпляра;
  - `modify` – изменение свойств существующего экземпляра;
  - `delete` – удаление экземпляра.

  Чтобы узнать полный список доступных действий для конкретного объекта или ресурса, выполните команду:

```
awx <resource> --help
```

- <arguments> – дополнительные параметры, которые можно передать вместе с командой. Например, идентификатор инвентаря или проекта, название задания и другие.

[159] https://jqlang.github.io/jq/

Примеры использования CLI

Вывод списка организаций:

```
awx organizations list -f human
```

Пример вывода команды:

```
id name
== ======================
1  Default
```

Вывод информации о необходимой организации:

```
awx organizations get <organizations_id>
```

Здесь <organizations_id> – идентификатор организации, например, 1.

Пример вывода команды:

```
{
  "id": 1,
  "type": "organization",
  "url": "/api/v2/organizations/1/",
  "summary_fields": {
    "created_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "modified_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "object_roles": {
      "admin_role": {
        "description": "Can manage all aspects of the organization",
        "name": "Admin",
        "id": 9,
        "user_only": true
      },
      "execute_role": {
        "description": "May run any executable resources in the organization",
        "name": "Execute",
        "id": 10
      },
      "project_admin_role": {
        "description": "Can manage all projects of the organization",
        "name": "Project Admin",
        "id": 11
      },
      "inventory_admin_role": {
        "description": "Can manage all inventories of the organization",
        "name": "Inventory Admin",
        "id": 12
      },
      "credential_admin_role": {
        "description": "Can manage all credentials of the organization",
        "name": "Credential Admin",
        "id": 13
```

(continues on next page)

(продолжение с предыдущей страницы)


```
      },
      "workflow_admin_role": {
        "description": "Can manage all workflows of the organization",
        "name": "Workflow Admin",
        "id": 14
      },
      "notification_admin_role": {
        "description": "Can manage all notifications of the organization",
        "name": "Notification Admin",
        "id": 15
      },
      "job_template_admin_role": {
        "description": "Can manage all job templates of the organization",
        "name": "Job Template Admin",
        "id": 16
      },
      "execution_environment_admin_role": {
        "description": "Can manage all execution environments of the organization",
        "name": "Execution Environment Admin",
        "id": 17
      },
      "auditor_role": {
        "description": "Can view all aspects of the organization",
        "name": "Auditor",
        "id": 18
      },
      "member_role": {
        "description": "User is a member of the organization",
        "name": "Member",
        "id": 19,
        "user_only": true
      },
      "read_role": {
        "description": "May view settings for the organization",
        "name": "Read",
        "id": 20
      },
      "approval_role": {
        "description": "Can approve or deny a workflow approval node",
        "name": "Approve",
        "id": 21
      }
    },
    "user_capabilities": {
      "edit": true,
      "delete": true
    },
    "related_field_counts": {
      "users": 2,
      "admins": 0,
      "inventories": 2,
      "teams": 3,
      "projects": 4,
      "job_templates": 2,
      "hosts": 2
    }
  },
  "created": "2024-02-16T00:06:02.349704Z",
  "modified": "2024-02-16T00:30:54.043146Z",
  "name": "Default",
  "description": "",
  "max_hosts": 0,
  "custom_virtualenv": null,
  "default_environment": null
```


(continues on next page)

(продолжение с предыдущей страницы)

```
}
```

Создание организации:

```
awx organizations create --name "<your_organization_name>"
```

Здесь <your_organization_name> – название организации, например, Saturn.

Пример вывода команды:

```
{
  "id": 4,
  "type": "organization",
  "url": "/api/v2/organizations/4/",
  "summary_fields": {
    "created_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "modified_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "object_roles": {
      "admin_role": {
        "description": "Can manage all aspects of the organization",
        "name": "Admin",
        "id": 171,
        "user_only": true
      },
      "execute_role": {
        "description": "May run any executable resources in the organization",
        "name": "Execute",
        "id": 172
      },
      "project_admin_role": {
        "description": "Can manage all projects of the organization",
        "name": "Project Admin",
        "id": 173
      },
      "inventory_admin_role": {
        "description": "Can manage all inventories of the organization",
        "name": "Inventory Admin",
        "id": 174
      },
      "credential_admin_role": {
        "description": "Can manage all credentials of the organization",
        "name": "Credential Admin",
        "id": 175
      },
      "workflow_admin_role": {
        "description": "Can manage all workflows of the organization",
        "name": "Workflow Admin",
        "id": 176
      },
      "notification_admin_role": {
        "description": "Can manage all notifications of the organization",
        "name": "Notification Admin",
```

(continues on next page)

(продолжение с предыдущей страницы)


```
        "id": 177
      },
      "job_template_admin_role": {
        "description": "Can manage all job templates of the organization",
        "name": "Job Template Admin",
        "id": 178
      },
      "execution_environment_admin_role": {
        "description": "Can manage all execution environments of the organization",
        "name": "Execution Environment Admin",
        "id": 179
      },
      "auditor_role": {
        "description": "Can view all aspects of the organization",
        "name": "Auditor",
        "id": 180
      },
      "member_role": {
        "description": "User is a member of the organization",
        "name": "Member",
        "id": 181,
        "user_only": true
      },
      "read_role": {
        "description": "May view settings for the organization",
        "name": "Read",
        "id": 182
      },
      "approval_role": {
        "description": "Can approve or deny a workflow approval node",
        "name": "Approve",
        "id": 183
      }
    },
    "user_capabilities": {
      "edit": true,
      "delete": true
    },
    "related_field_counts": {
      "inventories": 0,
      "teams": 0,
      "users": 0,
      "job_templates": 0,
      "admins": 0,
      "projects": 0
    }
  },
  "created": "2024-06-10T11:12:29.167911Z",
  "modified": "2024-06-10T11:12:29.167924Z",
  "name": "Saturn",
  "description": "",
  "max_hosts": 0,
  "custom_virtualenv": null,
  "default_environment": null
}
```


Изменение названия организации:

```
awx organizations modify <organization_id> --name "<new_name_organization>"
```

Здесь <new_name_organization> – новое название организации, например, Neptune.

Пример вывода команды:


```
{
  "id": 4,
  "type": "organization",
  "url": "/api/v2/organizations/4/",
  "summary_fields": {
    "created_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "modified_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "object_roles": {
      "admin_role": {
        "description": "Can manage all aspects of the organization",
        "name": "Admin",
        "id": 171,
        "user_only": true
      },
      "execute_role": {
        "description": "May run any executable resources in the organization",
        "name": "Execute",
        "id": 172
      },
      "project_admin_role": {
        "description": "Can manage all projects of the organization",
        "name": "Project Admin",
        "id": 173
      },
      "inventory_admin_role": {
        "description": "Can manage all inventories of the organization",
        "name": "Inventory Admin",
        "id": 174
      },
      "credential_admin_role": {
        "description": "Can manage all credentials of the organization",
        "name": "Credential Admin",
        "id": 175
      },
      "workflow_admin_role": {
        "description": "Can manage all workflows of the organization",
        "name": "Workflow Admin",
        "id": 176
      },
      "notification_admin_role": {
        "description": "Can manage all notifications of the organization",
        "name": "Notification Admin",
        "id": 177
      },
      "job_template_admin_role": {
        "description": "Can manage all job templates of the organization",
        "name": "Job Template Admin",
        "id": 178
      },
      "execution_environment_admin_role": {
        "description": "Can manage all execution environments of the organization",
        "name": "Execution Environment Admin",
        "id": 179
```


(continues on next page)

(продолжение с предыдущей страницы)


```
      },
      "auditor_role": {
        "description": "Can view all aspects of the organization",
        "name": "Auditor",
        "id": 180
      },
      "member_role": {
        "description": "User is a member of the organization",
        "name": "Member",
        "id": 181,
        "user_only": true
      },
      "read_role": {
        "description": "May view settings for the organization",
        "name": "Read",
        "id": 182
      },
      "approval_role": {
        "description": "Can approve or deny a workflow approval node",
        "name": "Approve",
        "id": 183
      }
    },
    "user_capabilities": {
      "edit": true,
      "delete": true
    },
    "related_field_counts": {
      "users": 0,
      "admins": 0,
      "inventories": 0,
      "teams": 0,
      "projects": 0,
      "job_templates": 0,
      "hosts": 0
    }
  },
  "created": "2024-06-10T11:12:29.167911Z",
  "modified": "2024-06-10T11:19:11.396969Z",
  "name": "Neptune",
  "description": "",
  "max_hosts": 0,
  "custom_virtualenv": null,
  "default_environment": null
}
```


Удаление организации:

```
awx organizations delete <organization_id>
```

7.5.4 Базовый сценарий

Базовый сценарий использования CLI Automation Controller включает в себя следующие действия:

1. Подготовка организационной структуры.
2. Подготовка инвентаря.
3. Подготовка проекта.
4. Подготовка задания.
5. Запуск задания и проверка его статуса.

Подготовка организационной структуры

Для подготовки организационной структуры выполните следующие действия:

1. Сгенерируйте персональный токен доступа с помощью команды *awx login*:

```
CONTROLLER_HOST=<address> CONTROLLER_USERNAME=<username> CONTROLLER_PASSWORD=
↪<password> awx login
```

   Подставьте собственные значения параметров вместо употребленных в примере:

   - <address> – URL контроллера, например, `https://192.168.56.11`;
   - <username> – название учетной записи пользователя, например, `admin`;
   - <password> – пароль, например, `awx`.

   Эта команда возвращает токен в формате JSON, например:

```
{
  "token": "70L1bQv6cj***EHkFaqSGrNZ2S5l4n"
}
```

   Сохраните полученный токен в надежном месте – в случае утраты его нужно будет сгенерировать заново. Восстановить существующий токен невозможно.

2. Экспортируйте токен в переменную окружения `CONTROLLER_OAUTH_TOKEN` с помощью команды *export*:

```
export CONTROLLER_OAUTH_TOKEN=<token>
```

3. Получите идентификатор среды исполнения с помощью команды *awx execution_environments list*:

```
awx execution_environments list -f human
```

   Эта команда возвращает информацию об имеющихся средах исполнения:

Пример ответа

```
id name
== ===================================
2  Control Plane Execution Environment
1  Default execution environment
```

4. Создайте организацию с помощью команды *awx organization create*:

```
awx organizations create \
   --name "<your_organization_name>" \
   --default_environment <execution_environment_id>
```

   Подставьте собственные значения параметров вместо употребленных в примере:

   - <your_organization_name> – название организации, например, `Jupiter`;
   - <execution_environment_id> – идентификатор среды исполнения, например, 1.

   Эта команда возвращает информацию о созданной организации:

Пример ответа


```
{
  "id": 3,
  "type": "organization",
  "url": "/api/v2/organizations/3/",
  "summary_fields": {
    "default_environment": {
      "id": 1,
      "name": "Default execution environment",
      "description": "",
      "image": "registry.astralinux.ru/aa/aa-base-ee:0.2.1"
    },
    "created_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "modified_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "object_roles": {
      "admin_role": {
        "description": "Can manage all aspects of the organization",
        "name": "Admin",
        "id": 115,
        "user_only": true
      },
      "execute_role": {
        "description": "May run any executable resources in the organization",
        "name": "Execute",
        "id": 116
      },
      "project_admin_role": {
        "description": "Can manage all projects of the organization",
        "name": "Project Admin",
        "id": 117
      },
      "inventory_admin_role": {
        "description": "Can manage all inventories of the organization",
        "name": "Inventory Admin",
        "id": 118
      },
      "credential_admin_role": {
        "description": "Can manage all credentials of the organization",
        "name": "Credential Admin",
        "id": 119
      },
      "workflow_admin_role": {
        "description": "Can manage all workflows of the organization",
        "name": "Workflow Admin",
        "id": 120
      },
      "notification_admin_role": {
        "description": "Can manage all notifications of the organization",
        "name": "Notification Admin",
        "id": 121
      },
      "job_template_admin_role": {
        "description": "Can manage all job templates of the organization",
        "name": "Job Template Admin",
```


(continues on next page)

(продолжение с предыдущей страницы)


```
        "id": 122
      },
      "execution_environment_admin_role": {
        "description": "Can manage all execution environments of the organization",
        "name": "Execution Environment Admin",
        "id": 123
      },
      "auditor_role": {
        "description": "Can view all aspects of the organization",
        "name": "Auditor",
        "id": 124
      },
      "member_role": {
        "description": "User is a member of the organization",
        "name": "Member",
        "id": 125,
        "user_only": true
      },
      "read_role": {
        "description": "May view settings for the organization",
        "name": "Read",
        "id": 126
      },
      "approval_role": {
        "description": "Can approve or deny a workflow approval node",
        "name": "Approve",
        "id": 127
      }
    },
    "user_capabilities": {
      "edit": true,
      "delete": true
    },
    "related_field counts": {
      "inventories": 0,
      "teams": 0,
      "users": 0,
      "job_templates": 0,
      "admins": 0,
      "projects": 0
    }
  },
  "created": "2024-06-04T20:16:52.466774Z",
  "modified": "2024-06-04T20:16:52.466805Z",
  "name": "Jupiter",
  "description": "",
  "max_hosts": 0,
  "custom_virtualenv": null,
  "default_environment": 1
}
```


5. Получите идентификатор полномочия Ansible Galaxy с помощью команды *awx credentials list*:

```
awx credentials list -f human
```

Эта команда возвращает информацию об имеющихся полномочиях:

Пример ответа

```
id name
== ===============
2  Ansible Galaxy
1  Demo Credential
```

6. Свяжите полномочие Ansible Galaxy с организацией при помощи команды *awx organization associate*:

```
awx organizations associate \
   --galaxy_credential <credential_id> \
   <organization_id>
```

Подставьте собственные значения параметров вместо употребленных в примере:

- <credential_id> – идентификатор полномочия для доступа к Ansible Galaxy, например, 2;
- <organization_id> – идентификатор организации, созданной ранее, например, 3.

Эта команда возвращает информацию о добавленном полномочии:

Пример ответа

```
{
  "count": 1,
  "next": null,
  "previous": null,
  "results": [
    {
      "id": 2,
      "type": "credential",
      "url": "/api/v2/credentials/2/",
      "related": {
        "created_by": "/api/v2/users/1/",
        "modified_by": "/api/v2/users/1/",
        "activity_stream": "/api/v2/credentials/2/activity_stream/",
        "access_list": "/api/v2/credentials/2/access_list/",
        "object_roles": "/api/v2/credentials/2/object_roles/",
        "owner_users": "/api/v2/credentials/2/owner_users/",
        "owner_teams": "/api/v2/credentials/2/owner_teams/",
        "copy": "/api/v2/credentials/2/copy/",
        "input_sources": "/api/v2/credentials/2/input_sources/",
        "credential_type": "/api/v2/credential_types/18/"
      },
      "summary_fields": {
        "credential_type": {
          "id": 18,
          "name": "Ansible Galaxy/Automation Hub API Token",
          "description": ""
        },
        "created_by": {
          "id": 1,
          "username": "admin",
          "first_name": "",
          "last_name": ""
        },
        "modified_by": {
          "id": 1,
          "username": "admin",
          "first_name": "",
          "last_name": ""
        },
```

(continues on next page)

(продолжение с предыдущей страницы)


```
        "object_roles": {
          "admin_role": {
            "description": "Can manage all aspects of the credential",
            "name": "Admin",
            "id": 29
          },
          "use_role": {
            "description": "Can use the credential in a job template",
            "name": "Use",
            "id": 30
          },
          "read_role": {
            "description": "May view settings for the credential",
            "name": "Read",
            "id": 31
          }
        },
        "user_capabilities": {
          "edit": false,
          "delete": false,
          "copy": true,
          "use": true
        },
        "owners": []
      },
      "created": "2024-02-16T00:06:03.063704Z",
      "modified": "2024-02-16T00:06:03.063729Z",
      "name": "Ansible Galaxy",
      "description": "",
      "organization": null,
      "credential_type": 18,
      "managed": true,
      "inputs": {
        "url": "https://galaxy.ansible.com/"
      },
      "kind": "galaxy_api_token",
      "cloud": false,
      "kubernetes": false
    }
  ]
}
```


Подготовка инвентаря

Для подготовки инвентаря выполните следующие действия:

1. Создайте инвентарь с помощью команды *awx inventories create*:

```
awx inventories create \
   --name "<your_inventory_name>" \
   --organization "<your_organization_name>"
```

   Подставьте собственные значения параметров вместо употребленных в примере. Здесь <your_inventory_name> – название инвентаря, например, `Demo Inventory`.

   Эта команда возвращает информацию об инвентаре:

Пример ответа


```
{
  "id": 3,
  "type": "inventory",
  "url": "/api/v2/inventories/3/",
  "summary_fields": {
    "organization": {
      "id": 3,
      "name": "Jupiter",
      "description": ""
    },
    "created_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "modified_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "object_roles": {
      "admin_role": {
        "description": "Can manage all aspects of the inventory",
        "name": "Admin",
        "id": 131
      },
      "update_role": {
        "description": "May update the inventory",
        "name": "Update",
        "id": 132
      },
      "adhoc_role": {
        "description": "May run ad hoc commands on the inventory",
        "name": "Ad Hoc",
        "id": 133
      },
      "use_role": {
        "description": "Can use the inventory in a job template",
        "name": "Use",
        "id": 134
      },
      "read_role": {
        "description": "May view settings for the inventory",
        "name": "Read",
        "id": 135
      }
    },
    "user_capabilities": {
      "edit": true,
      "delete": true,
      "copy": true,
      "adhoc": true
    },
    "labels": {
      "count": 0,
      "results": []
    }
  },
  "created": "2024-06-04T20:34:48.855589Z",
  "modified": "2024-06-04T20:34:48.855626Z",
  "name": "Demo Inventory",
```


(continues on next page)

(продолжение с предыдущей страницы)

```
  "description": "",
  "organization": 3,
  "kind": "",
  "host_filter": null,
  "variables": "",
  "has_active_failures": false,
  "total_hosts": 0,
  "hosts_with_active_failures": 0,
  "total_groups": 0,
  "has_inventory_sources": false,
  "total_inventory_sources": 0,
  "inventory_sources_with_failures": 0,
  "pending_deletion": false,
  "prevent_instance_group_fallback": false
}
```

2. Добавьте управляемый узел в инвентарь с помощью команды *awx host create*:

```
awx host create \
   --name "<node_name>" \
   --inventory <inventory_id> \
   --variables "<node_variables>"
```

Подставьте собственные значения параметров вместо употребленных в примере:

- <node_name> – название управляемого узла, например, `node01`;
- <inventory_id> – идентификатор инвентаря, например, 3;
- <node_variables> – переменные для доступа к управляемому узлу, например, `{'ansible_host': '192.168.56.101', 'ansible_user': 'vagrant'}`.

Эта команда возвращает информацию об управляемом узле:

Пример ответа

```
{
  "id": 15,
  "type": "host",
  "url": "/api/v2/hosts/15/",
  "summary_fields": {
    "inventory": {
      "id": 3,
      "name": "Demo Inventory",
      "description": "",
      "has_active_failures": false,
      "total_hosts": 2,
      "hosts_with_active_failures": 0,
      "total_groups": 0,
      "has_inventory_sources": false,
      "total_inventory_sources": 0,
      "inventory_sources_with_failures": 0,
      "organization_id": 3,
      "kind": ""
    },
    "created_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "modified_by": {
      "id": 1,
      "username": "admin",
```

(continues on next page)

(продолжение с предыдущей страницы)

```
      "first_name": "",
      "last_name": ""
    },
    "user_capabilities": {
      "edit": true,
      "delete": true
    },
    "groups": {
      "count": 0,
      "results": []
    },
    "recent_jobs": []
  },
  "created": "2024-06-13T12:18:36.814082Z",
  "modified": "2024-06-13T12:18:36.814093Z",
  "name": "node01",
  "description": "",
  "inventory": 3,
  "enabled": true,
  "instance_id": "",
  "variables": "{\"ansible_host\": \"192.168.56.101\", \"ansible_user\": \"vagrant\
↪"}",
  "has_active_failures": false,
  "has_inventory_sources": false,
  "last_job": null,
  "last_job_host_summary": null,
  "ansible_facts_modified": null
}
```

3. Создайте полномочие для доступа к управляемому узлу с помощью команды *awx credentials create*:

```
awx credentials create \
   --name '<credential_name>' \
   --credential_type 'Machine' \
   --organization <organization_id> \
   --inputs '<node_inputs>'
```

Подставьте собственные значения параметров вместо употребленных в примере:

- <credential_name> – название полномочия для доступа к управляемому узлу, например, `NGINX`;
- <node_inputs> – данные для доступа к управляемому узлу, например, `{"username": "vagrant", "password": "vagrant"}`.

Эта команда возвращает информацию о созданном полномочии:

Пример ответа

```
{
  "id": 4,
  "type": "credential",
  "url": "/api/v2/credentials/4/",
  "summary_fields": {
    "organization": {
      "id": 3,
      "name": "Jupiter",
      "description": ""
    },
    "credential_type": {
      "id": 1,
      "name": "Machine",
```

(continues on next page)

(продолжение с предыдущей страницы)


```
      "description": ""
    },
    "created_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "modified_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "object_roles": {
      "admin_role": {
        "description": "Can manage all aspects of the credential",
        "name": "Admin",
        "id": 197
      },
      "use_role": {
        "description": "Can use the credential in a job template",
        "name": "Use",
        "id": 198
      },
      "read_role": {
        "description": "May view settings for the credential",
        "name": "Read",
        "id": 199
      }
    },
    "user_capabilities": {
      "edit": true,
      "delete": true,
      "copy": true,
      "use": true
    },
    "owners": [
      {
        "id": 3,
        "type": "organization",
        "name": "Jupiter",
        "description": "",
        "url": "/api/v2/organizations/3/"
      }
    ]
  },
  "created": "2024-06-17T14:34:40.684019Z",
  "modified": "2024-06-17T14:34:40.684031Z",
  "name": "NGINX",
  "description": "",
  "organization": 3,
  "credential_type": 1,
  "managed": false,
  "inputs": {
    "password": "$encrypted$",
    "username": "vagrant"
  },
  "kind": "ssh",
  "cloud": false,
  "kubernetes": false
}
```

Подготовка проекта

Для подготовки проекта выполните следующие действия:

1. Создайте полномочие для доступа к Private Automation Hub с помощью команды *awx credentials create*

```
awx credentials create \
   --name '<credential_name>' \
   --credential_type 'Ansible Hub' \
   --organization <organization_id> \
   --inputs '<token>'
```

Подставьте собственные значения параметров вместо употребленных в примере:

- <credential_name> – название полномочия для доступа к Private Automation Hub, например, `Private Automation Hub`.
- <token> – токен для доступа к Private Automation Hub.

Примечание: С подробной информацией о создании токена доступа можно ознакомиться в секции *Токен API*.

Эта команда возвращает информацию о созданном полномочии:

Пример ответа

```
{
  "id": 5,
  "type": "credential",
  "url": "/api/v2/credentials/5/",
  "summary_fields": {
    "organization": {
      "id": 3,
      "name": "Jupiter",
      "description": ""
    },
    "credential_type": {
      "id": 2,
      "name": "Source Control",
      "description": ""
    },
    "created_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "modified_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "object_roles": {
      "admin_role": {
        "description": "Can manage all aspects of the credential",
        "name": "Admin",
        "id": 200
      },
      "use_role": {
        "description": "Can use the credential in a job template",
        "name": "Use",
```

(continues on next page)

(продолжение с предыдущей страницы)

```
        "id": 201
      },
      "read_role": {
        "description": "May view settings for the credential",
        "name": "Read",
        "id": 202
      }
    },
    "user_capabilities": {
      "edit": true,
      "delete": true,
      "copy": true,
      "use": true
    },
    "owners": [
      {
        "id": 3,
        "type": "organization",
        "name": "Jupiter",
        "description": "",
        "url": "/api/v2/organizations/3/"
      }
    ]
  },
  "created": "2024-06-17T15:33:12.168817Z",
  "modified": "2024-06-17T15:33:12.168837Z",
  "name": "Automation Hub",
  "description": "",
  "organization": 3,
  "credential_type": 2,
  "managed": false,
  "inputs": {
    "ssh_key_data": "$encrypted$",
    "ssh_key_unlock": "$encrypted$"
  },
  "kind": "scm",
  "cloud": false,
  "kubernetes": false
}
```

2. Создайте проект с помощью команды *awx projects create*:

```
awx projects create \
   --wait \
   --organization <organization_id> \
   --name='<your_project_name>' \
   --scm_type git \
   --scm_url 'ssh://git@source.astragroup.ru:2222/aa-gca/AA/aac-samples.git' \
   --credential <credential_id>
```

Подставьте собственные значения параметров вместо употребленных в примере:

- <organization_id> – идентификатор организации, созданной ранее, например, 3;
- <your_project_name> – название проекта, например, `Demo NGINX Project`;
- <credential_id> – идентификатор полномочия для доступа к Private Automation Hub, например, 5.

Примечание: В качестве источника при создании проекта используется репозиторий с примерами[160].

Эта команда возвращает информацию о созданном проекте:

[160] https://source.astragroup.ru/

Пример ответа


```
{
  "id": 18,
  "type": "project",
  "url": "/api/v2/projects/18/",
  "summary_fields": {
    "organization": {
      "id": 3,
      "name": "Jupiter",
      "description": ""
    },
    "credential": {
      "id": 4,
      "name": "Automation Hub",
      "description": "",
      "kind": "scm",
      "cloud": false,
      "kubernetes": false,
      "credential_type_id": 2
    },
    "current_update": {
      "id": 53,
      "name": "Demo NGINX Project",
      "description": "",
      "status": "pending",
      "failed": false
    },
    "current_job": {
      "id": 53,
      "name": "Demo NGINX Project",
      "description": "",
      "status": "pending",
      "failed": false
    },
    "created_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "modified_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "object_roles": {
      "admin_role": {
        "description": "Can manage all aspects of the project",
        "name": "Admin",
        "id": 143
      },
      "use_role": {
        "description": "Can use the project in a job template",
        "name": "Use",
        "id": 144
      },
      "update_role": {
        "description": "May update the project",
        "name": "Update",
        "id": 145
      },
      "read_role": {
        "description": "May view settings for the project",
```


(continues on next page)

(продолжение с предыдущей страницы)


```
        "name": "Read",
        "id": 146
      }
    },
    "user_capabilities": {
      "edit": true,
      "delete": true,
      "start": true,
      "schedule": true,
      "copy": true
    }
  },
  "created": "2024-06-09T21:03:24.140327Z",
  "modified": "2024-06-09T21:03:24.140351Z",
  "name": "Demo NGINX Project",
  "description": "",
  "local_path": "_18__demo_nginx_project",
  "scm_type": "git",
  "scm_url": "ssh://git@hub.astra-automation.ru:2222/aa-gca/AA/aac-samples.git",
  "scm_branch": "",
  "scm_refspec": "",
  "scm_clean": false,
  "scm_track_submodules": false,
  "scm_delete_on_update": false,
  "credential": 4,
  "timeout": 0,
  "scm_revision": "",
  "last_job_run": null,
  "last_job_failed": false,
  "next_job_run": null,
  "status": "pending",
  "organization": 3,
  "scm_update_on_launch": false,
  "scm_update_cache_timeout": 0,
  "allow_override": false,
  "custom_virtualenv": null,
  "default_environment": null,
  "signature_validation_credential": null,
  "last_update_failed": false,
  "last_updated": null
}
```


Подготовка задания

Для подготовки задания выполните следующие действия:

1. Создайте шаблон задания для установки NGINX с помощью команды *awx job_templates create*:

```
awx job_templates create \
   --name='<job_templates_name>' \
   --project '<your_project_name>' \
   --playbook nginx/site.yml \
   --inventory '<your_inventory_id>'
```

   Подставьте собственные значения параметров вместо употребленных в примере:

   - <job_templates_name> – название шаблона задания, например, `NGINX Deployment`;
   - <your_inventory_id> – идентификатор инвентаря, созданного ранее, например, 3.

   Эта команда возвращает информацию о созданном шаблоне задания в формате:

Пример ответа


```
{
  "id": 19,
  "type": "job_template",
  "url": "/api/v2/job_templates/19/",
  "summary_fields": {
    "organization": {
      "id": 3,
      "name": "Jupiter",
      "description": ""
    },
    "inventory": {
      "id": 3,
      "name": "Demo Inventory",
      "description": "",
      "has_active_failures": false,
      "total_hosts": 1,
      "hosts_with_active_failures": 0,
      "total_groups": 0,
      "has_inventory_sources": false,
      "total_inventory_sources": 0,
      "inventory_sources_with_failures": 0,
      "organization_id": 3,
      "kind": ""
    },
    "project": {
      "id": 18,
      "name": "Demo NGINX Project",
      "description": "",
      "status": "successful",
      "scm_type": "git",
      "allow_override": false
    },
    "created_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "modified_by": {
      "id": 1,
      "username": "admin",
      "first_name": "",
      "last_name": ""
    },
    "object_roles": {
      "admin_role": {
        "description": "Can manage all aspects of the job template",
        "name": "Admin",
        "id": 147
      },
      "execute_role": {
        "description": "May run the job template",
        "name": "Execute",
        "id": 148
      },
      "read_role": {
        "description": "May view settings for the job template",
        "name": "Read",
        "id": 149
      }
    },
    "user_capabilities": {
      "edit": true,
```


(continues on next page)

(продолжение с предыдущей страницы)


```
      "delete": true,
      "start": true,
      "schedule": true,
      "copy": true
    },
    "labels": {
      "count": 0,
      "results": []
    },
    "recent_jobs": [],
    "credentials": []
  },
  "created": "2024-06-09T21:06:16.520636Z",
  "modified": "2024-06-09T21:06:16.520659Z",
  "name": "NGINX Deployment",
  "description": "",
  "job_type": "run",
  "inventory": 3,
  "project": 18,
  "playbook": "nginx/site.yml",
  "scm_branch": "",
  "forks": 0,
  "limit": "",
  "verbosity": 0,
  "extra_vars": "",
  "job_tags": "",
  "force_handlers": false,
  "skip_tags": "",
  "start_at_task": "",
  "timeout": 0,
  "use_fact_cache": false,
  "organization": 3,
  "last_job_run": null,
  "last_job_failed": false,
  "next_job_run": null,
  "status": "never updated",
  "execution_environment": null,
  "host_config_key": "",
  "ask_scm_branch_on_launch": false,
  "ask_diff_mode_on_launch": false,
  "ask_variables_on_launch": false,
  "ask_limit_on_launch": false,
  "ask_tags_on_launch": false,
  "ask_skip_tags_on_launch": false,
  "ask_job_type_on_launch": false,
  "ask_verbosity_on_launch": false,
  "ask_inventory_on_launch": false,
  "ask_credential_on_launch": false,
  "ask_execution_environment_on_launch": false,
  "ask_labels_on_launch": false,
  "ask_forks_on_launch": false,
  "ask_job_slice_count_on_launch": false,
  "ask_timeout_on_launch": false,
  "ask_instance_groups_on_launch": false,
  "survey_enabled": false,
  "become_enabled": false,
  "diff_mode": false,
  "allow_simultaneous": false,
  "custom_virtualenv": null,
  "job_slice_count": 1,
  "webhook_service": "",
  "webhook_credential": null,
  "prevent_instance_group_fallback": false
}
```

2. Свяжите шаблон задания с полномочием для доступа к управляемому узлу с помощью команды *awx job_templates associate*:

```
awx job_templates associate \
   --id <job_template_id> \
   --credential <credential_id>
```

Подставьте собственные значения параметров вместо употребленных в примере:

- <job_template_id> – идентификатор созданного ранее шаблона задания, например, 19;
- <credential_id> – идентификатор полномочия для доступа к управляемому узлу, например, 4.

Эта команда возвращает информацию о добавленном полномочии:

Пример ответа


```
{
  "count": 1,
  "next": null,
  "previous": null,
  "results": [
    {
      "id": 1,
      "type": "credential",
      "url": "/api/v2/credentials/1/",
      "related": {
        "created_by": "/api/v2/users/1/",
        "modified_by": "/api/v2/users/1/",
        "activity_stream": "/api/v2/credentials/1/activity_stream/",
        "access_list": "/api/v2/credentials/1/access_list/",
        "object_roles": "/api/v2/credentials/1/object_roles/",
        "owner_users": "/api/v2/credentials/1/owner_users/",
        "owner_teams": "/api/v2/credentials/1/owner_teams/",
        "copy": "/api/v2/credentials/1/copy/",
        "input_sources": "/api/v2/credentials/1/input_sources/",
        "credential_type": "/api/v2/credential_types/1/",
        "user": "/api/v2/users/1/"
      },
      "summary_fields": {
        "credential_type": {
          "id": 1,
          "name": "Machine",
          "description": ""
        },
        "created_by": {
          "id": 1,
          "username": "admin",
          "first_name": "",
          "last_name": ""
        },
        "modified_by": {
          "id": 1,
          "username": "admin",
          "first_name": "",
          "last_name": ""
        },
        "object_roles": {
          "admin_role": {
            "description": "Can manage all aspects of the credential",
            "name": "Admin",
            "id": 26
          },
```


(continues on next page)

(продолжение с предыдущей страницы)


```
          "use_role": {
            "description": "Can use the credential in a job template",
            "name": "Use",
            "id": 27
          },
          "read_role": {
            "description": "May view settings for the credential",
            "name": "Read",
            "id": 28
          }
        },
        "user_capabilities": {
          "edit": true,
          "delete": true,
          "copy": true,
          "use": true
        },
        "owners": [
          {
            "id": 1,
            "type": "user",
            "name": "admin",
            "description": " ",
            "url": "/api/v2/users/1/"
          }
        ]
      },
      "created": "2024-02-16T00:06:02.881275Z",
      "modified": "2024-02-16T00:06:02.881304Z",
      "name": "Demo Credential",
      "description": "",
      "organization": null,
      "credential_type": 1,
      "managed": false,
      "inputs": {
        "username": "admin"
      },
      "kind": "ssh",
      "cloud": false,
      "kubernetes": false
    }
  ]
}
```


Запуск задания и проверка его статуса

Для запуска и проверки статуса задания воспользуйтесь командой *awx job_templates launch*:

```
awx job_templates launch '<job_templates_name>' --monitor -f human
```

Эта команда возвращает информацию о запущенном шаблоне задания:

Пример ответа

```
------Starting Standard Out Stream------
SSH password:

PLAY [Install NGINX server] ****************************************************

TASK [Gathering Facts] *********************************************************
[WARNING]: Platform linux on host node01 is using the discovered Python
interpreter at /usr/bin/python3.9, but future installation of another Python
interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-
core/2.13/reference_appendices/interpreter_discovery.html for more information.
ok: [node01]

TASK [astra.nginx.nginx : Install required nginx version.] *********************
ok: [node01]

TASK [astra.nginx.nginx : Install modules] *************************************

TASK [astra.nginx.nginx : Remove default nginx vhost config file (if configured).] ***
skipping: [node01]

TASK [astra.nginx.nginx : Ensure nginx_ssl_conf_path exists.] ******************
ok: [node01]

TASK [astra.nginx.nginx : Add ssl config file.] ********************************

TASK [astra.nginx.nginx : Remove ssl config file.] *****************************

TASK [astra.nginx.nginx : Add managed vhost config files.] *********************

TASK [astra.nginx.nginx : Remove managed vhost config files.] ******************

TASK [astra.nginx.nginx : Copy nginx configuration in place.] ******************
ok: [node01]

TASK [astra.nginx.nginx : Ensure nginx_service_config_path exists.] ************
ok: [node01]

TASK [astra.nginx.nginx : Copy nginx service config in place.] *****************
skipping: [node01]

TASK [astra.nginx.nginx : Add nginx snippets] **********************************

TASK [astra.nginx.nginx : Copy styled error pages] *****************************
ok: [node01] => (item=403.html)
ok: [node01] => (item=404.html)
ok: [node01] => (item=500.html)
ok: [node01] => (item=underconstruction.html)

TASK [astra.nginx.nginx : Ensure nginx service is running as configured.] ******
ok: [node01]

TASK [Create index.html] *******************************************************
ok: [node01]

PLAY RECAP *********************************************************************
node01                     : ok=8    changed=0    unreachable=0    failed=0    skipped=8␣
↪   rescued=0    ignored=0
------End of Standard Out Stream--------

id name
== ===================
68 Nginx Job Template4
```

Если в поле `failed` указано значение `0`, значит задание завершилось успешно.

7.5.5 CLI specification

Здесь представлена полная формальная спецификация интерфейса командной строки (*CLI*), представляемого утилитой awx. Подобное описание для любого сочетания ресурс-действие (resource-action) можно получить с помощью аргумента `--help`, выполнив соответствующую команду по отношению к работающему контроллеру.

Для получения синтаксиса команды по управлению конкретным ресурсом выберите в панели навигации слева название этого ресурса. Локальная панель навигации справа позволит переключиться на описание синтаксиса необходимого действия.

Ниже представлен обобщенный синтаксис команд утилиты awx.

```
usage: awx [--help] [--version] [--conf.host https://example.awx.org]
           [--conf.token TEXT] [--conf.username TEXT] [--conf.password TEXT]
           [-k] [-f {json,yaml,jq,human}] [--filter TEXT]
           [--conf.color BOOLEAN] [-v]
           resource ...

positional arguments:
  resource
    login               authenticate and retrieve an OAuth2 token
    config              print current configuration values
    import              import resources into Tower
    export              export resources from Tower
    ping
    instances (instance)
    instance_groups (instance_group)
    settings (setting)
    me
    organizations (organization)
    users (user)
    execution_environments (execution_environment)
    projects (project)
    project_updates (project_update)
    teams (team)
    credentials (credential)
    credential_types (credential_type)
    credential_input_sources
    applications (application)
    tokens
    metrics
    inventory (inventories)
    constructed_inventory
    inventory_sources (inventory_source)
    inventory_updates (inventory_update)
    groups (group)
    hosts (host)
    host_metrics
    host_metric_summary_monthly
    job_templates (job_template)
    jobs (job)
    ad_hoc_commands (ad_hoc)
    system_job_templates
    system_jobs
    schedules (schedule)
    roles (role)
    notification_templates (notification_template)
    notifications
    labels (label)
    unified_job_templates
    unified_jobs
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    activity_stream
    workflow_job_templates (workflow)
    workflow_jobs (workflow_job)
    workflow_approvals
    workflow_job_template_nodes (node)
    workflow_job_nodes
    mesh_visualizer
    bulk
    analytics

optional arguments:
  --help                prints usage information for the awx tool
  --version             display awx CLI version

authentication:
  --conf.host https://example.awx.org
  --conf.token TEXT     an OAuth2.0 token (get one by using `awx login`)
  --conf.username TEXT
  --conf.password TEXT
  -k, --conf.insecure   Allow insecure server connections when using SSL

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made
```

activity_stream

```
usage: awx activity_stream [-h] action ...

positional arguments:
  action
    list
    get

optional arguments:
  -h, --help  show this help message and exit

awx activity_stream: the following arguments are required: action
```

activity_stream list

```
usage: awx activity_stream list [-h] [--all] [--order_by ORDER_BY]
                                [-f {json,yaml,jq,human}] [--filter TEXT]
                                [--conf.color BOOLEAN] [-v]
                                [--type {activity_stream}]
                                [--timestamp TIMESTAMP]
                                [--operation {create,update,delete,associate,
↪disassociate}]
                                [--changes JSON/YAML] [--object1 TEXT]
                                [--object2 TEXT] [--action_node TEXT]

optional arguments:
  -h, --help            show this help message and exit
  --all                 fetch all pages of content from the API when returning
                        results (instead of just the first page)
  --order_by ORDER_BY   order results by given field name, prefix the field
```

(continues on next page)

(продолжение с предыдущей страницы)

```
                        name with a dash (-) to sort in reverse eg
                        --order_by='-name',multiple sorting fields may be
                        specified by separating the field names with a comma
                        (,)
  --type {activity_stream}
                        only list activity_stream with the specified type
  --timestamp TIMESTAMP
                        only list activity_stream with the specified timestamp
  --operation {create,update,delete,associate,disassociate}
                        only list activity_stream with the specified operation
  --changes JSON/YAML   only list activity_stream with the specified changes
  --object1 TEXT        only list activity_stream with the specified object1
  --object2 TEXT        only list activity_stream with the specified object2
  --action_node TEXT    only list activity_stream with the specified
                        action_node

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made
```

activity_stream get

```
usage: awx activity_stream get [-h] [-f {json,yaml,jq,human}] [--filter TEXT]
                               [--conf.color BOOLEAN] [-v]
                               id

positional arguments:
  id                    the ID (or unique name) of the resource

optional arguments:
  -h, --help            show this help message and exit

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made

awx activity_stream get: the following arguments are required: id
```

ad_hoc_commands

```
usage: awx ad_hoc_commands [-h] action ...

positional arguments:
  action
    list
    create
    get
    delete
    stdout

optional arguments:
  -h, --help  show this help message and exit
```

(continues on next page)

(продолжение с предыдущей страницы)

```
awx ad_hoc_commands: the following arguments are required: action
```

ad_hoc_commands list

```
usage: awx ad_hoc_commands list [-h] [--all] [--order_by ORDER_BY]
                                [-f {json,yaml,jq,human}] [--filter TEXT]
                                [--conf.color BOOLEAN] [-v]
                                [--type {ad_hoc_command}] [--created CREATED]
                                [--modified MODIFIED] [--name TEXT]
                                [--launch_type {manual,relaunch,callback,scheduled,
↪dependency,workflow,webhook,sync,scm}]
                                [--status {new,pending,waiting,running,successful,failed,
↪error,canceled}]
                                [--execution_environment ID]
                                [--failed BOOLEAN] [--started STARTED]
                                [--finished FINISHED]
                                [--canceled_on CANCELED_ON]
                                [--elapsed ELAPSED] [--job_explanation TEXT]
                                [--execution_node TEXT]
                                [--controller_node TEXT] [--work_unit_id TEXT]
                                [--job_type {run,check}] [--inventory ID]
                                [--limit TEXT] [--credential ID]
                                [--module_name {command,shell,yum,apt,apt_key,apt_
↪repository,apt_rpm,service,group,user,mount,ping,selinux,setup,win_ping,win_service,
↪win_updates,win_group,win_user}]
                                [--module_args TEXT] [--forks INTEGER]
                                [--verbosity {0,1,2,3,4,5}]
                                [--extra_vars TEXT] [--become_enabled BOOLEAN]
                                [--diff_mode BOOLEAN]

optional arguments:
  -h, --help            show this help message and exit
  --all                 fetch all pages of content from the API when returning
                        results (instead of just the first page)
  --order_by ORDER_BY   order results by given field name, prefix the field
                        name with a dash (-) to sort in reverse eg
                        --order_by='-name',multiple sorting fields may be
                        specified by separating the field names with a comma
                        (,)
  --type {ad_hoc_command}
                        only list ad_hoc_commands with the specified type
  --created CREATED     only list ad_hoc_commands with the specified created
  --modified MODIFIED   only list ad_hoc_commands with the specified modified
  --name TEXT           only list ad_hoc_commands with the specified name
  --launch_type {manual,relaunch,callback,scheduled,dependency,workflow,webhook,sync,scm}
                        only list ad_hoc_commands with the specified
                        launch_type
  --status {new,pending,waiting,running,successful,failed,error,canceled}
                        only list ad_hoc_commands with the specified status
  --execution_environment ID
                        only list ad_hoc_commands with the specified
                        execution_environment
  --failed BOOLEAN      only list ad_hoc_commands with the specified failed
  --started STARTED     only list ad_hoc_commands with the specified started
  --finished FINISHED   only list ad_hoc_commands with the specified finished
  --canceled_on CANCELED_ON
                        only list ad_hoc_commands with the specified
                        canceled_on
  --elapsed ELAPSED     only list ad_hoc_commands with the specified elapsed
  --job_explanation TEXT
                        only list ad_hoc_commands with the specified
```

(continues on next page)

(продолжение с предыдущей страницы)

```
                        job_explanation
  --execution_node TEXT
                        only list ad_hoc_commands with the specified
                        execution_node
  --controller_node TEXT
                        only list ad_hoc_commands with the specified
                        controller_node
  --work_unit_id TEXT   only list ad_hoc_commands with the specified
                        work_unit_id
  --job_type {run,check}
                        only list ad_hoc_commands with the specified job_type
  --inventory ID        only list ad_hoc_commands with the specified inventory
  --limit TEXT          only list ad_hoc_commands with the specified limit
  --credential ID       only list ad_hoc_commands with the specified
                        credential
  --module_name {command,shell,yum,apt,apt_key,apt_repository,apt_rpm,service,group,user,
↪mount,ping,selinux,setup,win_ping,win_service,win_updates,win_group,win_user}
                        only list ad_hoc_commands with the specified
                        module_name
  --module_args TEXT    only list ad_hoc_commands with the specified
                        module_args
  --forks INTEGER       only list ad_hoc_commands with the specified forks
  --verbosity {0,1,2,3,4,5}
                        only list ad_hoc_commands with the specified verbosity
  --extra_vars TEXT     only list ad_hoc_commands with the specified
                        extra_vars
  --become_enabled BOOLEAN
                        only list ad_hoc_commands with the specified
                        become_enabled
  --diff_mode BOOLEAN   only list ad_hoc_commands with the specified diff_mode

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made
```

ad_hoc_commands create

```
usage: awx ad_hoc_commands create [-h] [--monitor]
                                  [--action-timeout ACTION_TIMEOUT] [--wait]
                                  [--interval INTERVAL]
                                  [--execution_environment ID]
                                  [--job_type {run,check}] --inventory ID
                                  [--limit TEXT] --credential ID
                                  [--module_name {command,shell,yum,apt,apt_key,apt_
↪repository,apt_rpm,service,group,user,mount,ping,selinux,setup,win_ping,win_service,
↪win_updates,win_group,win_user}]
                                  [--module_args TEXT] [--forks INTEGER]
                                  [--verbosity {0,1,2,3,4,5}]
                                  [--extra_vars TEXT]
                                  [--become_enabled BOOLEAN]
                                  [--diff_mode BOOLEAN]

required arguments:
  --inventory ID        the ID of the associated inventory
  --credential ID       the ID of the associated credential

optional arguments:
  -h, --help            show this help message and exit
```

(continues on next page)

(продолжение с предыдущей страницы)

```
  --monitor             If set, prints stdout of the launched job until it
                        finishes.
  --action-timeout ACTION_TIMEOUT
                        If set with --monitor or --wait, time out waiting on
                        job completion.
  --wait                If set, waits until the launched job finishes.
  --interval INTERVAL   If set with --monitor or --wait, amount of time to
                        wait in seconds between api calls. Minimum value is
                        2.5 seconds to avoid overwhelming the api
  --execution_environment ID
                        The container image to be used for execution.
  --job_type {run,check}
  --limit TEXT
  --module_name {command,shell,yum,apt,apt_key,apt_repository,apt_rpm,service,group,user,
↪mount,ping,selinux,setup,win_ping,win_service,win_updates,win_group,win_user}
  --module_args TEXT
  --forks INTEGER
  --verbosity {0,1,2,3,4,5}
  --extra_vars TEXT
  --become_enabled BOOLEAN
  --diff_mode BOOLEAN

awx ad_hoc_commands create: the following arguments are required: --inventory, --
↪credential
```

ad_hoc_commands get

```
usage: awx ad_hoc_commands get [-h] [-f {json,yaml,jq,human}] [--filter TEXT]
                               [--conf.color BOOLEAN] [-v]
                               id

positional arguments:
  id                    the ID (or unique name) of the resource

optional arguments:
  -h, --help            show this help message and exit

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made

awx ad_hoc_commands get: the following arguments are required: id
```

ad_hoc_commands delete

```
usage: awx ad_hoc_commands delete [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx ad_hoc_commands delete: the following arguments are required: id
```

ad_hoc_commands stdout

```
usage: awx ad_hoc_commands stdout [-h] id

positional arguments:
  id

optional arguments:
  -h, --help  show this help message and exit

awx ad_hoc_commands stdout: the following arguments are required: id
```

analytics

```
usage: awx analytics [-h] action ...

positional arguments:
  action
    list
    get

optional arguments:
  -h, --help  show this help message and exit

awx analytics: the following arguments are required: action
```

analytics list

```
usage: awx analytics list [-h] [--all] [--order_by ORDER_BY]
                          [-f {json,yaml,jq,human}] [--filter TEXT]
                          [--conf.color BOOLEAN] [-v]

optional arguments:
  -h, --help            show this help message and exit
  --all                 fetch all pages of content from the API when returning
                        results (instead of just the first page)
  --order_by ORDER_BY   order results by given field name, prefix the field
                        name with a dash (-) to sort in reverse eg
                        --order_by='-name',multiple sorting fields may be
                        specified by separating the field names with a comma
                        (,)

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made
```

analytics get

```
usage: awx analytics get [-h] [-f {json,yaml,jq,human}] [--filter TEXT]
                         [--conf.color BOOLEAN] [-v]
                         id

positional arguments:
  id                    the ID (or unique name) of the resource

optional arguments:
  -h, --help            show this help message and exit

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made

awx analytics get: the following arguments are required: id
```

applications

```
usage: awx applications [-h] action ...

positional arguments:
  action
    list
    create
    get
    modify
    delete

optional arguments:
  -h, --help  show this help message and exit

awx applications: the following arguments are required: action
```

applications list

```
usage: awx applications list [-h] [--all] [--order_by ORDER_BY]
                             [-f {json,yaml,jq,human}] [--filter TEXT]
                             [--conf.color BOOLEAN] [-v]
                             [--type {o_auth2_application}]
                             [--created CREATED] [--name TEXT]
                             [--description TEXT] [--client_id TEXT]
                             [--client_secret TEXT]
                             [--client_type {confidential,public}]
                             [--redirect_uris TEXT]
                             [--authorization_grant_type {authorization-code,password}]
                             [--skip_authorization BOOLEAN]
                             [--organization ID]

optional arguments:
  -h, --help            show this help message and exit
  --all                 fetch all pages of content from the API when returning
                        results (instead of just the first page)
  --order_by ORDER_BY   order results by given field name, prefix the field
                        name with a dash (-) to sort in reverse eg
```

(continues on next page)

(продолжение с предыдущей страницы)

```
                        --order_by='-name',multiple sorting fields may be
                        specified by separating the field names with a comma
                        (,)
  --type {o_auth2_application}
                        only list applications with the specified type
  --created CREATED     only list applications with the specified created
  --name TEXT           only list applications with the specified name
  --description TEXT    only list applications with the specified description
  --client_id TEXT      only list applications with the specified client_id
  --client_secret TEXT  only list applications with the specified
                        client_secret
  --client_type {confidential,public}
                        only list applications with the specified client_type
  --redirect_uris TEXT  only list applications with the specified
                        redirect_uris
  --authorization_grant_type {authorization-code,password}
                        only list applications with the specified
                        authorization_grant_type
  --skip_authorization BOOLEAN
                        only list applications with the specified
                        skip_authorization
  --organization ID     only list applications with the specified organization

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made
```

applications create

```
usage: awx applications create [-h] --name TEXT [--description TEXT]
                               --client_type {confidential,public}
                               [--redirect_uris TEXT]
                               --authorization_grant_type
                               {authorization-code,password}
                               [--skip_authorization BOOLEAN] --organization
                               ID

required arguments:
  --name TEXT           Name of this application.
  --client_type {confidential,public}
                        Set to Public or Confidential depending on how secure
                        the client device is.
  --authorization_grant_type {authorization-code,password}
                        The Grant type the user must use for acquire tokens
                        for this application.
  --organization ID     Organization containing this application.

optional arguments:
  -h, --help            show this help message and exit
  --description TEXT    Optional description of this application.
  --redirect_uris TEXT  Allowed URIs list, space separated
  --skip_authorization BOOLEAN
                        Set True to skip authorization step for completely
                        trusted applications.

awx applications create: the following arguments are required: --name, --client_type, --
↪authorization_grant_type, --organization
```

applications get

```
usage: awx applications get [-h] [-f {json,yaml,jq,human}] [--filter TEXT]
                            [--conf.color BOOLEAN] [-v]
                            id

positional arguments:
  id                    the ID (or unique name) of the resource

optional arguments:
  -h, --help            show this help message and exit

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made

awx applications get: the following arguments are required: id
```

applications modify

```
usage: awx applications modify [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx applications modify: the following arguments are required: id
```

applications delete

```
usage: awx applications delete [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx applications delete: the following arguments are required: id
```

bulk

```
usage: awx bulk [-h] action ...

positional arguments:
  action
    list
    get
    job_launch
    host_create
    host_delete
```

(continues on next page)

(продолжение с предыдущей страницы)

```
optional arguments:
  -h, --help   show this help message and exit

awx bulk: the following arguments are required: action
```

bulk list

```
usage: awx bulk list [-h] [--all] [--order_by ORDER_BY]
                     [-f {json,yaml,jq,human}] [--filter TEXT]
                     [--conf.color BOOLEAN] [-v]

optional arguments:
  -h, --help            show this help message and exit
  --all                 fetch all pages of content from the API when returning
                        results (instead of just the first page)
  --order_by ORDER_BY   order results by given field name, prefix the field
                        name with a dash (-) to sort in reverse eg
                        --order_by='-name',multiple sorting fields may be
                        specified by separating the field names with a comma
                        (,)

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made
```

bulk get

```
usage: awx bulk get [-h] [-f {json,yaml,jq,human}] [--filter TEXT]
                    [--conf.color BOOLEAN] [-v]
                    id

positional arguments:
  id                    the ID (or unique name) of the resource

optional arguments:
  -h, --help            show this help message and exit

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made

awx bulk get: the following arguments are required: id
```

bulk job_launch

```
usage: awx bulk job_launch [-h] [--monitor] [--action-timeout ACTION_TIMEOUT]
                           [--wait] [--interval INTERVAL] [--name TEXT] --jobs
                           JOBS [--description TEXT] [--extra_vars JSON/YAML]
                           [--organization ID] [--inventory ID] [--limit TEXT]
                           [--scm_branch TEXT] [--skip_tags TEXT]
                           [--job_tags TEXT]

required arguments:
  --jobs JOBS           List of jobs to be launched, JSON. e.g.
                        [{"unified_job_template": 7}, {"unified_job_template":
                        10}]

optional arguments:
  -h, --help            show this help message and exit
  --monitor             If set, prints stdout of the launched job until it
                        finishes.
  --action-timeout ACTION_TIMEOUT
                        If set with --monitor or --wait, time out waiting on
                        job completion.
  --wait                If set, waits until the launched job finishes.
  --interval INTERVAL   If set with --monitor or --wait, amount of time to
                        wait in seconds between api calls. Minimum value is
                        2.5 seconds to avoid overwhelming the api
  --name TEXT           Name of this workflow job.
  --description TEXT    Optional description of this workflow job.
  --extra_vars JSON/YAML
                        a JSON or YAML string. You can optionally specify a
                        file path e.g., @path/to/file.yml
  --organization ID     Inherit permissions from this organization. If not
                        provided, a organization the user is a member of will
                        be selected automatically.
  --inventory ID        the ID of the associated inventory
  --limit TEXT
  --scm_branch TEXT
  --skip_tags TEXT
  --job_tags TEXT

awx bulk job_launch: the following arguments are required: --jobs
```

bulk host_create

```
usage: awx bulk host_create [-h] --inventory ID --hosts HOSTS

required arguments:
  --inventory ID  Primary Key ID of inventory to add hosts to.
  --hosts HOSTS   List of hosts to be created, JSON. e.g. [{"name":
                  "example.com"}, {"name": "127.0.0.1"}]

optional arguments:
  -h, --help      show this help message and exit

awx bulk host_create: the following arguments are required: --inventory, --hosts
```

bulk host_delete

```
usage: awx bulk host_delete [-h] --hosts HOSTS

required arguments:
  --hosts HOSTS  List of hosts ids to be deleted, e.g. [105, 130, 131, 200]

optional arguments:
  -h, --help     show this help message and exit

awx bulk host_delete: the following arguments are required: --hosts
```

config

```
usage: awx config [-h]

optional arguments:
  -h, --help  show this help message and exit
```

constructed_inventory

```
usage: awx constructed_inventory [-h] action ...

positional arguments:
  action
    list
    create
    get
    modify
    delete

optional arguments:
  -h, --help  show this help message and exit

awx constructed_inventory: the following arguments are required: action
```

constructed_inventory list

```
usage: awx constructed_inventory list [-h] [--all] [--order_by ORDER_BY]
                                      [-f {json,yaml,jq,human}]
                                      [--filter TEXT] [--conf.color BOOLEAN]
                                      [-v] [--type {inventory}]
                                      [--created CREATED]
                                      [--modified MODIFIED] [--name TEXT]
                                      [--description TEXT] [--organization ID]
                                      [--kind {,smart,constructed}]
                                      [--variables JSON/YAML]
                                      [--has_active_failures BOOLEAN]
                                      [--total_hosts INTEGER]
                                      [--hosts_with_active_failures INTEGER]
                                      [--total_groups INTEGER]
                                      [--has_inventory_sources BOOLEAN]
                                      [--total_inventory_sources INTEGER]
                                      [--inventory_sources_with_failures INTEGER]
                                      [--pending_deletion BOOLEAN]
                                      [--prevent_instance_group_fallback BOOLEAN]

optional arguments:
```

(continues on next page)

(продолжение с предыдущей страницы)

```
  -h, --help            show this help message and exit
  --all                 fetch all pages of content from the API when returning
                        results (instead of just the first page)
  --order_by ORDER_BY   order results by given field name, prefix the field
                        name with a dash (-) to sort in reverse eg
                        --order_by='-name',multiple sorting fields may be
                        specified by separating the field names with a comma
                        (,)
  --type {inventory}    only list constructed_inventory with the specified
                        type
  --created CREATED     only list constructed_inventory with the specified
                        created
  --modified MODIFIED   only list constructed_inventory with the specified
                        modified
  --name TEXT           only list constructed_inventory with the specified
                        name
  --description TEXT    only list constructed_inventory with the specified
                        description
  --organization ID     only list constructed_inventory with the specified
                        organization
  --kind {,smart,constructed}
                        only list constructed_inventory with the specified
                        kind
  --variables JSON/YAML
                        only list constructed_inventory with the specified
                        variables
  --has_active_failures BOOLEAN
                        only list constructed_inventory with the specified
                        has_active_failures
  --total_hosts INTEGER
                        only list constructed_inventory with the specified
                        total_hosts
  --hosts_with_active_failures INTEGER
                        only list constructed_inventory with the specified
                        hosts_with_active_failures
  --total_groups INTEGER
                        only list constructed_inventory with the specified
                        total_groups
  --has_inventory_sources BOOLEAN
                        only list constructed_inventory with the specified
                        has_inventory_sources
  --total_inventory_sources INTEGER
                        only list constructed_inventory with the specified
                        total_inventory_sources
  --inventory_sources_with_failures INTEGER
                        only list constructed_inventory with the specified
                        inventory_sources_with_failures
  --pending_deletion BOOLEAN
                        only list constructed_inventory with the specified
                        pending_deletion
  --prevent_instance_group_fallback BOOLEAN
                        only list constructed_inventory with the specified
                        prevent_instance_group_fallback

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made
```

constructed_inventory create

```
usage: awx constructed_inventory create [-h] --name TEXT [--description TEXT]
                                        --organization ID
                                        [--variables JSON/YAML]
                                        [--prevent_instance_group_fallback BOOLEAN]
                                        [--source_vars TEXT]
                                        [--update_cache_timeout INTEGER]
                                        [--limit TEXT] [--verbosity INTEGER]

required arguments:
  --name TEXT           Name of this inventory.
  --organization ID     Organization containing this inventory.

optional arguments:
  -h, --help            show this help message and exit
  --description TEXT    Optional description of this inventory.
  --variables JSON/YAML
                        Inventory variables in JSON or YAML format. You can
                        optionally specify a file path e.g., @path/to/file.yml
  --prevent_instance_group_fallback BOOLEAN
                        If enabled, the inventory will prevent adding any
                        organization instance groups to the list of preferred
                        instances groups to run associated job templates on.If
                        this setting is enabled and you provided an empty
                        list, the global instance groups will be applied.
  --source_vars TEXT    The source_vars for the related auto-created inventory
                        source, special to constructed inventory.
  --update_cache_timeout INTEGER
                        The cache timeout for the related auto-created
                        inventory source, special to constructed inventory
  --limit TEXT          The limit to restrict the returned hosts for the
                        related auto-created inventory source, special to
                        constructed inventory.
  --verbosity INTEGER   The verbosity level for the related auto-created
                        inventory source, special to constructed inventory

awx constructed_inventory create: the following arguments are required: --name, --
↪organization
```

constructed_inventory get

```
usage: awx constructed_inventory get [-h] [-f {json,yaml,jq,human}]
                                     [--filter TEXT] [--conf.color BOOLEAN]
                                     [-v]
                                     id

positional arguments:
  id                    the ID (or unique name) of the resource

optional arguments:
  -h, --help            show this help message and exit

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made

awx constructed_inventory get: the following arguments are required: id
```

constructed_inventory modify

```
usage: awx constructed_inventory modify [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx constructed_inventory modify: the following arguments are required: id
```

constructed_inventory delete

```
usage: awx constructed_inventory delete [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx constructed_inventory delete: the following arguments are required: id
```

credential_input_sources

```
usage: awx credential_input_sources [-h] action ...

positional arguments:
  action
    list
    create
    get
    modify
    delete

optional arguments:
  -h, --help  show this help message and exit

awx credential_input_sources: the following arguments are required: action
```

credential_input_sources list

```
usage: awx credential_input_sources list [-h] [--all] [--order_by ORDER_BY]
                                         [-f {json,yaml,jq,human}]
                                         [--filter TEXT]
                                         [--conf.color BOOLEAN] [-v]
                                         [--type {credential_input_source}]
                                         [--created CREATED]
                                         [--modified MODIFIED]
                                         [--description TEXT]
                                         [--input_field_name TEXT]
                                         [--metadata JSON/YAML]
                                         [--target_credential ID]
                                         [--source_credential ID]

optional arguments:
  -h, --help            show this help message and exit
```

(continues on next page)

(продолжение с предыдущей страницы)

```
  --all                 fetch all pages of content from the API when returning
                        results (instead of just the first page)
  --order_by ORDER_BY   order results by given field name, prefix the field
                        name with a dash (-) to sort in reverse eg
                        --order_by='-name',multiple sorting fields may be
                        specified by separating the field names with a comma
                        (,)
  --type {credential_input_source}
                        only list credential_input_sources with the specified
                        type
  --created CREATED     only list credential_input_sources with the specified
                        created
  --modified MODIFIED   only list credential_input_sources with the specified
                        modified
  --description TEXT    only list credential_input_sources with the specified
                        description
  --input_field_name TEXT
                        only list credential_input_sources with the specified
                        input_field_name
  --metadata JSON/YAML  only list credential_input_sources with the specified
                        metadata
  --target_credential ID
                        only list credential_input_sources with the specified
                        target_credential
  --source_credential ID
                        only list credential_input_sources with the specified
                        source_credential

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made
```

credential_input_sources create

```
usage: awx credential_input_sources create [-h] [--description TEXT]
                                           --input_field_name TEXT
                                           [--metadata JSON/YAML]
                                           --target_credential ID
                                           --source_credential ID

required arguments:
  --input_field_name TEXT
  --target_credential ID
                        the ID of the associated target_credential
  --source_credential ID
                        the ID of the associated source_credential

optional arguments:
  -h, --help            show this help message and exit
  --description TEXT    Optional description of this credential input source.
  --metadata JSON/YAML  a JSON or YAML string. You can optionally specify a
                        file path e.g., @path/to/file.yml

awx credential_input_sources create: the following arguments are required: --input_field_
↪name, --target_credential, --source_credential
```

credential_input_sources get

```
usage: awx credential_input_sources get [-h] [-f {json,yaml,jq,human}]
                                        [--filter TEXT] [--conf.color BOOLEAN]
                                        [-v]
                                        id

positional arguments:
  id                    the ID (or unique name) of the resource

optional arguments:
  -h, --help            show this help message and exit

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made

awx credential_input_sources get: the following arguments are required: id
```

credential_input_sources modify

```
usage: awx credential_input_sources modify [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx credential_input_sources modify: the following arguments are required: id
```

credential_input_sources delete

```
usage: awx credential_input_sources delete [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx credential_input_sources delete: the following arguments are required: id
```

credential_types

```
usage: awx credential_types [-h] action ...

positional arguments:
  action
    list
    create
    get
    modify
    delete
```

(continues on next page)

(продолжение с предыдущей страницы)

```
optional arguments:
  -h, --help  show this help message and exit

awx credential_types: the following arguments are required: action
```

credential_types list

```
usage: awx credential_types list [-h] [--all] [--order_by ORDER_BY]
                                 [-f {json,yaml,jq,human}] [--filter TEXT]
                                 [--conf.color BOOLEAN] [-v]
                                 [--type {credential_type}]
                                 [--created CREATED] [--modified MODIFIED]
                                 [--name TEXT] [--description TEXT]
                                 [--kind {ssh,vault,net,scm,cloud,registry,token,
↪insights,external,kubernetes,galaxy,cryptography}]
                                 [--namespace TEXT] [--managed BOOLEAN]
                                 [--inputs JSON/YAML] [--injectors JSON/YAML]

optional arguments:
  -h, --help            show this help message and exit
  --all                 fetch all pages of content from the API when returning
                        results (instead of just the first page)
  --order_by ORDER_BY   order results by given field name, prefix the field
                        name with a dash (-) to sort in reverse eg
                        --order_by='-name',multiple sorting fields may be
                        specified by separating the field names with a comma
                        (,)
  --type {credential_type}
                        only list credential_types with the specified type
  --created CREATED     only list credential_types with the specified created
  --modified MODIFIED   only list credential_types with the specified modified
  --name TEXT           only list credential_types with the specified name
  --description TEXT    only list credential_types with the specified
                        description
  --kind {ssh,vault,net,scm,cloud,registry,token,insights,external,kubernetes,galaxy,
↪cryptography}
                        only list credential_types with the specified kind
  --namespace TEXT      only list credential_types with the specified
                        namespace
  --managed BOOLEAN     only list credential_types with the specified managed
  --inputs JSON/YAML    only list credential_types with the specified inputs
  --injectors JSON/YAML
                        only list credential_types with the specified
                        injectors

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made
```

credential_types create

```
usage: awx credential_types create [-h] --name TEXT [--description TEXT]
                                   --kind {net,cloud} [--inputs JSON/YAML]
                                   [--injectors JSON/YAML]

required arguments:
  --name TEXT           Name of this credential type.
  --kind {net,cloud}

optional arguments:
  -h, --help            show this help message and exit
  --description TEXT    Optional description of this credential type.
  --inputs JSON/YAML    Enter inputs using either JSON or YAML syntax. Refer
                        to the documentation for example syntax. You can
                        optionally specify a file path e.g., @path/to/file.yml
  --injectors JSON/YAML
                        Enter injectors using either JSON or YAML syntax.
                        Refer to the documentation for example syntax. You can
                        optionally specify a file path e.g., @path/to/file.yml

awx credential_types create: the following arguments are required: --name, --kind
```

credential_types get

```
usage: awx credential_types get [-h] [-f {json,yaml,jq,human}] [--filter TEXT]
                                [--conf.color BOOLEAN] [-v]
                                id

positional arguments:
  id                    the ID (or unique name) of the resource

optional arguments:
  -h, --help            show this help message and exit

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made

awx credential_types get: the following arguments are required: id
```

credential_types modify

```
usage: awx credential_types modify [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx credential_types modify: the following arguments are required: id
```

credential_types delete

```
usage: awx credential_types delete [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx credential_types delete: the following arguments are required: id
```

credentials

```
usage: awx credentials [-h] action ...

positional arguments:
  action
    list
    create
    get
    modify
    delete

optional arguments:
  -h, --help  show this help message and exit

awx credentials: the following arguments are required: action
```

credentials list

```
usage: awx credentials list [-h] [--all] [--order_by ORDER_BY]
                            [-f {json,yaml,jq,human}] [--filter TEXT]
                            [--conf.color BOOLEAN] [-v] [--type {credential}]
                            [--created CREATED] [--modified MODIFIED]
                            [--name TEXT] [--description TEXT]
                            [--organization ID] [--credential_type ID]
                            [--managed BOOLEAN] [--inputs JSON/YAML]

optional arguments:
  -h, --help            show this help message and exit
  --all                 fetch all pages of content from the API when returning
                        results (instead of just the first page)
  --order_by ORDER_BY   order results by given field name, prefix the field
                        name with a dash (-) to sort in reverse eg
                        --order_by='-name',multiple sorting fields may be
                        specified by separating the field names with a comma
                        (,)
  --type {credential}   only list credentials with the specified type
  --created CREATED     only list credentials with the specified created
  --modified MODIFIED   only list credentials with the specified modified
  --name TEXT           only list credentials with the specified name
  --description TEXT    only list credentials with the specified description
  --organization ID     only list credentials with the specified organization
  --credential_type ID  only list credentials with the specified
                        credential_type
  --managed BOOLEAN     only list credentials with the specified managed
  --inputs JSON/YAML    only list credentials with the specified inputs

input/output formatting:
```

(continues on next page)

(продолжение с предыдущей страницы)

```
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made
```

credentials create

```
usage: awx credentials create [-h] --name TEXT [--description TEXT]
                              [--organization ID] --credential_type ID
                              [--inputs JSON/YAML] [--user ID] [--team ID]

required arguments:
  --name TEXT           Name of this credential.
  --credential_type ID  Specify the type of credential you want to create.
                        Refer to the documentation for details on each type.

optional arguments:
  -h, --help            show this help message and exit
  --description TEXT    Optional description of this credential.
  --organization ID     Inherit permissions from organization roles. If
                        provided on creation, do not give either user or team.
  --inputs JSON/YAML    Enter inputs using either JSON or YAML syntax. Refer
                        to the documentation for example syntax. You can
                        optionally specify a file path e.g., @path/to/file.yml
  --user ID             Write-only field used to add user to owner role. If
                        provided, do not give either team or organization.
                        Only valid for creation.
  --team ID             Write-only field used to add team to owner role. If
                        provided, do not give either user or organization.
                        Only valid for creation.

awx credentials create: the following arguments are required: --name, --credential_type
```

credentials get

```
usage: awx credentials get [-h] [-f {json,yaml,jq,human}] [--filter TEXT]
                           [--conf.color BOOLEAN] [-v]
                           id

positional arguments:
  id                    the ID (or unique name) of the resource

optional arguments:
  -h, --help            show this help message and exit

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made

awx credentials get: the following arguments are required: id
```

credentials modify

```
usage: awx credentials modify [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx credentials modify: the following arguments are required: id
```

credentials delete

```
usage: awx credentials delete [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx credentials delete: the following arguments are required: id
```

execution_environments

```
usage: awx execution_environments [-h] action ...

positional arguments:
  action
    list
    create
    get
    modify
    delete

optional arguments:
  -h, --help  show this help message and exit

awx execution_environments: the following arguments are required: action
```

execution_environments list

```
usage: awx execution_environments list [-h] [--all] [--order_by ORDER_BY]
                                       [-f {json,yaml,jq,human}]
                                       [--filter TEXT] [--conf.color BOOLEAN]
                                       [-v] [--type {execution_environment}]
                                       [--created CREATED]
                                       [--modified MODIFIED] [--name TEXT]
                                       [--description TEXT]
                                       [--organization ID] [--image TEXT]
                                       [--managed BOOLEAN] [--credential ID]
                                       [--pull {,always,missing,never}]

optional arguments:
  -h, --help            show this help message and exit
  --all                 fetch all pages of content from the API when returning
                        results (instead of just the first page)
```

(continues on next page)

(продолжение с предыдущей страницы)

```
  --order_by ORDER_BY   order results by given field name, prefix the field
                        name with a dash (-) to sort in reverse eg
                        --order_by='-name',multiple sorting fields may be
                        specified by separating the field names with a comma
                        (,)
  --type {execution_environment}
                        only list execution_environments with the specified
                        type
  --created CREATED     only list execution_environments with the specified
                        created
  --modified MODIFIED   only list execution_environments with the specified
                        modified
  --name TEXT           only list execution_environments with the specified
                        name
  --description TEXT    only list execution_environments with the specified
                        description
  --organization ID     only list execution_environments with the specified
                        organization
  --image TEXT          only list execution_environments with the specified
                        image
  --managed BOOLEAN     only list execution_environments with the specified
                        managed
  --credential ID       only list execution_environments with the specified
                        credential
  --pull {,always,missing,never}
                        only list execution_environments with the specified
                        pull

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made
```

execution_environments create

```
usage: awx execution_environments create [-h] --name TEXT [--description TEXT]
                                         [--organization ID] --image TEXT
                                         [--credential ID]
                                         [--pull {,always,missing,never}]

required arguments:
  --name TEXT           Name of this execution environment.
  --image TEXT          The full image location, including the container
                        registry, image name, and version tag.

optional arguments:
  -h, --help            show this help message and exit
  --description TEXT    Optional description of this execution environment.
  --organization ID     The organization used to determine access to this
                        execution environment.
  --credential ID       the ID of the associated credential
  --pull {,always,missing,never}
                        Pull image before running?

awx execution_environments create: the following arguments are required: --name, --image
```

execution_environments get

```
usage: awx execution_environments get [-h] [-f {json,yaml,jq,human}]
                                      [--filter TEXT] [--conf.color BOOLEAN]
                                      [-v]
                                      id

positional arguments:
  id                    the ID (or unique name) of the resource

optional arguments:
  -h, --help            show this help message and exit

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made

awx execution_environments get: the following arguments are required: id
```

execution_environments modify

```
usage: awx execution_environments modify [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx execution_environments modify: the following arguments are required: id
```

execution_environments delete

```
usage: awx execution_environments delete [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx execution_environments delete: the following arguments are required: id
```

export

```
usage: awx export > exportfile

export resources to stdout

optional arguments:
  -h, --help            show this help message and exit

resources:
  --users [USERS [USERS ...]]
```

(continues on next page)

(продолжение с предыдущей страницы)

```
  --organizations [ORGANIZATIONS [ORGANIZATIONS ...]]
  --teams [TEAMS [TEAMS ...]]
  --credential_types [CREDENTIAL_TYPES [CREDENTIAL_TYPES ...]]
  --credentials [CREDENTIALS [CREDENTIALS ...]]
  --notification_templates [NOTIFICATION_TEMPLATES [NOTIFICATION_TEMPLATES ...]]
  --projects [PROJECTS [PROJECTS ...]]
  --inventory [INVENTORY [INVENTORY ...]]
  --inventory_sources [INVENTORY_SOURCES [INVENTORY_SOURCES ...]]
  --job_templates [JOB_TEMPLATES [JOB_TEMPLATES ...]]
  --workflow_job_templates [WORKFLOW_JOB_TEMPLATES [WORKFLOW_JOB_TEMPLATES ...]]
  --execution_environments [EXECUTION_ENVIRONMENTS [EXECUTION_ENVIRONMENTS ...]]
  --applications [APPLICATIONS [APPLICATIONS ...]]
  --schedules [SCHEDULES [SCHEDULES ...]]

input/output formatting:
  -f {json,yaml}, --conf.format {json,yaml}
                        specify a format for the input and output
  -v, --verbose         print debug-level logs, including requests made
```

groups

```
usage: awx groups [-h] action ...

positional arguments:
  action
    list
    create
    get
    modify
    delete

optional arguments:
  -h, --help  show this help message and exit

awx groups: the following arguments are required: action
```

groups list

```
usage: awx groups list [-h] [--all] [--order_by ORDER_BY]
                       [-f {json,yaml,jq,human}] [--filter TEXT]
                       [--conf.color BOOLEAN] [-v] [--type {group}]
                       [--created CREATED] [--modified MODIFIED] [--name TEXT]
                       [--description TEXT] [--inventory ID]
                       [--variables JSON/YAML]

optional arguments:
  -h, --help            show this help message and exit
  --all                 fetch all pages of content from the API when returning
                        results (instead of just the first page)
  --order_by ORDER_BY   order results by given field name, prefix the field
                        name with a dash (-) to sort in reverse eg
                        --order_by='-name',multiple sorting fields may be
                        specified by separating the field names with a comma
                        (,)
  --type {group}        only list groups with the specified type
  --created CREATED     only list groups with the specified created
  --modified MODIFIED   only list groups with the specified modified
  --name TEXT           only list groups with the specified name
  --description TEXT    only list groups with the specified description
  --inventory ID        only list groups with the specified inventory
```

(continues on next page)

(продолжение с предыдущей страницы)

```
  --variables JSON/YAML
                        only list groups with the specified variables

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made
```

groups create

```
usage: awx groups create [-h] --name TEXT [--description TEXT] --inventory ID
                         [--variables JSON/YAML]

required arguments:
  --name TEXT           Name of this group.
  --inventory ID        the ID of the associated inventory

optional arguments:
  -h, --help            show this help message and exit
  --description TEXT    Optional description of this group.
  --variables JSON/YAML
                        Group variables in JSON or YAML format. You can
                        optionally specify a file path e.g., @path/to/file.yml

awx groups create: the following arguments are required: --name, --inventory
```

groups get

```
usage: awx groups get [-h] [-f {json,yaml,jq,human}] [--filter TEXT]
                      [--conf.color BOOLEAN] [-v]
                      id

positional arguments:
  id                    the ID (or unique name) of the resource

optional arguments:
  -h, --help            show this help message and exit

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made

awx groups get: the following arguments are required: id
```

groups modify

```
usage: awx groups modify [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx groups modify: the following arguments are required: id
```

groups delete

```
usage: awx groups delete [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx groups delete: the following arguments are required: id
```

host_metric_summary_monthly

```
usage: awx host_metric_summary_monthly [-h] action ...

positional arguments:
  action
    list
    get

optional arguments:
  -h, --help  show this help message and exit

awx host_metric_summary_monthly: the following arguments are required: action
```

host_metric_summary_monthly list

```
usage: awx host_metric_summary_monthly list [-h] [--all] [--order_by ORDER_BY]
                                            [-f {json,yaml,jq,human}]
                                            [--filter TEXT]
                                            [--conf.color BOOLEAN] [-v]
                                            [--date DATE]
                                            [--license_consumed INTEGER]
                                            [--license_capacity INTEGER]
                                            [--hosts_added INTEGER]
                                            [--hosts_deleted INTEGER]
                                            [--indirectly_managed_hosts INTEGER]

optional arguments:
  -h, --help            show this help message and exit
  --all                 fetch all pages of content from the API when returning
                        results (instead of just the first page)
  --order_by ORDER_BY   order results by given field name, prefix the field
                        name with a dash (-) to sort in reverse eg
                        --order_by='-name',multiple sorting fields may be
```

(continues on next page)

(продолжение с предыдущей страницы)

```
                        specified by separating the field names with a comma
                        (,)
  --date DATE           only list host_metric_summary_monthly with the
                        specified date
  --license_consumed INTEGER
                        only list host_metric_summary_monthly with the
                        specified license_consumed
  --license_capacity INTEGER
                        only list host_metric_summary_monthly with the
                        specified license_capacity
  --hosts_added INTEGER
                        only list host_metric_summary_monthly with the
                        specified hosts_added
  --hosts_deleted INTEGER
                        only list host_metric_summary_monthly with the
                        specified hosts_deleted
  --indirectly_managed_hosts INTEGER
                        only list host_metric_summary_monthly with the
                        specified indirectly_managed_hosts

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made
```

host_metric_summary_monthly get

```
usage: awx host_metric_summary_monthly get [-h] [-f {json,yaml,jq,human}]
                                           [--filter TEXT]
                                           [--conf.color BOOLEAN] [-v]
                                           id

positional arguments:
  id                    the ID (or unique name) of the resource

optional arguments:
  -h, --help            show this help message and exit

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made

awx host_metric_summary_monthly get: the following arguments are required: id
```

host_metrics

```
usage: awx host_metrics [-h] action ...

positional arguments:
  action
    list
    get
    delete

optional arguments:
  -h, --help  show this help message and exit

awx host_metrics: the following arguments are required: action
```

host_metrics list

```
usage: awx host_metrics list [-h] [--all] [--order_by ORDER_BY]
                             [-f {json,yaml,jq,human}] [--filter TEXT]
                             [--conf.color BOOLEAN] [-v] [--hostname TEXT]
                             [--first_automation FIRST_AUTOMATION]
                             [--last_automation LAST_AUTOMATION]
                             [--last_deleted LAST_DELETED]
                             [--automated_counter INTEGER]
                             [--deleted_counter INTEGER] [--deleted BOOLEAN]
                             [--used_in_inventories INTEGER]

optional arguments:
  -h, --help            show this help message and exit
  --all                 fetch all pages of content from the API when returning
                        results (instead of just the first page)
  --order_by ORDER_BY   order results by given field name, prefix the field
                        name with a dash (-) to sort in reverse eg
                        --order_by='-name',multiple sorting fields may be
                        specified by separating the field names with a comma
                        (,)
  --hostname TEXT       only list host_metrics with the specified hostname
  --first_automation FIRST_AUTOMATION
                        only list host_metrics with the specified
                        first_automation
  --last_automation LAST_AUTOMATION
                        only list host_metrics with the specified
                        last_automation
  --last_deleted LAST_DELETED
                        only list host_metrics with the specified last_deleted
  --automated_counter INTEGER
                        only list host_metrics with the specified
                        automated_counter
  --deleted_counter INTEGER
                        only list host_metrics with the specified
                        deleted_counter
  --deleted BOOLEAN     only list host_metrics with the specified deleted
  --used_in_inventories INTEGER
                        only list host_metrics with the specified
                        used_in_inventories

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made
```

host_metrics get

```
usage: awx host_metrics get [-h] [-f {json,yaml,jq,human}] [--filter TEXT]
                            [--conf.color BOOLEAN] [-v]
                            id

positional arguments:
  id                    the ID (or unique name) of the resource

optional arguments:
  -h, --help            show this help message and exit

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made

awx host_metrics get: the following arguments are required: id
```

host_metrics delete

```
usage: awx host_metrics delete [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx host_metrics delete: the following arguments are required: id
```

hosts

```
usage: awx hosts [-h] action ...

positional arguments:
  action
    list
    create
    get
    modify
    delete

optional arguments:
  -h, --help  show this help message and exit

awx hosts: the following arguments are required: action
```

hosts list

```
usage: awx hosts list [-h] [--all] [--order_by ORDER_BY]
                      [-f {json,yaml,jq,human}] [--filter TEXT]
                      [--conf.color BOOLEAN] [-v] [--type {host}]
                      [--created CREATED] [--modified MODIFIED] [--name TEXT]
                      [--description TEXT] [--inventory ID]
                      [--enabled BOOLEAN] [--instance_id TEXT]
                      [--variables JSON/YAML] [--last_job ID]
                      [--last_job_host_summary ID]
                      [--ansible_facts_modified ANSIBLE_FACTS_MODIFIED]

optional arguments:
  -h, --help            show this help message and exit
  --all                 fetch all pages of content from the API when returning
                        results (instead of just the first page)
  --order_by ORDER_BY   order results by given field name, prefix the field
                        name with a dash (-) to sort in reverse eg
                        --order_by='-name',multiple sorting fields may be
                        specified by separating the field names with a comma
                        (,)
  --type {host}         only list hosts with the specified type
  --created CREATED     only list hosts with the specified created
  --modified MODIFIED   only list hosts with the specified modified
  --name TEXT           only list hosts with the specified name
  --description TEXT    only list hosts with the specified description
  --inventory ID        only list hosts with the specified inventory
  --enabled BOOLEAN     only list hosts with the specified enabled
  --instance_id TEXT    only list hosts with the specified instance_id
  --variables JSON/YAML
                        only list hosts with the specified variables
  --last_job ID         only list hosts with the specified last_job
  --last_job_host_summary ID
                        only list hosts with the specified
                        last_job_host_summary
  --ansible_facts_modified ANSIBLE_FACTS_MODIFIED
                        only list hosts with the specified
                        ansible_facts_modified

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made
```

hosts create

```
usage: awx hosts create [-h] --name TEXT [--description TEXT] --inventory ID
                        [--enabled BOOLEAN] [--instance_id TEXT]
                        [--variables JSON/YAML]

required arguments:
  --name TEXT           Name of this host.
  --inventory ID        the ID of the associated inventory

optional arguments:
  -h, --help            show this help message and exit
  --description TEXT    Optional description of this host.
  --enabled BOOLEAN     Is this host online and available for running jobs?
  --instance_id TEXT    The value used by the remote inventory source to
```

(continues on next page)

(продолжение с предыдущей страницы)

```
                        uniquely identify the host
  --variables JSON/YAML
                        Host variables in JSON or YAML format. You can
                        optionally specify a file path e.g., @path/to/file.yml

awx hosts create: the following arguments are required: --name, --inventory
```

hosts get

```
usage: awx hosts get [-h] [-f {json,yaml,jq,human}] [--filter TEXT]
                     [--conf.color BOOLEAN] [-v]
                     id

positional arguments:
  id                    the ID (or unique name) of the resource

optional arguments:
  -h, --help            show this help message and exit

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made

awx hosts get: the following arguments are required: id
```

hosts modify

```
usage: awx hosts modify [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx hosts modify: the following arguments are required: id
```

hosts delete

```
usage: awx hosts delete [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx hosts delete: the following arguments are required: id
```

import

```
usage: awx import < exportfile

import resources from stdin

optional arguments:
  -h, --help            show this help message and exit

input/output formatting:
  -f {json,yaml}, --conf.format {json,yaml}
                        specify a format for the input and output
  -v, --verbose         print debug-level logs, including requests made
```

instance_groups

```
usage: awx instance_groups [-h] action ...

positional arguments:
  action
    list
    create
    get
    modify
    delete

optional arguments:
  -h, --help  show this help message and exit

awx instance_groups: the following arguments are required: action
```

instance_groups list

```
usage: awx instance_groups list [-h] [--all] [--order_by ORDER_BY]
                                [-f {json,yaml,jq,human}] [--filter TEXT]
                                [--conf.color BOOLEAN] [-v]
                                [--type {instance_group}] [--name TEXT]
                                [--created CREATED] [--modified MODIFIED]
                                [--max_concurrent_jobs INTEGER]
                                [--max_forks INTEGER]
                                [--is_container_group BOOLEAN]
                                [--credential ID]
                                [--policy_instance_percentage INTEGER]
                                [--policy_instance_minimum INTEGER]
                                [--policy_instance_list JSON/YAML]
                                [--pod_spec_override TEXT]

optional arguments:
  -h, --help            show this help message and exit
  --all                 fetch all pages of content from the API when returning
                        results (instead of just the first page)
  --order_by ORDER_BY   order results by given field name, prefix the field
                        name with a dash (-) to sort in reverse eg
                        --order_by='-name',multiple sorting fields may be
                        specified by separating the field names with a comma
                        (,)
  --type {instance_group}
                        only list instance_groups with the specified type
  --name TEXT           only list instance_groups with the specified name
  --created CREATED     only list instance_groups with the specified created
  --modified MODIFIED   only list instance_groups with the specified modified
```

(continues on next page)

(продолжение с предыдущей страницы)

```
  --max_concurrent_jobs INTEGER
                        only list instance_groups with the specified
                        max_concurrent_jobs
  --max_forks INTEGER   only list instance_groups with the specified max_forks
  --is_container_group BOOLEAN
                        only list instance_groups with the specified
                        is_container_group
  --credential ID       only list instance_groups with the specified
                        credential
  --policy_instance_percentage INTEGER
                        only list instance_groups with the specified
                        policy_instance_percentage
  --policy_instance_minimum INTEGER
                        only list instance_groups with the specified
                        policy_instance_minimum
  --policy_instance_list JSON/YAML
                        only list instance_groups with the specified
                        policy_instance_list
  --pod_spec_override TEXT
                        only list instance_groups with the specified
                        pod_spec_override

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made
```

instance_groups create

```
usage: awx instance_groups create [-h] --name TEXT
                                  [--max_concurrent_jobs INTEGER]
                                  [--max_forks INTEGER]
                                  [--is_container_group BOOLEAN]
                                  [--credential ID]
                                  [--policy_instance_percentage INTEGER]
                                  [--policy_instance_minimum INTEGER]
                                  [--policy_instance_list JSON/YAML]
                                  [--pod_spec_override TEXT]

required arguments:
  --name TEXT           Name of this instance group.

optional arguments:
  -h, --help            show this help message and exit
  --max_concurrent_jobs INTEGER
                        Maximum number of concurrent jobs to run on a group.
                        When set to zero, no maximum is enforced.
  --max_forks INTEGER   Maximum number of forks to execute concurrently on a
                        group. When set to zero, no maximum is enforced.
  --is_container_group BOOLEAN
                        Indicates whether instances in this group are
                        containerized.Containerized groups have a designated
                        Openshift or Kubernetes cluster.
  --credential ID       the ID of the associated credential
  --policy_instance_percentage INTEGER
                        Minimum percentage of all instances that will be
                        automatically assigned to this group when new
                        instances come online.
  --policy_instance_minimum INTEGER
```

(continues on next page)

(продолжение с предыдущей страницы)

```
                        Static minimum number of Instances that will be
                        automatically assign to this group when new instances
                        come online.
  --policy_instance_list JSON/YAML
                        List of exact-match Instances that will be assigned to
                        this group You can optionally specify a file path
                        e.g., @path/to/file.yml
  --pod_spec_override TEXT

awx instance_groups create: the following arguments are required: --name
```

instance_groups get

```
usage: awx instance_groups get [-h] [-f {json,yaml,jq,human}] [--filter TEXT]
                               [--conf.color BOOLEAN] [-v]
                               id

positional arguments:
  id                    the ID (or unique name) of the resource

optional arguments:
  -h, --help            show this help message and exit

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made

awx instance_groups get: the following arguments are required: id
```

instance_groups modify

```
usage: awx instance_groups modify [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx instance_groups modify: the following arguments are required: id
```

instance_groups delete

```
usage: awx instance_groups delete [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx instance_groups delete: the following arguments are required: id
```

instances

```
usage: awx instances [-h] action ...

positional arguments:
  action
    list
    create
    get
    modify

optional arguments:
  -h, --help  show this help message and exit

awx instances: the following arguments are required: action
```

instances list

```
usage: awx instances list [-h] [--all] [--order_by ORDER_BY]
                          [-f {json,yaml,jq,human}] [--filter TEXT]
                          [--conf.color BOOLEAN] [-v] [--hostname TEXT]
                          [--type {instance}] [--uuid TEXT]
                          [--created CREATED] [--modified MODIFIED]
                          [--last_seen LAST_SEEN]
                          [--health_check_started HEALTH_CHECK_STARTED]
                          [--last_health_check LAST_HEALTH_CHECK]
                          [--errors TEXT]
                          [--capacity_adjustment CAPACITY_ADJUSTMENT]
                          [--version TEXT] [--capacity INTEGER] [--cpu CPU]
                          [--memory INTEGER] [--cpu_capacity INTEGER]
                          [--mem_capacity INTEGER] [--enabled BOOLEAN]
                          [--managed_by_policy BOOLEAN]
                          [--node_type {control,execution,hybrid,hop}]
                          [--node_state {provisioning,provision-fail,installed,ready,
↪unavailable,deprovisioning,deprovision-fail}]
                          [--ip_address TEXT] [--listener_port INTEGER]
                          [--peers_from_control_nodes BOOLEAN]

optional arguments:
  -h, --help            show this help message and exit
  --all                 fetch all pages of content from the API when returning
                        results (instead of just the first page)
  --order_by ORDER_BY   order results by given field name, prefix the field
                        name with a dash (-) to sort in reverse eg
                        --order_by='-name',multiple sorting fields may be
                        specified by separating the field names with a comma
                        (,)
  --hostname TEXT       only list instances with the specified hostname
  --type {instance}     only list instances with the specified type
  --uuid TEXT           only list instances with the specified uuid
  --created CREATED     only list instances with the specified created
  --modified MODIFIED   only list instances with the specified modified
  --last_seen LAST_SEEN
                        only list instances with the specified last_seen
  --health_check_started HEALTH_CHECK_STARTED
                        only list instances with the specified
                        health_check_started
  --last_health_check LAST_HEALTH_CHECK
                        only list instances with the specified
                        last_health_check
  --errors TEXT         only list instances with the specified errors
  --capacity_adjustment CAPACITY_ADJUSTMENT
                        only list instances with the specified
```

(continues on next page)

(продолжение с предыдущей страницы)

```
                        capacity_adjustment
  --version TEXT        only list instances with the specified version
  --capacity INTEGER    only list instances with the specified capacity
  --cpu CPU             only list instances with the specified cpu
  --memory INTEGER      only list instances with the specified memory
  --cpu_capacity INTEGER
                        only list instances with the specified cpu_capacity
  --mem_capacity INTEGER
                        only list instances with the specified mem_capacity
  --enabled BOOLEAN     only list instances with the specified enabled
  --managed_by_policy BOOLEAN
                        only list instances with the specified
                        managed_by_policy
  --node_type {control,execution,hybrid,hop}
                        only list instances with the specified node_type
  --node_state {provisioning,provision-fail,installed,ready,unavailable,deprovisioning,
↪deprovision-fail}
                        only list instances with the specified node_state
  --ip_address TEXT     only list instances with the specified ip_address
  --listener_port INTEGER
                        only list instances with the specified listener_port
  --peers_from_control_nodes BOOLEAN
                        only list instances with the specified
                        peers_from_control_nodes

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made
```

instances create

```
usage: awx instances create [-h] --hostname TEXT
                            [--capacity_adjustment CAPACITY_ADJUSTMENT]
                            [--enabled BOOLEAN] [--managed_by_policy BOOLEAN]
                            [--node_type {control,execution,hybrid,hop}]
                            [--node_state {provisioning,provision-fail,installed,ready,
↪unavailable,deprovisioning,deprovision-fail}]
                            [--listener_port INTEGER] [--peers PEERS]
                            [--peers_from_control_nodes BOOLEAN]

required arguments:
  --hostname TEXT

optional arguments:
  -h, --help            show this help message and exit
  --capacity_adjustment CAPACITY_ADJUSTMENT
  --enabled BOOLEAN
  --managed_by_policy BOOLEAN
  --node_type {control,execution,hybrid,hop}
                        Role that this node plays in the mesh.
  --node_state {provisioning,provision-fail,installed,ready,unavailable,deprovisioning,
↪deprovision-fail}
                        Indicates the current life cycle stage of this
                        instance.
  --listener_port INTEGER
                        Port that Receptor will listen for incoming
                        connections on.
  --peers PEERS
```

(continues on next page)

(продолжение с предыдущей страницы)

```
  --peers_from_control_nodes BOOLEAN
                        If True, control plane cluster nodes should
                        automatically peer to it.

awx instances create: the following arguments are required: --hostname
```

instances get

```
usage: awx instances get [-h] [-f {json,yaml,jq,human}] [--filter TEXT]
                         [--conf.color BOOLEAN] [-v]
                         id

positional arguments:
  id                    the ID (or unique name) of the resource

optional arguments:
  -h, --help            show this help message and exit

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made

awx instances get: the following arguments are required: id
```

instances modify

```
usage: awx instances modify [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx instances modify: the following arguments are required: id
```

inventory

```
usage: awx inventory [-h] action ...

positional arguments:
  action
    list
    create
    get
    modify
    delete

optional arguments:
  -h, --help  show this help message and exit

awx inventory: the following arguments are required: action
```

inventory list

```
usage: awx inventory list [-h] [--all] [--order_by ORDER_BY]
                          [-f {json,yaml,jq,human}] [--filter TEXT]
                          [--conf.color BOOLEAN] [-v] [--type {inventory}]
                          [--created CREATED] [--modified MODIFIED]
                          [--name TEXT] [--description TEXT]
                          [--organization ID] [--kind {,smart,constructed}]
                          [--host_filter TEXT] [--variables JSON/YAML]
                          [--has_active_failures BOOLEAN]
                          [--total_hosts INTEGER]
                          [--hosts_with_active_failures INTEGER]
                          [--total_groups INTEGER]
                          [--has_inventory_sources BOOLEAN]
                          [--total_inventory_sources INTEGER]
                          [--inventory_sources_with_failures INTEGER]
                          [--pending_deletion BOOLEAN]
                          [--prevent_instance_group_fallback BOOLEAN]

optional arguments:
  -h, --help            show this help message and exit
  --all                 fetch all pages of content from the API when returning
                        results (instead of just the first page)
  --order_by ORDER_BY   order results by given field name, prefix the field
                        name with a dash (-) to sort in reverse eg
                        --order_by='-name',multiple sorting fields may be
                        specified by separating the field names with a comma
                        (,)
  --type {inventory}    only list inventory with the specified type
  --created CREATED     only list inventory with the specified created
  --modified MODIFIED   only list inventory with the specified modified
  --name TEXT           only list inventory with the specified name
  --description TEXT    only list inventory with the specified description
  --organization ID     only list inventory with the specified organization
  --kind {,smart,constructed}
                        only list inventory with the specified kind
  --host_filter TEXT    only list inventory with the specified host_filter
  --variables JSON/YAML
                        only list inventory with the specified variables
  --has_active_failures BOOLEAN
                        only list inventory with the specified
                        has_active_failures
  --total_hosts INTEGER
                        only list inventory with the specified total_hosts
  --hosts_with_active_failures INTEGER
                        only list inventory with the specified
                        hosts_with_active_failures
  --total_groups INTEGER
                        only list inventory with the specified total_groups
  --has_inventory_sources BOOLEAN
                        only list inventory with the specified
                        has_inventory_sources
  --total_inventory_sources INTEGER
                        only list inventory with the specified
                        total_inventory_sources
  --inventory_sources_with_failures INTEGER
                        only list inventory with the specified
                        inventory_sources_with_failures
  --pending_deletion BOOLEAN
                        only list inventory with the specified
                        pending_deletion
  --prevent_instance_group_fallback BOOLEAN
                        only list inventory with the specified
                        prevent_instance_group_fallback
```

(continues on next page)

(продолжение с предыдущей страницы)

```
input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made
```

inventory create

```
usage: awx inventory create [-h] --name TEXT [--description TEXT]
                            --organization ID [--kind {,smart,constructed}]
                            [--host_filter TEXT] [--variables JSON/YAML]
                            [--prevent_instance_group_fallback BOOLEAN]

required arguments:
  --name TEXT           Name of this inventory.
  --organization ID     Organization containing this inventory.

optional arguments:
  -h, --help            show this help message and exit
  --description TEXT    Optional description of this inventory.
  --kind {,smart,constructed}
                        Kind of inventory being represented.
  --host_filter TEXT    Filter that will be applied to the hosts of this
                        inventory.
  --variables JSON/YAML
                        Inventory variables in JSON or YAML format. You can
                        optionally specify a file path e.g., @path/to/file.yml
  --prevent_instance_group_fallback BOOLEAN
                        If enabled, the inventory will prevent adding any
                        organization instance groups to the list of preferred
                        instances groups to run associated job templates on.If
                        this setting is enabled and you provided an empty
                        list, the global instance groups will be applied.

awx inventory create: the following arguments are required: --name, --organization
```

inventory get

```
usage: awx inventory get [-h] [-f {json,yaml,jq,human}] [--filter TEXT]
                         [--conf.color BOOLEAN] [-v]
                         id

positional arguments:
  id                    the ID (or unique name) of the resource

optional arguments:
  -h, --help            show this help message and exit

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made

awx inventory get: the following arguments are required: id
```

inventory modify

```
usage: awx inventory modify [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx inventory modify: the following arguments are required: id
```

inventory delete

```
usage: awx inventory delete [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx inventory delete: the following arguments are required: id
```

inventory_sources

```
usage: awx inventory_sources [-h] action ...

positional arguments:
  action
    list
    create
    get
    modify
    delete
    update
    associate
    disassociate

optional arguments:
  -h, --help    show this help message and exit

awx inventory_sources: the following arguments are required: action
```

inventory_sources list

```
usage: awx inventory_sources list [-h] [--all] [--order_by ORDER_BY]
                                  [-f {json,yaml,jq,human}] [--filter TEXT]
                                  [--conf.color BOOLEAN] [-v]
                                  [--type {inventory_source}]
                                  [--created CREATED] [--modified MODIFIED]
                                  [--name TEXT] [--description TEXT]
                                  [--source {file,constructed,scm,ec2,gce,azure_rm,
↪vmware,satellite6,openstack,rhv,controller,insights}]
                                  [--source_path TEXT] [--source_vars TEXT]
                                  [--scm_branch TEXT] [--enabled_var TEXT]
                                  [--enabled_value TEXT] [--host_filter TEXT]
                                  [--overwrite BOOLEAN]
```

(continues on next page)

(продолжение с предыдущей страницы)

```
                                  [--overwrite_vars BOOLEAN]
                                  [--custom_virtualenv TEXT]
                                  [--timeout INTEGER] [--verbosity {0,1,2}]
                                  [--limit TEXT] [--last_job_run LAST_JOB_RUN]
                                  [--last_job_failed BOOLEAN]
                                  [--next_job_run NEXT_JOB_RUN]
                                  [--status {new,pending,waiting,running,successful,
↪failed,error,canceled,never updated,none}]
                                  [--execution_environment ID]
                                  [--inventory ID]
                                  [--update_on_launch BOOLEAN]
                                  [--update_cache_timeout INTEGER]
                                  [--source_project ID]

optional arguments:
  -h, --help            show this help message and exit
  --all                 fetch all pages of content from the API when returning
                        results (instead of just the first page)
  --order_by ORDER_BY   order results by given field name, prefix the field
                        name with a dash (-) to sort in reverse eg
                        --order_by='-name',multiple sorting fields may be
                        specified by separating the field names with a comma
                        (,)
  --type {inventory_source}
                        only list inventory_sources with the specified type
  --created CREATED     only list inventory_sources with the specified created
  --modified MODIFIED   only list inventory_sources with the specified
                        modified
  --name TEXT           only list inventory_sources with the specified name
  --description TEXT    only list inventory_sources with the specified
                        description
  --source {file,constructed,scm,ec2,gce,azure_rm,vmware,satellite6,openstack,rhv,
↪controller,insights}
                        only list inventory_sources with the specified source
  --source_path TEXT    only list inventory_sources with the specified
                        source_path
  --source_vars TEXT    only list inventory_sources with the specified
                        source_vars
  --scm_branch TEXT     only list inventory_sources with the specified
                        scm_branch
  --enabled_var TEXT    only list inventory_sources with the specified
                        enabled_var
  --enabled_value TEXT  only list inventory_sources with the specified
                        enabled_value
  --host_filter TEXT    only list inventory_sources with the specified
                        host_filter
  --overwrite BOOLEAN   only list inventory_sources with the specified
                        overwrite
  --overwrite_vars BOOLEAN
                        only list inventory_sources with the specified
                        overwrite_vars
  --custom_virtualenv TEXT
                        only list inventory_sources with the specified
                        custom_virtualenv
  --timeout INTEGER     only list inventory_sources with the specified timeout
  --verbosity {0,1,2}   only list inventory_sources with the specified
                        verbosity
  --limit TEXT          only list inventory_sources with the specified limit
  --last_job_run LAST_JOB_RUN
                        only list inventory_sources with the specified
                        last_job_run
  --last_job_failed BOOLEAN
                        only list inventory_sources with the specified
                        last_job_failed
```

(continues on next page)

(продолжение с предыдущей страницы)

```
  --next_job_run NEXT_JOB_RUN
                        only list inventory_sources with the specified
                        next_job_run
  --status {new,pending,waiting,running,successful,failed,error,canceled,never updated,
↪none}
                        only list inventory_sources with the specified status
  --execution_environment ID
                        only list inventory_sources with the specified
                        execution_environment
  --inventory ID        only list inventory_sources with the specified
                        inventory
  --update_on_launch BOOLEAN
                        only list inventory_sources with the specified
                        update_on_launch
  --update_cache_timeout INTEGER
                        only list inventory_sources with the specified
                        update_cache_timeout
  --source_project ID   only list inventory_sources with the specified
                        source_project

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made
```

inventory_sources create

```
usage: awx inventory_sources create [-h] --name TEXT [--description TEXT]
                                    [--source {file,constructed,scm,ec2,gce,azure_rm,
↪vmware,satellite6,openstack,rhv,controller,insights}]
                                    [--source_path TEXT] [--source_vars TEXT]
                                    [--scm_branch TEXT] [--credential INTEGER]
                                    [--enabled_var TEXT]
                                    [--enabled_value TEXT]
                                    [--host_filter TEXT] [--overwrite BOOLEAN]
                                    [--overwrite_vars BOOLEAN]
                                    [--timeout INTEGER] [--verbosity {0,1,2}]
                                    [--limit TEXT]
                                    [--execution_environment ID] --inventory
                                    ID [--update_on_launch BOOLEAN]
                                    [--update_cache_timeout INTEGER]
                                    [--source_project ID]

required arguments:
  --name TEXT           Name of this inventory source.
  --inventory ID        the ID of the associated inventory

optional arguments:
  -h, --help            show this help message and exit
  --description TEXT    Optional description of this inventory source.
  --source {file,constructed,scm,ec2,gce,azure_rm,vmware,satellite6,openstack,rhv,
↪controller,insights}
  --source_path TEXT
  --source_vars TEXT    Inventory source variables in YAML or JSON format.
  --scm_branch TEXT     Inventory source SCM branch. Project default used if
                        blank. Only allowed if project allow_override field is
                        set to true.
  --credential INTEGER  Cloud credential to use for inventory updates.
  --enabled_var TEXT    Retrieve the enabled state from the given dict of host
```

(continues on next page)

(продолжение с предыдущей страницы)

```
                        variables. The enabled variable may be specified as
                        "foo.bar", in which case the lookup will traverse into
                        nested dicts, equivalent to: from_dict.get("foo",
                        {}).get("bar", default)
  --enabled_value TEXT  Only used when enabled_var is set. Value when the host
                        is considered enabled. For example if
                        enabled_var="status.power_state"and
                        enabled_value="powered_on" with host variables:{
                        "status": { "power_state": "powered_on", "created":
                        "2020-08-04T18:13:04+00:00", "healthy": true },
                        "name": "foobar", "ip_address": "192.168.2.1"}The host
                        would be marked enabled. If power_state where any
                        value other than powered_on then the host would be
                        disabled when imported. If the key is not found then
                        the host will be enabled
  --host_filter TEXT    This field is deprecated and will be removed in a
                        future release. Regex where only matching hosts will
                        be imported.
  --overwrite BOOLEAN   Overwrite local groups and hosts from remote inventory
                        source.
  --overwrite_vars BOOLEAN
                        Overwrite local variables from remote inventory
                        source.
  --timeout INTEGER     The amount of time (in seconds) to run before the task
                        is canceled.
  --verbosity {0,1,2}
  --limit TEXT          Enter host, group or pattern match
  --execution_environment ID
                        The container image to be used for execution.
  --update_on_launch BOOLEAN
  --update_cache_timeout INTEGER
  --source_project ID   Project containing inventory file used as source.

awx inventory_sources create: the following arguments are required: --name, --inventory
```

inventory_sources get

```
usage: awx inventory_sources get [-h] [-f {json,yaml,jq,human}]
                                 [--filter TEXT] [--conf.color BOOLEAN] [-v]
                                 id

positional arguments:
  id                    the ID (or unique name) of the resource

optional arguments:
  -h, --help            show this help message and exit

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made

awx inventory_sources get: the following arguments are required: id
```

inventory_sources modify

```
usage: awx inventory_sources modify [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx inventory_sources modify: the following arguments are required: id
```

inventory_sources delete

```
usage: awx inventory_sources delete [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx inventory_sources delete: the following arguments are required: id
```

inventory_sources update

```
usage: awx inventory_sources update [-h] [--monitor]
                                    [--action-timeout ACTION_TIMEOUT] [--wait]
                                    [--interval INTERVAL]
                                    id

positional arguments:
  id

optional arguments:
  -h, --help            show this help message and exit
  --monitor             If set, prints stdout of the launched job until it
                        finishes.
  --action-timeout ACTION_TIMEOUT
                        If set with --monitor or --wait, time out waiting on
                        job completion.
  --wait                If set, waits until the launched job finishes.
  --interval INTERVAL   If set with --monitor or --wait, amount of time to
                        wait in seconds between api calls. Minimum value is
                        2.5 seconds to avoid overwhelming the api

awx inventory_sources update: the following arguments are required: id
```

inventory_sources associate

```
usage: awx inventory_sources associate [-h]
                                       (--start_notification  | --success_notification ␣
↪| --failure_notification )
                                       id

positional arguments:
  id

optional arguments:
  -h, --help            show this help message and exit
  --start_notification
                        The ID (or name) of the notification_template to
                        associate
  --success_notification
                        The ID (or name) of the notification_template to
                        associate
  --failure_notification
                        The ID (or name) of the notification_template to
                        associate

awx inventory_sources associate: the following arguments are required: id
```

inventory_sources disassociate

```
usage: awx inventory_sources disassociate [-h]
                                          (--start_notification  | --success_
↪notification  | --failure_notification )
                                          id

positional arguments:
  id

optional arguments:
  -h, --help            show this help message and exit
  --start_notification
                        The ID (or name) of the notification_template to
                        disassociate
  --success_notification
                        The ID (or name) of the notification_template to
                        disassociate
  --failure_notification
                        The ID (or name) of the notification_template to
                        disassociate

awx inventory_sources disassociate: the following arguments are required: id
```

inventory_updates

```
usage: awx inventory_updates [-h] action ...

positional arguments:
  action
    list
    get
    delete
    stdout

optional arguments:
```

(continues on next page)

(продолжение с предыдущей страницы)

```
  -h, --help  show this help message and exit

awx inventory_updates: the following arguments are required: action
```

inventory_updates list

```
usage: awx inventory_updates list [-h] [--all] [--order_by ORDER_BY]
                                  [-f {json,yaml,jq,human}] [--filter TEXT]
                                  [--conf.color BOOLEAN] [-v]
                                  [--type {inventory_update}]
                                  [--created CREATED] [--modified MODIFIED]
                                  [--name TEXT] [--description TEXT]
                                  [--unified_job_template ID]
                                  [--launch_type {manual,relaunch,callback,scheduled,
↪dependency,workflow,webhook,sync,scm}]
                                  [--status {new,pending,waiting,running,successful,
↪failed,error,canceled}]
                                  [--execution_environment ID]
                                  [--failed BOOLEAN] [--started STARTED]
                                  [--finished FINISHED]
                                  [--canceled_on CANCELED_ON]
                                  [--elapsed ELAPSED] [--job_explanation TEXT]
                                  [--execution_node TEXT]
                                  [--controller_node TEXT]
                                  [--work_unit_id TEXT]
                                  [--source {file,constructed,scm,ec2,gce,azure_rm,
↪vmware,satellite6,openstack,rhv,controller,insights}]
                                  [--source_path TEXT] [--source_vars TEXT]
                                  [--scm_branch TEXT] [--enabled_var TEXT]
                                  [--enabled_value TEXT] [--host_filter TEXT]
                                  [--overwrite BOOLEAN]
                                  [--overwrite_vars BOOLEAN]
                                  [--custom_virtualenv TEXT]
                                  [--timeout INTEGER] [--verbosity {0,1,2}]
                                  [--limit TEXT] [--inventory ID]
                                  [--inventory_source ID]
                                  [--license_error BOOLEAN]
                                  [--org_host_limit_error BOOLEAN]
                                  [--source_project_update ID]
                                  [--instance_group ID] [--scm_revision TEXT]

optional arguments:
  -h, --help            show this help message and exit
  --all                 fetch all pages of content from the API when returning
                        results (instead of just the first page)
  --order_by ORDER_BY   order results by given field name, prefix the field
                        name with a dash (-) to sort in reverse eg
                        --order_by='-name',multiple sorting fields may be
                        specified by separating the field names with a comma
                        (,)
  --type {inventory_update}
                        only list inventory_updates with the specified type
  --created CREATED     only list inventory_updates with the specified created
  --modified MODIFIED   only list inventory_updates with the specified
                        modified
  --name TEXT           only list inventory_updates with the specified name
  --description TEXT    only list inventory_updates with the specified
                        description
  --unified_job_template ID
                        only list inventory_updates with the specified
                        unified_job_template
  --launch_type {manual,relaunch,callback,scheduled,dependency,workflow,webhook,sync,scm}
```

(continues on next page)

(продолжение с предыдущей страницы)

```
                        only list inventory_updates with the specified
                        launch_type
  --status {new,pending,waiting,running,successful,failed,error,canceled}
                        only list inventory_updates with the specified status
  --execution_environment ID
                        only list inventory_updates with the specified
                        execution_environment
  --failed BOOLEAN      only list inventory_updates with the specified failed
  --started STARTED     only list inventory_updates with the specified started
  --finished FINISHED   only list inventory_updates with the specified
                        finished
  --canceled_on CANCELED_ON
                        only list inventory_updates with the specified
                        canceled_on
  --elapsed ELAPSED     only list inventory_updates with the specified elapsed
  --job_explanation TEXT
                        only list inventory_updates with the specified
                        job_explanation
  --execution_node TEXT
                        only list inventory_updates with the specified
                        execution_node
  --controller_node TEXT
                        only list inventory_updates with the specified
                        controller_node
  --work_unit_id TEXT   only list inventory_updates with the specified
                        work_unit_id
  --source {file,constructed,scm,ec2,gce,azure_rm,vmware,satellite6,openstack,rhv,
↪controller,insights}
                        only list inventory_updates with the specified source
  --source_path TEXT    only list inventory_updates with the specified
                        source_path
  --source_vars TEXT    only list inventory_updates with the specified
                        source_vars
  --scm_branch TEXT     only list inventory_updates with the specified
                        scm_branch
  --enabled_var TEXT    only list inventory_updates with the specified
                        enabled_var
  --enabled_value TEXT  only list inventory_updates with the specified
                        enabled_value
  --host_filter TEXT    only list inventory_updates with the specified
                        host_filter
  --overwrite BOOLEAN   only list inventory_updates with the specified
                        overwrite
  --overwrite_vars BOOLEAN
                        only list inventory_updates with the specified
                        overwrite_vars
  --custom_virtualenv TEXT
                        only list inventory_updates with the specified
                        custom_virtualenv
  --timeout INTEGER     only list inventory_updates with the specified timeout
  --verbosity {0,1,2}   only list inventory_updates with the specified
                        verbosity
  --limit TEXT          only list inventory_updates with the specified limit
  --inventory ID        only list inventory_updates with the specified
                        inventory
  --inventory_source ID
                        only list inventory_updates with the specified
                        inventory_source
  --license_error BOOLEAN
                        only list inventory_updates with the specified
                        license_error
  --org_host_limit_error BOOLEAN
                        only list inventory_updates with the specified
                        org_host_limit_error
```

(continues on next page)

(продолжение с предыдущей страницы)

```
  --source_project_update ID
                        only list inventory_updates with the specified
                        source_project_update
  --instance_group ID   only list inventory_updates with the specified
                        instance_group
  --scm_revision TEXT   only list inventory_updates with the specified
                        scm_revision

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made
```

inventory_updates get

```
usage: awx inventory_updates get [-h] [-f {json,yaml,jq,human}]
                                 [--filter TEXT] [--conf.color BOOLEAN] [-v]
                                 id

positional arguments:
  id                    the ID (or unique name) of the resource

optional arguments:
  -h, --help            show this help message and exit

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made

awx inventory_updates get: the following arguments are required: id
```

inventory_updates delete

```
usage: awx inventory_updates delete [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx inventory_updates delete: the following arguments are required: id
```

inventory_updates stdout

```
usage: awx inventory_updates stdout [-h] id

positional arguments:
  id

optional arguments:
  -h, --help  show this help message and exit

awx inventory_updates stdout: the following arguments are required: id
```

job_templates

```
usage: awx job_templates [-h] action ...

positional arguments:
  action
    list
    create
    get
    modify
    delete
    launch
    associate
    disassociate

optional arguments:
  -h, --help    show this help message and exit

awx job_templates: the following arguments are required: action
```

job_templates list

```
usage: awx job_templates list [-h] [--all] [--order_by ORDER_BY]
                              [-f {json,yaml,jq,human}] [--filter TEXT]
                              [--conf.color BOOLEAN] [-v]
                              [--type {job_template}] [--created CREATED]
                              [--modified MODIFIED] [--name TEXT]
                              [--description TEXT] [--job_type {run,check}]
                              [--inventory ID] [--project ID]
                              [--playbook TEXT] [--scm_branch TEXT]
                              [--forks INTEGER] [--limit TEXT]
                              [--verbosity {0,1,2,3,4,5}]
                              [--extra_vars JSON/YAML] [--job_tags TEXT]
                              [--force_handlers BOOLEAN] [--skip_tags TEXT]
                              [--start_at_task TEXT] [--timeout INTEGER]
                              [--use_fact_cache BOOLEAN] [--organization ID]
                              [--last_job_run LAST_JOB_RUN]
                              [--last_job_failed BOOLEAN]
                              [--next_job_run NEXT_JOB_RUN]
                              [--status {new,pending,waiting,running,successful,failed,
↪error,canceled,never updated}]
                              [--execution_environment ID]
                              [--host_config_key TEXT]
                              [--ask_scm_branch_on_launch BOOLEAN]
                              [--ask_diff_mode_on_launch BOOLEAN]
                              [--ask_variables_on_launch BOOLEAN]
                              [--ask_limit_on_launch BOOLEAN]
                              [--ask_tags_on_launch BOOLEAN]
```

(continues on next page)

(продолжение с предыдущей страницы)

```
                              [--ask_skip_tags_on_launch BOOLEAN]
                              [--ask_job_type_on_launch BOOLEAN]
                              [--ask_verbosity_on_launch BOOLEAN]
                              [--ask_inventory_on_launch BOOLEAN]
                              [--ask_credential_on_launch BOOLEAN]
                              [--ask_execution_environment_on_launch BOOLEAN]
                              [--ask_labels_on_launch BOOLEAN]
                              [--ask_forks_on_launch BOOLEAN]
                              [--ask_job_slice_count_on_launch BOOLEAN]
                              [--ask_timeout_on_launch BOOLEAN]
                              [--ask_instance_groups_on_launch BOOLEAN]
                              [--survey_enabled BOOLEAN]
                              [--become_enabled BOOLEAN] [--diff_mode BOOLEAN]
                              [--allow_simultaneous BOOLEAN]
                              [--custom_virtualenv TEXT]
                              [--job_slice_count INTEGER]
                              [--webhook_service {,github,gitlab,bitbucket_dc}]
                              [--webhook_credential ID]
                              [--prevent_instance_group_fallback BOOLEAN]

optional arguments:
  -h, --help            show this help message and exit
  --all                 fetch all pages of content from the API when returning
                        results (instead of just the first page)
  --order_by ORDER_BY   order results by given field name, prefix the field
                        name with a dash (-) to sort in reverse eg
                        --order_by='-name',multiple sorting fields may be
                        specified by separating the field names with a comma
                        (,)
  --type {job_template}
                        only list job_templates with the specified type
  --created CREATED     only list job_templates with the specified created
  --modified MODIFIED   only list job_templates with the specified modified
  --name TEXT           only list job_templates with the specified name
  --description TEXT    only list job_templates with the specified description
  --job_type {run,check}
                        only list job_templates with the specified job_type
  --inventory ID        only list job_templates with the specified inventory
  --project ID          only list job_templates with the specified project
  --playbook TEXT       only list job_templates with the specified playbook
  --scm_branch TEXT     only list job_templates with the specified scm_branch
  --forks INTEGER       only list job_templates with the specified forks
  --limit TEXT          only list job_templates with the specified limit
  --verbosity {0,1,2,3,4,5}
                        only list job_templates with the specified verbosity
  --extra_vars JSON/YAML, -e JSON/YAML
                        only list job_templates with the specified extra_vars
  --job_tags TEXT       only list job_templates with the specified job_tags
  --force_handlers BOOLEAN
                        only list job_templates with the specified
                        force_handlers
  --skip_tags TEXT      only list job_templates with the specified skip_tags
  --start_at_task TEXT  only list job_templates with the specified
                        start_at_task
  --timeout INTEGER     only list job_templates with the specified timeout
  --use_fact_cache BOOLEAN
                        only list job_templates with the specified
                        use_fact_cache
  --organization ID     only list job_templates with the specified
                        organization
  --last_job_run LAST_JOB_RUN
                        only list job_templates with the specified
                        last_job_run
  --last_job_failed BOOLEAN
```

(continues on next page)

(продолжение с предыдущей страницы)

```
                        only list job_templates with the specified
                        last_job_failed
  --next_job_run NEXT_JOB_RUN
                        only list job_templates with the specified
                        next_job_run
  --status {new,pending,waiting,running,successful,failed,error,canceled,never updated}
                        only list job_templates with the specified status
  --execution_environment ID
                        only list job_templates with the specified
                        execution_environment
  --host_config_key TEXT
                        only list job_templates with the specified
                        host_config_key
  --ask_scm_branch_on_launch BOOLEAN
                        only list job_templates with the specified
                        ask_scm_branch_on_launch
  --ask_diff_mode_on_launch BOOLEAN
                        only list job_templates with the specified
                        ask_diff_mode_on_launch
  --ask_variables_on_launch BOOLEAN
                        only list job_templates with the specified
                        ask_variables_on_launch
  --ask_limit_on_launch BOOLEAN
                        only list job_templates with the specified
                        ask_limit_on_launch
  --ask_tags_on_launch BOOLEAN
                        only list job_templates with the specified
                        ask_tags_on_launch
  --ask_skip_tags_on_launch BOOLEAN
                        only list job_templates with the specified
                        ask_skip_tags_on_launch
  --ask_job_type_on_launch BOOLEAN
                        only list job_templates with the specified
                        ask_job_type_on_launch
  --ask_verbosity_on_launch BOOLEAN
                        only list job_templates with the specified
                        ask_verbosity_on_launch
  --ask_inventory_on_launch BOOLEAN
                        only list job_templates with the specified
                        ask_inventory_on_launch
  --ask_credential_on_launch BOOLEAN
                        only list job_templates with the specified
                        ask_credential_on_launch
  --ask_execution_environment_on_launch BOOLEAN
                        only list job_templates with the specified
                        ask_execution_environment_on_launch
  --ask_labels_on_launch BOOLEAN
                        only list job_templates with the specified
                        ask_labels_on_launch
  --ask_forks_on_launch BOOLEAN
                        only list job_templates with the specified
                        ask_forks_on_launch
  --ask_job_slice_count_on_launch BOOLEAN
                        only list job_templates with the specified
                        ask_job_slice_count_on_launch
  --ask_timeout_on_launch BOOLEAN
                        only list job_templates with the specified
                        ask_timeout_on_launch
  --ask_instance_groups_on_launch BOOLEAN
                        only list job_templates with the specified
                        ask_instance_groups_on_launch
  --survey_enabled BOOLEAN
                        only list job_templates with the specified
                        survey_enabled
```

(continues on next page)

(продолжение с предыдущей страницы)

```
  --become_enabled BOOLEAN
                        only list job_templates with the specified
                        become_enabled
  --diff_mode BOOLEAN   only list job_templates with the specified diff_mode
  --allow_simultaneous BOOLEAN
                        only list job_templates with the specified
                        allow_simultaneous
  --custom_virtualenv TEXT
                        only list job_templates with the specified
                        custom_virtualenv
  --job_slice_count INTEGER
                        only list job_templates with the specified
                        job_slice_count
  --webhook_service {,github,gitlab,bitbucket_dc}
                        only list job_templates with the specified
                        webhook_service
  --webhook_credential ID
                        only list job_templates with the specified
                        webhook_credential
  --prevent_instance_group_fallback BOOLEAN
                        only list job_templates with the specified
                        prevent_instance_group_fallback

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made
```

job_templates create

```
usage: awx job_templates create [-h] --name TEXT [--description TEXT]
                                [--job_type {run,check}] [--inventory ID]
                                --project ID --playbook TEXT
                                [--scm_branch TEXT] [--forks INTEGER]
                                [--limit TEXT] [--verbosity {0,1,2,3,4,5}]
                                [--extra_vars JSON/YAML] [--job_tags TEXT]
                                [--force_handlers BOOLEAN] [--skip_tags TEXT]
                                [--start_at_task TEXT] [--timeout INTEGER]
                                [--use_fact_cache BOOLEAN]
                                [--execution_environment ID]
                                [--host_config_key TEXT]
                                [--ask_scm_branch_on_launch BOOLEAN]
                                [--ask_diff_mode_on_launch BOOLEAN]
                                [--ask_variables_on_launch BOOLEAN]
                                [--ask_limit_on_launch BOOLEAN]
                                [--ask_tags_on_launch BOOLEAN]
                                [--ask_skip_tags_on_launch BOOLEAN]
                                [--ask_job_type_on_launch BOOLEAN]
                                [--ask_verbosity_on_launch BOOLEAN]
                                [--ask_inventory_on_launch BOOLEAN]
                                [--ask_credential_on_launch BOOLEAN]
                                [--ask_execution_environment_on_launch BOOLEAN]
                                [--ask_labels_on_launch BOOLEAN]
                                [--ask_forks_on_launch BOOLEAN]
                                [--ask_job_slice_count_on_launch BOOLEAN]
                                [--ask_timeout_on_launch BOOLEAN]
                                [--ask_instance_groups_on_launch BOOLEAN]
                                [--survey_enabled BOOLEAN]
                                [--become_enabled BOOLEAN]
```

(continues on next page)

(продолжение с предыдущей страницы)

```
                                [--diff_mode BOOLEAN]
                                [--allow_simultaneous BOOLEAN]
                                [--job_slice_count INTEGER]
                                [--webhook_service {,github,gitlab,bitbucket_dc}]
                                [--webhook_credential ID]
                                [--prevent_instance_group_fallback BOOLEAN]

required arguments:
  --name TEXT           Name of this job template.
  --project ID          the ID of the associated project
  --playbook TEXT

optional arguments:
  -h, --help            show this help message and exit
  --description TEXT    Optional description of this job template.
  --job_type {run,check}
  --inventory ID        the ID of the associated inventory
  --scm_branch TEXT     Branch to use in job run. Project default used if
                        blank. Only allowed if project allow_override field is
                        set to true.
  --forks INTEGER
  --limit TEXT
  --verbosity {0,1,2,3,4,5}
  --extra_vars JSON/YAML, -e JSON/YAML
                        a JSON or YAML string. You can optionally specify a
                        file path e.g., @path/to/file.yml
  --job_tags TEXT
  --force_handlers BOOLEAN
  --skip_tags TEXT
  --start_at_task TEXT
  --timeout INTEGER     The amount of time (in seconds) to run before the task
                        is canceled.
  --use_fact_cache BOOLEAN
                        If enabled, the service will act as an Ansible Fact
                        Cache Plugin; persisting facts at the end of a
                        playbook run to the database and caching facts for use
                        by Ansible.
  --execution_environment ID
                        The container image to be used for execution.
  --host_config_key TEXT
  --ask_scm_branch_on_launch BOOLEAN
  --ask_diff_mode_on_launch BOOLEAN
  --ask_variables_on_launch BOOLEAN
  --ask_limit_on_launch BOOLEAN
  --ask_tags_on_launch BOOLEAN
  --ask_skip_tags_on_launch BOOLEAN
  --ask_job_type_on_launch BOOLEAN
  --ask_verbosity_on_launch BOOLEAN
  --ask_inventory_on_launch BOOLEAN
  --ask_credential_on_launch BOOLEAN
  --ask_execution_environment_on_launch BOOLEAN
  --ask_labels_on_launch BOOLEAN
  --ask_forks_on_launch BOOLEAN
  --ask_job_slice_count_on_launch BOOLEAN
  --ask_timeout_on_launch BOOLEAN
  --ask_instance_groups_on_launch BOOLEAN
  --survey_enabled BOOLEAN
  --become_enabled BOOLEAN
  --diff_mode BOOLEAN   If enabled, textual changes made to any templated
                        files on the host are shown in the standard output
  --allow_simultaneous BOOLEAN
  --job_slice_count INTEGER
                        The number of jobs to slice into at runtime. Will
                        cause the Job Template to launch a workflow if value
```

(continues on next page)

(продолжение с предыдущей страницы)

```
                        is greater than 1.
  --webhook_service {,github,gitlab,bitbucket_dc}
                        Service that webhook requests will be accepted from
  --webhook_credential ID
                        Personal Access Token for posting back the status to
                        the service API
  --prevent_instance_group_fallback BOOLEAN
                        If enabled, the job template will prevent adding any
                        inventory or organization instance groups to the list
                        of preferred instances groups to run on.If this
                        setting is enabled and you provided an empty list, the
                        global instance groups will be applied.

awx job_templates create: the following arguments are required: --name, --project, --
↪playbook
```

job_templates get

```
usage: awx job_templates get [-h] [-f {json,yaml,jq,human}] [--filter TEXT]
                             [--conf.color BOOLEAN] [-v]
                             id

positional arguments:
  id                    the ID (or unique name) of the resource

optional arguments:
  -h, --help            show this help message and exit

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made

awx job_templates get: the following arguments are required: id
```

job_templates modify

```
usage: awx job_templates modify [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx job_templates modify: the following arguments are required: id
```

job_templates delete

```
usage: awx job_templates delete [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx job_templates delete: the following arguments are required: id
```

job_templates launch

```
usage: awx job_templates launch [-h] [--monitor]
                                [--action-timeout ACTION_TIMEOUT] [--wait]
                                [--interval INTERVAL] [--extra_vars JSON/YAML]
                                [--inventory ID] [--scm_branch TEXT]
                                [--limit TEXT] [--job_tags TEXT]
                                [--skip_tags TEXT] [--job_type {run,check}]
                                [--verbosity {0,1,2,3,4,5}]
                                [--diff_mode BOOLEAN]
                                [--credentials [ID, ID, ...]]
                                [--credential_passwords JSON/YAML]
                                [--execution_environment ID] [--labels LABELS]
                                [--forks INTEGER] [--job_slice_count INTEGER]
                                [--timeout INTEGER]
                                [--instance_groups INSTANCE_GROUPS]
                                id

positional arguments:
  id

optional arguments:
  -h, --help            show this help message and exit
  --monitor             If set, prints stdout of the launched job until it
                        finishes.
  --action-timeout ACTION_TIMEOUT
                        If set with --monitor or --wait, time out waiting on
                        job completion.
  --wait                If set, waits until the launched job finishes.
  --interval INTERVAL   If set with --monitor or --wait, amount of time to
                        wait in seconds between api calls. Minimum value is
                        2.5 seconds to avoid overwhelming the api
  --extra_vars JSON/YAML, -e JSON/YAML
                        a JSON or YAML string. You can optionally specify a
                        file path e.g., @path/to/file.yml
  --inventory ID        the ID of the associated inventory
  --scm_branch TEXT
  --limit TEXT
  --job_tags TEXT
  --skip_tags TEXT
  --job_type {run,check}
  --verbosity {0,1,2,3,4,5}
  --diff_mode BOOLEAN
  --credentials [ID, ID, ...]
                        a list of comma-delimited credentials to associate
                        (IDs or unique names)
  --credential_passwords JSON/YAML
                        a JSON or YAML string. You can optionally specify a
                        file path e.g., @path/to/file.yml
  --execution_environment ID
```

(continues on next page)

(продолжение с предыдущей страницы)

```
                        the ID of the associated execution_environment
  --labels LABELS
  --forks INTEGER
  --job_slice_count INTEGER
  --timeout INTEGER
  --instance_groups INSTANCE_GROUPS

awx job_templates launch: the following arguments are required: id
```

job_templates associate

```
usage: awx job_templates associate [-h]
                                   (--start_notification  | --success_notification  | --
↪failure_notification  | --credential )
                                   id

positional arguments:
  id

optional arguments:
  -h, --help            show this help message and exit
  --start_notification
                        The ID (or name) of the notification_template to
                        associate
  --success_notification
                        The ID (or name) of the notification_template to
                        associate
  --failure_notification
                        The ID (or name) of the notification_template to
                        associate
  --credential          The ID (or name) of the credential to associate

awx job_templates associate: the following arguments are required: id
```

job_templates disassociate

```
usage: awx job_templates disassociate [-h]
                                      (--start_notification  | --success_notification  |␣
↪--failure_notification  | --credential )
                                      id

positional arguments:
  id

optional arguments:
  -h, --help            show this help message and exit
  --start_notification
                        The ID (or name) of the notification_template to
                        disassociate
  --success_notification
                        The ID (or name) of the notification_template to
                        disassociate
  --failure_notification
                        The ID (or name) of the notification_template to
                        disassociate
  --credential          The ID (or name) of the credential to disassociate

awx job_templates disassociate: the following arguments are required: id
```

jobs

```
usage: awx jobs [-h] action ...

positional arguments:
  action
    list
    get
    delete
    stdout
    monitor

optional arguments:
  -h, --help  show this help message and exit

awx jobs: the following arguments are required: action
```

jobs list

```
usage: awx jobs list [-h] [--all] [--order_by ORDER_BY]
                     [-f {json,yaml,jq,human}] [--filter TEXT]
                     [--conf.color BOOLEAN] [-v] [--type {job}]
                     [--created CREATED] [--modified MODIFIED] [--name TEXT]
                     [--description TEXT] [--unified_job_template ID]
                     [--launch_type {manual,relaunch,callback,scheduled,dependency,
↪workflow,webhook,sync,scm}]
                     [--status {new,pending,waiting,running,successful,failed,error,
↪canceled}]
                     [--execution_environment ID] [--failed BOOLEAN]
                     [--started STARTED] [--finished FINISHED]
                     [--canceled_on CANCELED_ON] [--elapsed ELAPSED]
                     [--job_explanation TEXT] [--execution_node TEXT]
                     [--controller_node TEXT] [--work_unit_id TEXT]
                     [--job_type {run,check,scan}] [--inventory ID]
                     [--project ID] [--playbook TEXT] [--scm_branch TEXT]
                     [--forks INTEGER] [--limit TEXT]
                     [--verbosity {0,1,2,3,4,5}] [--extra_vars JSON/YAML]
                     [--job_tags TEXT] [--force_handlers BOOLEAN]
                     [--skip_tags TEXT] [--start_at_task TEXT]
                     [--timeout INTEGER] [--use_fact_cache BOOLEAN]
                     [--organization ID] [--job_template ID]
                     [--allow_simultaneous BOOLEAN] [--artifacts JSON/YAML]
                     [--scm_revision TEXT] [--instance_group ID]
                     [--diff_mode BOOLEAN] [--job_slice_number INTEGER]
                     [--job_slice_count INTEGER]
                     [--webhook_service {,github,gitlab,bitbucket_dc}]
                     [--webhook_credential ID] [--webhook_guid TEXT]

optional arguments:
  -h, --help            show this help message and exit
  --all                 fetch all pages of content from the API when returning
                        results (instead of just the first page)
  --order_by ORDER_BY   order results by given field name, prefix the field
                        name with a dash (-) to sort in reverse eg
                        --order_by='-name',multiple sorting fields may be
                        specified by separating the field names with a comma
                        (,)
  --type {job}          only list jobs with the specified type
  --created CREATED     only list jobs with the specified created
  --modified MODIFIED   only list jobs with the specified modified
  --name TEXT           only list jobs with the specified name
  --description TEXT    only list jobs with the specified description
  --unified_job_template ID
```

(continues on next page)

(продолжение с предыдущей страницы)

```
                        only list jobs with the specified unified_job_template
  --launch_type {manual,relaunch,callback,scheduled,dependency,workflow,webhook,sync,scm}
                        only list jobs with the specified launch_type
  --status {new,pending,waiting,running,successful,failed,error,canceled}
                        only list jobs with the specified status
  --execution_environment ID
                        only list jobs with the specified
                        execution_environment
  --failed BOOLEAN      only list jobs with the specified failed
  --started STARTED     only list jobs with the specified started
  --finished FINISHED   only list jobs with the specified finished
  --canceled_on CANCELED_ON
                        only list jobs with the specified canceled_on
  --elapsed ELAPSED     only list jobs with the specified elapsed
  --job_explanation TEXT
                        only list jobs with the specified job_explanation
  --execution_node TEXT
                        only list jobs with the specified execution_node
  --controller_node TEXT
                        only list jobs with the specified controller_node
  --work_unit_id TEXT   only list jobs with the specified work_unit_id
  --job_type {run,check,scan}
                        only list jobs with the specified job_type
  --inventory ID        only list jobs with the specified inventory
  --project ID          only list jobs with the specified project
  --playbook TEXT       only list jobs with the specified playbook
  --scm_branch TEXT     only list jobs with the specified scm_branch
  --forks INTEGER       only list jobs with the specified forks
  --limit TEXT          only list jobs with the specified limit
  --verbosity {0,1,2,3,4,5}
                        only list jobs with the specified verbosity
  --extra_vars JSON/YAML
                        only list jobs with the specified extra_vars
  --job_tags TEXT       only list jobs with the specified job_tags
  --force_handlers BOOLEAN
                        only list jobs with the specified force_handlers
  --skip_tags TEXT      only list jobs with the specified skip_tags
  --start_at_task TEXT  only list jobs with the specified start_at_task
  --timeout INTEGER     only list jobs with the specified timeout
  --use_fact_cache BOOLEAN
                        only list jobs with the specified use_fact_cache
  --organization ID     only list jobs with the specified organization
  --job_template ID     only list jobs with the specified job_template
  --allow_simultaneous BOOLEAN
                        only list jobs with the specified allow_simultaneous
  --artifacts JSON/YAML
                        only list jobs with the specified artifacts
  --scm_revision TEXT   only list jobs with the specified scm_revision
  --instance_group ID   only list jobs with the specified instance_group
  --diff_mode BOOLEAN   only list jobs with the specified diff_mode
  --job_slice_number INTEGER
                        only list jobs with the specified job_slice_number
  --job_slice_count INTEGER
                        only list jobs with the specified job_slice_count
  --webhook_service {,github,gitlab,bitbucket_dc}
                        only list jobs with the specified webhook_service
  --webhook_credential ID
                        only list jobs with the specified webhook_credential
  --webhook_guid TEXT   only list jobs with the specified webhook_guid

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
```

(continues on next page)

(продолжение с предыдущей страницы)

```
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made
```

jobs get

```
usage: awx jobs get [-h] [-f {json,yaml,jq,human}] [--filter TEXT]
                    [--conf.color BOOLEAN] [-v]
                    id

positional arguments:
  id                    the ID (or unique name) of the resource

optional arguments:
  -h, --help            show this help message and exit

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made

awx jobs get: the following arguments are required: id
```

jobs delete

```
usage: awx jobs delete [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx jobs delete: the following arguments are required: id
```

jobs stdout

```
usage: awx jobs stdout [-h] id

positional arguments:
  id

optional arguments:
  -h, --help  show this help message and exit

awx jobs stdout: the following arguments are required: id
```

jobs monitor

```
usage: awx jobs monitor [-h] id

positional arguments:
  id

optional arguments:
  -h, --help  show this help message and exit

awx jobs monitor: the following arguments are required: id
```

labels

```
usage: awx labels [-h] action ...

positional arguments:
  action
    list
    create
    get
    modify

optional arguments:
  -h, --help  show this help message and exit

awx labels: the following arguments are required: action
```

labels list

```
usage: awx labels list [-h] [--all] [--order_by ORDER_BY]
                       [-f {json,yaml,jq,human}] [--filter TEXT]
                       [--conf.color BOOLEAN] [-v] [--type {label}]
                       [--created CREATED] [--modified MODIFIED] [--name TEXT]
                       [--organization ID]

optional arguments:
  -h, --help            show this help message and exit
  --all                 fetch all pages of content from the API when returning
                        results (instead of just the first page)
  --order_by ORDER_BY   order results by given field name, prefix the field
                        name with a dash (-) to sort in reverse eg
                        --order_by='-name',multiple sorting fields may be
                        specified by separating the field names with a comma
                        (,)
  --type {label}        only list labels with the specified type
  --created CREATED     only list labels with the specified created
  --modified MODIFIED   only list labels with the specified modified
  --name TEXT           only list labels with the specified name
  --organization ID     only list labels with the specified organization

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made
```

labels create

```
usage: awx labels create [-h] --name TEXT --organization ID

required arguments:
  --name TEXT        Name of this label.
  --organization ID  Organization this label belongs to.

optional arguments:
  -h, --help         show this help message and exit

awx labels create: the following arguments are required: --name, --organization
```

labels get

```
usage: awx labels get [-h] [-f {json,yaml,jq,human}] [--filter TEXT]
                      [--conf.color BOOLEAN] [-v]
                      id

positional arguments:
  id                    the ID (or unique name) of the resource

optional arguments:
  -h, --help            show this help message and exit

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made

awx labels get: the following arguments are required: id
```

labels modify

```
usage: awx labels modify [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx labels modify: the following arguments are required: id
```

login

```
usage: awx login [-h] [--description TEXT] [--conf.client_id TEXT]
                 [--conf.client_secret TEXT] [--conf.scope {read,write}]
                 [--conf.host https://example.awx.org] [--conf.token TEXT]
                 [--conf.username TEXT] [--conf.password TEXT] [-k]

optional arguments:
  -h, --help            show this help message and exit

OAuth2.0 Options:
```

(continues on next page)

(продолжение с предыдущей страницы)

```
  --description TEXT    description of the generated OAuth2.0 token
  --conf.client_id TEXT
  --conf.client_secret TEXT
  --conf.scope {read,write}

authentication:
  --conf.host https://example.awx.org
  --conf.token TEXT     an OAuth2.0 token (get one by using `awx login`)
  --conf.username TEXT
  --conf.password TEXT
  -k, --conf.insecure   Allow insecure server connections when using SSL
```

me

```
usage: awx me [-h]

optional arguments:
  -h, --help  show this help message and exit
```

mesh_visualizer

```
usage: awx mesh_visualizer [-h]

optional arguments:
  -h, --help  show this help message and exit
```

metrics

```
usage: awx metrics [-h]

optional arguments:
  -h, --help  show this help message and exit
```

notification_templates

```
usage: awx notification_templates [-h] action ...

positional arguments:
  action
    list
    create
    get
    modify
    delete

optional arguments:
  -h, --help  show this help message and exit

awx notification_templates: the following arguments are required: action
```

notification_templates list

```
usage: awx notification_templates list [-h] [--all] [--order_by ORDER_BY]
                                       [-f {json,yaml,jq,human}]
                                       [--filter TEXT] [--conf.color BOOLEAN]
                                       [-v] [--type {notification_template}]
                                       [--created CREATED]
                                       [--modified MODIFIED] [--name TEXT]
                                       [--description TEXT]
                                       [--organization ID]
                                       [--notification_type {email,grafana,irc,
↪mattermost,pagerduty,rocketchat,slack,twilio,webhook}]
                                       [--notification_configuration JSON/YAML]
                                       [--messages JSON/YAML]

optional arguments:
  -h, --help            show this help message and exit
  --all                 fetch all pages of content from the API when returning
                        results (instead of just the first page)
  --order_by ORDER_BY   order results by given field name, prefix the field
                        name with a dash (-) to sort in reverse eg
                        --order_by='-name',multiple sorting fields may be
                        specified by separating the field names with a comma
                        (,)
  --type {notification_template}
                        only list notification_templates with the specified
                        type
  --created CREATED     only list notification_templates with the specified
                        created
  --modified MODIFIED   only list notification_templates with the specified
                        modified
  --name TEXT           only list notification_templates with the specified
                        name
  --description TEXT    only list notification_templates with the specified
                        description
  --organization ID     only list notification_templates with the specified
                        organization
  --notification_type {email,grafana,irc,mattermost,pagerduty,rocketchat,slack,twilio,
↪webhook}
                        only list notification_templates with the specified
                        notification_type
  --notification_configuration JSON/YAML
                        only list notification_templates with the specified
                        notification_configuration
  --messages JSON/YAML  only list notification_templates with the specified
                        messages

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made
```

notification_templates create

```
usage: awx notification_templates create [-h] --name TEXT [--description TEXT]
                                         --organization ID --notification_type
                                         {email,grafana,irc,mattermost,pagerduty,
↪rocketchat,slack,twilio,webhook}
                                         [--notification_configuration JSON/YAML]
                                         [--messages JSON/YAML]

required arguments:
  --name TEXT           Name of this notification template.
  --organization ID     the ID of the associated organization
  --notification_type {email,grafana,irc,mattermost,pagerduty,rocketchat,slack,twilio,
↪webhook}

optional arguments:
  -h, --help            show this help message and exit
  --description TEXT    Optional description of this notification template.
  --notification_configuration JSON/YAML
                        a JSON or YAML string. You can optionally specify a
                        file path e.g., @path/to/file.yml
  --messages JSON/YAML  Optional custom messages for notification template.
                        You can optionally specify a file path e.g.,
                        @path/to/file.yml

awx notification_templates create: the following arguments are required: --name, --
↪organization, --notification_type
```

notification_templates get

```
usage: awx notification_templates get [-h] [-f {json,yaml,jq,human}]
                                      [--filter TEXT] [--conf.color BOOLEAN]
                                      [-v]
                                      id

positional arguments:
  id                    the ID (or unique name) of the resource

optional arguments:
  -h, --help            show this help message and exit

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made

awx notification_templates get: the following arguments are required: id
```

notification_templates modify

```
usage: awx notification_templates modify [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx notification_templates modify: the following arguments are required: id
```

notification_templates delete

```
usage: awx notification_templates delete [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx notification_templates delete: the following arguments are required: id
```

notifications

```
usage: awx notifications [-h] action ...

positional arguments:
  action
    list
    get

optional arguments:
  -h, --help  show this help message and exit

awx notifications: the following arguments are required: action
```

notifications list

```
usage: awx notifications list [-h] [--all] [--order_by ORDER_BY]
                              [-f {json,yaml,jq,human}] [--filter TEXT]
                              [--conf.color BOOLEAN] [-v]
                              [--type {notification}] [--created CREATED]
                              [--modified MODIFIED]
                              [--notification_template ID] [--error TEXT]
                              [--status {pending,successful,failed}]
                              [--notifications_sent INTEGER]
                              [--notification_type {email,grafana,irc,mattermost,
↪pagerduty,rocketchat,slack,twilio,webhook}]
                              [--recipients TEXT] [--subject TEXT]
                              [--body JSON/YAML]

optional arguments:
  -h, --help            show this help message and exit
  --all                 fetch all pages of content from the API when returning
                        results (instead of just the first page)
  --order_by ORDER_BY   order results by given field name, prefix the field
```

(continues on next page)

(продолжение с предыдущей страницы)

```
                        name with a dash (-) to sort in reverse eg
                        --order_by='-name',multiple sorting fields may be
                        specified by separating the field names with a comma
                        (,)
  --type {notification}
                        only list notifications with the specified type
  --created CREATED     only list notifications with the specified created
  --modified MODIFIED   only list notifications with the specified modified
  --notification_template ID
                        only list notifications with the specified
                        notification_template
  --error TEXT          only list notifications with the specified error
  --status {pending,successful,failed}
                        only list notifications with the specified status
  --notifications_sent INTEGER
                        only list notifications with the specified
                        notifications_sent
  --notification_type {email,grafana,irc,mattermost,pagerduty,rocketchat,slack,twilio,
↪webhook}
                        only list notifications with the specified
                        notification_type
  --recipients TEXT     only list notifications with the specified recipients
  --subject TEXT        only list notifications with the specified subject
  --body JSON/YAML      only list notifications with the specified body

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made
```

notifications get

```
usage: awx notifications get [-h] [-f {json,yaml,jq,human}] [--filter TEXT]
                             [--conf.color BOOLEAN] [-v]
                             id

positional arguments:
  id                    the ID (or unique name) of the resource

optional arguments:
  -h, --help            show this help message and exit

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made

awx notifications get: the following arguments are required: id
```

organizations

```
usage: awx organizations [-h] action ...

positional arguments:
  action
    list
    create
    get
    modify
    delete
    associate
    disassociate

optional arguments:
  -h, --help    show this help message and exit

awx organizations: the following arguments are required: action
```

organizations list

```
usage: awx organizations list [-h] [--all] [--order_by ORDER_BY]
                              [-f {json,yaml,jq,human}] [--filter TEXT]
                              [--conf.color BOOLEAN] [-v]
                              [--type {organization}] [--created CREATED]
                              [--modified MODIFIED] [--name TEXT]
                              [--description TEXT] [--max_hosts INTEGER]
                              [--custom_virtualenv TEXT]
                              [--default_environment ID]

optional arguments:
  -h, --help            show this help message and exit
  --all                 fetch all pages of content from the API when returning
                        results (instead of just the first page)
  --order_by ORDER_BY   order results by given field name, prefix the field
                        name with a dash (-) to sort in reverse eg
                        --order_by='-name',multiple sorting fields may be
                        specified by separating the field names with a comma
                        (,)
  --type {organization}
                        only list organizations with the specified type
  --created CREATED     only list organizations with the specified created
  --modified MODIFIED   only list organizations with the specified modified
  --name TEXT           only list organizations with the specified name
  --description TEXT    only list organizations with the specified description
  --max_hosts INTEGER   only list organizations with the specified max_hosts
  --custom_virtualenv TEXT
                        only list organizations with the specified
                        custom_virtualenv
  --default_environment ID
                        only list organizations with the specified
                        default_environment

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made
```

organizations create

```
usage: awx organizations create [-h] --name TEXT [--description TEXT]
                                [--max_hosts INTEGER]
                                [--default_environment ID]

required arguments:
  --name TEXT           Name of this organization.

optional arguments:
  -h, --help            show this help message and exit
  --description TEXT    Optional description of this organization.
  --max_hosts INTEGER   Maximum number of hosts allowed to be managed by this
                        organization.
  --default_environment ID
                        The default execution environment for jobs run by this
                        organization.

awx organizations create: the following arguments are required: --name
```

organizations get

```
usage: awx organizations get [-h] [-f {json,yaml,jq,human}] [--filter TEXT]
                             [--conf.color BOOLEAN] [-v]
                             id

positional arguments:
  id                    the ID (or unique name) of the resource

optional arguments:
  -h, --help            show this help message and exit

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made

awx organizations get: the following arguments are required: id
```

organizations modify

```
usage: awx organizations modify [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx organizations modify: the following arguments are required: id
```

organizations delete

```
usage: awx organizations delete [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx organizations delete: the following arguments are required: id
```

organizations associate

```
usage: awx organizations associate [-h]
                                   (--start_notification  | --success_notification  | --
↪failure_notification  | --approval_notification  | --galaxy_credential )
                                   id

positional arguments:
  id

optional arguments:
  -h, --help            show this help message and exit
  --start_notification
                        The ID (or name) of the notification_template to
                        associate
  --success_notification
                        The ID (or name) of the notification_template to
                        associate
  --failure_notification
                        The ID (or name) of the notification_template to
                        associate
  --approval_notification
                        The ID (or name) of the notification_template to
                        associate
  --galaxy_credential   The ID (or name) of the credential to associate

awx organizations associate: the following arguments are required: id
```

organizations disassociate

```
usage: awx organizations disassociate [-h]
                                      (--start_notification  | --success_notification  |␣
↪--failure_notification  | --approval_notification  | --galaxy_credential )
                                      id

positional arguments:
  id

optional arguments:
  -h, --help            show this help message and exit
  --start_notification
                        The ID (or name) of the notification_template to
                        disassociate
  --success_notification
                        The ID (or name) of the notification_template to
                        disassociate
  --failure_notification
                        The ID (or name) of the notification_template to
```

(continues on next page)

(продолжение с предыдущей страницы)

```
                        disassociate
  --approval_notification
                        The ID (or name) of the notification_template to
                        disassociate
  --galaxy_credential   The ID (or name) of the credential to disassociate

awx organizations disassociate: the following arguments are required: id
```

ping

```
usage: awx ping [-h]

optional arguments:
  -h, --help  show this help message and exit
```

project_updates

```
usage: awx project_updates [-h] action ...

positional arguments:
  action
    list
    get
    delete
    stdout

optional arguments:
  -h, --help  show this help message and exit

awx project_updates: the following arguments are required: action
```

project_updates list

```
usage: awx project_updates list [-h] [--all] [--order_by ORDER_BY]
                                [-f {json,yaml,jq,human}] [--filter TEXT]
                                [--conf.color BOOLEAN] [-v]
                                [--type {project_update}] [--created CREATED]
                                [--modified MODIFIED] [--name TEXT]
                                [--description TEXT]
                                [--unified_job_template ID]
                                [--launch_type {manual,relaunch,callback,scheduled,
↪dependency,workflow,webhook,sync,scm}]
                                [--status {new,pending,waiting,running,successful,failed,
↪error,canceled}]
                                [--execution_environment ID]
                                [--failed BOOLEAN] [--started STARTED]
                                [--finished FINISHED]
                                [--canceled_on CANCELED_ON]
                                [--elapsed ELAPSED] [--job_explanation TEXT]
                                [--execution_node TEXT] [--work_unit_id TEXT]
                                [--local_path TEXT]
                                [--scm_type {,git,svn,insights,archive}]
                                [--scm_url TEXT] [--scm_branch TEXT]
                                [--scm_refspec TEXT] [--scm_clean BOOLEAN]
                                [--scm_track_submodules BOOLEAN]
                                [--scm_delete_on_update BOOLEAN]
                                [--credential ID] [--timeout INTEGER]
                                [--scm_revision TEXT] [--project ID]
```

(continues on next page)

(продолжение с предыдущей страницы)

```
                                [--job_type {run,check}] [--job_tags TEXT]

optional arguments:
  -h, --help            show this help message and exit
  --all                 fetch all pages of content from the API when returning
                        results (instead of just the first page)
  --order_by ORDER_BY   order results by given field name, prefix the field
                        name with a dash (-) to sort in reverse eg
                        --order_by='-name',multiple sorting fields may be
                        specified by separating the field names with a comma
                        (,)
  --type {project_update}
                        only list project_updates with the specified type
  --created CREATED     only list project_updates with the specified created
  --modified MODIFIED   only list project_updates with the specified modified
  --name TEXT           only list project_updates with the specified name
  --description TEXT    only list project_updates with the specified
                        description
  --unified_job_template ID
                        only list project_updates with the specified
                        unified_job_template
  --launch_type {manual,relaunch,callback,scheduled,dependency,workflow,webhook,sync,scm}
                        only list project_updates with the specified
                        launch_type
  --status {new,pending,waiting,running,successful,failed,error,canceled}
                        only list project_updates with the specified status
  --execution_environment ID
                        only list project_updates with the specified
                        execution_environment
  --failed BOOLEAN      only list project_updates with the specified failed
  --started STARTED     only list project_updates with the specified started
  --finished FINISHED   only list project_updates with the specified finished
  --canceled_on CANCELED_ON
                        only list project_updates with the specified
                        canceled_on
  --elapsed ELAPSED     only list project_updates with the specified elapsed
  --job_explanation TEXT
                        only list project_updates with the specified
                        job_explanation
  --execution_node TEXT
                        only list project_updates with the specified
                        execution_node
  --work_unit_id TEXT   only list project_updates with the specified
                        work_unit_id
  --local_path TEXT     only list project_updates with the specified
                        local_path
  --scm_type {,git,svn,insights,archive}
                        only list project_updates with the specified scm_type
  --scm_url TEXT        only list project_updates with the specified scm_url
  --scm_branch TEXT     only list project_updates with the specified
                        scm_branch
  --scm_refspec TEXT    only list project_updates with the specified
                        scm_refspec
  --scm_clean BOOLEAN   only list project_updates with the specified scm_clean
  --scm_track_submodules BOOLEAN
                        only list project_updates with the specified
                        scm_track_submodules
  --scm_delete_on_update BOOLEAN
                        only list project_updates with the specified
                        scm_delete_on_update
  --credential ID       only list project_updates with the specified
                        credential
  --timeout INTEGER     only list project_updates with the specified timeout
  --scm_revision TEXT   only list project_updates with the specified
```

(continues on next page)

(продолжение с предыдущей страницы)

```
                        scm_revision
  --project ID          only list project_updates with the specified project
  --job_type {run,check}
                        only list project_updates with the specified job_type
  --job_tags TEXT       only list project_updates with the specified job_tags

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made
```

project_updates get

```
usage: awx project_updates get [-h] [-f {json,yaml,jq,human}] [--filter TEXT]
                               [--conf.color BOOLEAN] [-v]
                               id

positional arguments:
  id                    the ID (or unique name) of the resource

optional arguments:
  -h, --help            show this help message and exit

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made

awx project_updates get: the following arguments are required: id
```

project_updates delete

```
usage: awx project_updates delete [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx project_updates delete: the following arguments are required: id
```

project_updates stdout

```
usage: awx project_updates stdout [-h] id

positional arguments:
  id

optional arguments:
  -h, --help  show this help message and exit

awx project_updates stdout: the following arguments are required: id
```

projects

```
usage: awx projects [-h] action ...

positional arguments:
  action
    list
    create
    get
    modify
    delete
    update
    associate
    disassociate

optional arguments:
  -h, --help    show this help message and exit

awx projects: the following arguments are required: action
```

projects list

```
usage: awx projects list [-h] [--all] [--order_by ORDER_BY]
                         [-f {json,yaml,jq,human}] [--filter TEXT]
                         [--conf.color BOOLEAN] [-v] [--type {project}]
                         [--created CREATED] [--modified MODIFIED]
                         [--name TEXT] [--description TEXT]
                         [--local_path TEXT]
                         [--scm_type {,git,svn,insights,archive}]
                         [--scm_url TEXT] [--scm_branch TEXT]
                         [--scm_refspec TEXT] [--scm_clean BOOLEAN]
                         [--scm_track_submodules BOOLEAN]
                         [--scm_delete_on_update BOOLEAN] [--credential ID]
                         [--timeout INTEGER] [--scm_revision TEXT]
                         [--last_job_run LAST_JOB_RUN]
                         [--last_job_failed BOOLEAN]
                         [--next_job_run NEXT_JOB_RUN]
                         [--status {new,pending,waiting,running,successful,failed,error,
↪canceled,never updated,ok,missing}]
                         [--organization ID] [--scm_update_on_launch BOOLEAN]
                         [--scm_update_cache_timeout INTEGER]
                         [--allow_override BOOLEAN] [--custom_virtualenv TEXT]
                         [--default_environment ID]
                         [--signature_validation_credential ID]

optional arguments:
  -h, --help            show this help message and exit
  --all                 fetch all pages of content from the API when returning
```

(continues on next page)

(продолжение с предыдущей страницы)

```
                        results (instead of just the first page)
  --order_by ORDER_BY   order results by given field name, prefix the field
                        name with a dash (-) to sort in reverse eg
                        --order_by='-name',multiple sorting fields may be
                        specified by separating the field names with a comma
                        (,)
  --type {project}      only list projects with the specified type
  --created CREATED     only list projects with the specified created
  --modified MODIFIED   only list projects with the specified modified
  --name TEXT           only list projects with the specified name
  --description TEXT    only list projects with the specified description
  --local_path TEXT     only list projects with the specified local_path
  --scm_type {,git,svn,insights,archive}
                        only list projects with the specified scm_type
  --scm_url TEXT        only list projects with the specified scm_url
  --scm_branch TEXT     only list projects with the specified scm_branch
  --scm_refspec TEXT    only list projects with the specified scm_refspec
  --scm_clean BOOLEAN   only list projects with the specified scm_clean
  --scm_track_submodules BOOLEAN
                        only list projects with the specified
                        scm_track_submodules
  --scm_delete_on_update BOOLEAN
                        only list projects with the specified
                        scm_delete_on_update
  --credential ID       only list projects with the specified credential
  --timeout INTEGER     only list projects with the specified timeout
  --scm_revision TEXT   only list projects with the specified scm_revision
  --last_job_run LAST_JOB_RUN
                        only list projects with the specified last_job_run
  --last_job_failed BOOLEAN
                        only list projects with the specified last_job_failed
  --next_job_run NEXT_JOB_RUN
                        only list projects with the specified next_job_run
  --status {new,pending,waiting,running,successful,failed,error,canceled,never updated,
↪ok,missing}
                        only list projects with the specified status
  --organization ID     only list projects with the specified organization
  --scm_update_on_launch BOOLEAN
                        only list projects with the specified
                        scm_update_on_launch
  --scm_update_cache_timeout INTEGER
                        only list projects with the specified
                        scm_update_cache_timeout
  --allow_override BOOLEAN
                        only list projects with the specified allow_override
  --custom_virtualenv TEXT
                        only list projects with the specified
                        custom_virtualenv
  --default_environment ID
                        only list projects with the specified
                        default_environment
  --signature_validation_credential ID
                        only list projects with the specified
                        signature_validation_credential

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made
```

projects create

```
usage: awx projects create [-h] [--monitor] [--wait] --name TEXT
                           [--description TEXT] [--local_path TEXT]
                           [--scm_type {,git,svn,insights,archive}]
                           [--scm_url TEXT] [--scm_branch TEXT]
                           [--scm_refspec TEXT] [--scm_clean BOOLEAN]
                           [--scm_track_submodules BOOLEAN]
                           [--scm_delete_on_update BOOLEAN] [--credential ID]
                           [--timeout INTEGER] [--organization ID]
                           [--scm_update_on_launch BOOLEAN]
                           [--scm_update_cache_timeout INTEGER]
                           [--allow_override BOOLEAN]
                           [--default_environment ID]
                           [--signature_validation_credential ID]

required arguments:
  --name TEXT           Name of this project.

optional arguments:
  -h, --help            show this help message and exit
  --monitor             If set, prints stdout of the project update until it
                        finishes.
  --wait                If set, waits until the new project has updated.
  --description TEXT    Optional description of this project.
  --local_path TEXT     Local path (relative to PROJECTS_ROOT) containing
                        playbooks and related files for this project.
  --scm_type {,git,svn,insights,archive}
                        Specifies the source control system used to store the
                        project.
  --scm_url TEXT        The location where the project is stored.
  --scm_branch TEXT     Specific branch, tag or commit to checkout.
  --scm_refspec TEXT    For git projects, an additional refspec to fetch.
  --scm_clean BOOLEAN   Discard any local changes before syncing the project.
  --scm_track_submodules BOOLEAN
                        Track submodules latest commits on defined branch.
  --scm_delete_on_update BOOLEAN
                        Delete the project before syncing.
  --credential ID       the ID of the associated credential
  --timeout INTEGER     The amount of time (in seconds) to run before the task
                        is canceled.
  --organization ID     The organization used to determine access to this
                        template.
  --scm_update_on_launch BOOLEAN
                        Update the project when a job is launched that uses
                        the project.
  --scm_update_cache_timeout INTEGER
                        The number of seconds after the last project update
                        ran that a new project update will be launched as a
                        job dependency.
  --allow_override BOOLEAN
                        Allow changing the SCM branch or revision in a job
                        template that uses this project.
  --default_environment ID
                        The default execution environment for jobs run using
                        this project.
  --signature_validation_credential ID
                        An optional credential used for validating files in
                        the project against unexpected changes.

awx projects create: the following arguments are required: --name
```

projects get

```
usage: awx projects get [-h] [-f {json,yaml,jq,human}] [--filter TEXT]
                        [--conf.color BOOLEAN] [-v]
                        id

positional arguments:
  id                    the ID (or unique name) of the resource

optional arguments:
  -h, --help            show this help message and exit

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made

awx projects get: the following arguments are required: id
```

projects modify

```
usage: awx projects modify [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx projects modify: the following arguments are required: id
```

projects delete

```
usage: awx projects delete [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx projects delete: the following arguments are required: id
```

projects update

```
usage: awx projects update [-h] [--monitor] [--action-timeout ACTION_TIMEOUT]
                           [--wait] [--interval INTERVAL]
                           id

positional arguments:
  id

optional arguments:
  -h, --help            show this help message and exit
  --monitor             If set, prints stdout of the launched job until it
```

(continues on next page)

(продолжение с предыдущей страницы)

```
                        finishes.
  --action-timeout ACTION_TIMEOUT
                        If set with --monitor or --wait, time out waiting on
                        job completion.
  --wait                If set, waits until the launched job finishes.
  --interval INTERVAL   If set with --monitor or --wait, amount of time to
                        wait in seconds between api calls. Minimum value is
                        2.5 seconds to avoid overwhelming the api

awx projects update: the following arguments are required: id
```

projects associate

```
usage: awx projects associate [-h]
                              (--start_notification  | --success_notification  | --
↪failure_notification )
                              id

positional arguments:
  id

optional arguments:
  -h, --help            show this help message and exit
  --start_notification
                        The ID (or name) of the notification_template to
                        associate
  --success_notification
                        The ID (or name) of the notification_template to
                        associate
  --failure_notification
                        The ID (or name) of the notification_template to
                        associate

awx projects associate: the following arguments are required: id
```

projects disassociate

```
usage: awx projects disassociate [-h]
                                 (--start_notification  | --success_notification  | --
↪failure_notification )
                                 id

positional arguments:
  id

optional arguments:
  -h, --help            show this help message and exit
  --start_notification
                        The ID (or name) of the notification_template to
                        disassociate
  --success_notification
                        The ID (or name) of the notification_template to
                        disassociate
  --failure_notification
                        The ID (or name) of the notification_template to
                        disassociate

awx projects disassociate: the following arguments are required: id
```

roles

```
usage: awx roles [-h] action ...

positional arguments:
  action
    list
    get

optional arguments:
  -h, --help  show this help message and exit

awx roles: the following arguments are required: action
```

roles list

```
usage: awx roles list [-h] [--all] [--order_by ORDER_BY]
                      [-f {json,yaml,jq,human}] [--filter TEXT]
                      [--conf.color BOOLEAN] [-v] [--type {role}]

optional arguments:
  -h, --help            show this help message and exit
  --all                 fetch all pages of content from the API when returning
                        results (instead of just the first page)
  --order_by ORDER_BY   order results by given field name, prefix the field
                        name with a dash (-) to sort in reverse eg
                        --order_by='-name',multiple sorting fields may be
                        specified by separating the field names with a comma
                        (,)
  --type {role}         only list roles with the specified type

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made
```

roles get

```
usage: awx roles get [-h] [-f {json,yaml,jq,human}] [--filter TEXT]
                     [--conf.color BOOLEAN] [-v]
                     id

positional arguments:
  id                    the ID (or unique name) of the resource

optional arguments:
  -h, --help            show this help message and exit

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made

awx roles get: the following arguments are required: id
```

schedules

```
usage: awx schedules [-h] action ...

positional arguments:
  action
    list
    create
    get
    modify
    delete

optional arguments:
  -h, --help  show this help message and exit

awx schedules: the following arguments are required: action
```

schedules list

```
usage: awx schedules list [-h] [--all] [--order_by ORDER_BY]
                          [-f {json,yaml,jq,human}] [--filter TEXT]
                          [--conf.color BOOLEAN] [-v] [--rrule TEXT]
                          [--type {schedule}] [--created CREATED]
                          [--modified MODIFIED] [--name TEXT]
                          [--description TEXT] [--extra_data JSON/YAML]
                          [--inventory ID] [--execution_environment ID]
                          [--unified_job_template ID] [--enabled BOOLEAN]
                          [--dtstart DTSTART] [--dtend DTEND]
                          [--next_run NEXT_RUN]

optional arguments:
  -h, --help            show this help message and exit
  --all                 fetch all pages of content from the API when returning
                        results (instead of just the first page)
  --order_by ORDER_BY   order results by given field name, prefix the field
                        name with a dash (-) to sort in reverse eg
                        --order_by='-name',multiple sorting fields may be
                        specified by separating the field names with a comma
                        (,)
  --rrule TEXT          only list schedules with the specified rrule
  --type {schedule}     only list schedules with the specified type
  --created CREATED     only list schedules with the specified created
  --modified MODIFIED   only list schedules with the specified modified
  --name TEXT           only list schedules with the specified name
  --description TEXT    only list schedules with the specified description
  --extra_data JSON/YAML
                        only list schedules with the specified extra_data
  --inventory ID        only list schedules with the specified inventory
  --execution_environment ID
                        only list schedules with the specified
                        execution_environment
  --unified_job_template ID
                        only list schedules with the specified
                        unified_job_template
  --enabled BOOLEAN     only list schedules with the specified enabled
  --dtstart DTSTART     only list schedules with the specified dtstart
  --dtend DTEND         only list schedules with the specified dtend
  --next_run NEXT_RUN   only list schedules with the specified next_run

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
```

(continues on next page)

(продолжение с предыдущей страницы)

```
                          format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made
```

schedules create

```
usage: awx schedules create [-h] --rrule TEXT --name TEXT [--description TEXT]
                            [--extra_data JSON/YAML] [--inventory ID]
                            [--scm_branch TEXT] [--job_type {None,,run,check}]
                            [--job_tags TEXT] [--skip_tags TEXT]
                            [--limit TEXT] [--diff_mode BOOLEAN]
                            [--verbosity {None,0,1,2,3,4,5}]
                            [--execution_environment ID] [--forks INTEGER]
                            [--job_slice_count INTEGER] [--timeout INTEGER]
                            --unified_job_template ID [--enabled BOOLEAN]

required arguments:
  --rrule TEXT          A value representing the schedules iCal recurrence
                        rule.
  --name TEXT           Name of this schedule.
  --unified_job_template ID
                        the ID of the associated unified_job_template

optional arguments:
  -h, --help            show this help message and exit
  --description TEXT    Optional description of this schedule.
  --extra_data JSON/YAML
                        a JSON or YAML string. You can optionally specify a
                        file path e.g., @path/to/file.yml
  --inventory ID        Inventory applied as a prompt, assuming job template
                        prompts for inventory
  --scm_branch TEXT
  --job_type {None,,run,check}
  --job_tags TEXT
  --skip_tags TEXT
  --limit TEXT
  --diff_mode BOOLEAN
  --verbosity {None,0,1,2,3,4,5}
  --execution_environment ID
                        The container image to be used for execution.
  --forks INTEGER
  --job_slice_count INTEGER
  --timeout INTEGER
  --enabled BOOLEAN     Enables processing of this schedule.

awx schedules create: the following arguments are required: --rrule, --name, --unified_
↪job_template
```

schedules get

```
usage: awx schedules get [-h] [-f {json,yaml,jq,human}] [--filter TEXT]
                         [--conf.color BOOLEAN] [-v]
                         id

positional arguments:
  id                    the ID (or unique name) of the resource

optional arguments:
  -h, --help            show this help message and exit
```

(continues on next page)

(продолжение с предыдущей страницы)

```
input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made

awx schedules get: the following arguments are required: id
```

schedules modify

```
usage: awx schedules modify [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx schedules modify: the following arguments are required: id
```

schedules delete

```
usage: awx schedules delete [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx schedules delete: the following arguments are required: id
```

settings

```
usage: awx settings [-h] action ...

positional arguments:
  action
    list
    modify

optional arguments:
  -h, --help  show this help message and exit

awx settings: the following arguments are required: action
```

settings list

```
usage: awx settings list [-h] [--slug SLUG]

optional arguments:
  -h, --help   show this help message and exit
  --slug SLUG  optional setting category/slug
```

settings modify

```
usage: awx settings modify [-h] key value

positional arguments:
  key
  value

optional arguments:
  -h, --help  show this help message and exit

awx settings modify: the following arguments are required: key, value
```

system_job_templates

```
usage: awx system_job_templates [-h] action ...

positional arguments:
  action
    list
    get

optional arguments:
  -h, --help  show this help message and exit

awx system_job_templates: the following arguments are required: action
```

system_job_templates list

```
usage: awx system_job_templates list [-h] [--all] [--order_by ORDER_BY]
                                     [-f {json,yaml,jq,human}] [--filter TEXT]
                                     [--conf.color BOOLEAN] [-v]
                                     [--type {system_job_template}]
                                     [--created CREATED] [--modified MODIFIED]
                                     [--name TEXT] [--description TEXT]
                                     [--last_job_run LAST_JOB_RUN]
                                     [--last_job_failed BOOLEAN]
                                     [--next_job_run NEXT_JOB_RUN]
                                     [--status {new,pending,waiting,running,successful,
↪failed,error,canceled,never updated,ok,missing,none,updating}]
                                     [--execution_environment ID]
                                     [--job_type {,cleanup_jobs,cleanup_activitystream,
↪cleanup_sessions,cleanup_tokens}]

optional arguments:
  -h, --help            show this help message and exit
  --all                 fetch all pages of content from the API when returning
                        results (instead of just the first page)
  --order_by ORDER_BY   order results by given field name, prefix the field
                        name with a dash (-) to sort in reverse eg
```

(continues on next page)

(продолжение с предыдущей страницы)

```
                        --order_by='-name',multiple sorting fields may be
                        specified by separating the field names with a comma
                        (,)
  --type {system_job_template}
                        only list system_job_templates with the specified type
  --created CREATED     only list system_job_templates with the specified
                        created
  --modified MODIFIED   only list system_job_templates with the specified
                        modified
  --name TEXT           only list system_job_templates with the specified name
  --description TEXT    only list system_job_templates with the specified
                        description
  --last_job_run LAST_JOB_RUN
                        only list system_job_templates with the specified
                        last_job_run
  --last_job_failed BOOLEAN
                        only list system_job_templates with the specified
                        last_job_failed
  --next_job_run NEXT_JOB_RUN
                        only list system_job_templates with the specified
                        next_job_run
  --status {new,pending,waiting,running,successful,failed,error,canceled,never updated,
↪ok,missing,none,updating}
                        only list system_job_templates with the specified
                        status
  --execution_environment ID
                        only list system_job_templates with the specified
                        execution_environment
  --job_type {,cleanup_jobs,cleanup_activitystream,cleanup_sessions,cleanup_tokens}
                        only list system_job_templates with the specified
                        job_type

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made
```

system_job_templates get

```
usage: awx system_job_templates get [-h] [-f {json,yaml,jq,human}]
                                    [--filter TEXT] [--conf.color BOOLEAN]
                                    [-v]
                                    id

positional arguments:
  id                    the ID (or unique name) of the resource

optional arguments:
  -h, --help            show this help message and exit

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made

awx system_job_templates get: the following arguments are required: id
```

system_jobs

```
usage: awx system_jobs [-h] action ...

positional arguments:
  action
    list
    get
    delete

optional arguments:
  -h, --help  show this help message and exit

awx system_jobs: the following arguments are required: action
```

system_jobs list

```
usage: awx system_jobs list [-h] [--all] [--order_by ORDER_BY]
                            [-f {json,yaml,jq,human}] [--filter TEXT]
                            [--conf.color BOOLEAN] [-v] [--type {system_job}]
                            [--created CREATED] [--modified MODIFIED]
                            [--name TEXT] [--description TEXT]
                            [--unified_job_template ID]
                            [--launch_type {manual,relaunch,callback,scheduled,
↪dependency,workflow,webhook,sync,scm}]
                            [--status {new,pending,waiting,running,successful,failed,
↪error,canceled}]
                            [--execution_environment ID] [--failed BOOLEAN]
                            [--started STARTED] [--finished FINISHED]
                            [--canceled_on CANCELED_ON] [--elapsed ELAPSED]
                            [--job_explanation TEXT] [--execution_node TEXT]
                            [--work_unit_id TEXT] [--system_job_template ID]
                            [--job_type {,cleanup_jobs,cleanup_activitystream,cleanup_
↪sessions,cleanup_tokens}]
                            [--extra_vars TEXT]

optional arguments:
  -h, --help            show this help message and exit
  --all                 fetch all pages of content from the API when returning
                        results (instead of just the first page)
  --order_by ORDER_BY   order results by given field name, prefix the field
                        name with a dash (-) to sort in reverse eg
                        --order_by='-name',multiple sorting fields may be
                        specified by separating the field names with a comma
                        (,)
  --type {system_job}   only list system_jobs with the specified type
  --created CREATED     only list system_jobs with the specified created
  --modified MODIFIED   only list system_jobs with the specified modified
  --name TEXT           only list system_jobs with the specified name
  --description TEXT    only list system_jobs with the specified description
  --unified_job_template ID
                        only list system_jobs with the specified
                        unified_job_template
  --launch_type {manual,relaunch,callback,scheduled,dependency,workflow,webhook,sync,scm}
                        only list system_jobs with the specified launch_type
  --status {new,pending,waiting,running,successful,failed,error,canceled}
                        only list system_jobs with the specified status
  --execution_environment ID
                        only list system_jobs with the specified
                        execution_environment
  --failed BOOLEAN      only list system_jobs with the specified failed
  --started STARTED     only list system_jobs with the specified started
  --finished FINISHED   only list system_jobs with the specified finished
```

(continues on next page)

(продолжение с предыдущей страницы)

```
  --canceled_on CANCELED_ON
                        only list system_jobs with the specified canceled_on
  --elapsed ELAPSED     only list system_jobs with the specified elapsed
  --job_explanation TEXT
                        only list system_jobs with the specified
                        job_explanation
  --execution_node TEXT
                        only list system_jobs with the specified
                        execution_node
  --work_unit_id TEXT   only list system_jobs with the specified work_unit_id
  --system_job_template ID
                        only list system_jobs with the specified
                        system_job_template
  --job_type {,cleanup_jobs,cleanup_activitystream,cleanup_sessions,cleanup_tokens}
                        only list system_jobs with the specified job_type
  --extra_vars TEXT     only list system_jobs with the specified extra_vars

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made
```

system_jobs get

```
usage: awx system_jobs get [-h] [-f {json,yaml,jq,human}] [--filter TEXT]
                           [--conf.color BOOLEAN] [-v]
                           id

positional arguments:
  id                    the ID (or unique name) of the resource

optional arguments:
  -h, --help            show this help message and exit

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made

awx system_jobs get: the following arguments are required: id
```

system_jobs delete

```
usage: awx system_jobs delete [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx system_jobs delete: the following arguments are required: id
```

teams

```
usage: awx teams [-h] action ...

positional arguments:
  action
    list
    create
    get
    modify
    delete
    grant
    revoke

optional arguments:
  -h, --help  show this help message and exit

awx teams: the following arguments are required: action
```

teams list

```
usage: awx teams list [-h] [--all] [--order_by ORDER_BY]
                      [-f {json,yaml,jq,human}] [--filter TEXT]
                      [--conf.color BOOLEAN] [-v] [--type {team}]
                      [--created CREATED] [--modified MODIFIED] [--name TEXT]
                      [--description TEXT] [--organization ID]

optional arguments:
  -h, --help            show this help message and exit
  --all                 fetch all pages of content from the API when returning
                        results (instead of just the first page)
  --order_by ORDER_BY   order results by given field name, prefix the field
                        name with a dash (-) to sort in reverse eg
                        --order_by='-name',multiple sorting fields may be
                        specified by separating the field names with a comma
                        (,)
  --type {team}         only list teams with the specified type
  --created CREATED     only list teams with the specified created
  --modified MODIFIED   only list teams with the specified modified
  --name TEXT           only list teams with the specified name
  --description TEXT    only list teams with the specified description
  --organization ID     only list teams with the specified organization

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made
```

teams create

```
usage: awx teams create [-h] --name TEXT [--description TEXT] --organization
                        ID

required arguments:
  --name TEXT         Name of this team.
  --organization ID   the ID of the associated organization

optional arguments:
  -h, --help          show this help message and exit
  --description TEXT  Optional description of this team.

awx teams create: the following arguments are required: --name, --organization
```

teams get

```
usage: awx teams get [-h] [-f {json,yaml,jq,human}] [--filter TEXT]
                     [--conf.color BOOLEAN] [-v]
                     id

positional arguments:
  id                    the ID (or unique name) of the resource

optional arguments:
  -h, --help            show this help message and exit

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made

awx teams get: the following arguments are required: id
```

teams modify

```
usage: awx teams modify [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx teams modify: the following arguments are required: id
```

teams delete

```
usage: awx teams delete [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx teams delete: the following arguments are required: id
```

teams grant

```
usage: awx teams grant [-h]
                       (--organization ID | --project ID | --inventory ID | --team ID | -
↪-credential ID | --job_template ID | --workflow_job_template ID | --instance_group ID)
                       --role
                       {read,notification_admin,workflow_admin,job_template_admin,adhoc,
↪auditor,execute,update,approval,inventory_admin,execution_environment_admin,credential_
↪admin,admin,project_admin,member,use}
                       id

positional arguments:
  id                    The ID (or name) of the teams to grant access to/from

optional arguments:
  -h, --help            show this help message and exit
  --organization ID     The ID (or name) of the target organization
  --project ID          The ID (or name) of the target project
  --inventory ID        The ID (or name) of the target inventory
  --team ID             The ID (or name) of the target team
  --credential ID       The ID (or name) of the target credential
  --job_template ID     The ID (or name) of the target job_template
  --workflow_job_template ID
                        The ID (or name) of the target workflow_job_template
  --instance_group ID   The ID (or name) of the target instance_group
  --role {read,notification_admin,workflow_admin,job_template_admin,adhoc,auditor,
↪execute,update,approval,inventory_admin,execution_environment_admin,credential_admin,
↪admin,project_admin,member,use}
                        The name of the role to grant

awx teams grant: the following arguments are required: id, --role
```

teams revoke

```
usage: awx teams revoke [-h]
                        (--organization ID | --project ID | --inventory ID | --team ID |␣
↪--credential ID | --job_template ID | --workflow_job_template ID | --instance_group ID)
                        --role
                        {workflow_admin,approval,credential_admin,notification_admin,
↪update,inventory_admin,execute,job_template_admin,read,use,admin,auditor,adhoc,
↪execution_environment_admin,member,project_admin}
                        id

positional arguments:
  id                    The ID (or name) of the teams to revoke access to/from

optional arguments:
  -h, --help            show this help message and exit
```

(continues on next page)

(продолжение с предыдущей страницы)

```
  --organization ID     The ID (or name) of the target organization
  --project ID          The ID (or name) of the target project
  --inventory ID        The ID (or name) of the target inventory
  --team ID             The ID (or name) of the target team
  --credential ID       The ID (or name) of the target credential
  --job_template ID     The ID (or name) of the target job_template
  --workflow_job_template ID
                        The ID (or name) of the target workflow_job_template
  --instance_group ID   The ID (or name) of the target instance_group
  --role {workflow_admin,approval,credential_admin,notification_admin,update,inventory_
↪admin,execute,job_template_admin,read,use,admin,auditor,adhoc,execution_environment_
↪admin,member,project_admin}
                        The name of the role to revoke

awx teams revoke: the following arguments are required: id, --role
```

tokens

```
usage: awx tokens [-h] action ...

positional arguments:
  action
    list
    create
    get
    modify
    delete

optional arguments:
  -h, --help  show this help message and exit

awx tokens: the following arguments are required: action
```

tokens list

```
usage: awx tokens list [-h] [--all] [--order_by ORDER_BY]
                       [-f {json,yaml,jq,human}] [--filter TEXT]
                       [--conf.color BOOLEAN] [-v]
                       [--type {o_auth2_access_token}] [--created CREATED]
                       [--modified MODIFIED] [--description TEXT] [--user ID]
                       [--token TEXT] [--application ID] [--expires EXPIRES]
                       [--scope TEXT]

optional arguments:
  -h, --help            show this help message and exit
  --all                 fetch all pages of content from the API when returning
                        results (instead of just the first page)
  --order_by ORDER_BY   order results by given field name, prefix the field
                        name with a dash (-) to sort in reverse eg
                        --order_by='-name',multiple sorting fields may be
                        specified by separating the field names with a comma
                        (,)
  --type {o_auth2_access_token}
                        only list tokens with the specified type
  --created CREATED     only list tokens with the specified created
  --modified MODIFIED   only list tokens with the specified modified
  --description TEXT    only list tokens with the specified description
  --user ID             only list tokens with the specified user
  --token TEXT          only list tokens with the specified token
  --application ID      only list tokens with the specified application
```

(continues on next page)

(продолжение с предыдущей страницы)

```
  --expires EXPIRES     only list tokens with the specified expires
  --scope TEXT          only list tokens with the specified scope

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made
```

tokens create

```
usage: awx tokens create [-h] [--description TEXT] [--application ID]
                         [--scope TEXT]

optional arguments:
  -h, --help          show this help message and exit
  --description TEXT  Optional description of this access token.
  --application ID    the ID of the associated application
  --scope TEXT        Allowed scopes, further restricts user's permissions.
                      Must be a simple space-separated string with allowed
                      scopes ['read', 'write'].
```

tokens get

```
usage: awx tokens get [-h] [-f {json,yaml,jq,human}] [--filter TEXT]
                      [--conf.color BOOLEAN] [-v]
                      id

positional arguments:
  id                    the ID (or unique name) of the resource

optional arguments:
  -h, --help            show this help message and exit

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made

awx tokens get: the following arguments are required: id
```

tokens modify

```
usage: awx tokens modify [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx tokens modify: the following arguments are required: id
```

tokens delete

```
usage: awx tokens delete [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx tokens delete: the following arguments are required: id
```

unified_job_templates

```
usage: awx unified_job_templates [-h] action ...

positional arguments:
  action
    list
    get

optional arguments:
  -h, --help  show this help message and exit

awx unified_job_templates: the following arguments are required: action
```

unified_job_templates list

```
usage: awx unified_job_templates list [-h] [--all] [--order_by ORDER_BY]
                                      [-f {json,yaml,jq,human}]
                                      [--filter TEXT] [--conf.color BOOLEAN]
                                      [-v]
                                      [--type {project,inventory_source,job_template,
↪system_job_template,workflow_job_template}]
                                      [--created CREATED]
                                      [--modified MODIFIED] [--name TEXT]
                                      [--description TEXT]
                                      [--last_job_run LAST_JOB_RUN]
                                      [--last_job_failed BOOLEAN]
                                      [--next_job_run NEXT_JOB_RUN]
                                      [--status {new,pending,waiting,running,successful,
↪failed,error,canceled,never updated,ok,missing,none,updating}]
                                      [--execution_environment ID]

optional arguments:
  -h, --help            show this help message and exit
  --all                 fetch all pages of content from the API when returning
                        results (instead of just the first page)
  --order_by ORDER_BY   order results by given field name, prefix the field
                        name with a dash (-) to sort in reverse eg
                        --order_by='-name',multiple sorting fields may be
                        specified by separating the field names with a comma
                        (,)
  --type {project,inventory_source,job_template,system_job_template,workflow_job_
↪template}
                        only list unified_job_templates with the specified
                        type
  --created CREATED     only list unified_job_templates with the specified
                        created
  --modified MODIFIED   only list unified_job_templates with the specified
```

(continues on next page)

(продолжение с предыдущей страницы)

```
                        modified
  --name TEXT           only list unified_job_templates with the specified
                        name
  --description TEXT    only list unified_job_templates with the specified
                        description
  --last_job_run LAST_JOB_RUN
                        only list unified_job_templates with the specified
                        last_job_run
  --last_job_failed BOOLEAN
                        only list unified_job_templates with the specified
                        last_job_failed
  --next_job_run NEXT_JOB_RUN
                        only list unified_job_templates with the specified
                        next_job_run
  --status {new,pending,waiting,running,successful,failed,error,canceled,never updated,
↪ok,missing,none,updating}
                        only list unified_job_templates with the specified
                        status
  --execution_environment ID
                        only list unified_job_templates with the specified
                        execution_environment

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made
```

unified_job_templates get

```
usage: awx unified_job_templates get [-h] [-f {json,yaml,jq,human}]
                                     [--filter TEXT] [--conf.color BOOLEAN]
                                     [-v]
                                     id

positional arguments:
  id                    the ID (or unique name) of the resource

optional arguments:
  -h, --help            show this help message and exit

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made

awx unified_job_templates get: the following arguments are required: id
```

unified_jobs

```
usage: awx unified_jobs [-h] action ...

positional arguments:
  action
    list
    get

optional arguments:
  -h, --help  show this help message and exit

awx unified_jobs: the following arguments are required: action
```

unified_jobs list

```
usage: awx unified_jobs list [-h] [--all] [--order_by ORDER_BY]
                             [-f {json,yaml,jq,human}] [--filter TEXT]
                             [--conf.color BOOLEAN] [-v]
                             [--type {project_update,inventory_update,job,ad_hoc_command,
↪system_job,workflow_job}]
                             [--created CREATED] [--modified MODIFIED]
                             [--name TEXT] [--description TEXT]
                             [--unified_job_template ID]
                             [--launch_type {manual,relaunch,callback,scheduled,
↪dependency,workflow,webhook,sync,scm}]
                             [--status {new,pending,waiting,running,successful,failed,
↪error,canceled}]
                             [--execution_environment ID] [--failed BOOLEAN]
                             [--started STARTED] [--finished FINISHED]
                             [--canceled_on CANCELED_ON] [--elapsed ELAPSED]
                             [--job_explanation TEXT] [--execution_node TEXT]
                             [--controller_node TEXT] [--work_unit_id TEXT]

optional arguments:
  -h, --help            show this help message and exit
  --all                 fetch all pages of content from the API when returning
                        results (instead of just the first page)
  --order_by ORDER_BY   order results by given field name, prefix the field
                        name with a dash (-) to sort in reverse eg
                        --order_by='-name',multiple sorting fields may be
                        specified by separating the field names with a comma
                        (,)
  --type {project_update,inventory_update,job,ad_hoc_command,system_job,workflow_job}
                        only list unified_jobs with the specified type
  --created CREATED     only list unified_jobs with the specified created
  --modified MODIFIED   only list unified_jobs with the specified modified
  --name TEXT           only list unified_jobs with the specified name
  --description TEXT    only list unified_jobs with the specified description
  --unified_job_template ID
                        only list unified_jobs with the specified
                        unified_job_template
  --launch_type {manual,relaunch,callback,scheduled,dependency,workflow,webhook,sync,scm}
                        only list unified_jobs with the specified launch_type
  --status {new,pending,waiting,running,successful,failed,error,canceled}
                        only list unified_jobs with the specified status
  --execution_environment ID
                        only list unified_jobs with the specified
                        execution_environment
  --failed BOOLEAN      only list unified_jobs with the specified failed
  --started STARTED     only list unified_jobs with the specified started
  --finished FINISHED   only list unified_jobs with the specified finished
  --canceled_on CANCELED_ON
```

(continues on next page)

(продолжение с предыдущей страницы)

```
                        only list unified_jobs with the specified canceled_on
  --elapsed ELAPSED     only list unified_jobs with the specified elapsed
  --job_explanation TEXT
                        only list unified_jobs with the specified
                        job_explanation
  --execution_node TEXT
                        only list unified_jobs with the specified
                        execution_node
  --controller_node TEXT
                        only list unified_jobs with the specified
                        controller_node
  --work_unit_id TEXT   only list unified_jobs with the specified work_unit_id

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made
```

unified_jobs get

```
usage: awx unified_jobs get [-h] [-f {json,yaml,jq,human}] [--filter TEXT]
                            [--conf.color BOOLEAN] [-v]
                            id

positional arguments:
  id                    the ID (or unique name) of the resource

optional arguments:
  -h, --help            show this help message and exit

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made

awx unified_jobs get: the following arguments are required: id
```

users

```
usage: awx users [-h] action ...

positional arguments:
  action
    list
    create
    get
    modify
    delete
    grant
    revoke

optional arguments:
  -h, --help  show this help message and exit
```

(continues on next page)

(продолжение с предыдущей страницы)

```
awx users: the following arguments are required: action
```

users list

```
usage: awx users list [-h] [--all] [--order_by ORDER_BY]
                      [-f {json,yaml,jq,human}] [--filter TEXT]
                      [--conf.color BOOLEAN] [-v] [--type {user}]
                      [--username TEXT] [--first_name TEXT] [--last_name TEXT]
                      [--email TEXT] [--is_superuser BOOLEAN]
                      [--password TEXT] [--last_login LAST_LOGIN]

optional arguments:
  -h, --help            show this help message and exit
  --all                 fetch all pages of content from the API when returning
                        results (instead of just the first page)
  --order_by ORDER_BY   order results by given field name, prefix the field
                        name with a dash (-) to sort in reverse eg
                        --order_by='-name',multiple sorting fields may be
                        specified by separating the field names with a comma
                        (,)
  --type {user}         only list users with the specified type
  --username TEXT       only list users with the specified username
  --first_name TEXT     only list users with the specified first_name
  --last_name TEXT      only list users with the specified last_name
  --email TEXT          only list users with the specified email
  --is_superuser BOOLEAN
                        only list users with the specified is_superuser
  --password TEXT       only list users with the specified password
  --last_login LAST_LOGIN
                        only list users with the specified last_login

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made
```

users create

```
usage: awx users create [-h] --username TEXT [--first_name TEXT]
                        [--last_name TEXT] [--email TEXT]
                        [--is_superuser BOOLEAN] [--is_system_auditor BOOLEAN]
                        --password TEXT

required arguments:
  --username TEXT       Required. 150 characters or fewer. Letters, digits and
                        @/./+/-/_ only.
  --password TEXT       Field used to change the password.

optional arguments:
  -h, --help            show this help message and exit
  --first_name TEXT
  --last_name TEXT
  --email TEXT
  --is_superuser BOOLEAN
                        Designates that this user has all permissions without
```

(continues on next page)

(продолжение с предыдущей страницы)

```
                        explicitly assigning them.
  --is_system_auditor BOOLEAN

awx users create: the following arguments are required: --username, --password
```

users get

```
usage: awx users get [-h] [-f {json,yaml,jq,human}] [--filter TEXT]
                     [--conf.color BOOLEAN] [-v]
                     id

positional arguments:
  id                    the ID (or unique name) of the resource

optional arguments:
  -h, --help            show this help message and exit

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made

awx users get: the following arguments are required: id
```

users modify

```
usage: awx users modify [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx users modify: the following arguments are required: id
```

users delete

```
usage: awx users delete [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx users delete: the following arguments are required: id
```

users grant

```
usage: awx users grant [-h]
                       (--organization ID | --project ID | --inventory ID | --team ID | -
↪-credential ID | --job_template ID | --workflow_job_template ID | --instance_group ID)
                       --role
                       {inventory_admin,job_template_admin,execution_environment_admin,
↪auditor,approval,update,use,project_admin,workflow_admin,member,read,credential_admin,
↪adhoc,admin,notification_admin,execute}
                       id

positional arguments:
  id                    The ID (or name) of the users to grant access to/from

optional arguments:
  -h, --help            show this help message and exit
  --organization ID     The ID (or name) of the target organization
  --project ID          The ID (or name) of the target project
  --inventory ID        The ID (or name) of the target inventory
  --team ID             The ID (or name) of the target team
  --credential ID       The ID (or name) of the target credential
  --job_template ID     The ID (or name) of the target job_template
  --workflow_job_template ID
                        The ID (or name) of the target workflow_job_template
  --instance_group ID   The ID (or name) of the target instance_group
  --role {inventory_admin,job_template_admin,execution_environment_admin,auditor,
↪approval,update,use,project_admin,workflow_admin,member,read,credential_admin,adhoc,
↪admin,notification_admin,execute}
                        The name of the role to grant

awx users grant: the following arguments are required: id, --role
```

users revoke

```
usage: awx users revoke [-h]
                        (--organization ID | --project ID | --inventory ID | --team ID |␣
↪--credential ID | --job_template ID | --workflow_job_template ID | --instance_group ID)
                        --role
                        {credential_admin,job_template_admin,execute,update,workflow_
↪admin,adhoc,inventory_admin,approval,admin,execution_environment_admin,member,read,
↪project_admin,auditor,notification_admin,use}
                        id

positional arguments:
  id                    The ID (or name) of the users to revoke access to/from

optional arguments:
  -h, --help            show this help message and exit
  --organization ID     The ID (or name) of the target organization
  --project ID          The ID (or name) of the target project
  --inventory ID        The ID (or name) of the target inventory
  --team ID             The ID (or name) of the target team
  --credential ID       The ID (or name) of the target credential
  --job_template ID     The ID (or name) of the target job_template
  --workflow_job_template ID
                        The ID (or name) of the target workflow_job_template
  --instance_group ID   The ID (or name) of the target instance_group
  --role {credential_admin,job_template_admin,execute,update,workflow_admin,adhoc,
↪inventory_admin,approval,admin,execution_environment_admin,member,read,project_admin,
↪auditor,notification_admin,use}
                        The name of the role to revoke

awx users revoke: the following arguments are required: id, --role
```

workflow_approvals

```
usage: awx workflow_approvals [-h] action ...

positional arguments:
  action
    list
    get
    delete

optional arguments:
  -h, --help  show this help message and exit

awx workflow_approvals: the following arguments are required: action
```

workflow_approvals list

```
usage: awx workflow_approvals list [-h] [--all] [--order_by ORDER_BY]
                                   [-f {json,yaml,jq,human}] [--filter TEXT]
                                   [--conf.color BOOLEAN] [-v]
                                   [--type {workflow_approval}]
                                   [--created CREATED] [--modified MODIFIED]
                                   [--name TEXT] [--description TEXT]
                                   [--unified_job_template ID]
                                   [--launch_type {manual,relaunch,callback,scheduled,
↪dependency,workflow,webhook,sync,scm}]
                                   [--status {new,pending,waiting,running,successful,
↪failed,error,canceled}]
                                   [--execution_environment ID]
                                   [--failed BOOLEAN] [--started STARTED]
                                   [--finished FINISHED]
                                   [--canceled_on CANCELED_ON]
                                   [--elapsed ELAPSED]
                                   [--job_explanation TEXT]
                                   [--work_unit_id TEXT] [--timed_out BOOLEAN]

optional arguments:
  -h, --help            show this help message and exit
  --all                 fetch all pages of content from the API when returning
                        results (instead of just the first page)
  --order_by ORDER_BY   order results by given field name, prefix the field
                        name with a dash (-) to sort in reverse eg
                        --order_by='-name',multiple sorting fields may be
                        specified by separating the field names with a comma
                        (,)
  --type {workflow_approval}
                        only list workflow_approvals with the specified type
  --created CREATED     only list workflow_approvals with the specified
                        created
  --modified MODIFIED   only list workflow_approvals with the specified
                        modified
  --name TEXT           only list workflow_approvals with the specified name
  --description TEXT    only list workflow_approvals with the specified
                        description
  --unified_job_template ID
                        only list workflow_approvals with the specified
                        unified_job_template
  --launch_type {manual,relaunch,callback,scheduled,dependency,workflow,webhook,sync,scm}
                        only list workflow_approvals with the specified
                        launch_type
  --status {new,pending,waiting,running,successful,failed,error,canceled}
                        only list workflow_approvals with the specified status
  --execution_environment ID
```

(continues on next page)

(продолжение с предыдущей страницы)

```
                        only list workflow_approvals with the specified
                        execution_environment
  --failed BOOLEAN      only list workflow_approvals with the specified failed
  --started STARTED     only list workflow_approvals with the specified
                        started
  --finished FINISHED   only list workflow_approvals with the specified
                        finished
  --canceled_on CANCELED_ON
                        only list workflow_approvals with the specified
                        canceled_on
  --elapsed ELAPSED     only list workflow_approvals with the specified
                        elapsed
  --job_explanation TEXT
                        only list workflow_approvals with the specified
                        job_explanation
  --work_unit_id TEXT   only list workflow_approvals with the specified
                        work_unit_id
  --timed_out BOOLEAN   only list workflow_approvals with the specified
                        timed_out

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made
```

workflow_approvals get

```
usage: awx workflow_approvals get [-h] [-f {json,yaml,jq,human}]
                                  [--filter TEXT] [--conf.color BOOLEAN] [-v]
                                  id

positional arguments:
  id                    the ID (or unique name) of the resource

optional arguments:
  -h, --help            show this help message and exit

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made

awx workflow_approvals get: the following arguments are required: id
```

workflow_approvals delete

```
usage: awx workflow_approvals delete [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx workflow_approvals delete: the following arguments are required: id
```

workflow_job_nodes

```
usage: awx workflow_job_nodes [-h] action ...

positional arguments:
  action
    list
    get

optional arguments:
  -h, --help  show this help message and exit

awx workflow_job_nodes: the following arguments are required: action
```

workflow_job_nodes list

```
usage: awx workflow_job_nodes list [-h] [--all] [--order_by ORDER_BY]
                                   [-f {json,yaml,jq,human}] [--filter TEXT]
                                   [--conf.color BOOLEAN] [-v]
                                   [--type {workflow_job_node}]
                                   [--created CREATED] [--modified MODIFIED]
                                   [--extra_data JSON/YAML] [--inventory ID]
                                   [--execution_environment ID] [--job ID]
                                   [--workflow_job ID]
                                   [--unified_job_template ID]
                                   [--all_parents_must_converge BOOLEAN]
                                   [--do_not_run BOOLEAN] [--identifier TEXT]

optional arguments:
  -h, --help            show this help message and exit
  --all                 fetch all pages of content from the API when returning
                        results (instead of just the first page)
  --order_by ORDER_BY   order results by given field name, prefix the field
                        name with a dash (-) to sort in reverse eg
                        --order_by='-name',multiple sorting fields may be
                        specified by separating the field names with a comma
                        (,)
  --type {workflow_job_node}
                        only list workflow_job_nodes with the specified type
  --created CREATED     only list workflow_job_nodes with the specified
                        created
  --modified MODIFIED   only list workflow_job_nodes with the specified
                        modified
  --extra_data JSON/YAML
                        only list workflow_job_nodes with the specified
                        extra_data
  --inventory ID        only list workflow_job_nodes with the specified
                        inventory
```

(continues on next page)

(продолжение с предыдущей страницы)

```
  --execution_environment ID
                        only list workflow_job_nodes with the specified
                        execution_environment
  --job ID              only list workflow_job_nodes with the specified job
  --workflow_job ID     only list workflow_job_nodes with the specified
                        workflow_job
  --unified_job_template ID
                        only list workflow_job_nodes with the specified
                        unified_job_template
  --all_parents_must_converge BOOLEAN
                        only list workflow_job_nodes with the specified
                        all_parents_must_converge
  --do_not_run BOOLEAN  only list workflow_job_nodes with the specified
                        do_not_run
  --identifier TEXT     only list workflow_job_nodes with the specified
                        identifier

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made
```

workflow_job_nodes get

```
usage: awx workflow_job_nodes get [-h] [-f {json,yaml,jq,human}]
                                  [--filter TEXT] [--conf.color BOOLEAN] [-v]
                                  id

positional arguments:
  id                    the ID (or unique name) of the resource

optional arguments:
  -h, --help            show this help message and exit

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made

awx workflow_job_nodes get: the following arguments are required: id
```

workflow_job_template_nodes

```
usage: awx workflow_job_template_nodes [-h] action ...

positional arguments:
  action
    list
    create
    get
    modify
    delete

optional arguments:
```

(continues on next page)

(продолжение с предыдущей страницы)

```
  -h, --help  show this help message and exit

awx workflow_job_template_nodes: the following arguments are required: action
```

workflow_job_template_nodes list

```
usage: awx workflow_job_template_nodes list [-h] [--all] [--order_by ORDER_BY]
                                             [-f {json,yaml,jq,human}]
                                             [--filter TEXT]
                                             [--conf.color BOOLEAN] [-v]
                                             [--type {workflow_job_template_node}]
                                             [--created CREATED]
                                             [--modified MODIFIED]
                                             [--extra_data JSON/YAML]
                                             [--inventory ID]
                                             [--execution_environment ID]
                                             [--workflow_job_template ID]
                                             [--unified_job_template ID]
                                             [--all_parents_must_converge BOOLEAN]
                                             [--identifier TEXT]

optional arguments:
  -h, --help            show this help message and exit
  --all                 fetch all pages of content from the API when returning
                        results (instead of just the first page)
  --order_by ORDER_BY   order results by given field name, prefix the field
                        name with a dash (-) to sort in reverse eg
                        --order_by='-name',multiple sorting fields may be
                        specified by separating the field names with a comma
                        (,)
  --type {workflow_job_template_node}
                        only list workflow_job_template_nodes with the
                        specified type
  --created CREATED     only list workflow_job_template_nodes with the
                        specified created
  --modified MODIFIED   only list workflow_job_template_nodes with the
                        specified modified
  --extra_data JSON/YAML
                        only list workflow_job_template_nodes with the
                        specified extra_data
  --inventory ID        only list workflow_job_template_nodes with the
                        specified inventory
  --execution_environment ID
                        only list workflow_job_template_nodes with the
                        specified execution_environment
  --workflow_job_template ID
                        only list workflow_job_template_nodes with the
                        specified workflow_job_template
  --unified_job_template ID
                        only list workflow_job_template_nodes with the
                        specified unified_job_template
  --all_parents_must_converge BOOLEAN
                        only list workflow_job_template_nodes with the
                        specified all_parents_must_converge
  --identifier TEXT     only list workflow_job_template_nodes with the
                        specified identifier

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
```

(continues on next page)

(продолжение с предыдущей страницы)

```
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made
```

workflow_job_template_nodes create

```
usage: awx workflow_job_template_nodes create [-h] [--extra_data JSON/YAML]
                                              [--inventory ID]
                                              [--scm_branch TEXT]
                                              [--job_type {None,,run,check}]
                                              [--job_tags TEXT]
                                              [--skip_tags TEXT]
                                              [--limit TEXT]
                                              [--diff_mode BOOLEAN]
                                              [--verbosity {None,0,1,2,3,4,5}]
                                              [--execution_environment ID]
                                              [--forks INTEGER]
                                              [--job_slice_count INTEGER]
                                              [--timeout INTEGER]
                                              --workflow_job_template ID
                                              [--unified_job_template ID]
                                              [--all_parents_must_converge BOOLEAN]
                                              [--identifier TEXT]

required arguments:
  --workflow_job_template ID
                        the ID of the associated workflow_job_template

optional arguments:
  -h, --help            show this help message and exit
  --extra_data JSON/YAML
                        a JSON or YAML string. You can optionally specify a
                        file path e.g., @path/to/file.yml
  --inventory ID        Inventory applied as a prompt, assuming job template
                        prompts for inventory
  --scm_branch TEXT
  --job_type {None,,run,check}
  --job_tags TEXT
  --skip_tags TEXT
  --limit TEXT
  --diff_mode BOOLEAN
  --verbosity {None,0,1,2,3,4,5}
  --execution_environment ID
                        The container image to be used for execution.
  --forks INTEGER
  --job_slice_count INTEGER
  --timeout INTEGER
  --unified_job_template ID
                        the ID of the associated unified_job_template
  --all_parents_must_converge BOOLEAN
                        If enabled then the node will only run if all of the
                        parent nodes have met the criteria to reach this node
  --identifier TEXT     An identifier for this node that is unique within its
                        workflow. It is copied to workflow job nodes
                        corresponding to this node.

awx workflow_job_template_nodes create: the following arguments are required: --workflow_
↪job_template
```

workflow_job_template_nodes get

```
usage: awx workflow_job_template_nodes get [-h] [-f {json,yaml,jq,human}]
                                            [--filter TEXT]
                                            [--conf.color BOOLEAN] [-v]
                                            id

positional arguments:
  id                    the ID (or unique name) of the resource

optional arguments:
  -h, --help            show this help message and exit

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made

awx workflow_job_template_nodes get: the following arguments are required: id
```

workflow_job_template_nodes modify

```
usage: awx workflow_job_template_nodes modify [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx workflow_job_template_nodes modify: the following arguments are required: id
```

workflow_job_template_nodes delete

```
usage: awx workflow_job_template_nodes delete [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx workflow_job_template_nodes delete: the following arguments are required: id
```

workflow_job_templates

```
usage: awx workflow_job_templates [-h] action ...

positional arguments:
  action
    list
    create
    get
    modify
    delete
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    launch
    associate
    disassociate

optional arguments:
  -h, --help    show this help message and exit

awx workflow_job_templates: the following arguments are required: action
```

workflow_job_templates list

```
usage: awx workflow_job_templates list [-h] [--all] [--order_by ORDER_BY]
                                       [-f {json,yaml,jq,human}]
                                       [--filter TEXT] [--conf.color BOOLEAN]
                                       [-v] [--type {workflow_job_template}]
                                       [--created CREATED]
                                       [--modified MODIFIED] [--name TEXT]
                                       [--description TEXT]
                                       [--last_job_run LAST_JOB_RUN]
                                       [--last_job_failed BOOLEAN]
                                       [--next_job_run NEXT_JOB_RUN]
                                       [--status {new,pending,waiting,running,successful,
↪failed,error,canceled,never updated,ok,missing,none,updating}]
                                       [--extra_vars JSON/YAML]
                                       [--organization ID]
                                       [--survey_enabled BOOLEAN]
                                       [--allow_simultaneous BOOLEAN]
                                       [--ask_variables_on_launch BOOLEAN]
                                       [--inventory ID]
                                       [--ask_inventory_on_launch BOOLEAN]
                                       [--ask_scm_branch_on_launch BOOLEAN]
                                       [--ask_limit_on_launch BOOLEAN]
                                       [--webhook_service {,github,gitlab,bitbucket_dc}]
                                       [--webhook_credential ID]
                                       [--ask_labels_on_launch BOOLEAN]
                                       [--ask_skip_tags_on_launch BOOLEAN]
                                       [--ask_tags_on_launch BOOLEAN]

optional arguments:
  -h, --help            show this help message and exit
  --all                 fetch all pages of content from the API when returning
                        results (instead of just the first page)
  --order_by ORDER_BY   order results by given field name, prefix the field
                        name with a dash (-) to sort in reverse eg
                        --order_by='-name',multiple sorting fields may be
                        specified by separating the field names with a comma
                        (,)
  --type {workflow_job_template}
                        only list workflow_job_templates with the specified
                        type
  --created CREATED     only list workflow_job_templates with the specified
                        created
  --modified MODIFIED   only list workflow_job_templates with the specified
                        modified
  --name TEXT           only list workflow_job_templates with the specified
                        name
  --description TEXT    only list workflow_job_templates with the specified
                        description
  --last_job_run LAST_JOB_RUN
                        only list workflow_job_templates with the specified
                        last_job_run
  --last_job_failed BOOLEAN
```

(continues on next page)

(продолжение с предыдущей страницы)

```
                        only list workflow_job_templates with the specified
                        last_job_failed
  --next_job_run NEXT_JOB_RUN
                        only list workflow_job_templates with the specified
                        next_job_run
  --status {new,pending,waiting,running,successful,failed,error,canceled,never updated,
↪ok,missing,none,updating}
                        only list workflow_job_templates with the specified
                        status
  --extra_vars JSON/YAML, -e JSON/YAML
                        only list workflow_job_templates with the specified
                        extra_vars
  --organization ID     only list workflow_job_templates with the specified
                        organization
  --survey_enabled BOOLEAN
                        only list workflow_job_templates with the specified
                        survey_enabled
  --allow_simultaneous BOOLEAN
                        only list workflow_job_templates with the specified
                        allow_simultaneous
  --ask_variables_on_launch BOOLEAN
                        only list workflow_job_templates with the specified
                        ask_variables_on_launch
  --inventory ID        only list workflow_job_templates with the specified
                        inventory
  --ask_inventory_on_launch BOOLEAN
                        only list workflow_job_templates with the specified
                        ask_inventory_on_launch
  --ask_scm_branch_on_launch BOOLEAN
                        only list workflow_job_templates with the specified
                        ask_scm_branch_on_launch
  --ask_limit_on_launch BOOLEAN
                        only list workflow_job_templates with the specified
                        ask_limit_on_launch
  --webhook_service {,github,gitlab,bitbucket_dc}
                        only list workflow_job_templates with the specified
                        webhook_service
  --webhook_credential ID
                        only list workflow_job_templates with the specified
                        webhook_credential
  --ask_labels_on_launch BOOLEAN
                        only list workflow_job_templates with the specified
                        ask_labels_on_launch
  --ask_skip_tags_on_launch BOOLEAN
                        only list workflow_job_templates with the specified
                        ask_skip_tags_on_launch
  --ask_tags_on_launch BOOLEAN
                        only list workflow_job_templates with the specified
                        ask_tags_on_launch

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made
```

workflow_job_templates create

```
usage: awx workflow_job_templates create [-h] --name TEXT [--description TEXT]
                                         [--extra_vars JSON/YAML]
                                         [--organization ID]
                                         [--survey_enabled BOOLEAN]
                                         [--allow_simultaneous BOOLEAN]
                                         [--ask_variables_on_launch BOOLEAN]
                                         [--inventory ID] [--limit TEXT]
                                         [--scm_branch TEXT]
                                         [--ask_inventory_on_launch BOOLEAN]
                                         [--ask_scm_branch_on_launch BOOLEAN]
                                         [--ask_limit_on_launch BOOLEAN]
                                         [--webhook_service {,github,gitlab,bitbucket_dc}
↪]
                                         [--webhook_credential ID]
                                         [--ask_labels_on_launch BOOLEAN]
                                         [--ask_skip_tags_on_launch BOOLEAN]
                                         [--ask_tags_on_launch BOOLEAN]
                                         [--skip_tags TEXT] [--job_tags TEXT]

required arguments:
  --name TEXT           Name of this workflow job template.

optional arguments:
  -h, --help            show this help message and exit
  --description TEXT    Optional description of this workflow job template.
  --extra_vars JSON/YAML, -e JSON/YAML
                        a JSON or YAML string. You can optionally specify a
                        file path e.g., @path/to/file.yml
  --organization ID     The organization used to determine access to this
                        template.
  --survey_enabled BOOLEAN
  --allow_simultaneous BOOLEAN
  --ask_variables_on_launch BOOLEAN
  --inventory ID        Inventory applied as a prompt, assuming job template
                        prompts for inventory
  --limit TEXT
  --scm_branch TEXT
  --ask_inventory_on_launch BOOLEAN
  --ask_scm_branch_on_launch BOOLEAN
  --ask_limit_on_launch BOOLEAN
  --webhook_service {,github,gitlab,bitbucket_dc}
                        Service that webhook requests will be accepted from
  --webhook_credential ID
                        Personal Access Token for posting back the status to
                        the service API
  --ask_labels_on_launch BOOLEAN
  --ask_skip_tags_on_launch BOOLEAN
  --ask_tags_on_launch BOOLEAN
  --skip_tags TEXT
  --job_tags TEXT

awx workflow_job_templates create: the following arguments are required: --name
```

workflow_job_templates get

```
usage: awx workflow_job_templates get [-h] [-f {json,yaml,jq,human}]
                                      [--filter TEXT] [--conf.color BOOLEAN]
                                      [-v]
                                      id

positional arguments:
  id                    the ID (or unique name) of the resource

optional arguments:
  -h, --help            show this help message and exit

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made

awx workflow_job_templates get: the following arguments are required: id
```

workflow_job_templates modify

```
usage: awx workflow_job_templates modify [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx workflow_job_templates modify: the following arguments are required: id
```

workflow_job_templates delete

```
usage: awx workflow_job_templates delete [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx workflow_job_templates delete: the following arguments are required: id
```

workflow_job_templates launch

```
usage: awx workflow_job_templates launch [-h] [--monitor]
                                         [--action-timeout ACTION_TIMEOUT]
                                         [--wait] [--interval INTERVAL]
                                         [--extra_vars JSON/YAML]
                                         [--inventory ID] [--limit TEXT]
                                         [--scm_branch TEXT] [--labels LABELS]
                                         [--skip_tags TEXT] [--job_tags TEXT]
                                         id
```

(continues on next page)

(продолжение с предыдущей страницы)

```
positional arguments:
  id

optional arguments:
  -h, --help            show this help message and exit
  --monitor             If set, prints stdout of the launched job until it
                        finishes.
  --action-timeout ACTION_TIMEOUT
                        If set with --monitor or --wait, time out waiting on
                        job completion.
  --wait                If set, waits until the launched job finishes.
  --interval INTERVAL   If set with --monitor or --wait, amount of time to
                        wait in seconds between api calls. Minimum value is
                        2.5 seconds to avoid overwhelming the api
  --extra_vars JSON/YAML, -e JSON/YAML
                        a JSON or YAML string. You can optionally specify a
                        file path e.g., @path/to/file.yml
  --inventory ID        the ID of the associated inventory
  --limit TEXT
  --scm_branch TEXT
  --labels LABELS
  --skip_tags TEXT
  --job_tags TEXT

awx workflow_job_templates launch: the following arguments are required: id
```

workflow_job_templates associate

```
usage: awx workflow_job_templates associate [-h]
                                            (--start_notification  | --success_
↪notification  | --failure_notification  | --approval_notification )
                                            id

positional arguments:
  id

optional arguments:
  -h, --help            show this help message and exit
  --start_notification
                        The ID (or name) of the notification_template to
                        associate
  --success_notification
                        The ID (or name) of the notification_template to
                        associate
  --failure_notification
                        The ID (or name) of the notification_template to
                        associate
  --approval_notification
                        The ID (or name) of the notification_template to
                        associate

awx workflow_job_templates associate: the following arguments are required: id
```

workflow_job_templates disassociate

```
usage: awx workflow_job_templates disassociate [-h]
                                               (--start_notification  | --success_
↪notification  | --failure_notification  | --approval_notification )
                                               id

positional arguments:
  id

optional arguments:
  -h, --help            show this help message and exit
  --start_notification
                        The ID (or name) of the notification_template to
                        disassociate
  --success_notification
                        The ID (or name) of the notification_template to
                        disassociate
  --failure_notification
                        The ID (or name) of the notification_template to
                        disassociate
  --approval_notification
                        The ID (or name) of the notification_template to
                        disassociate

awx workflow_job_templates disassociate: the following arguments are required: id
```

workflow_jobs

```
usage: awx workflow_jobs [-h] action ...

positional arguments:
  action
    list
    get
    delete
    monitor

optional arguments:
  -h, --help  show this help message and exit

awx workflow_jobs: the following arguments are required: action
```

workflow_jobs list

```
usage: awx workflow_jobs list [-h] [--all] [--order_by ORDER_BY]
                              [-f {json,yaml,jq,human}] [--filter TEXT]
                              [--conf.color BOOLEAN] [-v]
                              [--type {workflow_job}] [--created CREATED]
                              [--modified MODIFIED] [--name TEXT]
                              [--description TEXT] [--unified_job_template ID]
                              [--launch_type {manual,relaunch,callback,scheduled,
↪dependency,workflow,webhook,sync,scm}]
                              [--status {new,pending,waiting,running,successful,failed,
↪error,canceled}]
                              [--failed BOOLEAN] [--started STARTED]
                              [--finished FINISHED]
                              [--canceled_on CANCELED_ON] [--elapsed ELAPSED]
                              [--job_explanation TEXT] [--work_unit_id TEXT]
                              [--workflow_job_template ID]
```

(continues on next page)

(продолжение с предыдущей страницы)

```
                              [--extra_vars JSON/YAML]
                              [--allow_simultaneous BOOLEAN]
                              [--job_template ID] [--is_sliced_job BOOLEAN]
                              [--inventory ID]
                              [--webhook_service {,github,gitlab,bitbucket_dc}]
                              [--webhook_credential ID] [--webhook_guid TEXT]

optional arguments:
  -h, --help            show this help message and exit
  --all                 fetch all pages of content from the API when returning
                        results (instead of just the first page)
  --order_by ORDER_BY   order results by given field name, prefix the field
                        name with a dash (-) to sort in reverse eg
                        --order_by='-name',multiple sorting fields may be
                        specified by separating the field names with a comma
                        (,)
  --type {workflow_job}
                        only list workflow_jobs with the specified type
  --created CREATED     only list workflow_jobs with the specified created
  --modified MODIFIED   only list workflow_jobs with the specified modified
  --name TEXT           only list workflow_jobs with the specified name
  --description TEXT    only list workflow_jobs with the specified description
  --unified_job_template ID
                        only list workflow_jobs with the specified
                        unified_job_template
  --launch_type {manual,relaunch,callback,scheduled,dependency,workflow,webhook,sync,scm}
                        only list workflow_jobs with the specified launch_type
  --status {new,pending,waiting,running,successful,failed,error,canceled}
                        only list workflow_jobs with the specified status
  --failed BOOLEAN      only list workflow_jobs with the specified failed
  --started STARTED     only list workflow_jobs with the specified started
  --finished FINISHED   only list workflow_jobs with the specified finished
  --canceled_on CANCELED_ON
                        only list workflow_jobs with the specified canceled_on
  --elapsed ELAPSED     only list workflow_jobs with the specified elapsed
  --job_explanation TEXT
                        only list workflow_jobs with the specified
                        job_explanation
  --work_unit_id TEXT   only list workflow_jobs with the specified
                        work_unit_id
  --workflow_job_template ID
                        only list workflow_jobs with the specified
                        workflow_job_template
  --extra_vars JSON/YAML
                        only list workflow_jobs with the specified extra_vars
  --allow_simultaneous BOOLEAN
                        only list workflow_jobs with the specified
                        allow_simultaneous
  --job_template ID     only list workflow_jobs with the specified
                        job_template
  --is_sliced_job BOOLEAN
                        only list workflow_jobs with the specified
                        is_sliced_job
  --inventory ID        only list workflow_jobs with the specified inventory
  --webhook_service {,github,gitlab,bitbucket_dc}
                        only list workflow_jobs with the specified
                        webhook_service
  --webhook_credential ID
                        only list workflow_jobs with the specified
                        webhook_credential
  --webhook_guid TEXT   only list workflow_jobs with the specified
                        webhook_guid

input/output formatting:
```

(continues on next page)

(продолжение с предыдущей страницы)

```
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made
```

workflow_jobs get

```
usage: awx workflow_jobs get [-h] [-f {json,yaml,jq,human}] [--filter TEXT]
                             [--conf.color BOOLEAN] [-v]
                             id

positional arguments:
  id                    the ID (or unique name) of the resource

optional arguments:
  -h, --help            show this help message and exit

input/output formatting:
  -f {json,yaml,jq,human}, --conf.format {json,yaml,jq,human}
                        specify a format for the input and output
  --filter TEXT         specify an output filter (only valid with jq or human
                        format)
  --conf.color BOOLEAN  Display colorized output. Defaults to True
  -v, --verbose         print debug-level logs, including requests made

awx workflow_jobs get: the following arguments are required: id
```

workflow_jobs delete

```
usage: awx workflow_jobs delete [-h] id

positional arguments:
  id          the ID (or unique name) of the resource

optional arguments:
  -h, --help  show this help message and exit

awx workflow_jobs delete: the following arguments are required: id
```

workflow_jobs monitor

```
usage: awx workflow_jobs monitor [-h] id

positional arguments:
  id

optional arguments:
  -h, --help  show this help message and exit

awx workflow_jobs monitor: the following arguments are required: id
```

7.6 Утилита awx-manage

awx-manage является утилитой командной строки, используемой для получения подробной информации о контроллере и управления им. Команды утилиты awx-manage необходимо запускать на управляющем узле контроллера от имени пользователя awx или суперпользователя. Если управляющих узлов несколько, то запускать команду можно на любом из них.

Важно: Запуск команд awx-manage через playbook не поддерживается.

Вызов утилиты имеет следующий вид:

```
sudo awx-manage <command> [<arguments>]
```

Для вывода справочной информации воспользуйтесь командой:

```
sudo awx-manage --help
```

7.6.1 Импорт инвентаря

Утилита awx-manage предоставляет администраторам возможность импортировать инвентарь в Automation Controller с помощью следующих шагов:

1. В контроллере создайте инвентарь, который будет служить местом назначения для импорта.
2. Подключитесь по SSH к управляющему узлу контроллера и выполните команду:

   ```
   sudo awx-manage inventory_import --source=<path_to_inventory> --inventory-id=<id>
   ```

   Здесь:

   - <path_to_inventory> – путь к файлу инвентаря, который будет использован в качестве источника.
   - <id> – идентификатор инвентаря, созданного ранее.

Команда inventory_import позволяет синхронизировать существующий инвентарь в контроллере с текстовым файлом инвентаря в формате INI или YAML, динамическим инвентарем или каталогом, содержащим такие файлы и инвентари.

При импорте можно использовать аргументы --overwrite и --overwrite-vars, чтобы управлять тем, как данные объединяются или заменяются.

По умолчанию данные из внешнего источника перезаписывают имеющиеся. Данные, которые отсутствуют во внешнем источнике, но присутствуют в контроллере, сохраняются.

При запуске утилиты с аргументом --overwrite используются только данные из внешнего источника. Данные, отсутствующие во внешнем источнике, будут удалены.

При запуске утилиты с аргументом --overwrite-vars используются переменные исключительно из внешнего источника. Узлы, которые отсутствуют во внешнем источнике, но присутствуют в контроллере, сохраняются.

7.6.2 Удаление устаревших данных

Команды `cleanup_jobs` и `cleanup_activitystream` используются для удаления данных о выполненных заданиях и данных ленты активности старше количества дней, указанного в аргументе `--days`:

- Удаление данных о выполненных заданиях:

```
sudo awx-manage cleanup_jobs --days <n>
```

- Удаление данных ленты активности:

```
sudo awx-manage cleanup_activitystream --days <n>
```

7.6.3 Удаление узла

Команда `deprovision_instance` используется для удаления узлов:

```
sudo awx-manage deprovision_instance --hostname=<hostname>
```

Здесь <hostname> – название удаляемого узла в инвентаре установщика платформы.

Примечание: Узел необходимо удалить из инвентаря установщика платформы, если он не должен появиться в кластере при повторном запуске сценария настройки.

7.6.4 Метрики узлов

Утилита `awx-manage` предоставляет возможность генерировать данные метрик узлов в формате CSV.

Создание файлов с данными метрик узла и информацией о кластере:

```
sudo awx-manage host_metric --csv
```

Чтобы упаковать все файлы в один архив, воспользуйтесь командой:

```
sudo awx-manage host_metric --tarball
```

Чтобы указать количество строк (<N>) для вывода в каждый файл, используйте аргумент `--pows_per_file`:

```
sudo awx-manage host_metric --rows_per_file <N>
```

7.7 Настройка контроллера через Ansible

Важно: Эта часть документации находится в стадии разработки.

7.8 Среда исполнения

Automation Controller использует *среды исполнения* для запуска заданий на исполняющих и гибридных узлах.

При развертывании контроллера в нем автоматически создаются две среды исполнения:

- Control Plane Execution Environment:
  - Используется при развертывании и обновлении платформы.
  - Используется для выполнения служебных заданий Control Plane.
  - Образ среды невозможно изменить через пользовательский интерфейс.
- Default execution environment:
  - Используется по умолчанию для выполнения заданий из шаблонов.
  - Его образ можно заменить на другой.

Для создания дополнительных сред исполнения можно использовать как образы из реестра Astra Automation[161] (рекомендуется), так и собранные самостоятельно.

Поддерживается использование *полномочий* для доступа к реестрам образов, требующим аутентификацию пользователей.

Вместо среды по умолчанию можно назначить другую среду на следующих уровнях (в порядке возрастания приоритета использования):

1. контроллер;
2. организация;
3. проект;
4. шаблон задания.

При создании среды исполнения необходимо указать следующие параметры:

- Название. Должно быть уникальным в рамках контроллера.
- Образ. Ссылка на образ в реестре образов. Указывается в следующем формате:

  ```
  <registry>/<name>:<tag>
  ```

  где:

  - <registry> – URL реестра образов;
  - <name> – название образа;
  - <tag> – тег, указывающий версию образа.

Для замены Control Plane Execution Environment на уже установленном контроллере воспользуйтесь утилитой `aa-setup` с дополнительной переменной `control_plane_execution_environment`, например:

```
./aa-setup -- -e control_plane_execution_environment=<image_name>
```

где <image_name> – полное название образа контейнера, как оно представлено в Private Automation Hub, например, `hub.example.com/aa/aa-base-ee:0.6.2`.

После выполнения команды контроллер будет недоступен 2-3 минуты.

[161] https://registry.astralinux.ru/browse/aa/#!/taglist/aa-base-ee

7.8.1 Параметры загрузки

Automation Controller позволяет задать настройки загрузки образа для каждой среды исполнения. Поддерживаются три варианта управления загрузкой:

- Перед запуском контейнера образ всегда загружается заново.

  Эта настройка может быть полезна при использовании в ссылке на образ тега `latest`.
- Перед запуском контейнера образ загружается только в том случае, если он отсутствует.
- Никогда – образ загружается только один раз, в момент создания среды исполнения.

7.8.2 Получение сведений о составе образа

Для получения сведений о версиях компонентов воспользуйтесь инструкцией из секции *Получение сведений об образах среды исполнения*.

Архитектура *Структура управления* Элементы управления и связь между ними.

Графический интерфейс Управление через веб-интерфейс.

Программный интерфейс Формальное описание API в стандарте Open API.

ГЛАВА 8

Разработка контента и среды исполнения

Значимой частью Astra Automation является набор средств создания собственного контента Ansible и контейнерных образов для среды исполнения. Данный раздел представляет концепции и инструкции по созданию образов среды исполнения и коллекций Ansible, удовлетворяющих потребностям организации.

8.1 Инструменты

В состав Astra Automation входят следующие инструментальные средства для работы с Ansible:

- Ansible Navigator – создание, проверка и запуск контента, а также поиск и устранение проблем в его работе.
- Ansible Builder – создание образов среды исполнения.
- Ansible Lint – статический анализ кода на наличие ошибок.
- Ansible Molecule – тестирование ролей.

8.1.1 Установка

Для установки всех доступных инструментов выполните следующие действия:

С доступом к интернету

1. Подключите репозиторий Astra Automation.

Инструкция по подключению репозитория

1. В каталоге `/etc/apt/sources.list.d/` создайте файл `astra-automation.list` со ссылкой на репозиторий Astra Automation:

   ```
   deb https://dl.astralinux.ru/aa/aa-debs-for-alse-1.7 <version> main
   ```

   Вместо <version> необходимо подставить версию устанавливаемой платформы, например, `1.1-upd1`.

   Доступные версии продукта опубликованы в таблице *История обновлений*.

2. Обновите список доступных пакетов:

   ```
   sudo apt update
   ```

2. Установите пакет `astra-automation-cdk`:

   ```
   sudo apt install astra-automation-cdk --yes
   ```

Если необходима установка отдельных утилит, следуйте инструкции в описании соответствующей утилиты.

Без доступа к интернету (Offline Bundle)

Added in version 1.1-upd1.

1. Загрузите архив с установщиком на установочный узел.

   Примечание: Архив доступен для загрузки в Личном кабинете[162] при наличии действующей лицензии на продукт.

2. Создайте каталог `/opt/rbta/aa/CDK-setup/`:

   ```
   sudo mkdir -p /opt/rbta/aa/CDK-setup/
   ```

3. Распакуйте содержимое архива с установщиком в каталог `/opt/rbta/aa/CDK-setup/`:

   ```
   sudo tar -xvzf <archive>.tar.gz -C /opt/rbta/aa/CDK-setup/
   ```

4. Перейдите в каталог с распакованными файлами:

   ```
   cd /opt/rbta/aa/CDK-setup/
   ```

5. Сделайте файл `install_cdk_bundle.sh` исполняемым:

   ```
   sudo chmod +x install_cdk_bundle.sh
   ```

6. Для установки всех доступных утилит выполните команду:

   ```
   sudo ./install_cdk_bundle.sh
   ```

7. Чтобы установить отдельную утилиту, выполните команду:

   ```
   sudo ./install_cdk_bundle.sh <utility_name>
   ```

   Здесь <utility_name> – название утилиты, которую необходимо установить. Поддерживаемые значения:

   - `ansible-navigator`;
   - `ansible-builder`;

[162] https://lk-new.astralinux.ru

- `ansible-lint`;
- `molecule`.

8. Проведите дополнительные настройки Podman согласно *инструкции*.
9. Если у вас установлен Podman версии 4.3.1, добавьте `aa-creator-ee` в список локальных образов:

```
sudo podman load -i /opt/rbta/aa/CDK-setup/aa-creator-ee/aa-creator-ee.tar
```

Здесь `/opt/rbta/aa/CDK-setup/aa-creator-ee/aa-creator-ee.tar` – путь к архиву с образом `aa-creator-ee`.

Ansible Navigator

Ansible Navigator предоставляет удобный единый интерфейс для использования возможностей многих утилит Ansible в командной строке. Ansible Navigator может вызывать эти утилиты из среды исполнения (основной рекомендуемый вариант использования) или непосредственно из рабочей станции.

Установка

Для установки Ansible Navigator выполните следующие действия:

1. Подключите репозиторий Astra Automation.

Инструкция по подключению репозитория

1. В каталоге `/etc/apt/sources.list.d/` создайте файл `astra-automation.list` со ссылкой на репозиторий Astra Automation:

```
deb https://dl.astralinux.ru/aa/aa-debs-for-alse-1.7 <version> main
```

Вместо <version> необходимо подставить версию устанавливаемой платформы, например, `1.1-upd1`.

Доступные версии продукта опубликованы в таблице *История обновлений*.

2. Обновите список доступных пакетов:

```
sudo apt update
```

2. Выполните команду:

```
sudo apt-get install ansible-navigator --yes
```

Вместе с Ansible Navigator также устанавливаются:

- *Ansible Builder*;
- *Ansible Lint*;
- Ansible Runner;
- Docker.

Совет: Вместо Docker рекомендуется использовать *Podman*. Необходимо учесть, что для корректной работы Podman следует провести дополнительные *настройки*.

Установка при отсутствии доступа к интернету описана в документе *Инструменты*.

Команды и утилиты

Основу Ansible Navigator составляет утилита `ansible-navigator`, вызов которой имеет следующий вид:

```
ansible-navigator <command> [<arguments>]
```

Здесь:

- <command> – команда;
- <arguments> – аргументы.

Команды, используемые для вызова других утилит Ansible:

Команда	Вызываемая утилита	Пример
`builder`	*ansible-builder*	`ansible-navigator builder␣ ↪build --tag ee:0.0.1`
`config`	ansible-config[163]	`ansible-navigator config␣ ↪init`
`doc`	ansible-doc[164]	`ansible-navigator doc␣ ↪ansible.builtin.apt`
`inventory`	ansible-inventory[165]	`ansible-navigator␣ ↪inventory -i ./inventory. ↪ini`
`lint`	ansible-lint[166]	`ansible-navigator lint ./ ↪playbooks/*.yml`
`run`	ansible-playbook[167]	`ansible-navigator run ./ ↪playbook.yml -i ./ ↪inventory.yml`

Собственные команды Ansible Navigator:

- `collections` – просмотр сведений об имеющихся коллекциях Ansible.
- exec – запуск команды Ansible в *среде исполнения*.
- `images` – управление образами среды исполнения.
- `replay` – просмотр артефактов, собранных во время выполнения playbook.
- `settings` – просмотр действующих значений настроек Ansible Navigator.
- `welcome` – интерактивный режим работы.

Подробное описание команд и возможных аргументов см. в *справочных данных*.

[163] https://docs.ansible.com/ansible/latest/cli/ansible-config.html
[164] https://docs.ansible.com/ansible/latest/cli/ansible-doc.html
[165] https://docs.ansible.com/ansible/latest/cli/ansible-inventory.html
[166] https://ansible.readthedocs.io/projects/lint/
[167] https://docs.ansible.com/ansible/latest/cli/ansible-playbook.html

Режимы работы

Ansible Navigator может работать в интерактивном (по умолчанию) и неинтерактивном режимах.

Для запуска Ansible Navigator в неинтерактивном режиме добавьте аргумент `--mode` со значением `stdout`, например:

```
ansible-navigator images list --mode stdout
```

Интерактивный режим работы

При запуске в интерактивном режиме Ansible Navigator предоставляет псевдографический интерфейс следующего вида:

```
   Name                                  Default Source                   Current
 0|Ansible runner artifact dir           True    Not set                  Not set
 1|Ansible runner rotate artifacts count True    Not set                  Not set
 2|Ansible runner timeout                True    Not set                  Not set
 3|Ansible runner write job events       True    Defaults                 False
 4|App                                   False   Command line             settings
 5|Cmdline                               True    Not set                  Not set
 6|Collection doc cache path             True    Defaults                 /home/vagrant/.
↪cache/ansible-navigator/c
 7|Config                                True    Not set                  Not set
 8|Container engine                      False   Automatically determined docker
 9|Container options                     True    Not set                  Not set
10|Current settings file                 True    None                     None
11|Display color                         True    Defaults                 True
12|Editor command                        True    Defaults                 vi +{line_
↪number} {filename}
13|Editor console                        True    Defaults                 True
14|Enable prompts                        True    Defaults                 False
15|Exec command                          True    Defaults                 /bin/bash
16|Exec shell                            True    Defaults                 True
17|Execution environment                 True    Defaults                 True
18|Execution environment image           True    Defaults                 registry.
↪astralinux.ru/aa/aa-creator-ee:
19|Execution environment volume mounts   True    Not set                  Not set
^b/PgUp page up       ^f/PgDn page down       ↑↓ scroll       esc back       [0-9] goto ␣
↪     :help help
```

Этот интерфейс состоит из следующих элементов, сверху вниз:

1. Заголовки столбцов. Первый столбец не имеет названия и используется для вывода номера пункта списка.
2. Список записей.
3. Строка с подсказками.

Интерактивный режим обладает следующими особенностями:

- Для прокрутки списка используются клавиши ↑, ↓, PgUp и PgDn.
- Для завершения работы с Ansible Navigator или перехода на уровень выше используется клавиша Esc.
- Для ввода команд и перехода к пунктам списка используется следующая последовательность действий:
  1. Ввести символ `:` и сразу же после него название команды или номер строки, например, `:help` или `:19`.
  2. Нажать Enter.

> **Совет:** Для перехода к строкам с номерами от 0 до 9 можно использовать быстрый ввод – достаточно нажать соответствующую клавишу цифрового ряда клавиатуры.

- При работе с некоторыми разделами можно использовать клавиши `-` и `+` для перехода к предыдущему и следующему пунктам списка соответственно.

Интерактивный режим может быть полезен для получения детальных сведений о действующих настройках Ansible Navigator, образах среды исполнения, коллекциях и так далее.

Неинтерактивный режим работы

При запуске в неинтерактивном режиме Ansible Navigator выполняет команды без взаимодействия с пользователем. Вывод выполняемых команд направляется в стандартный поток вывода `stdout`.

Неинтерактивный режим может быть полезен при использовании Ansible Navigator в сценариях командной строки, когда взаимодействие с пользователем не требуется.

Настройки

Значения настроек Ansible Navigator могут быть заданы следующими способами:

- аргументы командной строки;
- переменные окружения;
- конфигурационный файл.

При запуске Ansible Navigator выполняет поиск своего конфигурационного файла в следующем порядке:

1. Путь, указанный в переменной окружения `ANSIBLE_NAVIGATOR_CONFIG`.
2. Файл `ansible-navigator.<ext>` в каталоге проекта.
3. Файл `.ansible-navigator.<ext>` в домашнем каталоге.

Здесь `<ext>` – расширение имени файла, `.yml`, `.yaml` или `.json`.

Поиск прекращается при первом совпадении. Например, если в каталоге проекта существует файл `ansible-navigator.yml`, настройки из файла `~/.ansible-navigator.yml`, расположенного в домашнем каталоге, загружены **не будут**.

Особенности хранения настроек в конфигурационных файлах:

- Доступны форматы YAML и JSON.
- При использовании формата JSON название файла должно иметь расширение `.json`.
- При использовании формата YAML название файла должно иметь расширение `.yml` или `.yaml`.
- В каталоге проекта и домашнем каталоге должен быть только один конфигурационный файл Ansible Navigator. В противном случае Ansible Navigator выводит сообщение об ошибке.

Пример файла с настройками

Список 1: ansible-navigator.yml

```
---
ansible-navigator:
  #
  #   ansible:
  #     config:
  #       help: False
  #       path: /tmp/ansible.cfg
  #     cmdline: "--forks 15"
  #     doc:
  #       help: False
  #       plugin:
  #         name: shell
  #         type: become
  #     inventory:
  #       help: False
  #       entries:
  #         - /tmp/test_inventory.yml
  #     playbook:
  #       help: False
  #       path: /tmp/test_playbook.yml
  #
  #   ansible-builder:
  #     help: False
  #     workdir: /tmp/
  #
  #   ansible-lint:
  #     config: ~/ansible-lint.yml
  #     lintables: ~/myproject/
  #
  #   ansible-runner:
  #     artifact-dir: /tmp/test1
  #     rotate-artifacts-count: 10
  #     timeout: 300
  #     job-events: False
  #
  #   app: run
  #
  #   collection-doc-cache-path: /tmp/cache.db
  #
  #   color:
  #     enable: False
  #     osc4: False
  #
  #   editor:
  #     command: vim_from_setting
  #     console: False
  #
  #   enable-prompts: False
  #
  #   exec:
  #     shell: False
  #     command: /bin/foo
  #
  #   execution-environment:
  #     container-engine: podman
  #     enabled: False
  #     environment-variables:
  #       pass:
  #         - ONE
  #         - TWO
  #         - THREE
```

(continues on next page)

(продолжение с предыдущей страницы)

```
  #       set:
  #         KEY1: VALUE1
  #         KEY2: VALUE2
  #         KEY3: VALUE3
  #     image: test_image:latest
  #     pull:
  #       arguments:
  #         - "--tls-verify=false"
  #       policy: never
  #     volume-mounts:
  #       - src: "/tmp"
  #         dest: "/test1"
  #         options: "Z"
  #     container-options:
  #       - "--net=host"
  #
  #   format: json
  #
  #   images:
  #     details:
  #       - ansible_version
  #       - python_version
  #
  #   inventory-columns:
  #     - ansible_network_os
  #     - ansible_network_cli_ssh_type
  #     - ansible_connection
  #
  logging:
    level: critical
#     append: False
#     file: /tmp/log.txt
#
#   mode: stdout
#
#   playbook-artifact:
#     enable: True
#     replay: /tmp/test_artifact.json
#     save-as: /tmp/test_artifact.json
#
#   settings:
#     effective: False
#     sample: False
#     schema: json
#     sources: False
#
#   time-zone: Japan
```

Подробное описание настроек см. в *справочных данных*.

Примеры

Изучите использование Ansible Navigator в неинтерактивном и интерактивном режимах на примерах.

Использование Ansible Navigator в неинтерактивном режиме

Для работы с Ansible Navigator в неинтерактивном режиме используйте аргумент `--mode` со значением `stdout`.

- Получение справочной информации об использовании команды `ansible-config` в неинтерактивном режиме:

```
ansible-navigator config --hc
```

- Получение сведений о коллекциях, доступных в окружении используемой ОС:

```
ansible-navigator collections --mode stdout
```

- Получение сведений о коллекциях, доступных в образе `aa-creator-ee:latest`:

```
ansible-navigator collections \
   --eei registry.astralinux.ru/aa/aa-creator-ee:latest \
   --mode stdout
```

- Получение сведений о загруженных образах среды исполнения:

```
ansible-navigator images --mode stdout
```

- Загрузка самой новой версии образа `aa-creator-ee` из реестра:

```
ansible-navigator images \
   --mode stdout \
   --eei registry.astralinux.ru/aa/aa-creator-ee:latest
```

- Загрузка образа `aa-creator-ee:0.2.0`:

```
ansible-navigator images \
   --mode stdout \
   --eei registry.astralinux.ru/aa/aa-creator-ee:0.2.0
```

Использование Ansible Navigator в интерактивном режиме

Для использования Ansible Navigator в интерактивном режиме используйте аргумент `--mode` со значением `interactive`.

Получение сведений об образах среды исполнения

Для получения сведения о составе образов среды исполнения выполните следующие действия:

1. Выполните команду `ansible-navigator` с аргументами `images` и `--mode`:

```
ansible-navigator images --mode interactive
```

   В терминал выводится список образов, например:

```
  Image                    Tag         Execution environment                 ␣
↪Created                 Size
0|aa-base-ee               latest      True                                   6 days␣
↪ago             1.1GB
1|aa-base-ee               0.6.0       True                                   10␣
↪days ago            1.1GB
2|aa-base-ee               0.5.1       False                                  3␣
↪months ago           945MB
3|aa-creator-ee            latest      True                                   6 days␣
↪ago             1.76GB
4|aa-creator-ee            0.2.0       True                                   10␣
↪days ago            1.76GB
```

2. Для просмотра сведений об образе `aa-creator-ee:latest` выполните следующие действия:

   1. Нажмите 3.

      В терминал выводится меню следующего вида:

```
  Image: aa-creator-ee:latest                Description
0|Image information                          Information collected from image␣
↪inspection
1|General information                        OS and python version information
2|Ansible version and collections            Information about ansible and␣
↪ansible collections
3|Python packages                            Information about python and␣
↪python packages
4|Operating system packages                  Information about operating system␣
↪packages
5|Everything                                 All image information
```

   2. Нажмите соответствующую клавишу для перехода в нужный раздел. Например, для выбора раздела **Python packages** нажмите 3.

      Дождитесь загрузки сведений о пакетах Python. По окончании загрузки в терминал выводятся строки следующего вида:

```
    Name                Version  Summary
  0|aenum               3.1.15   Advanced Enumerations (compatible with Python's␣
↪stdlib Enum), NamedTuples, and N
  1|aiohttp             3.8.3    Async http client/server framework (asyncio)
  2|aiosignal           1.3.1    aiosignal: a list of registered asynchronous␣
↪callbacks
  3|ansible-compat      4.1.11   Ansible compatibility goodies
  4|ansible-core        2.15.10  Radically simple IT automation
  5|ansible-lint        6.22.2   Checks playbooks for practices and behavior␣
↪that could potentially be improved
  6|ansible-pylibssh    1.0.0    Python bindings for libssh client specific to␣
↪Ansible use case
  7|ansible-runner      2.3.6    "Consistent Ansible Python API and CLI with␣
↪container and process isolation runt
  8|appdirs             1.3.0    A small Python module for determining␣
↪appropriate " +         "platform-specific
  9|argcomplete         3.4.0    Bash tab completion for argparse
 10|async-timeout       4.0.2    Timeout context manager for asyncio programs
 11|attrs               23.2.0   Classes Without Boilerplate
 12|bcrypt              4.1.2    Modern password hashing for your software and␣
↪your servers
 13|black               24.4.2   The uncompromising code formatter.
 14|bracex              2.4      Bash style brace expander.
 15|cachetools          5.3.2    Extensible memoizing collections and decorators
 16|certifi             2024.2.2 Python package for providing Mozilla's CA␣
↪Bundle.
```

(continues on next page)

(продолжение с предыдущей страницы)

```
 17|cffi               1.16.0   Foreign Function Interface for Python calling C␣
↪code.
 18|charset-normalizer 2.1.1    The Real First Universal Charset Detector. Open,
↪ modern and actively maintained
 19|click              8.1.7    Composable command line interface toolkit
 20|click-help-colors  0.9.4    Colorization of help messages in Click
 21|coverage           7.5.3    Code coverage measurement for Python
 22|cryptography       42.0.7   cryptography is a package which provides␣
↪cryptographic recipes and primitives to
 23|Cython             3.0.10   The Cython compiler for writing C extensions in␣
↪the Python language.
 24|decorator          5.1.1    Decorators for Humans
 25|dicttoxml          1.7.16   Converts a Python dictionary or other native␣
↪data type into a valid XML string.
 26|distlib            0.3.8    Distribution utilities
 27|distro             1.9.0    Distro - an OS platform information API
 28|docutils           0.19     Docutils -- Python Documentation Utilities
 29|dogpile.cache      1.3.2    A caching front-end based on the Dogpile lock.
 30|dumb-init          1.2.5    Simple wrapper script which proxies signals to␣
↪a child
 31|enrich             1.2.7    enrich
^b/PgUp page up       ^f/PgDn page down       ↑↓ scroll       esc back       [0-
↪9] goto       :help help
```

3. Для просмотра сведений о пакете `cffi` введите строку `:17` и нажмите Enter.

3. Для возврата в предыдущее меню нажмите Esc.

Получение сведений о коллекциях

Для получения сведения о коллекциях, входящих в образ среды исполнения `aa-creator-ee:latest`, выполните следующие действия:

1. Выполните команду `ansible-navigator` с аргументами `collections` и `--eei`:

```
ansible-navigator collections \
   --eei registry.astralinux.ru/aa/aa-creator-ee:latest
```

В терминал выводится список коллекций, например:

```
  Name            Version Shadowed Type      Path
0|ansible.builtin 2.15.10 False    contained /usr/local/lib/python3.9/dist-packages/
↪ansible
1|ansible.posix   1.5.4   False    contained /usr/share/ansible/collections/ansible_
↪collections/ansible/posix
2|ansible.utils   3.0.0   False    contained /usr/share/ansible/collections/ansible_
↪collections/ansible/utils
```

2. Для просмотра сведений о коллекции `ansible.posix` нажмите 1.

В терминал выводится список модулей и функций обратного вызова (callback), входящих в коллекцию, например:

```
   Ansible.posix      Type     Added Deprecated Description
 0|acl                module   1.0.0 False      Set and retrieve file ACL␣
↪information.
 1|at                 module   1.0.0 False      Schedule the execution of a command␣
↪or script file via the at co
 2|authorized_key     module   1.0.0 False      Adds or removes an SSH authorized␣
↪key
 3|cgroup_perf_recap  callback None  False      Profiles system activity of tasks␣
↪and full execution using cgrou
```

(continues on next page)

(продолжение с предыдущей страницы)

```
 4|csh                shell    None  False      C shell (/bin/csh)
 5|debug              callback None  False      formatted stdout/stderr display
 6|firewalld          module   None  False      Manage arbitrary ports/services␣
↪with firewalld
 7|firewalld_info     module   None  False      Gather information about firewalld
 8|fish               shell    None  False      fish shell (/bin/fish)
 9|json               callback None  False      Ansible screen output as JSON
10|jsonl              callback None  False      Ansible screen output as JSONL␣
↪(lines in json format)
11|mount              module   1.0.0 False      Control active and configured mount␣
↪points
12|patch              module   1.0.0 False      Apply patch files using the GNU␣
↪patch tool
13|profile_roles      callback None  False      adds timing information to roles
14|profile_tasks      callback None  False      adds time information to tasks
15|rhel_facts         module   1.5.0 False      Facts module to set or override␣
↪RHEL specific facts.
16|rhel_rpm_ostree    module   1.5.0 False      Ensure packages exist in a RHEL for␣
↪Edge rpm-ostree based system
17|rpm_ostree_upgrade module   1.5.0 False      Manage rpm-ostree upgrade␣
↪transactions
18|seboolean          module   1.0.0 False      Toggles SELinux booleans
19|selinux            module   1.0.0 False      Change policy and state of SELinux
20|skippy             callback None  True       Ansible screen output that ignores␣
↪skipped status
21|synchronize        module   1.0.0 False      A wrapper around rsync to make␣
↪common tasks in your playbooks qu
22|sysctl             module   1.0.0 False      Manage entries in sysctl.conf.
23|timer              callback None  False      Adds time to play stats
```

3. Для просмотра сведений о модуле `mount` введите строку `:11` и нажмите Enter.
4. Для возврата в предыдущее меню нажмите Esc.

Ansible Builder

Ansible Builder – это инструмент командной строки, предназначенный для создания образов контейнеров, используемых в качестве среды исполнения (EE, execution environment). Непосредственное создание образов осуществляется с помощью *Podman* (рекомендуемый вариант) или *Docker* (например, когда необходима *настройка размера разделяемой памяти*, `--shm-size`).

Создаваемый образ необходимо описать с помощью метаданных в специальном *файле определения среды исполнения*.

Установка

Для установки Ansible Builder выполните следующие действия:

1. Подключите репозиторий Astra Automation.

   #### Инструкция по подключению репозитория

   1. В каталоге `/etc/apt/sources.list.d/` создайте файл `astra-automation.list` со ссылкой на репозиторий Astra Automation:

      ```
      deb https://dl.astralinux.ru/aa/aa-debs-for-alse-1.7 <version> main
      ```

      Вместо <version> необходимо подставить версию устанавливаемой платформы, например, `1.1-upd1`.

      Доступные версии продукта опубликованы в таблице *История обновлений*.

   2. Обновите список доступных пакетов:

      ```
      sudo apt update
      ```

2. Установите пакет `ansible-builder`:

   ```
   sudo apt install ansible-builder --yes
   ```

3. Установите и настройте *Podman* (рекомендуется) или *Docker*.

Установка при отсутствии доступа к интернету описана в документе *Инструменты*.

Этапы сборки

Ansible Builder выполняет несколько этапов при запуске инструмента контейнеризации для создания образа контейнера.

1. **Base**: использует Docker или Podman для загрузки базового образа, который был определен в *файле определения среды исполнения*, затем устанавливает версию Python (если она определена и отличается от всех версий Python в базовом образе), `pip`, `ansible-runner` и `ansible-core` или `ansible`. Все три последующих этапа сборки используют результаты этапа Base.
2. **Galaxy**: загружает определенные в *файле определения среды исполнения* коллекции и сохраняет их локально.
3. **Builder**: загружает Python-пакеты и системные пакеты, определенные в *файле определения среды исполнения*, и сохраняет их локально.
4. **Final**: производит окончательную сборку образа.

Файл определения среды исполнения

Файл определения среды исполнения (EE definition) содержит описание создаваемого образа среды исполнения. По умолчанию Ansible Builder использует для сборки образа описание из файла `execution-environment.yml` или `execution-environment.yaml`, хранящегося в текущем каталоге.

Файл определения среды исполнения может содержать следующие разделы:

- `version`;
- `build_arg_defaults`;
- `dependencies`;
- `images`;

- additional_build_files;
- additional_build_steps;
- options.

Подробное описание файла определения среды исполнения см. в *справочнике*.

Список 2: Пример заполнения

```yaml
---
version: 3

build_arg_defaults:
  ANSIBLE_GALAXY_CLI_COLLECTION_OPTS: '--pre'

dependencies:
  ansible_core:
    package_pip: ansible-core==2.15.6
  ansible_runner:
    package_pip: ansible-runner
  galaxy: requirements.yml
  python:
    - six
    - psutil
  system: bindep.txt
  exclude:
    python:
      - docker
    system:
      - python3-Cython

images:
  base_image:
    name: hub.astra-automation.ru/aa/aa-creator-ee:0.1.0

additional_build_files:
  - src: files/ansible.cfg
    dest: configs

additional_build_steps:
  prepend_base:
    - RUN echo This is a prepend base command!

  prepend_galaxy:
    - COPY _build/configs/ansible.cfg /etc/ansible/ansible.cfg

  prepend_final:
    - RUN whoami
    - RUN cat /etc/os-release

  append_final:
    - RUN echo This is a post-install command!
    - RUN ls -la /etc
```

Сборка образа

Для запуска сборки образа в корневом каталоге проекта выполните команду:

```
ansible-builder build \
  -t <image>:<tag>
```

Здесь:

- <image> – название создаваемого образа;
- <tag> – метка образа.

Полный перечень возможных команд и опций см. в *справочнике*.

Примеры

Изучите на примерах применение Ansible Builder для сборки собственных образов среды исполнения. Предварительно выполните следующие действия:

1. Настройте *доступ к Private Automation Hub*.
2. Создайте каталог для хранения файлов проекта.

Примечание: Создавать файлы и выполнять команды далее следует в каталоге проекта, если явно не указано иное.

Установка библиотек Python

Чтобы добавить в образ библиотеки Python, выполните следующие действия:

1. Создайте файл зависимостей Python `requirements.yml` и заполните его в соответствии с требованиями **PEP 508**[168], например:

   Список 3: `requirements.yml`

   ```
   yamllint
   black==24.10.0
   ```

2. Файл определения среды исполнения заполните следующим образом:

   Список 4: *execution-environment.yml*

   ```yaml
   ---
   version: 3
   images:
     base_image:
       name: registry.astralinux.ru/aa/aa-base-ee:latest
   depdendencies:
     python: requirements.yml
   ```

3. Запустите сборку образа.

[168] https://peps.python.org/pep-0508/

Добавление коллекции из архива

Чтобы добавить в образ коллекцию Ansible в формате архива `.tar.gz`, выполните следующие действия:

1. Создайте файл со списком зависимостей Ansible, например:

   Список 5: `requirements.yml`

   ```
   ---
   collections:
     - name: astra.ald_pro
     - name: astra.nginx
       version: 1.8.2
   ```

2. Загрузите коллекции:

   ```
   ansible-navigator exec -- ansible-galaxy collection download -r requirements.yml
   ```

   В каталоге проекта будет создан подкаталог `collections/`, в который будут загружены указанные коллекции.

3. Файл определения среды исполнения заполните следующим образом:

   Список 6: `execution-environment.yml`

   ```
   ---
   version: 3

   images:
     base_image:
       name: registry.astralinux.ru/aa/aa-base-ee:latest

   dependencies:
     galaxy: collections/requirements.yml
   ```

4. Создайте контекст сборки:

   ```
   ansible-builder create
   ```

5. Скопируйте загруженные коллекции в контекст сборки:

   ```
   cp collections/*.tar.gz context/_build/
   ```

6. Запустите сборку образа.

Добавление коллекций из приватного репозитория Git

Доступ к приватному репозиторию Git можно получить разными способами. В данном примере рассматривается использование приватного ключа SSH.

Чтобы добавить коллекцию из приватного репозитория Git, на этапе сборки необходимо передать полномочия в контейнер. Для этого выполните следующие действия:

1. В каталоге проекта разместите файл приватного ключа SSH `id_rsa`, позволяющий получить доступ к репозиторию.

   Примечание: Приватный ключ не должен быть защищен паролем.

2. Создайте файл зависимостей Ansible, например:

Список 7: requirements.yml

```
---
collections:
  - name: my.collection
    url: https://git.example.com/private/my-collection.git
```

3. Файл определения среды исполнения заполните следующим образом:

Список 8: execution-environment.yml

```
---
version: 3

images:
  base_image:
    name: registry.astralinux.ru/aa/aa-base-ee:latest

dependencies:
  galaxy: requirements.yml

additional_build_files:
  - src: id_rsa
    dest: ssh

additional_build_steps:
  prepend_galaxy:
    - RUN mkdir /root/.ssh/
    - RUN printf 'Host git.example.com\n\tIdentityFile /root/.ssh/id_rsa\
↪nStrictHostKeyChecking no\n' > /root/.ssh/config
    - COPY --chmod=0600 ./_build/ssh/id_rsa /root/.ssh/id_rsa
  append_galaxy:
    - RUN rm -rf /root/.ssh
```

4. Запустите сборку образа.

Добавление коллекций из Private Automation Hub

Чтобы добавить в образ среды исполнения коллекции, размещенные в Private Automation Hub, выполните следующие действия:

1. Создайте токен для доступа к Private Automation Hub согласно *инструкции*.
2. Создайте подкаталог `files/`, а в нем – файл `ansible.cfg` с настройками подключения к репозиториям Private Automation Hub, например:

```
[galaxy]
server_list = galaxy, validated

[galaxy_server.galaxy]
url = https://galaxy.ansible.com/

[galaxy_server.validated]
url = https://private-hub.example.com/api/galaxy/content/validated/
```

   Здесь указаны публично доступны репозиторий Ansible Galaxy[169] и репозиторий `validated`, размещенный в Private Automation Hub.

3. Создайте файл зависимостей Ansible, например:

[169] https://galaxy.ansible.com/

Список 9: `requirements.yml`

```yaml
collections:
  - name: astra.ald_pro
    version: 2.0.0
  - name: astra.cups
    version: 1.3.3
  - name: astra.rubackup
  - name: astra.rupost
```

4. Файл определения среды исполнения заполните следующим образом:

Список 10: `execution-environment.yml`

```yaml
---
version: 3

dependencies:
  galaxy: requirements.yml

images:
  base_image:
    name: registry.astralinux.ru/aa/aa-base-ee:latest

additional_build_files:
  - src: ansible.cfg
    dest: configs

additional_build_steps:
  prepend_galaxy:
    - COPY _build/configs/ansible.cfg /etc/ansible/ansible.cfg
    - ARG ANSIBLE_GALAXY_SERVER_VALIDATED_TOKEN
```

Для получения доступа к Private Automation Hub используется токен, заданный в переменной окружения `ANSIBLE_GALAXY_SERVER_VALIDATED_TOKEN`. Значение этой переменной импортируется в образ из хостовой системы.

Название переменной окружения, в которой хранится токен для доступа к реестру, формируется по шаблону:

```
ANSIBLE_GALAXY_SERVER_<NAME>_TOKEN
```

Здесь <NAME> – набранное в верхнем регистре название сервера, указанное в `ansible.cfg`. Например, если бы серверы назывались `published` и `testing_dev`, соответствующие переменные окружения имели бы названия `ANSIBLE_GALAXY_SERVER_PUBLISHED_TOKEN` и `ANSIBLE_GALAXY_SERVER_TESTING_DEV_TOKEN`.

5. Экспортируйте токен для доступа к Private Automation Hub в соответствующую переменную окружения:

```
export ANSIBLE_GALAXY_SERVER_VALIDATED_TOKEN=<token>
```

6. Запустите сборку образа, передав в аргументе `--build-arg` название переменной окружения, в которой хранится токен для доступа к Private Automation Hub:

```
ansible-navigator builder build \
  -t <image>:<tag> \
  --build-arg ANSIBLE_GALAXY_SERVER_VALIDATED_TOKEN
```

Ansible Lint

Ansible Lint – это утилита командной строки, предназначенная для проверки качества и стиля кода в сценариях, ролях и коллекциях Ansible. Она помогает следовать лучшим практикам при написании кода, выявлять распространенные ошибки и потенциальные проблемы, а также поддерживать чистоту и читаемость кода.

Основные возможности Ansible Lint

Основные возможности Ansible Lint:

- Проверка на соответствие стандартам.

  Ansible Lint содержит правила написания кода из лучших практик, что позволяет избежать ошибок, ухудшающих качество и поддерживаемость кода.

- Автоматическое выявление ошибок.

  Утилита анализирует контент Ansible, указывая на места, которые могут быть потенциально проблемными или нестандартными.

- Адаптация к новым версиям Ansible.

  Ansible Lint помогает адаптировать код к новым версиям Ansible, что упрощает его обновление и предотвращает возможные несовместимости.

Установка

Для установки Ansible Lint выполните следующие действия:

1. Подключите репозиторий Astra Automation.

   ##### Инструкция по подключению репозитория

   1. В каталоге `/etc/apt/sources.list.d/` создайте файл `astra-automation.list` со ссылкой на репозиторий Astra Automation:

      ```
      deb https://dl.astralinux.ru/aa/aa-debs-for-alse-1.7 <version> main
      ```

      Вместо <version> необходимо подставить версию устанавливаемой платформы, например, `1.1-upd1`.

      Доступные версии продукта опубликованы в таблице *История обновлений*.

   2. Обновите список доступных пакетов:

      ```
      sudo apt update
      ```

2. Выполните команду:

   ```
   sudo apt-get install ansible-lint --yes
   ```

Установка при отсутствии доступа к интернету описана в документе *Инструменты*.

Настройка Ansible Lint

Ansible Lint предоставляет различные возможности для настройки в проектах:

- игнорирование определенных правил;
- включение дополнительных проверок;
- настройка профилей под конкретные нужды.

Ansible Lint можно настроить с помощью следующих компонентов:

- *аргументы команды*;
- *переменные окружения*;
- *файл настроек*.

При вычислении действующих значений аргументов используется следующий порядок (значения, указанные на более поздних этапах, заменяют значения, указанные ранее):

1. Переменные окружения.
2. Файл настроек.
3. Аргументы команды.

По умолчанию Ansible Lint начинает поиск файла настроек с текущего каталога. Возможные названия и расположения файлов:

- `.ansible-lint`;
- `.config/ansible-lint.yml`;
- `.config/ansible-lint.yaml`.

Если файл настроек не найден в текущем каталоге, Ansible Lint попытается найти его в родительских каталогах. Если используется Git, то поиск осуществляется только в пределах текущего репозитория. Путь к файлу настроек можно указать явно с помощью аргумента `-c`, например:

```
ansible-lint -c ansible-lint-config.yml
```

Настройки в конфигурационном файле задаются следующим образом:

Пример файла настроек

Список 11: .ansible-lint

```
---
# .ansible-lint

profile: null # min, basic, moderate, safety, shared, production

# Allows dumping of results in SARIF format
# sarif_file: result.sarif

# exclude_paths included in this file are parsed relative to this file's location
# and not relative to the CWD of execution. CLI arguments passed to the --exclude
# option are parsed relative to the CWD of execution.
exclude_paths:
  - .cache/ # implicit unless exclude_paths is defined in config
  - test/fixtures/formatting-before/
  - test/fixtures/formatting-prettier/
# parseable: true
# quiet: true
# strict: true
# verbosity: 1
```

(continues on next page)

(продолжение с предыдущей страницы)

```
# Mock modules or roles in order to pass ansible-playbook --syntax-check
mock_modules:
  - zuul_return
  # note the foo.bar is invalid as being neither a module or a collection
  - fake_namespace.fake_collection.fake_module
  - fake_namespace.fake_collection.fake_module.fake_submodule
mock_roles:
  - mocked_role
  - author.role_name # old standalone galaxy role
  - fake_namespace.fake_collection.fake_role # role within a collection

# Enable checking of loop variable prefixes in roles
loop_var_prefix: "^(__|{role}_)"

# Enforce variable names to follow pattern below, in addition to Ansible own
# requirements, like avoiding python identifiers. To disable add `var-naming`
# to skip_list.
var_naming_pattern: "^[a-z_][a-z0-9_]*$"

use_default_rules: true
# Load custom rules from this specific folder
# rulesdir:
#   - ./rule/directory/

# Ansible-lint is able to recognize and load skip rules stored inside
# `.ansible-lint-ignore` (or `.config/ansible-lint-ignore.txt`) files.
# To skip a rule just enter filename and tag, like "playbook.yml package-latest"
# on a new line.
# Optionally you can add comments after the tag, prefixed by "#". We discourage
# the use of skip_list below because that will hide violations from the output.
# When putting ignores inside the ignore file, they are marked as ignored, but
# still visible, making it easier to address later.
skip_list:
  - skip_this_tag

# Ansible-lint does not automatically load rules that have the 'opt-in' tag.
# You must enable opt-in rules by listing each rule 'id' below.
enable_list:
  - args
  - empty-string-compare # opt-in
  - no-log-password # opt-in
  - no-same-owner # opt-in
  - name[prefix] # opt-in
  - galaxy-version-incorrect # opt-in
  # add yaml here if you want to avoid ignoring yaml checks when yamllint
  # library is missing. Normally its absence just skips using that rule.
  - yaml
# Report only a subset of tags and fully ignore any others
# tags:
#   - jinja[spacing]

# Ansible-lint does not fail on warnings from the rules or tags listed below
warn_list:
  - skip_this_tag
  - experimental # experimental is included in the implicit list
# - role-name
# - yaml[document-start]  # you can also use sub-rule matches

# Some rules can transform files to fix (or make it easier to fix) identified
# errors. `ansible-lint --fix` will reformat YAML files and run these transforms.
# By default it will run all transforms (effectively `write_list: ["all"]`).
# You can disable running transforms by setting `write_list: ["none"]`.
# Or only enable a subset of rule transforms by listing rules/tags here.
# write_list:
```

(continues on next page)

(продолжение с предыдущей страницы)

```
#   - all

# Offline mode disables installation of requirements.yml and schema refreshing
offline: true

# Define required Ansible's variables to satisfy syntax check
extra_vars:
foo: bar
multiline_string_variable: |
  line1
  line2
complex_variable: ":{;\t$()"
# Uncomment to enforce action validation with tasks, usually is not
# needed as Ansible syntax check also covers it.
# skip_action_validation: false

# List of additional kind:pattern to be added at the top of the default
# match list, first match determines the file kind.
kinds:
  # - playbook: "**/examples/*.{yml,yaml}"
  # - galaxy: "**/folder/galaxy.yml"
  # - tasks: "**/tasks/*.yml"
  # - vars: "**/vars/*.yml"
  # - meta: "**/meta/main.yml"
  - yaml: "**/*.yaml-too"

# List of additional collections to allow in only-builtins rule.
# only_builtins_allow_collections:
#   - example_ns.example_collection

# List of additions modules to allow in only-builtins rule.
# only_builtins_allow_modules:
#   - example_module

# Allow setting custom prefix for name[prefix] rule
task_name_prefix: "{stem} | "
# Complexity related settings

# Limit the depth of the nested blocks:
# max_block_depth: 20

# Also recognize these versions of Ansible as supported:
# supported_ansible_also:
#   - "2.14"
```

Подробное описание параметров см. в *справочнике*.

Игнорирование правил для определенных файлов

Чтобы при проверке файлов Ansible Lint не использовал определенные правила, создайте файл `.ansible-lint-ignore` или `.config/ansible-lint-ignore.txt`. Созданный файл должен быть размещен в том же каталоге, что и файл основных настроек Ansible Lint. Каждая строка содержит путь к файлу и через пробел идентификатор правила.

Пример заполнения:

```
# Игнорирование правила package-latest для файла playbook.yml
playbook.yml package-latest

# Игнорирование правила deprecated-module для playbook.yml
playbook.yml deprecated-module
```

(continues on next page)

(продолжение с предыдущей страницы)

```
# Игнорирование правила no-tabs для роли my_role
roles/my_role/tasks/main.yml no-tabs
```

Создание правил

Ansible Lint позволяет создавать правила, содержащие собственные уникальные проверки, для поддержки стиля кодирования, принятого в вашей организации. Каждое правило необходимо создавать в отдельном файле Python и описывать в виде класса, который наследует `AnsibleLintRule`. Файл с правилом следует именовать согласно цели правила, например, `DeprecatedVariableRule.py`. Внутри каждого класса должны быть следующие поля:

- `id` – уникальный идентификатор правила;
- `description` – краткое описание правила (какой тип ошибок оно выявляет);
- `tags` – список тегов для группировки правил, близких по смыслу.

Для проверки кода по заданному правилу используют следующие методы:

- `match`;
- `matchtask`.

Метод `match` работает с каждой строкой и возвращает результат проверки. Если строка не соответствует критерию, метод возвращает `None` или `False`. Если строка соответствует критерию, метод возвращает `True` или настраиваемое сообщение.

Пример правила, реализующего метод `match`:

```
class DeprecatedVariableRule(AnsibleLintRule):
    """Deprecated variable declarations."""
    id = 'DEPRECATED_VAR'
    description = 'Check for deprecated variable syntax ${var}'
    tags = ['deprecations']

    def match(self, line: str) -> Union[bool, str]:
        return '${' in line
```

Метод `matchtask` обрабатывает каждую задачу или обработчик (handler). Все задачи стандартизируются, чтобы включать ключи `module` и `module_arguments`. Если в задаче присутствуют модификаторы `when`, `with_items`, `tags` и другие, они также доступны для проверки.

Пример правила, реализующего метод `matchtask`:

```
from ansiblelint.rules import AnsibleLintRule
from typing import Union

class TaskHasTag(AnsibleLintRule):
    """Ensure every task has a tag for better organization."""
    id = 'TASK_TAG'
    description = 'Tasks must have a tag to improve organization and filtering'
    tags = ['department', 'team']

    def matchtask(self, task, file=None) -> Union[bool, str]:
        if 'tags' not in task:
           return "Task missing a tag"
        return False
```

Примечание: Для создания более сложных правил рекомендуется изучить методы и свойства класса `AnsibleLintRule` и его родительских классов.

Пример

Для изучения Ansible Lint на практике, сделайте клон репозитория, содержащего примеры сценариев:

```
git clone https://github.com/ansible/ansible-lint.git
```

Проверка сценария

Для проверки сценария `example.yml` выполните команду:

```
ansible-lint --offline -p ansible-lint/examples/playbooks/example.yml
```

Пример вывода:

```
ansible-lint/examples/playbooks/example.yml:1: schema[playbook]: $[0].tasks[13] None is␣
↪not of type 'object'
ansible-lint/examples/playbooks/example.yml:9: jinja[spacing]: Jinja2 spacing could be␣
↪improved: echo {{this_variable}} is not set in this playbook -> echo {{ this_variable }
↪} is not set in this playbook (warning)
ansible-lint/examples/playbooks/example.yml:9: no-changed-when: Commands should not␣
↪change things if nothing needs doing.
ansible-lint/examples/playbooks/example.yml:12: no-changed-when: Commands should not␣
↪change things if nothing needs doing.
ansible-lint/examples/playbooks/example.yml:13: yaml[trailing-spaces]: Trailing spaces
ansible-lint/examples/playbooks/example.yml:15: args[module]: missing required␣
↪arguments: repo (warning)
ansible-lint/examples/playbooks/example.yml:15: latest[git]: Result of the command may␣
↪vary on subsequent runs.
ansible-lint/examples/playbooks/example.yml:18: args[module]: missing required␣
↪arguments: repo (warning)
ansible-lint/examples/playbooks/example.yml:18: latest[git]: Result of the command may␣
↪vary on subsequent runs.
ansible-lint/examples/playbooks/example.yml:21: args[module]: Unsupported parameters for␣
↪(basic.py) module: bobbins. Supported parameters include: accept_hostkey, accept_
↪newhostkey, archive, archive_prefix, bare, clone, depth, dest, executable, force, gpg_
↪whitelist, key_file, recursive, reference, refspec, remote, repo, separate_git_dir,␣
↪single_branch, ssh_opts, track_submodules, umask, update, verify_commit, version␣
↪(name). (warning)
ansible-lint/examples/playbooks/example.yml:21: no-free-form: Avoid using free-form when␣
↪calling module actions. (ansible.builtin.git)
ansible-lint/examples/playbooks/example.yml:24: command-instead-of-module: git used in␣
↪place of git module
ansible-lint/examples/playbooks/example.yml:24: no-changed-when: Commands should not␣
↪change things if nothing needs doing.
ansible-lint/examples/playbooks/example.yml:27: command-instead-of-module: git used in␣
↪place of git module
ansible-lint/examples/playbooks/example.yml:30: latest[git]: Result of the command may␣
↪vary on subsequent runs.
ansible-lint/examples/playbooks/example.yml:33: jinja[spacing]: Jinja2 spacing could be␣
↪improved: {{item}} -> {{ item }} (warning)
ansible-lint/examples/playbooks/example.yml:36: yaml[indentation]: Wrong indentation:␣
↪expected 8 but found 6
ansible-lint/examples/playbooks/example.yml:39: no-free-form: Avoid using free-form when␣
↪calling module actions. (ansible.builtin.dnf)
ansible-lint/examples/playbooks/example.yml:39: package-latest: Package installs should␣
↪not use latest.
ansible-lint/examples/playbooks/example.yml:42: name[missing]: All tasks should be named.
ansible-lint/examples/playbooks/example.yml:42: no-free-form: Avoid using free-form when␣
↪calling module actions. (ansible.builtin.debug)
ansible-lint/examples/playbooks/example.yml:44: no-free-form: Avoid using free-form when␣
↪calling module actions. (ansible.builtin.apt)
ansible-lint/examples/playbooks/example.yml:44: package-latest: Package installs should␣
```

(continues on next page)

(продолжение с предыдущей страницы)

```
↪not use latest.
ansible-lint/examples/playbooks/example.yml:47: name[missing]: All tasks should be named.
Read documentation for instructions on how to ignore specific rule violations.

                       Rule Violation Summary
count tag                       profile rule associated tags
    2 command-instead-of-module basic   command-shell, idiom
    2 jinja[spacing]            basic   formatting (warning)
    4 no-free-form              basic   syntax, risk
    1 schema[playbook]          basic   core
    2 name[missing]             basic   idiom
    1 yaml[indentation]         basic   formatting, yaml
    1 yaml[trailing-spaces]     basic   formatting, yaml
    3 latest[git]               safety  idempotency
    2 package-latest            safety  idempotency
    3 no-changed-when           shared  command-shell, idempotency
    3 args[module]                      syntax, experimental (warning)

Failed: 19 failure(s), 5 warning(s) on 1 files. Last profile that met the validation␣
↪criteria was 'min'.
```

Пример создания правила

Чтобы создать правила на основе методов `match` и `matchtask` и проверить с помощью этих правил сценарий `example.com`, выполните следующие действия:

1. Создайте каталог для правил и перейдите в него, например:

```
mkdir my_custom_rules && cd my_custom_rules
```

2. Создайте файл `TaskHasTag.py`, содержащий правило для проверки наличия тега:

```
from __future__ import annotations
from typing import TYPE_CHECKING, Union

from ansiblelint.rules import AnsibleLintRule

if TYPE_CHECKING:
    from ansiblelint.file_utils import Lintable
    from ansiblelint.utils import Task


class TaskHasTag(AnsibleLintRule):
    """Tasks must have tag."""

    id = 'EXAMPLE001'
    description = 'Tasks must have tag'
    tags = ['custom', 'core']

    def matchtask(self,
                  task: Task,
                  file: Lintable | None = None
                  ) -> Union[bool, str]:
        # Эта проверка исключает задачи include и fail из требований наличия тегов,
        # так как для них теги не обязательны:
        # Если задача включает другие задачи (include) или намеренно завершает␣
↪выполнение (fail),
        # она исключается из проверки на наличие тегов, возвращая False.
        if not set(task.keys()).isdisjoint(['include', 'fail']):
            return False

        if not task.get("tags"):
```

(continues on next page)

(продолжение с предыдущей страницы)

```
            return True
        return False
```

3. Создайте файл `TestRule.py`, содержащий правило для проверки наличия определенного слова:

```
from typing import Union
from ansiblelint.rules import AnsibleLintRule


class TestRule(AnsibleLintRule):
    """Deprecated variable declarations."""

    id = 'EXAMPLE002'
    description = 'This rule triggers on tasks containing the word "task".'
    tags = ['custom', 'core']

    def match(self, line: str) -> Union[bool, str]:
        # Срабатывает только для строк, содержащих "task" — для тестирования
        return 'task' in line
```

4. Запустите проверку сценария `example.com`:

```
ansible-lint \
  --offline \
  -r my_custom_rules/ \
  -t custom ~/ansible-lint/examples/playbooks/example.yml
```

Пример вывода:

```
EXAMPLE002: Deprecated variable declarations.
ansible-lint/examples/playbooks/example.yml:8   tasks:

EXAMPLE001: Tasks must have tag.
ansible-lint/examples/playbooks/example.yml:9 Task/Handler: Unset variable

EXAMPLE001: Tasks must have tag.
ansible-lint/examples/playbooks/example.yml:12 Task/Handler: Trailing whitespace

EXAMPLE001: Tasks must have tag.
ansible-lint/examples/playbooks/example.yml:15 Task/Handler: Run git check

EXAMPLE001: Tasks must have tag.
ansible-lint/examples/playbooks/example.yml:18 Task/Handler: Run git check 2

EXAMPLE001: Tasks must have tag.
ansible-lint/examples/playbooks/example.yml:21 Task/Handler: Run git check 3

EXAMPLE001: Tasks must have tag.
ansible-lint/examples/playbooks/example.yml:24 Task/Handler: Executing git through␣
↪command

EXAMPLE001: Tasks must have tag.
ansible-lint/examples/playbooks/example.yml:27 Task/Handler: Executing git through␣
↪command

EXAMPLE001: Tasks must have tag.
ansible-lint/examples/playbooks/example.yml:30 Task/Handler: Using git module

EXAMPLE001: Tasks must have tag.
ansible-lint/examples/playbooks/example.yml:33 Task/Handler: Passing git as an␣
↪argument to another task

EXAMPLE002: Deprecated variable declarations.
```

(continues on next page)

(продолжение с предыдущей страницы)

```
ansible-lint/examples/playbooks/example.yml:33     - name: Passing git as an␣
↪argument to another task

EXAMPLE001: Tasks must have tag.
ansible-lint/examples/playbooks/example.yml:39 Task/Handler: Dnf latest

EXAMPLE001: Tasks must have tag.
ansible-lint/examples/playbooks/example.yml:42 Task/Handler: debug msg=debug task␣
↪without a name

EXAMPLE002: Deprecated variable declarations.
ansible-lint/examples/playbooks/example.yml:42     - ansible.builtin.debug: msg=
↪"debug task without a name"

EXAMPLE001: Tasks must have tag.
ansible-lint/examples/playbooks/example.yml:44 Task/Handler: Run apt latest

EXAMPLE001: Tasks must have tag.
ansible-lint/examples/playbooks/example.yml:47 Task/Handler: meta flush_handlers

Read documentation for instructions on how to ignore specific rule violations.

            Rule Violation Summary
count tag        profile rule associated tags
   13 EXAMPLE001         custom, core
   3 EXAMPLE002         custom, core

Failed: 16 failure(s), 0 warning(s) on 1 files. Last profile that met the␣
↪validation criteria was 'production'. Rating: 5/5 star
```

Ansible Molecule

Утилита Ansible Molecule предназначена для облегчения разработки и тестирования ролей Ansible, что способствует повышению качества и надежности проектов автоматизации инфраструктуры. Ansible Molecule позволяет создавать рабочую среду для тестирования ролей (тестовую среду) в различных виртуальных средах, как облачных, так и локальных, построенных как на виртуальных машинах, так и на контейнерах.

Ключевые особенности и достоинства

Ansible Molecule имеет следующие ключевые особенности и достоинства:

- Позволяет создавать тестовые среды в различных виртуальных окружениях.

  Ansible Molecule позволяет тестировать роли Ansible в различных средах, включая множество поставщиков виртуализации и контейнеризации, операционные системы и дистрибутивы. Эта универсальность позволяет тестировать роли в разнообразных условиях.

- Используется как самостоятельный инструмент так и в составе тестовых платформ, что позволяет значительно расширить возможности Ansible Molecule путем запуска параллельных заданий, форматирования отчетов и использования других возможностей платформ.

- Ansible Molecule позволяет создавать несколько сценариев тестирования. Основу каждого из них составляет набор сценариев Ansible в виде файлов playbook. Тестирование в пределах одного сценария основано на выполнении определенной последовательности функциональных тестов в виде сценариев из Ansible playbooks.

  Разработчику тестов необходимо определить каждый сценарий (Ansible play) в виде набора задач (Ansible tasks), используя знакомые ему возможности Ansible. Можно запускать полную последовательность тестов или часть ее.

- Обеспечивает объявленную политику поддержки версий Ansible – поддерживает две последние основные версии Ansible. Эта политика гарантирует, что тестовая среда остается актуальной с учетом последних функций Ansible, сохраняя при этом обратную совместимость с предыдущей основной версией.
- Обеспечивает понятный синтаксис командной строки на базе утилиты `molecule`, которую можно также запускать как модуль Python `python3 -m molecule`.

  Утилита предоставляет подсказки как на глобальном уровне, так и по отдельным ее командам. Возможна настройка режима автодополнения.

Применение

Утилиту можно применять для выполнения следующих задач:

- Разработка новых ролей Ansible – предоставляет стандартизированную среду для разработки ролей с нуля, обеспечивая их соответствие стандартам качества.
- Тестирование существующих ролей Ansible – может помочь выявить проблемы совместимости с новыми версиями Ansible или в различных средах, которые не были предусмотрены изначально.
- Интеграция в CI/CD конвейеры – можно интегрировать в процессы непрерывной интеграции и развертывания для автоматизации тестирования ролей Ansible, делая процесс более надежным и эффективным.

Принцип работы

Ansible Molecule построена на основе модульной архитектуры, которая позволяет ей быть гибкой и расширяемой. Основными компонентами Ansible Molecule являются:

- Сценарии (Scenarios).

  Являются основными строительными блоками данного функционального тестирования. Они выстроены в определенную последовательность, описываемую далее.
- Драйверы (Drivers или Providers).

  Отвечают за создание, управление и удаление тестовых сред. Примеры драйверов: Docker, Podman, Vagrant, Libvirt, LXC, EC2.
- Исполнитель сценариев (Provisioners).

  Используются для исполнения сценариев в созданных тестовых средах. По умолчанию Ansible Molecule использует Ansible в качестве исполнителя.
- Средства проверки (Verifiers).

  Инструменты для контроля успешности выполнения сценариев. Позволяет использовать Testinfra, Goss и Ansible.

Схема работы Ansible Molecule

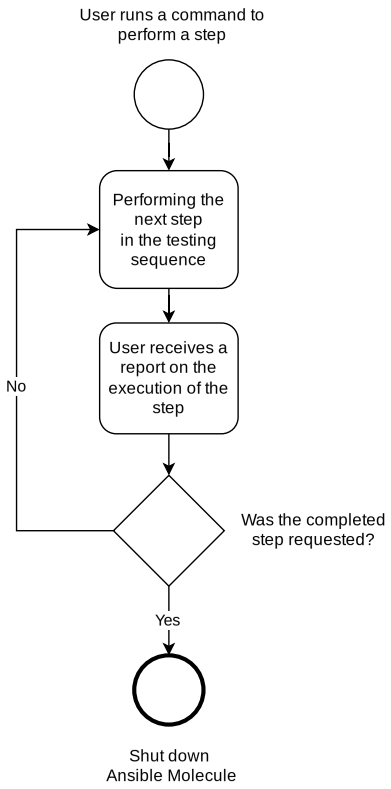
Ansible Molecule выполняет тестирование ролей с помощью последовательности этапов. На каждом этапе выполняется определенный сценарий. Название этапа и файла сценария (playbook) соответствуют друг другу. Например, на этапе `destroy` выполняется сценарий из файла `destroy.yml`.

1. *destroy* удаляет предыдущую тестовую среду, если она существует, чтобы обеспечить отсутствие изменений, сделанных на предыдущих запусках тестирования.
2. *dependency* устанавливает зависимости роли, например, указанные в файле `requirements.yml`.
3. *syntax* проверяет синтаксис сценариев и файлов роли.

4. *create* создает тестовую среду с использованием выбранного драйвера.
5. *prepare* выполняет дополнительные подготовительные задачи, если они определены.
6. *converge* применяет роль к тестовой среде с помощью Ansible.
7. *idempotence* проверяет идемпотентность роли, применяя ее повторно и ожидая отсутствия изменений.
8. *verify* выполняет тесты для проверки корректности работы роли.
9. *cleanup* выполняет задачи очистки, если они определены.
10. *destroy* удаляет тестовую среду после завершения тестирования.

Этапы осуществляются в строгой последовательности, после выполнения запрошенного этапа работа утилиты завершается.

Процесс тестирования показан на схеме:



На локальном узле выполняются следующие этапы:

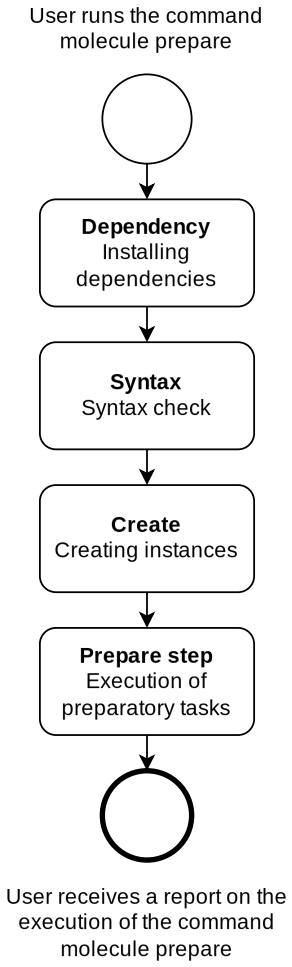
- *dependency*;
- *syntax*;
- *create*.

После создания тестовой среды в ней выполняются следующие этапы:

- *prepare*;
- *converge*;
- *idempotence*;
- *verify*;
- *cleanup*;
- *destroy*.

Если пользователь запрашивает выполнение определенного этапа, например, *prepare*, утилита выполнит также все предшествующие ему этапы.

Например, процесс выполнения этапа *prepare* показан на схеме:



Тестовые среды

В Ansible Molecule тестовые среды определяются драйвером. Их можно разделить на несколько типов в зависимости от того, какие инструменты используются:

- Контейнеры.

  При использовании драйверов типа Docker или Podman Ansible Molecule создает отдельные контейнеры для каждой тестовой платформы, определенной в `molecule.yml`. Если необходимо протестировать роль на нескольких операционных системах, Molecule создаст по одному контейнеру для каждой из них. Один контейнер используется для всех этапов (*create*, *converge*, *verify*) в рамках одного сценария. Это означает, что контейнер создается на этапе *create* и уничтожается на этапе *destroy*, а все остальные действия выполняются внутри этого контейнера.

- Виртуальные машины.

  При использовании драйверов типа Vagrant или Libvirt, Ansible Molecule создает виртуальные машины для каждой платформы.

- Виртуальные окружения.

  Ansible Molecule по умолчанию не использует виртуальные окружения Python (venv) внутри тестовых сред. Однако, если роль требует использования пакетов Python или виртуальных окружений на целевых узлах, вы можете настроить это в своих задачах Ansible.

Установка

Для установки Ansible Molecule выполните следующие действия:

1. Подключите репозиторий Astra Automation.

   #### Инструкция по подключению репозитория

   1. В каталоге `/etc/apt/sources.list.d/` создайте файл `astra-automation.list` со ссылкой на репозиторий Astra Automation:

      ```
      deb https://dl.astralinux.ru/aa/aa-debs-for-alse-1.7 <version> main
      ```

      Вместо <version> необходимо подставить версию устанавливаемой платформы, например, `1.1-upd1`.

      Доступные версии продукта опубликованы в таблице *История обновлений*.

   2. Обновите список доступных пакетов:

      ```
      sudo apt update
      ```

2. Установите пакет `molecule`:

   ```
   sudo apt install molecule --yes
   ```

Настройка

Ansible Molecule можно настроить с помощью следующих компонентов:

- *команды*;
- *файл настроек*.

При вычислении действующих значений аргументов используется следующий порядок (значения, указанные на более поздних этапах, заменяют значения, указанные ранее):

1. Файл настроек.
2. Команды.

По умолчанию Ansible Molecule начинает поиск файла настроек с текущего каталога. Если в каталоге несколько сценариев, утилита проверяет структуру подкаталогов. Если указан аргумент `--scenario-name`, утилита ищет файл `molecule.yml` в соответствующем каталоге сценария.

Параметры в файле настроек задаются следующим образом:

Список 12: molecule.yml

```
---
dependency:
  name: galaxy
  options:
    requirements-file: requirements.yml

platforms:
  - name: molecule-astra
    image: hub.astra-automation.ru/aa/aa-base-ee
    privileged: true
    groups:
      - all

driver:
  options:
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    managed: false
    login_cmd_template: "podman exec -ti {instance} bash"
    ansible_connection_options:
      ansible_connection: podman
```

Подробное описание параметров см. в *справочнике*.

Пример

В этом примере рассматривается создание роли Ansible, которая получает текущую версию Ansible и записывает ее в файл `/tmp/ansible_version_output.txt`. В качестве тестовой среды используется контейнер, запускаемый с помощью Podman. Подробное описание команд и возможных аргументов утилиты см. в *справочнике*.

Создание структуры роли

Для создания структуры роли выполните следующие действия:

1. Создайте каталог для роли и перейдите в него:

   ```
   mkdir ansible_version_role && cd ansible_version_role
   ```

2. Создайте сценарий:

   ```
   molecule init scenario
   ```

   Эта команда создаст структуру файлов и каталогов для сценария с поддержкой драйвера по умолчанию (`default`) в каталоге текущей роли.

3. Создайте каталог `tasks` для хранения задач роли:

   ```
   mkdir tasks
   ```

4. В каталоге `tasks` создайте файл `main.yml`.

В результате выполнения действий получится следующая структура:

```
ansible_version_role/
├── molecule
│   └── default
│       ├── converge.yml
│       ├── create.yml
│       ├── destroy.yml
│       ├── molecule.yml
│       ├── requirements.yml
│       └── verify.yml
└── tasks
    └── main.yml
```

Заполнение файлов роли

`molecule.yml` – файл настроек утилиты. Пример заполнения файла:

```yaml
---
dependency:
  name: galaxy
  options:
    requirements-file: requirements.yml

platforms:
  - name: molecule-astra
    image: hub.astra-automation.ru/aa/aa-base-ee
    privileged: true
    groups:
      - all

driver:
  options:
    managed: false
    login_cmd_template: "podman exec -ti {instance} bash"
    ansible_connection_options:
      ansible_connection: podman

provisioner:
  name: ansible
  config_options:
    defaults:
      remote_tmp: '/tmp/.ansible-${USER}/tmp'
  playbooks:
    create: create.yml
    converge: converge.yml
    destroy: destroy.yml
    verify: verify.yml
  inventory:
    group_vars:
      all:
        ansible_python_interpreter: /usr/bin/python3.9  # Версия Python контейнера
        ansible_user: '1000'  # Пользователь в контейнере
```

Файл `converge.yml` используется для запуска роли, которая выполняет задачу получения версии Ansible. Пример заполнения файла:

```yaml
- name: Converge
  hosts: all
  become: false
  gather_facts: false
  roles:
    - ansible_version_role
```

Файл `create.yml` отвечает за создание контейнера. В этом примере для создания контейнера используется модуль `containers.podman.podman_container`:

```yaml
- name: Create
  hosts: localhost
  gather_facts: false

  tasks:
    - name: Create a container
      containers.podman.podman_container:
        name: "{{ item.name }}"
        image: "{{ item.image }}"
        privileged: "{{ item.privileged | default(omit) }}"
        volumes: "{{ item.volumes | default(omit) }}"
        capabilities: "{{ item.capabilities | default(omit) }}"
```

(continues on next page)

(продолжение с предыдущей страницы)

```
        systemd: "{{ item.systemd | default(omit) }}"
        state: started
        command: "{{ item.command | default('sleep 1d') }}"
        log_driver: json-file
      register: result
      loop: "{{ molecule_yml.platforms }}"

    - name: Create remote_tmp directory
      ansible.builtin.file:
        path: '/tmp/.ansible-{{ ansible_user }}/tmp'
        state: directory
        mode: '0755'
```

Файл `destroy.yml` отвечает за удаление контейнера по окончании тестов. Пример заполнения:

```
- name: Destroy molecule containers
  hosts: localhost
  gather_facts: false
  tasks:
    - name: Stop and remove container
      containers.podman.podman_container:
        name: molecule-astra
        state: absent
        rm: true
```

В файле `verify.yml` содержится задача проверки существования файла с версией Ansible в тестовом окружении. Пример заполнения:

```
- name: Verify Ansible version output
  hosts: all
  tasks:
    - name: Check if ansible version output contains 'ansible'
      ansible.builtin.command: "cat /tmp/ansible_version_output.txt"
      register: ansible_version_check
      failed_when: "'ansible' not in ansible_version_check.stdout"
      changed_when: false
```

В файле `requirements.yml` указываются зависимости Ansible. В данном примере файл указывает на необходимость использования коллекции `containers.podman`:

```
collections:
  - containers.podman
```

В файле `main.yml` указываются задачи, которые выполняет роль. В данном примере роль выполняет команду `ansible  --version` и сохраняет результат в файл `/tmp/ansible_version_output.txt`.

```
- name: Get Ansible version
  ansible.builtin.command: ansible --version
  register: ansible_version_output
  changed_when: false

- name: Save Ansible version output to file in /tmp directory
  ansible.builtin.copy:
    content: "{{ ansible_version_output.stdout }}"
    dest: /tmp/ansible_version_output.txt
```

Запуск тестов

Чтобы запустить полный цикл тестирования, выполните команду:

```
molecule test
```

Запуск этой команды приведет к выполнению следующих этапов тестирования:

1. `create`;
2. `converge`;
3. `verify`;
4. `destroy`.

Для подключения к контейнеру и ручной проверки выполните команду:

```
molecule login
```

8.2 Среда исполнения

Среда исполнения Astra Automation (Execution Environment, EE) – это контейнер с образом на базе Astra Linux Special Edition, который рекомендуется использовать для развертывания и тестирования кода, запускаемого с помощью Ansible.

Инструментом для работы с образом может быть *Podman* (рекомендуется) или Docker.

Предупреждение: Для запуска контейнера с указанным образом необходима версия Docker не ниже 19.03.

Использование EE дает следующие преимущества:

- Нет необходимости устанавливать дополнительное ПО: все необходимое уже есть в составе образа.
- Не требуется ручная настройка окружения.
- Контейнер с образом EE может быть запущен в любой ОС, поддерживаемой Podman.

Совет: Актуальный список версий образа EE доступен в реестре Astra Automation[170].

8.2.1 Загрузка образа

Для загрузки образа среды исполнения используйте команду:

```
podman pull registry.astralinux.ru/aa/aa-base-ee:<version>
```

Здесь <version> – версия образа. Если версия не указана, загружается образ с меткой `latest`, которой соответствует новейшая стабильная версия образа.

Совет: С помощью команды `tag` образу можно задать дополнительное название, например:

```
podman tag registry.astralinux.ru/aa/aa-base-ee aa-base-ee
```

[170] https://registry.astralinux.ru/browse/aa/#!/taglist/aa-base-ee

Теперь к образу registry.astralinux.ru/aa/aa-base-ee:latest можно обращаться через aa-base-ee.

Для получения состава образа используйте команду:

```
ansible-navigator images <tag>
```

Здесь <tag> – название образа.

Например, для получения состава образа aa-base-ee:0.6.2 выполните следующие действия:

1. Загрузите образ среды исполнения:

   ```
   podman pull registry.astralinux.ru/aa/aa-base-ee:0.6.2
   ```

2. Получите состав образа:

   ```
   ansible-navigator images registry.astralinux.ru/aa/aa-base-ee:0.6.2
   ```

Примечание: С подробной информацией о составе рекомендуемых образов можно ознакомиться в *документации*.

8.2.2 Использование

Для выполнения определенной команды внутри контейнера используйте следующий синтаксис:

```
ansible-navigator exec \
   --eei registry.astralinux.ru/aa/aa-base-ee:<version> \
   -- <container_command> [options]
```

Здесь:

- <ansible_command> – команда Ansible.
- --eei – название образа среды исполнения.
- <container_command> – команда, которая должна быть выполнена в контейнере.

8.2.3 Примеры

Следующие примеры демонстрируют использование EE для выполнения различных команд.

Проверка доступности узлов

Для проверки доступности всех узлов инвентаря выполните в каталоге проекта команду:

```
ansible-navigator exec -- ansible all -m ping
```

Убедитесь, что по каждому узлу получен положительный ответ типа pong, например:

```
web01 | SUCCESS => {
   "ansible_facts": {
      "discovered_interpreter_python": "/usr/bin/python3.7"
   },
   "changed": false,
   "ping": "pong"
}
```

Установка зависимостей Ansible

Для установки необходимых ресурсов, согласно зависимостям, определенным в файле `requirements.yml`, используйте команду:

```
ansible-navigator exec \
   --eei registry.astralinux.ru/aa/aa-base-ee \
   -- ansible-galaxy install -r requirements.yml
```

Здесь:

- `registry.astralinux.ru/aa/aa-base-ee` – название образа среды исполнения;
- `ansible-galaxy` – утилита Ansible, используемая для управления зависимостями;
- `requirements.yml` – название файла с перечнем зависимостей Ansible.

Запуск playbook

Для запуска playbook используйте команду:

```
ansible-navigator run playbook.yml \
   -i inventory.yml \
   --eei registry.astralinux.ru/aa/aa-base-ee \
   -m stdout
```

Здесь:

- `registry.astralinux.ru/aa/aa-base-ee` – название образа среды исполнения;
- `inventory.yml` – файл инвентаря;
- `playbook.yml` – файл playbook.

8.3 Разработка коллекций

Использование Ansible Navigator и Ansible Creator для разработки контента.

Важно: Эта часть документации находится в стадии разработки.

8.4 Проверка стиля и синтаксиса

Использование Ansible Linter для проверки стиля и синтаксиса коллекций и playbook.

Важно: Эта часть документации находится в стадии разработки.

8.5 Тестирование

Использование Molecule и PyTest Ansible для тестирования контента.

8.5.1 Тестирование с помощью Pytest Ansible

Система тестирования PyTest[171] используется для тестирования модулей, написанных на языке программирования Python. Пакет `pytest-ansible` расширяет возможности PyTest и предоставляет следующие возможности:

- модульное тестирование коллекций Ansible;
- использование утилиты Molecule;
- интеграция Ansible с тестами PyTest.

Пример использования

В этом примере рассматривается создание теста, проверяющего работоспособность коллекции cisco.ios[172].

Для проверки работы PyTest Ansible выполните следующие действия:

1. Клонируйте исходный код коллекций:
   - `cisco.ios`:

     ```
     git clone https://github.com/ansible-collections/cisco.ios.git collections/
     ↪ansible_collections/cisco/ios
     ```

   - `ansible.netcommon`:

     ```
     git clone https://github.com/ansible-collections/ansible.netcommon.git␣
     ↪collections/ansible_collections/ansible/netcommon
     ```

     Примечание: Коллекция `ansible.netcommon` является зависимостью коллекции `cisco.ios`.

2. Перейдите в каталог с исходным кодом коллекции `cisco.ios`:

   ```
   cd collections/ansible_collections/cisco/ios
   ```

3. Для запуска тестов выполните команду:

   ```
   pytest tests
   ```

   Если тест пройден успешно, рядом с его названием в терминале выводится сообщение PASSED, например:

   ```
   [gw0] [ 90%] PASSED tests/unit/modules/network/ios/test_ios_vrf_global.
   ↪py::TestIosVrfGlobalModule::test_ios_vrf_global_replaced
   ```

[171] https://docs.pytest.org/en/stable/index.html
[172] https://docs.ansible.com/ansible/latest/collections/cisco/ios/index.html

8.5.2 Тестирование с помощью Molecule

Важно: Эта часть документации находится в стадии разработки.

8.6 Устранение неисправностей

В этом разделе описано использование Ansible Navigator для проверки и поиска неисправностей в контенте Ansible, включая:

- инвентарные списки;
- наборы playbook;
- коллекции;
- документацию;
- образы *среды исполнения*.

8.6.1 Настройки

Важно: Эта часть документации находится в стадии разработки.

8.6.2 Команды

Важно: Эта часть документации находится в стадии разработки.

8.6.3 Примеры использования

Важно: Эта часть документации находится в стадии разработки.

8.7 Контроль целостности

Использование Ansible Sign для контроля подлинности и целостности контента.

Важно: Эта часть документации находится в стадии разработки.

8.8 Справочные данные

Здесь приведены:

- справочные данные по назначению и применению утилит для создания и управления контентом Ansible;
- сведения о версиях компонентов в образах среды исполнения, созданных ПАО Группа Астра.

8.8.1 Ansible Navigator

Утилита Ansible Navigator является универсальным инструментом для разработки, тестирования и применения инфраструктурного кода в виде коллекций и playbook. Для этого она предоставляет несколько команд с различными аргументами.

Вызов утилиты имеет следующий вид:

```
ansible-navigator {<command>} {<arguments>}
```

Для понимания принципов работы утилиты обратитесь к ее *описанию*.

Общие аргументы

Ansible Navigator принимает аргументы, общие для всех команд, а также специфичные для утилит, которые используются для выполнения отдельных задач.

Аргументы, общие для всех команд:

`-h, --help`
: Вывод справочной информации о командах и аргументах Ansible Navigator.

`--version`
: Вывод информации о версии Ansible Navigator.

`--rad, --ansible-runner-artifact-dir`
: Путь к каталогу для хранения артефактов, созданных во время запуска `ansible-runner`.

`--rac, --ansible-runner-rotate-artifacts-count`
: Удаление каталогов с артефактами, кроме последних N запусков `ansible-runner`. При значении `0` не удаляется ни один каталог с артефактами.

`--rt, --ansible-runner-timeout`
: Период ожидания в секундах перед принудительным завершением выполнения `ansible-runner`.

`--rwje, --ansible-runner-write-job-events`
: Запись событий `job_events` в каталог артефактов:

  - `true` – запись включена;
  - `false` – запись выключена.

`--cdcp, --collection-doc-cache-path`
: Путь к каталогу для хранения кеша документации коллекций.

  Значение по умолчанию – `~/.cache/ansible-navigator/collection_doc_cache.db`.

`--ce, --container-engine`
: Инструмент контейнеризации:

- `auto` – тип инструмента контейнеризации выбирается автоматически.

  Примечание: Если в системе установлены Docker и Podman, будет использоваться Podman.

- `podman` – Podman.
- `docker` – Docker.

Значение по умолчанию – `auto`.

`--co`, `--container-options`

Дополнительные параметры, передаваемые в команду запуска инструмента контейнеризации.

`--dc`, `--display-color`

Использование цветового оформления:

- `true` – включено;
- `false` – выключено.

Значение по умолчанию – `true`.

`--ecmd`, `--editor-command`

Команда запуска текстового редактора.

Значение по умолчанию – `vi +{line_number} {filename}`.

`--econ`, `--editor-console`

Использование редактора в режиме *TUI*:

- `true` – текстовый интерфейс;
- `false` – режим работы определяется редактором.

Значение по умолчанию – `true`.

`--ee`, `--execution-environment`

Использование среды исполнения:

- `true` – разрешено.
- `false` – запрещено. Выполнение команд происходит с использованием компонентов, доступных в ОС.

Значение по умолчанию – `true`.

`--eei`, `--execution-environment-image`

Название образа среды исполнения.

Значение по умолчанию – `registry.astralinux.ru/aa/aa-creator-ee:0.2.1`.

`--eev` `<volume>:<mount_point>`,
`--execution-environment-volume-mounts` `<volume>:<mount_point>`

Монтирование тома <volume> в точку <mount_pount> контейнера при запуске среды исполнения.

Подробности о монтировании томов см. в документации используемой системы контейнеризации:

- Docker[173];
- Podman[174].

[173] https://docs.docker.com/engine/storage/volumes/
[174] https://docs.podman.io/en/latest/markdown/podman-volume-mount.1.html

--la, --log-append

Управление записью журнала сообщений в файл:

- `true` – для хранения всех записей используется один файл журнала.
- `false` – для каждой сессии создается отдельный файл журнала.

Значение по умолчанию – `true`.

--lf, --log-file

Путь к файлу хранения журнала Ansible Navigator.

Значение по умолчанию – `~/ansible-navigator.log`.

--ll, --log-level

Минимальный уровень важности сообщения, необходимый для его записи в журнал Ansible Navigator.

Возможные значения в порядке возрастания важности:

1. `debug`;
2. `info`;
3. `warning`;
4. `error`;
5. `critical`.

Значение по умолчанию – `warning`.

-m, --mode

Режим работы Ansible Navigator:

- `interactive` – интерактивный режим;
- `stdout` – неинтерактивный режим.

Значение по умолчанию – `interactive`.

--osc4

Использование в терминале цветового оформления в соответствии со стандартом OSC 4:

- `true` – включено;
- `false` – выключено.

Значение по умолчанию – `true`.

--penv, --pass-environment-variable

Передача существующей переменной окружения в среду исполнения.

--pa=[<params>], **--pull-arguments**=[<params>]

Дополнительные параметры для команды получения (pull) образа среды исполнения из реестра, например:

```
--pull-arguments='--tls-verify=false'
```

--pp, --pull-policy

Политика получения образа среды исполнения:

- `always` – всегда загружать образ из реестра.
- `missing` – загрузить образ из реестра только в том случае, когда нет его локальной копии.
- `never` – никогда не загружать образ из реестра.
- `tag` – если значение тега равно `latest`, образ будет загружен из реестра. Если значение тега отличается от `latest`, образ будет загружен из реестра только при отсутствии локальной копии.

Значение по умолчанию – tag.

--senv <variable>=<value>, **--set-environment-variable** <variable>=<value>

Передача в среду исполнения переменной окружения <varibable> со значением <value>.

--tz, --time-zone

Используемый часовой пояс:

- название часового пояса в формате IANA[175];
- local.

Значение по умолчанию – utc.

Команды

Здесь представлено описание команд Ansible Navigator. Для получения справочной информации об использовании связанных команд Ansible, Podman и Docker обратитесь к их документации.

builder

Синтаксис вызова команды builder:

```
ansible-navigator builder [arguments] <action>
```

Аргументы:

--hb, --help-builder

Вывод в stdout справочной информации по использованию команды ansible-builder:

- true – включен;
- false – выключен.

Значение по умолчанию – false.

--bwd, --workdir

Путь к каталогу, содержащему файл с манифестом образа среды исполнения.

Значение по умолчанию – текущий каталог.

collections

Синтаксис вызова команды collection:

```
ansible-navigator collections [arguments]
```

Аргументы:

--fmt, --format

Формат вывода сведений о коллекциях:

- json;
- yaml.

Значение по умолчанию – yaml.

[175] https://www.iana.org/time-zones

config

Синтаксис вызова команды config:

```
ansible-navigator config [arguments]
```

Аргументы:

-c, --config
Путь к файлу с настройками Ansible.

--hc, --help-config
Вывод в stdout справочной информации по использованию команды ansible-config:

- true – включен;
- false – выключен.

Значение по умолчанию – false.

doc

Синтаксис вызова команды doc:

```
ansible-navigator doc {<plugin>} [arguments]
```

Аргументы:

--hd, --help-doc
Вывод в stdout справочной информации по использованию команды ansible-doc:

- true – включен;
- false – выключен.

Значение по умолчанию – false.

<plugin>
Название расширения.

-t, --type
Тип расширения:

- become;
- cache;
- callback;
- cliconf;
- connection;
- filter;
- httpapi;
- inventory;
- keyword;
- lookup;
- module;
- netconf;
- role;
- shell;
- strategy;

- test;
- vars.

Значение по умолчанию – module.

exec

Синтаксис вызова команды exec:

```
ansible-navigator exec {<command>} [options]
```

Аргументы:

<command>

Запускаемая команда.

Значение по умолчанию – /bin/bash.

--exshell, --exec-shell

Запуск команды в оболочке командной строки.

Значение по умолчанию – true.

images

Сиктаксис вызова команды images:

```
ansible-navigator images [options]
```

Аргументы:

--fmt, --format

Формат вывода сообщений:

- json;
- yaml.

Значение по умолчанию – yaml.

-d, --details

Вывод подробных сведений о компонентах среды исполнения по умолчанию:

- ansible_collections – сведения о коллекциях;
- ansible_version – версии компонентов Ansible;
- everything – сведения обо всех компонентах образа среды исполнения;
- os_release – версия ОС;
- python_packages – версии пакетов Python;
- python_version – версия интерпретатора Python;
- system_packages – версии системных пакетов.

Значение по умолчанию – everything.

inventory

Синтаксис вызова команды `inventory`:

```
ansible-navigator inventory [options]
```

Аргументы:

--hi, --help-inventory

Вывод в stdout справочной информации по использованию команды `ansible-inventory`:

- `true` – включен;
- `false` – выключен.

Значение по умолчанию – `false`.

-i, --inventory

Путь к файлу инвентаря или список узлов.

При передаче списка узлов их названия разделяются запятыми, например:

```
ansible-navigator inventory node1.example.com,node2.example.com
```

--ic <column>, **--inventory-column** <column>

Отображение в таблице сведений об управляемых узлах дополнительной колонки со значением указанного атрибута, например:

```
ansible-navigator inventory -i inventory.ini --ic ansible_user
```

При запуске Ansible Navigator можно передать несколько значений аргумента, например:

```
ansible-navigator inventory -i inventory.ini \
   --ic ansible_user \
   --ic ansible_host
```

lint

Синтаксис вызова команды `lint`:

```
ansible-navigator lint [options] [<lintables>]
```

Аргументы:

--lic, --lint-config

Путь к файлу с настройками `ansible-lint`.

[<lintables>]

Путь к файлам, которые должны быть проверены с помощью `ansible-lint`.

По умолчанию поиск файла с настройками `ansible-lint` и проверяемых файлов выполняется в текущем каталоге.

replay

Синтаксис вызова команды:

```
ansible-navigator replay [<playbook_artifact>]
```

Аргументы:

<playbook_artifact>
Путь к файлу с артефактами, собранными при выполнении playbook.

run

Синтаксис вызова команды:

```
ansible-navigator run <playbook> [options]
```

Аргументы:

--ep, --enable-prompts
Интерактивный ввод пароля при выполнении playbook:

- `true` – разрешен;
- `false` – запрещен.

Значение по умолчанию – `true`.

Примечание: Использование этого аргумента перенаправляет вывод в `stdout` и запрещает создание артефактов.

--hp, --help-playbook
Вывод в `stdout` справочной информации по использованию команды `ansible-playbook`:

- `true` – включен;
- `false` – выключен.

Значение по умолчанию – `false`.

-i, --inventory
Путь к файлу инвентаря или список узлов.

При передаче списка узлов их названия разделяются запятыми, например:

```
ansible-navigator run playbook.yml -i node1.example.com,node2.example.com
```

<playbook>
Путь к файлу playbook.

--pae, --playbook-artifact-enable
Создание артефактов для выполненного playbook:

- `true` – разрешено;
- `false` – запрещено.

Значение по умолчанию – `true`.

Примечание: Если playbook использует интерактивный ввод данных пользователем, использование этого аргумента несовместимо со значением `stdout` для аргумента `--mode`.

--pas, --playbook-artifact-save-as

Шаблоны имен файлов для хранения артефактов, созданных при выполнении playbook.

При формировании имен файлов можно использовать следующие подстановочные блоки:

- `{playbook_dir}` – путь к каталогу с playbook;
- `{playbook_name}` – название playbook;
- `{playbook_status}` – статус выполнения playbook;
- `{time_stamp}` – время завершения выполнения playbook.

Значение по умолчанию – `{playbook_dir}/{playbook_name}-artifact-{time_stamp}.json`.

settings

Синтаксис вызова команды:

```
ansible-navigator settings [options]
```

Аргументы:

--se, --effective

Вывод сведений о действующих значениях настроек.

Действующий значения определяются путем комбинирования значений по умолчанию, аргументов командной строки, переменных окружения и значений, заданных в конфигурационном файле.

--gs, --sample

Создание примера файла настроек.

--ss `json`**, --schema** `json`

Создание примера схемы для файла настроек.

Значение по умолчанию – `json`.

--so, --sources

Вывод сведений об источнике, из которого получено значение каждой настройки.

welcome

Синтаксис вызова команды:

```
ansible-navigator welcome [options]
```

Настройки

Настройки Ansible Navigator делятся на две категории:

- основные – управляют работой самого Ansible Navigator;
- настройки субкоманд – управляют поведением других утилит, запускаемых с помощью Ansible Navigator.

Основные настройки

- `ansible-runner-artifact-dir`

  Default: not set

  CLI: `--rad` or `--ansible-runner-artifact-dir`

  ENV: `ANSIBLE_NAVIGATOR_ANSIBLE_RUNNER_ARTIFACT_DIR`
- `ansible-runner-rotate-artifacts-count`

  Default: not set

  CLI: `--rac` or `--ansible-runner-rotate-artifacts-count`

  ENV: `ANSIBLE_NAVIGATOR_ANSIBLE_RUNNER_ROTATE_ARTIFACTS_COUNT`
- `ansible-runner-timeout`

  Default: not set

  CLI: `--rt` or `--ansible-runner-timeout`

  ENV: `ANSIBLE_NAVIGATOR_ANSIBLE_RUNNER_TIMEOUT`
- `ansible-runner-write-job-events`

  Availiable values:

  - `True`
  - `False`

  Default: `False`

  CLI: `--rwje` or `--ansible-runner-write-job-events`

  ENV: `ANSIBLE_NAVIGATOR_ANSIBLE_RUNNER_WRITE_JOB_EVENTS`
- `app`

  Availiable values:

  - `builder`
  - `collections`
  - `config`
  - `doc`
  - `exec`
  - `images`
  - `inventory`
  - `lint`
  - `replay`
  - `run`
  - `settings`
  - `welcome`

  Default: `welcome`

  CLI: positional

  ENV: `ANSIBLE_NAVIGATOR_APP`
- `cmdline`

  Default: not set

  CLI: positional

ENV: ANSIBLE_NAVIGATOR_CMDLINE

- collection-doc-cache-path

  Default: ~/.cache/ansible-navigator/collection_doc_cache.db

  CLI: --cdcp or --collection-doc-cache-path

  ENV: ANSIBLE_NAVIGATOR_COLLECTION_DOC_CACHE_PATH

- container-engine

  Availiable values:

  - auto
  - podman
  - docker

  Default: auto

  CLI: --ce or --container-engine

  ENV: ANSIBLE_NAVIGATOR_CONTAINER_ENGINE

- container-options

  Default: not set

  CLI: --co or --container-options

  ENV: ANSIBLE_NAVIGATOR_CONTAINER_OPTIONS

- display-color

  Availiable values:

  - True
  - False

  Default: True

  CLI: --dc or --display-color

  ENV: NO_COLOR

- editor-command

  Default: vi +{line_number} {filename}

  CLI: --ecmd or --editor-command

  ENV: ANSIBLE_NAVIGATOR_EDITOR_COMMAND

- editor-console

  Availiable values:

  - True
  - False

  Default: True

  CLI: --econ or --editor-console

  ENV: ANSIBLE_NAVIGATOR_EDITOR_CONSOLE

- execution-environment

  Availiable values:

  - True
  - False

Default: True

CLI: --ee or --execution-environment

ENV: ANSIBLE_NAVIGATOR_EXECUTION_ENVIRONMENT

- execution-environment-image

  Default: None

  CLI: --eei or --execution-environment-image

  ENV: ANSIBLE_NAVIGATOR_EXECUTION_ENVIRONMENT_IMAGE

- execution-environment-volume-mounts

  Default: not set

  CLI: --eev or --execution-environment-volume-mounts

  ENV: ANSIBLE_NAVIGATOR_EXECUTION_ENVIRONMENT_VOLUME_MOUNTS

- log-append

  Availiable values:

  - True
  - False

  Default: True

  CLI: --la or --log-append

  ENV: ANSIBLE_NAVIGATOR_LOG_APPEND

- log-file

  Default: ./ansible-navigator.log

  CLI: --lf or --log-file

  ENV: ANSIBLE_NAVIGATOR_LOG_FILE

- log-level

  Availiable values:

  - debug
  - info
  - warning
  - error
  - critical

  Default: warning

  CLI: --ll or --log-level

  ENV: ANSIBLE_NAVIGATOR_LOG_LEVEL

- mode

  Availiable values:

  - stdout
  - interactive

  Default: interactive

  CLI: -m or --mode

  ENV: ANSIBLE_NAVIGATOR_MODE

- osc4

  Availiable values:

  - True
  - False

  Default: True

  CLI: --osc4 or --osc4

  ENV: ANSIBLE_NAVIGATOR_OSC4

- pass-environment-variable

  Default: not set

  CLI: --penv or --pass-environment-variable

  ENV: ANSIBLE_NAVIGATOR_PASS_ENVIRONMENT_VARIABLES

- pull-arguments

  Default: not set

  CLI: --pa or --pull-arguments

  ENV: ANSIBLE_NAVIGATOR_PULL_ARGUMENTS

- pull-policy

  Availiable values:

  - always
  - missing
  - never
  - tag

  Default: tag

  CLI: --pp or --pull-policy

  ENV: ANSIBLE_NAVIGATOR_PULL_POLICY

- set-environment-variable

  Default: not set

  CLI: --senv or --set-environment-variable

  ENV: ANSIBLE_NAVIGATOR_SET_ENVIRONMENT_VARIABLES

- time-zone

  Default: UTC

  CLI: --tz or --time-zone

  ENV: TZ

Subcommand builder

- help-builder

  Availiable values:

  - True
  - False

  Default: False

  CLI: --hb or --help-builder

  ENV: ANSIBLE_NAVIGATOR_HELP_BUILDER

- `workdir`

  Default: .

  CLI: `--bwd` or `--workdir`

  ENV: `ANSIBLE_NAVIGATOR_WORKDIR`

Subcommand `collections`

- `format`

  Availiable values:

  - `json`
  - `yaml`

  Default: `yaml`

  CLI: `--fmt` or `--format`

  ENV: `ANSIBLE_NAVIGATOR_FORMAT`

Subcommand `config`

- `config`

  Default: not set

  CLI: `-c` or `--config`

  ENV: `ANSIBLE_CONFIG`
- `help-config`

  Availiable values:

  - `True`
  - `False`

  Default: `False`

  CLI: `--hc` or `--help-config`

  ENV: `ANSIBLE_NAVIGATOR_HELP_CONFIG`

Subcommand `doc`

- `help-doc`

  Availiable values:

  - `True`
  - `False`

  Default: `False`

  CLI: `--hd` or `--help-doc`

  ENV: `ANSIBLE_NAVIGATOR_HELP_DOC`
- `plugin-name`

  Default: not set

  CLI: positional

  ENV: `ANSIBLE_NAVIGATOR_PLUGIN_NAME`

- `plugin-type`

  Availiable values:

  - `become`
  - `cache`
  - `callback`
  - `cliconf`
  - `connection`
  - `filter`
  - `httpapi`
  - `inventory`
  - `keyword`
  - `lookup`
  - `module`
  - `netconf`
  - `role`
  - `shell`
  - `strategy`
  - `test`
  - `vars`

  Default: `module`

  CLI: `-t` or `--type`

  ENV: `ANSIBLE_NAVIGATOR_PLUGIN_TYPE`

Subcommand exec

- `exec-command`

  Default: `/bin/bash`

  CLI: positional

  ENV: `ANSIBLE_NAVIGATOR_EXEC_COMMAND`

- `exec-shell`

  Availiable values:

  - `True`
  - `False`

  Default: `True`

  CLI: `--exshell` or `--exec-shell`

  ENV: `ANSIBLE_NAVIGATOR_EXEC_SHELL`

Subcommand `images`

- `format`

  Availiable values:

  - `json`
  - `yaml`

  Default: `yaml`

  CLI: `--fmt` or `--format`

  ENV: `ANSIBLE_NAVIGATOR_FORMAT`
- `images-details`

  Availiable values:

  - `ansible_collections`
  - `ansible_version`
  - `everything`
  - `os_release`
  - `python_packages`
  - `python_version`
  - `system_packages`

  Default: `['everything']`

  CLI: `-d` or `--details`

  ENV: `ANSIBLE_NAVIGATOR_IMAGES_DETAILS`

Subcommand `inventory`

- `help-inventory`

  Availiable values:

  - `True`
  - `False`

  Default: `False`

  CLI: `--hi` or `--help-inventory`

  ENV: `ANSIBLE_NAVIGATOR_HELP_INVENTORY`
- `inventory`

  Default: not set

  CLI: `-i` or `--inventory`

  ENV: `ANSIBLE_INVENTORY`
- `inventory-column`

  Default: not set

  CLI: `--ic` or `--inventory-column`

  ENV: `ANSIBLE_NAVIGATOR_INVENTORY_COLUMNS`

Subcommand `lint`

- `lint-config`

  Default: not set

  CLI: `--lic` or `--lint-config`

  ENV: `ANSIBLE_LINT_CONFIG`
- `lintables`

  Default: not set

  CLI: positional

  ENV: `ANSIBLE_NAVIGATOR_LINTABLES`

Subcommand `replay`

- `playbook-artifact-replay`

  Default: not set

  CLI: positional

  ENV: `ANSIBLE_NAVIGATOR_PLAYBOOK_ARTIFACT_REPLAY`

Subcommand `run`

- `enable-prompts`

  Availiable values:

  - `True`
  - `False`

  Default: `False`

  CLI: `--ep` or `--enable-prompts`

  ENV: `ANSIBLE_NAVIGATOR_ENABLE_PROMPTS`
- `help-playbook`

  Availiable values:

  - `True`
  - `False`

  Default: `False`

  CLI: `--hp` or `--help-playbook`

  ENV: `ANSIBLE_NAVIGATOR_HELP_PLAYBOOK`
- `inventory`

  Default: not set

  CLI: `-i` or `--inventory`

  ENV: `ANSIBLE_INVENTORY`
- `inventory-column`

  Default: not set

  CLI: `--ic` or `--inventory-column`

  ENV: `ANSIBLE_NAVIGATOR_INVENTORY_COLUMNS`

- playbook

  Default: not set

  CLI: positional

  ENV: ANSIBLE_NAVIGATOR_PLAYBOOK
- playbook-artifact-enable

  Availiable values:

  - True
  - False

  Default: True

  CLI: --pae or --playbook-artifact-enable

  ENV: ANSIBLE_NAVIGATOR_PLAYBOOK_ARTIFACT_ENABLE
- playbook-artifact-save-as

  Default: {playbook_dir}/{playbook_name}-artifact-{time_stamp}.json

  CLI: --pas or --playbook-artifact-save-as

  ENV: ANSIBLE_NAVIGATOR_PLAYBOOK_ARTIFACT_SAVE_AS

Subcommand settings

- settings-effective

  Default: False

  CLI: --se or --effective

  ENV: ANSIBLE_NAVIGATOR_SETTINGS_EFFECTIVE
- settings-sample

  Default: False

  CLI: --gs or --sample

  ENV: ANSIBLE_NAVIGATOR_SETTINGS_SAMPLE
- settings-schema

  Availiable values:

  - json

  Default: json

  CLI: --ss or --schema

  ENV: ANSIBLE_NAVIGATOR_SETTINGS_SCHEMA
- settings-sources

  Default: False

  CLI: --so or --sources

  ENV: ANSIBLE_NAVIGATOR_SETTINGS_SOURCES

8.8.2 Ansible Builder

Утилита Ansible Builder является универсальным инструментом для создания образов среды исполнения (EE (Execution Environment, среда исполнения)). Для этого она предоставляет несколько команд с различными аргументами, описание которых приводится далее.

Вызов утилиты имеет следующий вид:

```
ansible-builder <command> [<arguments>]
```

Здесь:

- <command> – команда;
- <arguments> – аргументы команды.

Возможные команды:

- *build* – создание образа контейнера.
- *create* – создание файла спецификации образа.
- *introspect* – поиск в каталоге проекта сведений о коллекциях и вывод информации об зависимостях.

build

Аргументы команды:

-h, --help
Справочная информация о команде.

-t TAG [TAG ...], **--tag** TAG [TAG ...]
Название создаваемого образа (по умолчанию – `ansible-execution-env:latest`).

--container-runtime {podman,docker}
Система управления контейнерами:

- `docker` – Docker;
- `podman` – Podman.

Значение по умолчанию – `podman`.

--build-arg BUILD_ARGS
Переменные, которые могут быть использованы во время сборки:

- `ANSIBLE_GALAXY_CLI_COLLECTION_OPTS` – позволяет пользователю передать параметр `–pre` для команды установки коллекций (`ansible-galaxy collections install`), чтобы разрешить установку предварительных выпусков коллекций.
- `ANSIBLE_GALAXY_CLI_ROLE_OPTS` – позволяет пользователю передавать любые параметры для команды установки ролей, такие как `–no-deps`.
- `EE_BASE_IMAGE` – указывает родительский образ для среды исполнения.
- `EE_BUILDER_IMAGE` – указывает образ, используемый для компиляции типовых задач.
- `PKGMGR_PRESERVE_CACHE` – контролирует частоту очистки кеша менеджера пакетов в процессе сборки образа. Если это значение не задано, кеш очищается наиболее часто. Если задано значение `always` , кеш никогда не очищается. Любое другое значение заставляет очищать кеш только после установки системных зависимостей на заключительном этапе сборки образа.

--no-cache
Отключение кеширования во время сборки образа.

--prune-images

Удаление неиспользуемых образов после создания образа.

--container-policy {system,ignore_all,signature_required}

Политика проверки образа контейнера:

- system – использование системной политики или подписи со встроенными путями к связкам ключей;
- ignore_all – игнорирование всех подписей;
- signature_required – использование подписей, указанных с помощью параметра --container-keyring.

Важно: Аргумент поддерживается только при использовании Podman.

--container-keyring CONTAINER_KEYRING

Путь к хранилищу контейнера, которое может использоваться для хранения и управления учетными данными контейнера, а также для обеспечения безопасного доступа к защищенным ресурсам.

--squash {new,all,off}

Объединение слоев во время сборки, что позволяет уменьшить размер конечного образа:

- new – существовавшие ранее слои не изменяются, а все новые слои объединяются в один;
- all – все слои образа объединяются в один;
- off – объединение слоев отключено.

Значение по умолчанию – off.

Важно: Аргумент поддерживается только при использовании Podman.

-f FILENAME, **--file** FILENAME

Путь к файлу определения среды исполнения.

Значение по умолчанию – execution-environment.(yml|yaml).

-c BUILD_CONTEXT, **--context** BUILD_CONTEXT

Каталог, который будет использоваться для хранения контекста сборки.

Значение по умолчанию – context.

--output-filename {Containerfile,Dockerfile}

Название файла для записи определения образа, зависит от --container-runtime:

- Containerfile – Podman;
- Dockerfile – Docker.

--galaxy-keyring GALAXY_KEYRING

Путь к хранилищу для проверки подписи коллекции при загрузке из Galaxy. Проверка отключена, если она не настроена.

--galaxy-ignore-signature-status-codes GALAXY_IGNORE_SIGNATURE_STATUS_CODES

Позволяет игнорировать определенные коды состояния подписи при работе с Galaxy. Может быть указано несколько раз.

--galaxy-required-valid-signature-count GALAXY_REQUIRED_VALID_SIGNATURE_COUNT

Количество действительных подписей, необходимых для успешной загрузки коллекции с Ansible Galaxy.

-v [VERBOSITY], **--verbosity** [VERBOSITY]

Детализация вывода. Добавление нескольких `-v` увеличит детализацию. Максимальное значение `-vvv`. Также принимаются целочисленные значения, например, `-v3` или `--verbosity 3`.

Значение по умолчанию – 2.

create

Аргументы команды:

- `-h`, `--help`;
- `-f FILENAME`, `--file FILENAME`;
- `-c BUILD_CONTEXT`, `--context BUILD_CONTEXT`;
- `--output-filename {Containerfile,Dockerfile}`;
- `--galaxy-keyring GALAXY_KEYRING`;
- `--galaxy-ignore-signature-status-codes GALAXY_IGNORE_SIGNATURE_STATUS_CODES`;
- `--galaxy-required-valid-signature-count GALAXY_REQUIRED_VALID_SIGNATURE_COUNT`;
- `-v [VERBOSITY]`, `--verbosity [VERBOSITY]`.

Подробное описание аргументов приведено в описании команды *build*.

introspect

Аргументы команды:

folder

Путь к каталогу с коллекциям Ansible. Внутри указанного каталога должен существовать подкаталог `ansible_collections`.

-h, --help

Справочная информация о работе команды.

--sanitize

Очистка зависимостей или данных, которые могут быть связаны с проектом. Например, удаление непроверенных или ненадежных зависимостей, чтобы гарантировать, что результирующий список остается безопасным и актуальным.

--user-pip USER_PIP

Путь к файлу `requirements.txt`.

--user-bindep USER_BINDEP

Путь к файлу `bindep.txt`.

--write-pip WRITE_PIP

Путь к файлу, в который будет записан список зависимостей Python.

--write-bindep WRITE_BINDEP

Путь к файлу, в который будет записан список системных зависимостей.

-v [VERBOSITY], **--verbosity** [VERBOSITY]

Детализация вывода. Добавление нескольких `-v` увеличит детализацию. Максимальное значение `-vvv`. Также принимаются целочисленные значения, например, `-v3` или `--verbosity 3`.

Значение по умолчанию – 2.

8.8.3 Ansible Lint

Утилита Ansible Lint является универсальным инструментом для проверки качества и стиля кода в сценариях, ролях и коллекциях Ansible. Для этого она предоставляет различные аргументы, описание которых приводится далее.

Вызов утилиты имеет следующий вид:

```
ansible-lint [-h] [-P | -L | -T] [-f {brief,full,md,json,codeclimate,quiet,pep8,sarif}]␣
↪[--sarif-file SARIF_FILE] [-q]
             [--profile {min,basic,moderate,safety,shared,production}] [-p] [--project-
↪dir PROJECT_DIR] [-r RULESDIR]
             [-R] [-s] [--fix [WRITE_LIST]] [--show-relpath] [-t TAGS] [-v] [-x SKIP_
↪LIST] [--generate-ignore] [-w WARN_LIST]
             [--enable-list ENABLE_LIST] [--nocolor] [--force-color] [--exclude EXCLUDE_
↪PATHS [EXCLUDE_PATHS ...]]
             [-c CONFIG_FILE] [-i IGNORE_FILE] [--offline] [--version] [lintables ...]
```

Для понимания принципов работы утилиты обратитесь к ее *описанию*.

Аргументы команды

При запуске Ansible Lint можно использовать следующие аргументы:

lintables

Позиционный аргумент, идущий в самом конце и определяющий список файлов (список путей), подлежащих проверке.

-h, --help

Вывод справочной информации.

-P, --list-profiles

Вывод списка всех профилей. Форматирование вывода не поддерживается.

-L, --list-rules

Вывод списка всех правил. Возможные значения:

- `brief` – краткое описание правил (по умолчанию);
- `full` – полное описание правил;
- `md` – формат Markdown.

-T, --list-tags

Вывод всех тегов и правил, к которым они относятся. Чтобы включить правила с тегом `opt-in`, необходимо повысить уровень детализации с помощью флага `-v`.

-f, --format

Формат вывода:

- `brief` – краткий;
- `codeclimate` – совместимый с Code Climate;
- `full` – полный с выводом детальной информацией о каждой найденной проблеме;
- `json` – JSON;
- `md` – Markdown;
- pep8 – соответствующий стандарту **PEP 8**[176];
- `quiet`– минимально возможное количество информации, то есть только сообщения об ошибках;
- `sarif` – SARIF.

Значение по умолчанию отсутствует.

--sarif-file

Файл для вывода в формате SARIF.

-q

Уменьшение уровня детализации вывода. Для минимального уровня укажите `-qq`.

--profile

Профиль проверки:

- `min` – минимальные проверки, чтобы Ansible мог загрузить контент;
- `basic` – стандартные правила форматирования и синтаксиса;
- `moderate` – рекомендации по читаемости и поддержке;
- `safety` – правила безопасности для избежания использования опасных модулей;
- `shared` – профиль для подготовки к публикации кода;
- `production` – наивысший уровень строгости.

-p, --parseable

Вывод результата в формате, удобном для разбора, эквивалентен `-f pep8`.

--project-dir

Каталог проекта или репозитория. По умолчанию – каталог, в котором расположен конфигурационный файл `.ansible-lint`.

-r, --rules-dir

Каталоги с пользовательскими правилами. Указанные правила заменяют собой встроенные. Чтобы использовать и встроенные правила, и пользовательские, используйте аргумент `--rules-dir` совместно с аргументом `-R`.

-R

Сохранение встроенных правил при использовании аргумента `-r`.

-s, --strict

Возвращение ненулевого кода выхода для предупреждений, а не только для ошибок.

---fix

Разрешение автоматического исправления проблем, включая форматирование YAML. Можно ограничить список правил, для которых разрешены исправления, указав `all`, `none`, список идентификаторов или тегов.

--show-relpath

Вывод относительного пути к текущему каталогу.

-t, --tags

Использование только тех правил, идентификаторы или теги которых совпадают с указанными.

-v

Повышение уровня детализации вывода. Для максимального уровня укажите `-vv`.

-x, --skip-list

Использование только тех правил, идентификаторы или теги которых не совпадают с указанными, например, `--skip-list=name`.

--generate-ignore

Генерация файла `.ansible-lint-ignore`, который содержит список файлов и правил, которые должны быть пропущены при проверке. Каждая строка содержит путь к файлу и через пробел название или идентификатор правила.

[176] https://peps.python.org/pep-0008/

-w, --warn-list

Показ предупреждений только для указанных правил, если это не переопределено в конфигурационном файле. Значение по умолчанию: `experimental, jinja[spacing], fqcn[deep]`.

--enable-list

Включение дополнительных правил. Позволяет указать идентификаторы или теги дополнительных правил, которые должны быть включены при проверке.

--nocolor

Отключение цветного вывода (эквивалентно `NO_COLOR=1`).

--force-color

Принудительное включение цветного вывода (эквивалентно `FORCE_COLOR=1`).

--exclude

Путь к файлу или каталогу, который следует пропустить (повторяемый параметр).

-c, --config-file

Конфигурационный файл. По умолчанию ищет файлы `.ansible-lint`, `config/ansible-lint.yml`, или `.config/ansible-lint.yaml`.

-i, --ignore-file

Файл игнорирования. В файле игнорирования содержится список файлов и правил, которые должны быть пропущены при проверке. Каждая строка содержит путь к файлу и через пробел идентификатор правила. По умолчанию ищет файлы `.ansible-lint-ignore` или `.config/ansible-lint-ignore.txt`.

--offline

Отключение установки зависимостей из `requirements.yml` и обновления схем.

--version

Вывод информации о версии Ansible Lint.

Переменные окружения

При запуске Ansible Lint можно использовать следующие переменные окружения:

ANSIBLE_LINT_CUSTOM_RULESDIR

Путь к каталогам с пользовательскими правилами.

ANSIBLE_LINT_IGNORE_FILE

Переопределение имени файла игнорирования.

Значение по умолчанию – `.ansible-lint-ignore`.

ANSIBLE_LINT_WRITE_TMP

Сохранение исправлений во временные файлы, не изменяя оригиналы. Созданные временные файлы используются для проверки.

ANSIBLE_LINT_SKIP_SCHEMA_UPDATE

Отключение обновления схем.

ANSIBLE_LINT_NODEPS

Отключение установки зависимостей и проверок, которые могут завершиться неудачей при отсутствии модулей.

Файл настроек

В файле настроек Ansible Lint можно задать следующие параметры:

profile

Профиль проверки. Возможные значения параметра приведены в описании аргумента `--profile`. При значении `null`, используется конфигурация, заданная другими параметрами.

Пример заполнения:

```
profile: null
```

sarif_file

Файл для вывода в формате SARIF.

Пример заполнения:

```
sarif_file: result.sarif
```

exclude_paths

Путь к файлу или каталогу, который следует пропустить (повторяемый параметр).

Пример заполнения:

```
exclude_paths:
  - .cache/
  - test/fixtures/formatting-before/
  - test/fixtures/formatting-prettier/
```

parseable

Включение форматирования, удобного для разбора.

Пример заполнения:

```
parseable: true
```

quiet

Понижение уровня детализации вывода.

Пример заполнения:

```
quiet: true
```

strict

Указание обрабатывать предупреждения как ошибки.

Пример заполнения:

```
strict: true
```

verbosity

Уровень подробности вывода. Если установлено значение 1, вывод будет наименее подробным. Для более детализированного вывода используйте более высокие значения.

Пример заполнения:

```
verbosity: 1
```

mock_modules

Список модулей, которые Ansible Lint должен имитировать для прохождения синтаксической проверки, если они отсутствуют.

Пример заполнения:

```
mock_modules:
  - zuul_return
  - fake_namespace.fake_collection.fake_module
  - fake_namespace.fake_collection.fake_module.fake_submodule
```

mock_roles

Список ролей, которые Ansible Lint должен имитировать для прохождения синтаксической проверки, если они отсутствуют.

Пример заполнения:

```
mock_roles:
  - mocked_role
  - author.role_name
  - fake_namespace.fake_collection.fake_role
```

loop_var_prefix

Определение префиксов для переменных цикла, используемых в ролях. Префикс должен начинаться с __ или названия роли, что предотвращает конфликт названий переменных.

Пример заполнения:

```
loop_var_prefix: "^(__|{role}_)"
```

var_naming_pattern

Регулярное выражение, определяющее допустимый шаблон для названий переменных. В данном примере название переменной должно начинаться с буквы или подчеркивания и содержать только строчные буквы, цифры и подчеркивания:

```
var_naming_pattern: "^[a-z_][a-z0-9_]*$"
```

use_default_rules

Указание необходимости использования встроенных правил Ansible Lint.

Пример заполнения:

```
use_default_rules: true
```

rulesdir

Список путей к каталогам с пользовательскими правилами.

Пример заполнения:

```
rulesdir:
  - ./rule/directory/
```

skip_list

Список тегов правил, которые должны быть проигнорированы. Чтобы правило было пропущено, достаточно совпадения одного тега из списка.

Пример заполнения:

```
skip_list:
  - skip_this_tag
```

enable_list

Список дополнительных правил.

Пример заполнения:

```
enable_list:
  - args
  - empty-string-compare
  - no-log-password
  - no-same-owner
  - name[prefix]
  - galaxy-version-incorrect
  - yaml
```

tags

Ограничение проверки только правилами с указанными тегами. Чтобы правило было выбрано, достаточно, чтобы оно содержало один из указанных тегов.

Пример заполнения:

```
tags:
  - jinja[spacing]
```

warn_list

Список тегов для правил, неудачную проверку которых нужно рассматривать как предупреждение, а не ошибку. Чтобы правило попало в предупреждения, достаточно совпадения одного тега из списка.

Пример заполнения:

```
warn_list:
  - skip_this_tag
  - experimental
```

write_list

Автоматическое исправление кода для соответствия указанным правилам. Возможные значения:

- `all` – разрешает все исправления;
- `none` – отключает исправления;
- список тегов.

Пример заполнения:

```
write_list:
  - all
```

offline

Отключение установки зависимостей и обновления схем.

Пример заполнения:

```
offline: true
```

extra_vars

Значения обязательных переменных Ansible, используемых в сценариях или ролях, чтобы избежать ошибок анализа, возникающих из-за их отсутствия.

Пример заполнения:

```
extra_vars:
  foo: bar
  multiline_string_variable: |
    line1
    line2
  complex_variable: ":{;\t$()"
```

skip_action_validation

Включение принудительной проверки действий с помощью задач. Обычно это не требуется, потому что проверка синтаксиса покрывает это.

Пример заполнения:

```
skip_action_validation: false
```

kinds

Добавление пользовательских шаблонов для определения типа файла.

Пример заполнения:

```
kinds:
  - yaml: "**/*.yaml-too"
```

only_builtins_allow_collections

Список сторонних коллекций, использование которых разрешено при включенном правиле `only_builtins`.

Пример заполнения:

```
only_builtins_allow_collections:
  - example_ns.example_collection
```

only_builtins_allow_modules

Список сторонних модулей, использование которых разрешено при включенном правиле `only_builtins`.

Пример заполнения:

```
only_builtins_allow_modules:
  - example_module
```

task_name_prefix

Определение префикса, который будет добавлен к названиям задач.

Пример заполнения:

```
task_name_prefix: "{stem} | "
```

max_block_depth

Ограничение максимальной вложенности блоков.

Пример заполнения:

```
max_block_depth: 20
```

supported_ansible_also

Добавление поддержки для версий Ansible, которые могут быть не указаны как поддерживаемые по умолчанию.

Пример заполнения:

```
supported_ansible_also:
  - "2.14"
```

8.8.4 Ansible Molecule

Утилита *Ansible Molecule* является универсальным инструментом для разработки и тестирования ролей Ansible. Для этого она предоставляет несколько команд с различными аргументами, описание которых приводится далее.

Вызов утилиты имеет следующий вид:

```
molecule [<options>] <command> [<arguments>]
```

Здесь:

- <options> – аргументы утилиты;
- <command> – команда;
- <arguments> – аргументы команды.

Возможные аргументы утилиты:

--help
: Вывод справочной информации о командах и аргументах Ansible Molecule.

--version
: Вывод информации о версии Ansible Molecule.

--debug, --no-debug
: Включение или отключение режима отладки. По умолчанию отключен.

-v, --verbose
: Увеличение уровня детализации Ansible. Значение по умолчанию – `0`.

--c, --base-config TEXT
: Путь к базовой конфигурации (можно указать несколько раз). Если указан, Ansible Molecule выполнит в указанном порядке слияние настроек и слияние внутри каждого сценария. По умолчанию Ansible Molecule ищет файл `.config/molecule/config.yml` в текущем каталоге, и если не найдет, будет искать в домашнем каталоге пользователя.

-e, --env-file TEXT
: Имя файла с переменными, используемыми при обработке *файла настроек*. Значение по умолчанию – `.env.yml`.

Команды

При запуске Ansible Molecule можно использовать следующие команды.

check

Эта команда позволяет увидеть, что будет сделано, не внося изменения в тестовую среду. Она приводит к выполнению следующих команд:

1. *destroy*;
2. *dependency*;
3. *create*;
4. *prepare*;
5. *converge*.

Синтаксис команды:

```
molecule check [-s, --parallel, --help]
```

Подробное описание аргументов команды см. *ниже*.

cleanup

Эта команда выполняет очистку любых изменений, внесенных в тестовую среду во время выполнения тестов. Обычно используют для возвращения системы в исходное состояние. Синтаксис команды:

```
molecule cleanup [-s, --help]
```

Подробное описание аргументов команды см. *ниже*.

converge

Эта команда настраивает тестовые среды, выполняя следующие команды:

1. *dependency*;
2. *create*;
3. *prepare*;
4. *converge*.

Синтаксис команды:

```
molecule converge [-s, --help]
```

Подробное описание аргументов команды см. *ниже*.

create

Эта команда создает тестовые среды. Синтаксис команды:

```
molecule create [-s, -d, --help]
```

Подробное описание аргументов команды см. *ниже*.

dependency

Эта команда выполняет установку необходимых зависимостей перед выполнением тестирования. Синтаксис команды:

```
molecule dependency [-s, --help]
```

Подробное описание аргументов команды см. *ниже*.

destroy

Эта команда удаляет тестовые среды. Синтаксис команды:

```
molecule destroy [-s, -d, --all, --parallel, --help]
```

Подробное описание аргументов команды см. *ниже*.

drivers

Эта команда отображает список доступных драйверов. Синтаксис команды:

```
molecule drivers [-f, --help]
```

Аргументы команды:

-f, --format [simple|plain]
Формат вывода. Значение по умолчанию – simple.

idempotence

Эта команда выполняет настройку тестовых сред и проверку идемпотентности. Синтаксис команды:

```
molecule idempotence [-s, --help]
```

Подробное описание аргументов команды см. *ниже*.

init

Эта команда создает структуру каталогов и файлов, необходимых для работы Ansible Molecule с новой ролью или сценарием. Синтаксис команды:

```
molecule init [--help, scenario] [--dependency-name, -d, --provisioner-name]
```

Подробное описание аргументов команды см. *ниже*.

list

Эта команда показывает статус тестовых сред. Синтаксис команды:

```
molecule list [-s, -f, --help]
```

Аргументы команды:

-f, --format [simple|plain]
Формат вывода. Значение по умолчанию – simple.

Подробное описание аргументов команды см. *ниже*.

login

Эта команда позволяет войти в тестовую среду. Синтаксис команды:

```
molecule login [-h, -s, --help]
```

Подробное описание аргументов команды см. *ниже*.

matrix

Эта команда отображает матрицу шагов, используемых для тестирования. Синтаксис команды:

```
molecule matrix [-s, --help]
```

Подробное описание аргументов команды см. *ниже*.

prepare

Эта команда подготавливает тестовые среды к выполнению тестов. Синтаксис команды:

```
molecule prepare [-s, -d, -f --help]
```

Аргументы команды:

-f, --force, --no-force
Включить или отключить режим принудительного выполнения. По умолчанию выключен.

Подробное описание аргументов команды см. *ниже*.

reset

Эта команда возвращает временные каталоги Ansible Molecule в исходное состояние. Синтаксис команды:

```
molecule reset [-s, --help]
```

Подробное описание аргументов команды см. *ниже*.

side-effect

Эта команда позволяет выполнить действия, которые не входят в основное применение роли. Синтаксис команды:

```
molecule side-effect [-s, --help]
```

Подробное описание аргументов команды см. *ниже*.

syntax

Эта команда проверяет синтаксис файлов роли. Синтаксис команды:

```
molecule syntax [-s, --help]
```

Подробное описание аргументов команды см. *ниже*.

test

Эта команда запускает полный цикл тестирования. Синтаксис команды:

```
molecule test [-s, -p, -d, --all, --destroy, --parallel, --help]
```

Подробное описание аргументов команды см. *ниже*.

verify

Эта команда выполняет проверку состояния тестовых сред после тестирования. Синтаксис команды:

```
molecule verify [-s, --help]
```

Подробное описание аргументов команды см. *ниже*.

Описание аргументов команд

Команды Ansible Molecule поддерживают следующие аргументы:

--all / `--no-all`
: Выполнение команды для всех сценариев. По умолчанию выключено (`--no-all`).

-d, --driver-name `[default|molecule_brest|molecule_yandexcloud|podman]`
: Название используемого драйвера. Значение по умолчанию – `default`.

--destroy `[always|never]`
: Политика удаления тестовых сред после завершения тестов:

  - `always` – удалять после каждого теста (по умолчанию);
  - `never` – не удалять.

-h, --host TEXT
: Узел для подключения.

-p, --platform-name TEXT
: Название платформы для тестирования. Значение по умолчанию – None.

--parallel / `--no-parallel`
: Включение или отключение параллельного режима. По умолчанию отключен.

-s, --scenario-name TEXT
: Название используемого сценария. Значение по умолчанию – `default`.

scenario
: Название инициализированного сценария. Значение по умолчанию: `default`. Возможные дополнительные аргументы:

  --dependency-name `[galaxy]`
  : Название используемой зависимости. Значение по умолчанию – `galaxy`.

  -d, --driver-name `[default|molecule_brest|molecule_yandexcloud|podman]`
  : Название используемого драйвера. Значение по умолчанию – `default`.

  --provisioner-name `[ansible]`
  : Название используемого *исполнителя сценариев*. Значение по умолчанию – `ansible`.

Файл настроек

Для настройки Ansible Molecule используется файл `molecule.yml`. В нем описывается, как и где будет выполнятся тестирование ролей. Он может содержать следующие параметры:

- драйвер;
- зависимости;
- исполнитель сценариев;
- платформы;
- средства проверки;
- сценарии.

Драйвер

Драйвер указывает какой инструмент утилита будет использовать для создания тестовой среды. Драйвер может быть изменен в командной строке. Утилита запоминает последний использованный драйвер и будет использовать его для всех последующих команд или до тех пор, пока не будут удалены тестовые среды.

Пример заполнения:

```yaml
driver:
  options:
    managed: False
    login_cmd_template: "podman exec -ti {instance} bash"
    ansible_connection_options:
      ansible_connection: podman
```

Возможные параметры:

name

Название используемого драйвера.

options

Дополнительные параметры для настройки драйвера:

managed

Указывает, должна ли утилита управлять созданием и уничтожением тестовой среды:

- `True` – утилита управляет тестовой средой (по умолчанию);
- `False` – управление передается внешнему инструменту.

login_cmd_template

Шаблон команды для входа в тестовую среду.

ansible_connection_options

Настройки соединения, которые будут использоваться для взаимодействия с тестовой средой.

Возможные опции:

ansible_connection

Тип соединения.

Зависимости

Для указания пути к файлу зависимостей используются параметры `role-file` и `requirements-file`. Если необходимо указать путь к файлу ролей, то используется параметр `role-file`. Если необходимо установить коллекции, то используйте параметр `requirements-file`.

Пример заполнения:

```
dependency:
  name: galaxy
  options:
    requirements-file: requirements.yml
```

Возможные параметры:

name

Тип зависимости:

- `galaxy` – для загрузки зависимостей из Ansible Galaxy.
- `shell` – для выполнения произвольных shell-команд.

enabled

Указывает, следует ли включить управление зависимостями:

- `True` (по умолчанию);
- `False`.

options

Дополнительные параметры для настройки зависимостей:

ignore-certs

Указывает, следует ли игнорировать ошибки, связанные с сертификатами SSL при загрузке зависимостей:

- `True`;
- `False`.

ignore-errors

Указывает, следует ли игнорировать ошибки, возникающие при загрузке зависимостей:

- `True`;
- `False`.

role-file

Указывает файл, содержащий список ролей, которые необходимо загрузить.

requirements-file

Указывает файл, содержащий список коллекций Ansible, которые нужно установить.

command

Команда, которую нужно выполнить для управления зависимостями. Только для типа зависимости `shell`.

Исполнитель сценариев

Ansible является исполнителем по умолчанию. Утилита управляет жизненным циклом тестовых сред. Однако пользователь может предоставить сценарии для выполнения этапов. Для этого в параметре playbooks необходимо указать путь к файлу, содержащему сценарий выполнения этапа.

Пример заполнения:

```
provisioner:
  name: ansible
  playbooks:
    create: create.yml
    converge: converge.yml
    destroy: destroy.yml
```

Платформы

Платформы определяют тестовые среды и группы, к которым они относятся. Можно использовать несколько тестовых сред.

Пример заполнения:

```
platforms:
  - name: molecule-astra
    image: hub.astra-automation.ru/aa/aa-base-ee
```

Возможные параметры:

name
: Название тестовой платформы.

image
: Образ контейнера, который будет использоваться для платформы.

privileged
: Указывает, нужно ли запускать контейнер в привилегированном режиме.

groups
: Группы, к которым принадлежит платформа.

Средства проверки

Утилита производит тестирование с помощью настраиваемых средств проверки. Ansible является средством проверки по умолчанию.

Пример заполнения:

```
verifier:
  name: testinfra
```

Возможные параметры:

name
: Название средства проверки:

enabled
: Указывает, нужно ли включить тестирование:

- True;
- False.

directory

Путь к тестовому каталогу.

Сценарии

Сценарий определяет этапы, которые будут выполняться в рамках тестирования. Сценарий по умолчанию – `default`.

Пример заполнения:

```
scenario:
  test_sequence:
    - dependency
    - cleanup
    - destroy
    - syntax
    - create
    - prepare
    - converge
    - idempotence
    - side_effect
    - verify
    - cleanup
```

8.8.5 Компоненты образов среды исполнения

Образы среды исполнения, созданные ПАО Группа Астра, доступны для загрузки из реестра registry.astralinux.ru/aa[177].

Образы aa-base-ee

0.6.2

Пакеты APT

Пакет	Версия	Описание
adduser	3.118.astra.se2	add and rem
apt	1.8.2.3+ci202309131156+astra26	commandlin
astra-version	8.1.41+v1.7.5.16	Update Astra
base-files	10.3.astra15	Debian base
base-passwd	3.5.46	Debian base
bash	5.0-4.astra2	GNU Bourne
bind9-host	1:9.11.5.P4+dfsg-5.1+deb10u9+ci202308311745+astra1	DNS lookup
binutils	2.31.1-16+ci202306051252+astra11	GNU assemb
binutils-common:amd64	2.31.1-16+ci202306051252+astra11	Common file
binutils-x86-64-linux-gnu	2.31.1-16+ci202306051252+astra11	GNU binary
bsdutils	1:2.33.1-0.1.astra6se5	basic utilitie
build-essential	12.6	Informationa
bzip2	1.0.6-9.2~deb10u1	high-quality
ca-certificates	20230311ubuntu0.20.04.1	Common CA
comerr-dev:amd64	2.1-1.44.5-1+deb10u3+ci202212051923+astra1	common err
coreutils	8.30-3	GNU core ut
cpp	4:8.3.0-1	GNU C prepr
cpp-8	8.3.0-6	GNU C prepr
dash	0.5.10.2-5	POSIX-comp

[177] https://registry.astralinux.ru/browse/aa/

Пакет	Версия	Описание
debconf	1.5.71+deb10u1+ci202303291646+astra1	Debian confi
debianutils	4.8.6.1	Miscellaneou
diffutils	1:3.7-3	File compari
dirmngr	2.2.27-2+deb11u2.astra5	GNU privacy
distro-info-data	0.41+deb10u3	information
dpkg	1.19.8.astra.se3	Debian pack
dpkg-dev	1.19.8.astra.se3	Debian pack
e2fsprogs	1.44.5-1+deb10u3+ci202212051923+astra1	ext2/ext3/ex
expect	5.45.4-2+ci202011031319	Automates i
fakeroot	1.23-1	tool for simu
fdisk	2.33.1-0.1.astra6se5	collection of
file	1:5.35-4+deb10u1	Recognize th
findutils	4.6.0+git+20190209-2	utilities for f
g++	4:8.3.0-1	GNU C++ co
g++-8	8.3.0-6	GNU C++ co
gawk	1:4.2.1+dfsg-1+ci202308181355+astra1	GNU awk, a
gcc	4:8.3.0-1	GNU C comp
gcc-8	8.3.0-6	GNU C comp
gcc-8-base:amd64	8.3.0-6	GCC, the GN
geoip-database	20181108-1	IP lookup co
git	1:2.30.2-1+deb11u2+ci202305151844+astra2+b1	fast, scalabl
git-man	1:2.30.2-1+deb11u2+ci202305151844+astra2+b1	fast, scalabl
gnupg	2.2.27-2+deb11u2.astra5	GNU privacy
gnupg-l10n	2.2.27-2+deb11u2.astra5	GNU privacy
gnupg-utils	2.2.27-2+deb11u2.astra5	GNU privacy
gpg	2.2.27-2+deb11u2.astra5	GNU Privacy
gpg-agent	2.2.27-2+deb11u2.astra5	GNU privacy
gpg-wks-client	2.2.27-2+deb11u2.astra5	GNU privacy
gpg-wks-server	2.2.27-2+deb11u2.astra5	GNU privacy
gpgconf	2.2.27-2+deb11u2.astra5	GNU privacy
gpgsm	2.2.27-2+deb11u2.astra5	GNU privacy
gpgv	2.2.27-2+deb11u2.astra5	GNU privacy
grep	3.3-1	GNU grep, e
gzip	1.9-3+deb10u1	GNU compre
hostname	3.21	utility to set/
init-system-helpers	1.56+nmu1	helper tools
iso-codes	4.2-1	ISO languag
krb5-config	2.6astra1	Configuratio
krb5-locales	1.18.3-6+deb11u3astra1	internationa
krb5-multidev:amd64	1.18.3-6+deb11u3astra1	developmen
less	487-0.1	pager progra
libacl1:amd64	2.2.53-4	access contr
libapt-inst2.0:amd64	1.8.2.3+ci202309131156+astra26	deb package
libapt-pkg5.0:amd64	1.8.2.3+ci202309131156+astra26	package ma
libasan5:amd64	8.3.0-6	AddressSani
libassuan0:amd64	2.5.2-1	IPC library fo
libatomic1:amd64	8.3.0-6	support libra
libattr1:amd64	1:2.4.48-4+ci202107131632+astra2	extended att
libaudit-common	1:2.8.5-2ubuntu6+ci202307261837+astra5	Dynamic libr
libaudit1:amd64	1:2.8.5-2ubuntu6+ci202307261837+astra5	Dynamic libr
libbind9-161:amd64	1:9.11.5.P4+dfsg-5.1+deb10u9+ci202308311745+astra1	BIND9 Share
libbinutils:amd64	2.31.1-16+ci202306051252+astra11	GNU binary
libblkid1:amd64	2.33.1-0.1.astra6se5	block device
libbsd0:amd64	0.9.1-2+deb10u1	utility functi
libbz2-1.0:amd64	1.0.6-9.2~deb10u1	high-quality
libc-bin	2.28-10+deb10u2+ci202309131632+astra7	GNU C Libra
libc-dev-bin	2.28-10+deb10u2+ci202309131632+astra7	GNU C Libra

Пакет	Версия	Описание
libc6:amd64	2.28-10+deb10u2+ci202309131632+astra7	GNU C Libra
libc6-dev:amd64	2.28-10+deb10u2+ci202309131632+astra7	GNU C Libra
libcap-ng0:amd64	0.7.9-2	An alternate
libcap2:amd64	1:2.25-2	POSIX 1003.
libcbor0.8:amd64	0.8.0-2astra1	library for pa
libcc1-0:amd64	8.3.0-6	GCC cc1 plu
libcom-err2:amd64	1.44.5-1+deb10u3+ci202212051923+astra1	common err
libcurl3-gnutls:amd64	7.64.0-4+deb10u5+astra6	easy-to-use
libdb5.3:amd64	5.3.28+dfsg2-1+ci202311141606+astra1	Berkeley v5.
libdebconfclient0:amd64	0.249astra11	Debian Conf
libdns1104:amd64	1:9.11.5.P4+dfsg-5.1+deb10u9+ci202308311745+astra1	DNS Shared
libdpkg-perl	1.19.8.astra.se3	Dpkg perl m
libedit2:amd64	3.1-20181209-1	BSD editline
liberror-perl	0.17027-2	Perl module
libexpat1:amd64	2.2.6-2+deb10u6	XML parsing
libexpat1-dev:amd64	2.2.6-2+deb10u6	XML parsing
libext2fs2:amd64	1.44.5-1+deb10u3+ci202212051923+astra1	ext2/ext3/ex
libfakeroot:amd64	1.23-1	tool for simu
libfdisk1:amd64	2.33.1-0.1.astra6se5	fdisk partitio
libffi6:amd64	3.2.1-9	Foreign Func
libfido2-1:amd64	1.6.0-2	library for ge
libfstrm0:amd64	0.4.0-1	Frame Strea
libgcc-8-dev:amd64	8.3.0-6	GCC support
libgcc1:amd64	1:8.3.0-6	GCC support
libgcrypt20:amd64	1.8.7-6.astra1	LGPL Crypto
libgdbm-compat4:amd64	1.18.1-4	GNU dbm da
libgdbm6:amd64	1.18.1-4	GNU dbm da
libgeoip1:amd64	1.6.12-1	non-DNS IP-t
libgmp10:amd64	2:6.1.2+dfsg-4+deb10u1	Multiprecisio
libgnutls30:amd64	3.6.7-4+deb10u11+astra1	GNU TLS libr
libgomp1:amd64	8.3.0-6	GCC OpenM
libgost	2.0.2-5+ci3	This packag
libgpg-error0:amd64	1.35-1	GnuPG deve
libgssapi-krb5-2:amd64	1.18.3-6+deb11u3astra1	MIT Kerbero
libgssrpc4:amd64	1.18.3-6+deb11u3astra1	MIT Kerbero
libhogweed4:amd64	3.4.1-1+deb10u1	low level cry
libicu63:amd64	63.1-6+deb10u2	Internationa
libidn2-0:amd64	2.0.5-1+deb10u1+ci202308141422+astra1	Internationa
libisc1100:amd64	1:9.11.5.P4+dfsg-5.1+deb10u9+ci202308311745+astra1	ISC Shared L
libisccc161:amd64	1:9.11.5.P4+dfsg-5.1+deb10u9+ci202308311745+astra1	Command C
libisccfg163:amd64	1:9.11.5.P4+dfsg-5.1+deb10u9+ci202308311745+astra1	Config File H
libisl19:amd64	0.20-2	manipulating
libitm1:amd64	8.3.0-6	GNU Transac
libjson-c3:amd64	0.12.1+ds-2+deb10u1	JSON manipu
libk5crypto3:amd64	1.18.3-6+deb11u3astra1	MIT Kerbero
libkadm5clnt-mit12:amd64	1.18.3-6+deb11u3astra1	MIT Kerbero
libkadm5srv-mit12:amd64	1.18.3-6+deb11u3astra1	MIT Kerbero
libkdb5-10:amd64	1.18.3-6+deb11u3astra1	MIT Kerbero
libkeyutils1:amd64	1.6-6	Linux Key Ma
libkrb5-3:amd64	1.18.3-6+deb11u3astra1	MIT Kerbero
libkrb5-dev:amd64	1.18.3-6+deb11u3astra1	headers and
libkrb5support0:amd64	1.18.3-6+deb11u3astra1	MIT Kerbero
libksba8:amd64	1.3.5-2+deb10u2	X.509 and C
libldap-2.4-2:amd64	2.4.57+dfsg-3+deb11u1~bpo10+1astra1	OpenLDAP li
libldap-common	2.4.57+dfsg-3+deb11u1~bpo10+1astra1	OpenLDAP c
liblmdb0:amd64	0.9.22-1	Lightning Me
liblocale-gettext-perl	1.07-3	module usin

Пакет	Версия	Описание
liblsan0:amd64	8.3.0-6	LeakSanitize
liblwres161:amd64	1:9.11.5.P4+dfsg-5.1+deb10u9+ci202308311745+astra1	Lightweight
liblz4-1:amd64	1.9.3-2	Fast LZ com
liblzma5:amd64	5.2.4-1+deb10u1	XZ-format c
libmagic-mgc	1:5.35-4+deb10u1	File type det
libmagic1:amd64	1:5.35-4+deb10u1	Recognize th
libmount1:amd64	2.33.1-0.1.astra6se5	device mour
libmpc3:amd64	1.1.0-1	multiple pre
libmpdec2:amd64	2.4.2-2	library for de
libmpfr6:amd64	4.0.2-1	multiple pre
libmpx2:amd64	8.3.0-6	Intel memor
libncursesw6:amd64	6.1+20181013-2+deb10u3+ci202309121705+astra1	shared libra
libnettle6:amd64	3.4.1-1+deb10u1	low level cry
libnghttp2-14:amd64	1.36.0-2+deb10u2	library imple
libnpth0:amd64	1.6-1	replacement
libp11-kit0:amd64	0.23.15-2+deb10u1	library for lo
libpam-modules:amd64	1.3.1-5.astra.se11	Pluggable Au
libpam-modules-bin	1.3.1-5.astra.se11	Pluggable Au
libpam-runtime	1.3.1-5.astra.se11	Runtime sup
libpam0g:amd64	1.3.1-5.astra.se11	Pluggable Au
libparsec-base3	3.4+ci41	base library
libparsec-cap3	3.4+ci41	capabilities
libpcre2-8-0:amd64	10.32-5+deb10u1+ci202308291416+astra1	New Perl Co
libpcre3:amd64	2:8.39-13	Old Perl 5 Co
libpdp	3.4+ci41	library for PA
libperl5.28:amd64	5.28.1-6+deb10u1+ci202305241311+astra2	shared Perl l
libprotobuf-c1:amd64	1.3.3-1+ci202311091556+astra2	Protocol Buf
libpsl5:amd64	0.20.2-2	Library for P
libpython3-stdlib:amd64	3.7.3-1	interactive h
libpython3.7-minimal:amd64	3.7.3-2+deb10u5+ci202309191152+astra10	Minimal subs
libpython3.7-stdlib:amd64	3.7.3-2+deb10u5+ci202309191152+astra10	Interactive h
libpython3.9:amd64	3.9.18-1+astra1	Shared Pyth
libpython3.9-dev:amd64	3.9.18-1+astra1	Header files
libpython3.9-minimal:amd64	3.9.18-1+astra1	Minimal subs
libpython3.9-stdlib:amd64	3.9.18-1+astra1	Interactive h
libquadmath0:amd64	8.3.0-6	GCC Quad-P
libreadline7:amd64	7.0-5	GNU readline
librtmp1:amd64	2.4+20151223.gitfa8646d.1-2	toolkit for RT
libsasl2-2:amd64	2.1.27+dfsg-1+deb10u2+b1	Cyrus SASL
libsasl2-modules-db:amd64	2.1.27+dfsg-1+deb10u2+b1	Cyrus SASL
libseccomp2:amd64	2.5.1-1~bpo10+1	high level in
libselinux1:amd64	2.8-1	SELinux runt
libsemanage-common	2.8-2	Common file
libsemanage1:amd64	2.8-2	SELinux poli
libsepol1:amd64	2.8-1	SELinux libra
libsigsegv2:amd64	2.12-2	Library for h
libsmartcols1:amd64	2.33.1-0.1.astra6se5	smart colum
libsqlite3-0:amd64	3.34.1-3+ci202212121148+astra1	SQLite 3 sha
libss2:amd64	1.44.5-1+deb10u3+ci202212051923+astra1	command-lin
libssh2-1:amd64	1.11.0-2	SSH2 client-
libssl1.1:amd64	1.1.1n-0+deb11u4-deb11u5-astra3	Secure Sock
libstdc++-8-dev:amd64	8.3.0-6	GNU Standa
libstdc++6:amd64	8.3.0-6	GNU Standa
libsystemd0:amd64	241-7~deb10u8astra.se32+ci4	systemd util
libtasn1-6:amd64	4.16.0-2+deb11u1	Manage ASN
libtcl8.6:amd64	8.6.9+dfsg-2+ci202302131725+astra1	Tcl (the Tool
libtext-charwidth-perl	0.04-7.1	get display w

Пакет	Версия	Описание
libtext-iconv-perl	1.7-5	converts bet
libtext-wrapi18n-perl	0.06-7.1	internationa
libtinfo6:amd64	6.1+20181013-2+deb10u3+ci202309121705+astra1	shared low-l
libtsan0:amd64	8.3.0-6	ThreadSanit
libubsan1:amd64	8.3.0-6	UBSan – und
libudev1:amd64	241-7~deb10u8astra.se32+ci4	libudev shar
libunistring2:amd64	0.9.10-1	Unicode strin
libuuid1:amd64	2.33.1-0.1.astra6se5	Universally U
libx11-6:amd64	2:1.8.7-1astra3	X11 client-si
libx11-data	2:1.8.7-1astra3	X11 client-si
libxau6:amd64	1:1.0.8-1	X11 authoris
libxcb1:amd64	1.15-1astra.se3	X C Binding
libxdmcp6:amd64	1:1.1.2-3	X11 Display
libxext6:amd64	2:1.3.3-1	X11 miscella
libxml2:amd64	2.9.4+dfsg1-7+deb10u6+ci202308291455+astra2	GNOME XML
libxmuu1:amd64	2:1.1.2-2	X11 miscella
libzstd1:amd64	1.5.2+dfsg-1	fast lossless
linux-libc-dev:amd64	5.4.0-54astra7+ci67	latest linux-l
linux-libc-dev-5.4.0-162:amd64	5.4.0-162.astra1+ci29	Linux Kernel
login	1:4.13+dfsg1-1.astra.se8	system login
lsb-base	10.2019051400+ci202012021351+astra2	Linux Standa
lsb-release	10.2019051400+ci202012021351+astra2	Linux Standa
make	4.2.1-1.2	utility for dir
manpages	4.16-2	Manual page
manpages-dev	4.16-2	Manual page
mawk	1.3.3-17	a pattern sca
mime-support	3.62	MIME files „r
mount	2.33.1-0.1.astra6se5	tools for mou
ncurses-base	6.1+20181013-2+deb10u3+ci202309121705+astra1	basic termin
ncurses-bin	6.1+20181013-2+deb10u3+ci202309121705+astra1	terminal-rela
netbase	6.1~bpo10+1	Basic TCP/IP
openssh-client	1:8.4p1-2~deb10u1astra6se10	secure shell
openssl	1.1.1n-0+deb11u4-deb11u5-astra3	Secure Sock
passwd	1:4.13+dfsg1-1.astra.se8	change and
patch	2.7.6-3+deb10u1	Apply a diff f
perl	5.28.1-6+deb10u1+ci202305241311+astra2	Larry Wall’s
perl-base	5.28.1-6+deb10u1+ci202305241311+astra2	minimal Perl
perl-modules-5.28	5.28.1-6+deb10u1+ci202305241311+astra2	Core Perl mo
pinentry-curses	1.1.0-2	curses-based
publicsuffix	20190415.1030-1	accurate, ma
python-apt-common	1.8.4.3+ci202309111239+astra1	Python inter
python3	3.7.3-1	interactive h
python3-apt	1.8.4.3+ci202309111239+astra1	Python 3 inte
python3-minimal	3.7.3-1	minimal sub
python3.7	3.7.3-2+deb10u5+ci202309191152+astra10	Interactive h
python3.7-minimal	3.7.3-2+deb10u5+ci202309191152+astra10	Minimal subs
python3.9	3.9.18-1+astra1	Interactive h
python3.9-dev	3.9.18-1+astra1	Header files
python3.9-distutils	3.9.18-1+astra1	distutils pac
python3.9-lib2to3	3.9.18-1+astra1	lib2to3 pack
python3.9-minimal	3.9.18-1+astra1	Minimal subs
python3.9-pip	23.1.2-1+aa1	The PyPA rec
python3.9-setuptools	70.1.0	Easily downl
python3.9-venv	3.9.18-1+astra1	Interactive h
readline-common	7.0-5	GNU readline
sed	4.7-1	GNU stream
sensible-utils	0.0.12	Utilities for s

Пакет	Версия	Описание
sshpass	1.06-1	Non-interact
sysvinit-utils	2.93-8	System-V-lik
tar	1.34+dfsg-1+ci202304071507+astra2	GNU version
tcl-expect:amd64	5.45.4-2+ci202011031319	Automates i
tcl8.6	8.6.9+dfsg-2+ci202302131725+astra1	Tcl (the Tool
tofu	1.6.2	OpenTofu is
tzdata	2021a-0+deb10u11	time zone ar
util-linux	2.33.1-0.1.astra6se5	miscellaneou
xauth	1:1.0.10-1	X authentica
xz-utils	5.2.4-1+deb10u1	XZ-format co
zlib1g:amd64	1:1.2.11.dfsg-1+deb10u2	compression

Пакеты Python

Пакет	Версия
Jinja2	3.1.4
MarkupSafe	2.1.5
PyNaCl	1.5.0
PyYAML	6.0
aenum	3.1.15
aiohttp	3.8.3
aiosignal	1.3.1
ansible-core	2.15.10
ansible-pylibssh	1.0.0
ansible-runner	2.3.6
appdirs	1.3.0
argcomplete	3.4.0
async-timeout	4.0.2
attrs	23.2.0
bcrypt	4.1.2
cachetools	5.3.2
certifi	2024.2.2
cffi	1.16.0
charset-normalizer	2.1.1
coverage	7.5.3
cryptography	42.0.7
decorator	5.1.1
dicttoxml	1.7.16
distlib	0.3.8
distro	1.9.0
docutils	0.19
dogpile.cache	1.3.2
dumb-init	1.2.5
frozenlist	1.3.3
future	0.18.3
gnupg	2.3.1
google-auth	2.22.0
grpcio	1.50.0
gssapi	1.8.3
hvac	1.1.1
idna	3.4
ijson	2.5.1
importlib-metadata	4.6.4
importlib_resources	5.0.7

continues on next page

Таблица 2 – продолжение с предыдущей страницы

Пакет	Версия
ipaddress	1.0.23
iso8601	0.1.11
jmespath	1.0.1
jsonpatch	1.32
jsonpointer	2.4
jsonschema	4.17.1
jxmlease	1.0.3
kerberos	1.3.1
keystoneauth1	3.18.0
kubernetes	25.3.0
lockfile	0.12.2
lxml	4.9.1
multidict	6.0.2
munch	4.0.0
ncclient	0.6.13
netaddr	0.8.0
netifaces	0.11.0
oauthlib	3.2.2
openssl-engine	1.3.1
openstacksdk	0.103.0
os-service-types	1.7.0
packaging	21.3
paramiko	2.12.0
passlib	1.7.4
pbr	5.11.1
pexpect	4.7.0
protobuf	4.21.9
psutil	5.9.8
psycopg2-binary	2.9.5
ptyprocess	0.7.0
pyOpenSSL	24.1.0
pyasn1-modules	0.2.8
pyasn1	0.4.8
pycparser	2.22
pyhcl	0.4.4
pynetbox	6.6.2
pyone	6.4.0
pyparsing	2.4.6
pypsrp	0.8.1
pyrsistent	0.19.2
pysnow	0.7.17
pyspnego	0.11.0
python-apt	=1.8.4.3-ci202309111239-astra1
python-daemon	3.0.1
python-dateutil	2.8.2
python-magic	0.4.27
pytz	2019.3
pyvmomi	8.0.0.1
pywinrm	0.4.3
requests-oauthlib	1.3.1
requests	2.28.1
requests_ntlm	1.3.0
requestsexceptions	1.2.0
resolvelib	1.0.1
rpm	0.1.0
rsa	4.9
scp	0.15.0

continues on next page

Таблица 2 – продолжение с предыдущей страницы

Пакет	Версия
selectors2	2.0.2
selinux	0.3.0
six	1.16.0
stevedore	3.0.0
tblib	3.0.0
textfsm	1.1.3
toml	0.10.2
tomli	2.0.1
tomli_w	1.0.0
ttp	0.9.1
typing_extensions	4.11.0
urllib3	1.26.17
urllib_gssapi	1.0.2
websocket-client	1.4.2
xmltodict	0.13.0
yarl	1.8.1
zabbix-api	0.5.5
zipp	3.11.0

Образы `aa-creator-ee`

`0.2.1`

Пакеты APT

Пакет	Версия	Описа
adduser	3.118.astra.se2	add ar
apt	1.8.2.3+ci202309131156+astra26	comma
astra-version	8.1.41+v1.7.5.16	Update
base-files	10.3.astra15	Debiar
base-passwd	3.5.46	Debiar
bash	5.0-4.astra2	GNU B
bind9-host	1:9.11.5.P4+dfsg-5.1+deb10u9+ci202308311745+astra1	DNS lo
binutils	2.31.1-16+ci202306051252+astra11	GNU a
binutils-common:amd64	2.31.1-16+ci202306051252+astra11	Comm
binutils-x86-64-linux-gnu	2.31.1-16+ci202306051252+astra11	GNU b
bsdutils	1:2.33.1-0.1.astra6se5	basic u
build-essential	12.6	Inform
buildah	1.28.0+ds1-2+b1	CLI too
bzip2	1.0.6-9.2~deb10u1	high-q
ca-certificates	20230311ubuntu0.20.04.1	Comm
catatonit	0.1.7-1	init pro
comerr-dev:amd64	2.1-1.44.5-1+deb10u3+ci202212051923+astra1	comme
conmon	2.1.3+ds1-1	OCI co
containernetworking-plugins	1.1.1+ds1-3	standa
coreutils	8.30-3	GNU c
cpp	4:8.3.0-1	GNU C
cpp-8	8.3.0-6	GNU C
dash	0.5.10.2-5	POSIX-
dbus	1.12.24-0+deb11u1.astra.se8+ci2	simple
dbus-user-session	1.12.24-0+deb11u1.astra.se8+ci2	simple
debconf	1.5.71+deb10u1+ci202303291646+astra1	Debiar
debianutils	4.8.6.1	Miscell
diffutils	1:3.7-3	File co

Пакет	Версия	Описа
dirmngr	2.2.27-2+deb11u2.astra5	GNU p
distro-info-data	0.41+deb10u3	inform
dmsetup	2:1.02.175-2	Linux k
dpkg	1.19.8.astra.se3	Debian
dpkg-dev	1.19.8.astra.se3	Debian
e2fsprogs	1.44.5-1+deb10u3+ci202212051923+astra1	ext2/e
expect	5.45.4-2+ci202011031319	Autom
fakeroot	1.23-1	tool fo
fdisk	2.33.1-0.1.astra6se5	collect
file	1:5.35-4+deb10u1	Recogn
findutils	4.6.0+git+20190209-2	utilities
fuse-overlayfs	1.4.0-1	implem
fuse3	3.4.1-1+deb10u1+b1	Filesys
g++	4:8.3.0-1	GNU C
g++-8	8.3.0-6	GNU C
gawk	1:4.2.1+dfsg-1+ci202308181355+astra1	GNU a
gcc	4:8.3.0-1	GNU C
gcc-8	8.3.0-6	GNU C
gcc-8-base:amd64	8.3.0-6	GCC, t
geoip-database	20181108-1	IP look
git	1:2.30.2-1+deb11u2+ci202305151844+astra2+b1	fast, sc
git-man	1:2.30.2-1+deb11u2+ci202305151844+astra2+b1	fast, sc
gnupg	2.2.27-2+deb11u2.astra5	GNU p
gnupg-l10n	2.2.27-2+deb11u2.astra5	GNU p
gnupg-utils	2.2.27-2+deb11u2.astra5	GNU p
golang-github-containers-common	0.50.1+ds1-2+b1	Comm
golang-github-containers-image	5.23.1-3+b1	Config
gpg	2.2.27-2+deb11u2.astra5	GNU P
gpg-agent	2.2.27-2+deb11u2.astra5	GNU p
gpg-wks-client	2.2.27-2+deb11u2.astra5	GNU p
gpg-wks-server	2.2.27-2+deb11u2.astra5	GNU p
gpgconf	2.2.27-2+deb11u2.astra5	GNU p
gpgsm	2.2.27-2+deb11u2.astra5	GNU p
gpgv	2.2.27-2+deb11u2.astra5	GNU p
grep	3.3-1	GNU g
gzip	1.9-3+deb10u1	GNU c
hostname	3.21	utility
init-system-helpers	1.56+nmu1	helper
iptables	1.8.5-3~bpo10+1	admini
iso-codes	4.2-1	ISO lan
krb5-config	2.6astra1	Config
krb5-locales	1.18.3-6+deb11u3astra1	interna
krb5-multidev:amd64	1.18.3-6+deb11u3astra1	develo
less	487-0.1	pager
libacl1:amd64	2.2.53-4	access
libapparmor1:amd64	2.13.2-10	change
libapt-inst2.0:amd64	1.8.2.3+ci202309131156+astra26	deb pa
libapt-pkg5.0:amd64	1.8.2.3+ci202309131156+astra26	packag
libargon2-1:amd64	0~20171227-0.2	memo
libasan5:amd64	8.3.0-6	Addres
libassuan0:amd64	2.5.2-1	IPC lib
libatomic1:amd64	8.3.0-6	suppor
libattr1:amd64	1:2.4.48-4+ci202107131632+astra2	extend
libaudit-common	1:2.8.5-2ubuntu6+ci202307261837+astra5	Dynam
libaudit1:amd64	1:2.8.5-2ubuntu6+ci202307261837+astra5	Dynam
libbind9-161:amd64	1:9.11.5.P4+dfsg-5.1+deb10u9+ci202308311745+astra1	BIND9
libbinutils:amd64	2.31.1-16+ci202306051252+astra11	GNU b

Пакет	Версия	Описа
libblkid1:amd64	2.33.1-0.1.astra6se5	block
libbsd0:amd64	0.9.1-2+deb10u1	utility
libbz2-1.0:amd64	1.0.6-9.2~deb10u1	high-q
libc-bin	2.28-10+deb10u2+ci202309131632+astra7	GNU C
libc-dev-bin	2.28-10+deb10u2+ci202309131632+astra7	GNU C
libc6:amd64	2.28-10+deb10u2+ci202309131632+astra7	GNU C
libc6-dev:amd64	2.28-10+deb10u2+ci202309131632+astra7	GNU C
libcap-ng0:amd64	0.7.9-2	An alte
libcap2:amd64	1:2.25-2	POSIX
libcbor0.8:amd64	0.8.0-2astra1	library
libcc1-0:amd64	8.3.0-6	GCC co
libcom-err2:amd64	1.44.5-1+deb10u3+ci202212051923+astra1	comm
libcryptsetup12:amd64	2:2.1.0-5+deb10u2	disk en
libcurl3-gnutls:amd64	7.64.0-4+deb10u5+astra6	easy-to
libdb5.3:amd64	5.3.28+dfsg2-1+ci202311141606+astra1	Berkel
libdbus-1-3:amd64	1.12.24-0+deb11u1.astra.se8+ci2	simple
libdebconfclient0:amd64	0.249astra11	Debian
libdevmapper1.02.1:amd64	2:1.02.175-2	Linux
libdns1104:amd64	1:9.11.5.P4+dfsg-5.1+deb10u9+ci202308311745+astra1	DNS S
libdpkg-perl	1.19.8.astra.se3	Dpkg p
libedit2:amd64	3.1-20181209-1	BSD ed
liberror-perl	0.17027-2	Perl m
libexpat1:amd64	2.2.6-2+deb10u6	XML pa
libexpat1-dev:amd64	2.2.6-2+deb10u6	XML pa
libext2fs2:amd64	1.44.5-1+deb10u3+ci202212051923+astra1	ext2/e
libfakeroot:amd64	1.23-1	tool fo
libfdisk1:amd64	2.33.1-0.1.astra6se5	fdisk p
libffi6:amd64	3.2.1-9	Foreign
libfido2-1:amd64	1.6.0-2	library
libflygetexe	1.0.13+ci3	libflyge
libfstrm0:amd64	0.4.0-1	Frame
libfuse3-3:amd64	3.4.1-1+deb10u1+b1	Filesys
libgcc-8-dev:amd64	8.3.0-6	GCC su
libgcc1:amd64	1:8.3.0-6	GCC su
libgcrypt20:amd64	1.8.7-6.astra1	LGPL C
libgdbm-compat4:amd64	1.18.1-4	GNU d
libgdbm6:amd64	1.18.1-4	GNU d
libgeoip1:amd64	1.6.12-1	non-DN
libglib2.0-0:amd64	2.58.3-2+deb10u4+ci202308281209+astra1	GLib li
libglib2.0-data	2.58.3-2+deb10u4+ci202308281209+astra1	Comm
libgmp10:amd64	2:6.1.2+dfsg-4+deb10u1	Multip
libgnutls30:amd64	3.6.7-4+deb10u11+astra1	GNU T
libgomp1:amd64	8.3.0-6	GCC O
libgost	2.0.2-5+ci3	This pa
libgpg-error0:amd64	1.35-1	GnuPG
libgpgme11:amd64	1.12.0-6	GPGME
libgpm2:amd64	1.20.7-5	Genera
libgssapi-krb5-2:amd64	1.18.3-6+deb11u3astra1	MIT Ke
libgssrpc4:amd64	1.18.3-6+deb11u3astra1	MIT Ke
libhogweed4:amd64	3.4.1-1+deb10u1	low lev
libicu63:amd64	63.1-6+deb10u2	Interna
libidn11:amd64	1.33-2.2	GNU Li
libidn2-0:amd64	2.0.5-1+deb10u1+ci202308141422+astra1	Interna
libip4tc2:amd64	1.8.5-3~bpo10+1	netfilte
libip6tc2:amd64	1.8.5-3~bpo10+1	netfilte
libisc1100:amd64	1:9.11.5.P4+dfsg-5.1+deb10u9+ci202308311745+astra1	ISC Sh
libisccc161:amd64	1:9.11.5.P4+dfsg-5.1+deb10u9+ci202308311745+astra1	Comm

Пакет	Версия	Описа
libisccfg163:amd64	1:9.11.5.P4+dfsg-5.1+deb10u9+ci202308311745+astra1	Config
libisl19:amd64	0.20-2	manip
libitm1:amd64	8.3.0-6	GNU T
libjansson4:amd64	2.12-1	C libra
libjson-c3:amd64	0.12.1+ds-2+deb10u1	JSON n
libk5crypto3:amd64	1.18.3-6+deb11u3astra1	MIT Ke
libkadm5clnt-mit12:amd64	1.18.3-6+deb11u3astra1	MIT Ke
libkadm5srv-mit12:amd64	1.18.3-6+deb11u3astra1	MIT Ke
libkdb5-10:amd64	1.18.3-6+deb11u3astra1	MIT Ke
libkeyutils1:amd64	1.6-6	Linux K
libkmod2:amd64	27+20200310-2ubuntu1	libkmo
libkrb5-3:amd64	1.18.3-6+deb11u3astra1	MIT Ke
libkrb5-dev:amd64	1.18.3-6+deb11u3astra1	header
libkrb5support0:amd64	1.18.3-6+deb11u3astra1	MIT Ke
libksba8:amd64	1.3.5-2+deb10u2	X.509
libldap-2.4-2:amd64	2.4.57+dfsg-3+deb11u1~bpo10+1astra1	OpenL
libldap-common	2.4.57+dfsg-3+deb11u1~bpo10+1astra1	OpenL
liblmdb0:amd64	0.9.22-1	Lightn
liblocale-gettext-perl	1.07-3	modul
liblsan0:amd64	8.3.0-6	LeakSa
liblwres161:amd64	1:9.11.5.P4+dfsg-5.1+deb10u9+ci202308311745+astra1	Lightw
liblz4-1:amd64	1.9.3-2	Fast LZ
liblzma5:amd64	5.2.4-1+deb10u1	XZ-for
libmagic-mgc	1:5.35-4+deb10u1	File typ
libmagic1:amd64	1:5.35-4+deb10u1	Recogn
libmnl0:amd64	1.0.4-2	minima
libmount1:amd64	2.33.1-0.1.astra6se5	device
libmpc3:amd64	1.1.0-1	multip
libmpdec2:amd64	2.4.2-2	library
libmpfr6:amd64	4.0.2-1	multip
libmpx2:amd64	8.3.0-6	Intel m
libncurses-dev:amd64	6.1+20181013-2+deb10u3+ci202309121705+astra1	develo
libncurses5-dev:amd64	6.1+20181013-2+deb10u3+ci202309121705+astra1	transit
libncurses6:amd64	6.1+20181013-2+deb10u3+ci202309121705+astra1	shared
libncursesw6:amd64	6.1+20181013-2+deb10u3+ci202309121705+astra1	shared
libnetfilter-conntrack3:amd64	1.0.7-1	Netfilte
libnettle6:amd64	3.4.1-1+deb10u1	low lev
libnfnetlink0:amd64	1.0.1-3	Netfilte
libnftables1:amd64	0.9.6-1~bpo10+1	Netfilte
libnftnl11:amd64	1.1.7-1~bpo10+1	Netfilte
libnghttp2-14:amd64	1.36.0-2+deb10u2	library
libnpth0:amd64	1.6-1	replace
libnss-systemd:amd64	241-7~deb10u8astra.se32+ci4	nss mo
libp11-kit0:amd64	0.23.15-2+deb10u1	library
libpam-modules:amd64	1.3.1-5.astra.se11	Plugga
libpam-modules-bin	1.3.1-5.astra.se11	Plugga
libpam-runtime	1.3.1-5.astra.se11	Runtim
libpam-systemd:amd64	241-7~deb10u8astra.se32+ci4	system
libpam0g:amd64	1.3.1-5.astra.se11	Plugga
libparsec-base3	3.4+ci41	base li
libparsec-cap3	3.4+ci41	capabi
libpcre2-8-0:amd64	10.32-5+deb10u1+ci202308291416+astra1	New Pe
libpcre3:amd64	2:8.39-13	Old Pe
libpdp	3.4+ci41	library
libperl5.28:amd64	5.28.1-6+deb10u1+ci202305241311+astra2	shared
libprotobuf-c1:amd64	1.3.3-1+ci202311091556+astra2	Protoc
libpsl5:amd64	0.20.2-2	Library

Пакет	Версия	Описа
libpython3-stdlib:amd64	3.7.3-1	interac
libpython3.7-minimal:amd64	3.7.3-2+deb10u5+ci202309191152+astra10	Minima
libpython3.7-stdlib:amd64	3.7.3-2+deb10u5+ci202309191152+astra10	Interac
libpython3.9:amd64	3.9.18-1+astra1	Shared
libpython3.9-dev:amd64	3.9.18-1+astra1	Heade
libpython3.9-minimal:amd64	3.9.18-1+astra1	Minima
libpython3.9-stdlib:amd64	3.9.18-1+astra1	Interac
libquadmath0:amd64	8.3.0-6	GCC Q
libreadline7:amd64	7.0-5	GNU re
librtmp1:amd64	2.4+20151223.gitfa8646d.1-2	toolkit
libsasl2-2:amd64	2.1.27+dfsg-1+deb10u2+b1	Cyrus
libsasl2-modules-db:amd64	2.1.27+dfsg-1+deb10u2+b1	Cyrus
libseccomp2:amd64	2.5.1-1~bpo10+1	high le
libselinux1:amd64	2.8-1	SELinu
libsemanage-common	2.8-2	Comm
libsemanage1:amd64	2.8-2	SELinu
libsepol1:amd64	2.8-1	SELinu
libsigsegv2:amd64	2.12-2	Library
libslirp0:amd64	4.4.0-1+deb11u2	Genera
libsmartcols1:amd64	2.33.1-0.1.astra6se5	smart
libsqlite3-0:amd64	3.34.1-3+ci202212121148+astra1	SQLite
libss2:amd64	1.44.5-1+deb10u3+ci202212051923+astra1	comma
libssh2-1:amd64	1.11.0-2	SSH2
libssl1.1:amd64	1.1.1n-0+deb11u4-deb11u5-astra3	Secure
libstdc++-8-dev:amd64	8.3.0-6	GNU S
libstdc++6:amd64	8.3.0-6	GNU S
libsubid4:amd64	1:4.13+dfsg1-1.astra.se8	subord
libsystemd0:amd64	241-7~deb10u8astra.se32+ci4	system
libtasn1-6:amd64	4.16.0-2+deb11u1	Manag
libtcl8.6:amd64	8.6.9+dfsg-2+ci202302131725+astra1	Tcl (the
libtext-charwidth-perl	0.04-7.1	get dis
libtext-iconv-perl	1.7-5	conver
libtext-wrapi18n-perl	0.06-7.1	interna
libtinfo6:amd64	6.1+20181013-2+deb10u3+ci202309121705+astra1	shared
libtsan0:amd64	8.3.0-6	Thread
libubsan1:amd64	8.3.0-6	UBSan
libudev1:amd64	241-7~deb10u8astra.se32+ci4	libudev
libunistring2:amd64	0.9.10-1	Unicod
libuuid1:amd64	2.33.1-0.1.astra6se5	Univer
libx11-6:amd64	2:1.8.7-1astra3	X11 cli
libx11-data	2:1.8.7-1astra3	X11 cli
libxau6:amd64	1:1.0.8-1	X11 au
libxcb1:amd64	1.15-1astra.se3	X C Bin
libxdmcp6:amd64	1:1.1.2-3	X11 Di
libxext6:amd64	2:1.3.3-1	X11 m
libxml2:amd64	2.9.4+dfsg1-7+deb10u6+ci202308291455+astra2	GNOM
libxmuu1:amd64	2:1.1.2-2	X11 m
libxtables12:amd64	1.8.5-3~bpo10+1	netfilte
libzstd1:amd64	1.5.2+dfsg-1	fast los
linux-libc-dev:amd64	5.4.0-54astra7+ci67	latest
linux-libc-dev-5.4.0-162:amd64	5.4.0-162.astra1+ci29	Linux
login	1:4.13+dfsg1-1.astra.se8	system
lsb-base	10.2019051400+ci202012021351+astra2	Linux S
lsb-release	10.2019051400+ci202012021351+astra2	Linux S
make	4.2.1-1.2	utility
manpages	4.16-2	Manua
manpages-dev	4.16-2	Manua

Пакет	Версия	Описа
mawk	1.3.3-17	a patte
mime-support	3.62	MIME f
mount	2.33.1-0.1.astra6se5	tools fo
ncurses-base	6.1+20181013-2+deb10u3+ci202309121705+astra1	basic t
ncurses-bin	6.1+20181013-2+deb10u3+ci202309121705+astra1	termin
netbase	6.1~bpo10+1	Basic T
nftables	0.9.6-1~bpo10+1	Progra
openssh-client	1:8.4p1-2~deb10u1astra6se10	secure
openssl	1.1.1n-0+deb11u4-deb11u5-astra3	Secure
passwd	1:4.13+dfsg1-1.astra.se8	change
patch	2.7.6-3+deb10u1	Apply a
perl	5.28.1-6+deb10u1+ci202305241311+astra2	Larry W
perl-base	5.28.1-6+deb10u1+ci202305241311+astra2	minima
perl-modules-5.28	5.28.1-6+deb10u1+ci202305241311+astra2	Core P
pinentry-curses	1.1.0-2	curses
podman	4.3.1+ds1-6astra1+ci5+b1	engine
publicsuffix	20190415.1030-1	accura
python-apt-common	1.8.4.3+ci202309111239+astra1	Python
python3	3.7.3-1	interac
python3-apt	1.8.4.3+ci202309111239+astra1	Python
python3-minimal	3.7.3-1	minima
python3.7	3.7.3-2+deb10u5+ci202309191152+astra10	Interac
python3.7-minimal	3.7.3-2+deb10u5+ci202309191152+astra10	Minima
python3.9	3.9.18-1+astra1	Interac
python3.9-ansible-compat	4.1.11	Ansible
python3.9-ansible-core	2.15.10	Radica
python3.9-ansible-lint	6.22.2	Checks
python3.9-attrs	23.2.0	Classe
python3.9-black	24.4.2	The un
python3.9-bracex	2.4	Bash s
python3.9-cffi	1.16.0	Foreign
python3.9-click	8.1.7	Compo
python3.9-click-help-colors	0.9.4	Coloriz
python3.9-cryptography	42.0.7	crypto
python3.9-dev	3.9.18-1+astra1	Heade
python3.9-distutils	3.9.18-1+astra1	distuti
python3.9-enrich	1.2.7	enrich
python3.9-filelock	3.14.0	A platf
python3.9-importlib-resources	5.0.7	Read r
python3.9-jinja2	3.1.4	A very
python3.9-jsonschema	4.22.0	An imp
python3.9-jsonschema-specifications	2023.12.1	The JS
python3.9-lib2to3	3.9.18-1+astra1	lib2to3
python3.9-markdown-it-py	3.0.0	Python
python3.9-markupsafe	2.1.5	Safely
python3.9-mdurl	0.1.2	Markd
python3.9-minimal	3.9.18-1+astra1	Minima
python3.9-molecule	6.0.3	Molecu
python3.9-mypy-extensions	1.0.0	Type sy
python3.9-packaging	24.0	Core u
python3.9-pathspec	0.12.1	Utility
python3.9-pip	23.1.2-1+aa1	The Py
python3.9-platformdirs	4.2.2	A smal
python3.9-pluggy	1.5.0	plugin
python3.9-pycparser	2.22	C pars
python3.9-pygments	2.18.0	Pygme
python3.9-pyyaml	6.0.1	YAML p

Пакет	Версия	Описа
python3.9-referencing	0.35.1	JSON R
python3.9-resolvelib	1.0.1	Resolv
python3.9-rich	13.7.1	Rende
python3.9-rpds-py	0.18.1	Python
python3.9-ruamel.yaml	0.18.6	ruame
python3.9-ruamel.yaml.clib	0.2.8	C vers
python3.9-setuptools	70.1.0	Easily
python3.9-subprocess-tee	0.4.1	subpro
python3.9-tomli	2.0.1	A lil“ T
python3.9-typing-extensions	4.11.0	Backpo
python3.9-venv	3.9.18-1+astra1	Interac
python3.9-wcmatch	8.5.2	Wildca
python3.9-yamllint	1.35.1	A linte
readline-common	7.0-5	GNU re
runc	1.1.7+astra5	Open C
sed	4.7-1	GNU st
sensible-utils	0.0.12	Utilitie
shared-mime-info	2.2-1	FreeDe
slirp4netns	1.0.1-1	User-m
sshpass	1.06-1	Non-in
systemd	241-7~deb10u8astra.se32+ci4	system
systemd-sysv	241-7~deb10u8astra.se32+ci4	system
sysvinit-utils	2.93-8	System
tar	1.34+dfsg-1+ci202304071507+astra2	GNU v
tcl-expect:amd64	5.45.4-2+ci202011031319	Autom
tcl8.6	8.6.9+dfsg-2+ci202302131725+astra1	Tcl (the
tofu	1.6.2	OpenT
tzdata	2021a-0+deb10u11	time z
uidmap	1:4.13+dfsg1-1.astra.se8	progra
util-linux	2.33.1-0.1.astra6se5	miscel
xauth	1:1.0.10-1	X auth
xdg-user-dirs	0.18-1astra2	tool to
xz-utils	5.2.4-1+deb10u1	XZ-for
zlib1g:amd64	1:1.2.11.dfsg-1+deb10u2	compr

Пакеты Python

Пакет
Cython
Jinja2
MarkupSafe
PyJWT
PyNaCl
PyYAML
Pygments
aenum
aiohttp
aiosignal
ansible-compat
ansible-core @ file:///docker_share/dist/ansible_core-2.15.10-py3-none-any.whl
ansible-lint
ansible-pylibssh
ansible-runner
appdirs

Таблица 4 – продолжение с пре

Пакет
argcomplete
async-timeout
attrs
bcrypt
black
bracex
cachetools
certifi
cffi
charset-normalizer
click-help-colors
click
coverage
cryptography
decorator
dicttoxml
distlib
distro
docutils
dogpile.cache
dumb-init
enrich
filelock
frozenlist
future
gnupg
google-auth
googleapis-common-protos
grpcio
gssapi
hvac
idna
ijson
importlib-metadata
importlib_resources
ipaddress
iso8601
jmespath
jsonpatch
jsonpointer
jsonschema-specifications
jsonschema
jxmlease
kerberos
keystoneauth1
kubernetes
lockfile
lxml
markdown-it-py
mdurl
molecule @ file:///docker_share/dist/molecule-6.0.3-py3-none-any.whl
molecule-plugins
molecule_molecule_brest @ file:///tmp/molecule-brest.tar.gz#sha256=f59386d697a371adc9a9523774965
molecule_molecule_yandexcloud @ file:///tmp/molecule-yandexcloud.tar.gz#sha256=741178b00a34610c
multidict
munch
mypy_extensions

Таблица 4 – продолжение с пре

Пакет
ncclient
netaddr
netifaces
oauthlib
openssl-engine
openstacksdk
os-service-types
packaging
paramiko
passlib
pathspec
pbr
pexpect
platformdirs
pluggy
protobuf
psutil
psycopg2-binary
ptyprocess
pyOpenSSL
pyasn1-modules
pyasn1
pycparser
pyhcl
pynetbox
pyone
pyparsing
pypsrp
pyrsistent
pysnow
pyspnego
python-apt
python-daemon
python-dateutil
python-magic
pytz
pyvmomi
pywinrm
referencing
requests-oauthlib
requests
requests_ntlm
requestsexceptions
resolvelib
rich
rpds-py
rpm
rsa
ruamel.yaml.clib
ruamel.yaml
scp
selectors2
selinux
six
stevedore
subprocess-tee
tblib

Таблица 4 – продолжение с пре

Пакет
textfsm
toml
tomli
tomli_w
ttp
typing_extensions
urllib3
urllib_gssapi
wcmatch
websocket-client
xmltodict
yamllint
yandexcloud
yarl
zabbix-api
zipp

8.8.6 Файл определения среды исполнения

Файл определения среды исполнения имеет формат YAML и содержит настройки, используемые *Ansible Builder* при создании образа контейнера среды исполнения.

version

Целочисленное значение, которое задает версию схемы файла определения среды исполнения. Значение по умолчанию – 1. При использовании Ansible Builder версии `3.x` рекомендуется в этом поле установить значение 3:

```
---
version: 3
```

build_arg_defaults

Значения по умолчанию для переменных окружения, используемых при сборке образа среды исполнения:

- `ANSIBLE_GALAXY_CLI_COLLECTION_OPTS`;
- `ANSIBLE_GALAXY_CLI_ROLE_OPTS`;
- `PKGMGR_PRESERVE_CACHE`.

Значения, указанные в `build_arg_defaults`, будут сохранены в файл определения образа `Containerfile` и будут применены даже при запуске сборки образа среды исполнения вручную.

Подробное описание аргументов см. в *справочнике параметров и команд Ansible Builder*.

Примечание: Значения, заданные в значении аргумента командной строки `--build-arg`, имеют более высокий приоритет, чем заданные в файле определения среды исполнения.

dependencies

В разделе dependencies указываются пакеты и библиотеки, которые необходимо установить в конечный образ.

dependencies.ansible_core

Версия пакета ansible-core.

Важно: Версия пакета указывается в соответствии с требованиями **PEP 508**[178].

Список 13: Пример заполнения

```
dependencies:
  ansible_core:
    package_pip: ansible-core==2.15.6
```

dependencies.ansible_runner

Версия пакета ansible_runner.

Важно: Версия пакета указывается в соответствии с требованиями **PEP 508**[179].

Список 14: Пример заполнения

```
dependencies:
  ansible_runner:
   package_pip: ansible-runner==2.3.2
```

dependencies.galaxy

Путь к файлу со списком зависимостей или список зависимостей.

Путь к файлу зависимостей указывается относительно каталога, содержащего файл определения среды исполнения. Рекомендуемое имя файла – requirements.yml.

При явном указании зависимостей в файле определения среды исполнения списки коллекций и ролей указываются в значениях параметров dependencies.galaxy.collections и dependencies.galaxy.roles соответственно.

Примеры заполнения:

Список 15: Имя файла со списком зависимостей

```
dependencies:
  galaxy: requirements.yml
```

Список 16: Список зависимостей

```
dependencies:
  galaxy:
    collections:
      - astra.ald_pro
      - astra.brest
```

(continues on next page)

[178] https://peps.python.org/pep-0508/
[179] https://peps.python.org/pep-0508/

(продолжение с предыдущей страницы)

```
      - astra.rupost
        version: 4.0.0
```

dependencies.python

Путь к файлу зависимостей Python.

Важно: Содержимое файла зависимостей Python должно быть оформлено в соответствии с требованиями **PEP 508**[180].

Путь к файлу зависимостей указывается относительно каталога, содержащего файл определения среды исполнения. Рекомендуемое имя файла – `requirements.txt`.

dependencies.python_interpreter.package_system

Название системного пакета, отвечающего за интерпретатор Python нужной версии.

Список 17: Пример заполнения

```
dependencies:
  python_interpreter:
    package_system: python3.11
```

dependencies.python_interpreter.python_path

Путь к интерпретатору Python, который должен использоваться по умолчанию.

Список 18: Пример заполнения

```
dependencies:
  python_interpreter:
    python_path: /usr/bin/python3.11
```

dependencies.system

Имя файла в формате BINDEP[181] со списком системных пакетов или список системных пакетов, которые должны быть установлены в образе.

Рекомендуемое имя файла – `bindep.txt`.

[180] https://peps.python.org/pep-0508/
[181] https://docs.opendev.org/opendev/bindep/latest/readme.html

Список 19: Пример заполнения

```
dependencies:
  system: bindep.txt
```

dependencies.exclude

Список зависимостей, которые *не* должны быть установлены в образ. Значения задаются в следующих параметрах:

- `dependencies.exclude.python` – список зависимостей Python.
- `dependencies.exclude.system` – список системных зависимостей.
- `dependencies.exclude.all_from_collections` – исключить все системные зависимости и зависимости Python, связанные с указанными коллекциями.

Сопоставление названий исключаемых зависимостей может выполняться по простому совпадению строки или регулярному выражению. В обоих случаях сравнение выполняется без учета регистра.

Чтобы строка интерпретировалась как регулярное выражение, добавьте в ее начало символ ~.

Примечание: Регулярное выражение должно соответствовать полному названию пакета или коллекции. Например, регулярное выражение `~foo.` соответствует строке `foobar` частично, а `~foo.+` – полностью.

При обеих формах сопоставления строка исключения будет сравниваться с простым названием пакета или коллекции. Например, чтобы исключить системный пакет с названием `foo [!platform:gentoo]`, строка исключения должна иметь значение `foo`. Чтобы исключить пакет Python `bar==1.0.0`, строка должна иметь значение `bar`.

Примеры:

Список 20: Простое сопоставление

```
dependencies:
 exclude:
   python:
     - docker
   system:
     - python3-Cython
```

Список 21: Регулярное выражение

```
dependencies:
  exclude:
    all_from_collections:
      - ~community\..+
```

Список 22: Имена файлов, содержащих различные зависимости

```
dependencies:
  python: requirements.txt
  system: bindep.txt
  galaxy: requirements.yml
  ansible_core:
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    package_pip: ansible-core==2.15.6
  ansible_runner:
    package_pip: ansible-runner==2.3.1
  python_interpreter:
    package_system: "python310"
    python_path: "/usr/bin/python3.10"
```

Список 23: Встроенные значения

```
dependencies:
  python:
    - pywinrm
  system:
    - iputils [platform:rpm]
  galaxy:
    collections:
      - name: community.windows
      - name: ansible.utils
        version: 2.10.1
  ansible_core:
    package_pip: ansible-core==2.15.6
  ansible_runner:
    package_pip: ansible-runner==2.3.1
  python_interpreter:
    package_system: "python310"
    python_path: "/usr/bin/python3.10"
```

images.base_image.name

Название базового образа контейнера.

Список 24: Пример заполнения

```
images:
  base_image:
    name: hub.astra-automation.ru/aa/aa-base-ee:latest
```

additional_build_files

Список файлов, которые должны быть добавлены в контекст сборки. Затем на них можно ссылаться или копировать в образ с помощью команд, указанных в `additional_build_steps`.

Каждый элемент списка должен быть словарем, содержащим следующие ключи:

- `src` – путь к исходному файлу или каталогу, который необходимо добавить в контекст сборки. Это может быть либо абсолютный путь (например, `/home/user/.ansible.cfg`), либо путь относительно файла описания среды исполнения. Относительные пути могут быть глобальным выражением, соответствующим одному или нескольким файлам (например, `files/*.cfg`).
- `dest` – путь к целевому размещению файла в подкаталоге контекста сборки. Если путь содержит названия несуществующих каталогов, они будут созданы автоматически.

Список 25: Пример заполнения

```
additional_build_files:
  - src: files/ansible.cfg
    dest: configs
```

additional_build_steps

Дополнительные команды, выполняемые на различных этапах сборки. Названия вложенных параметров состоят из префикса prepend_ или append_ и названия этапа.

Команды, перечисленные в разделах с префиксом prepend_, выполняются *до* соответствующих этапов.

Команды, перечисленные в разделах с префиксом append_, выполняются *после* соответствующих этапов.

Таким образом, необходимые команды могут быть указаны в следующих разделах, в порядке выполнения:

1. prepend_base;
2. append_base;
3. prepend_galaxy;
4. append_galaxy;
5. prepend_builder;
6. append_builder;
7. prepend_final;
8. after_final.

Команды могут быть заданы в виде многострочной записи или в отдельных элементах списка:

Список 26: Многострочная запись

```
additional_build_steps:
  prepend_final: |
    RUN whoami
    RUN cat /etc/os-release
```

Список 27: Список команд

```
additional_build_steps:
  append_final:
     - RUN echo
     - RUN ls -la /etc
```

ГЛАВА 9

Управление событиями

Обработчик событий Event-Driven Ansible оснащает платформу интеллектуальной составляющей по обработке различных событий, возникающих в различных компонентах инфраструктуры пользователя. Преимуществами применения обработчика событий являются:

- существенное снижение объема ручной работы;
- быстрое и своевременное реагирование на изменившуюся ситуацию в инфраструктуре.

Важно: Эта часть документации находится в стадии разработки.

ГЛАВА 10

Средства аналитики

Важно: Эта часть документации находится в стадии разработки.

ГЛАВА 11

Примеры решений

Приведенные примеры показывают различные области применения Automation Controller, Private Automation Hub, Ansible Navigator и коллекций Ansible для выполнения типовых задач управления IT-инфраструктурой.

11.1 Типовой проект

Использование типовой структуры проектов Ansible позволяет упростить их создание, использование и поддержку.

11.1.1 Файлы и каталоги

Структура типового проекта Ansible, для запуска которого используется *Automation Controller* или *Ansible Navigator* имеет следующий вид:

```
project/
├── playbooks/
│   ├── playbook1.yml
│   ├── playbook2.yml
│   │   ...
│   └── playbookK.yml
├── inventory/
│   ├── inventory1.yml
│   ├── inventory2.yml
│   │   ...
│   └── inventoryL.yml
├── files/
│   ├── file1
│   ├── file2
│   │   ...
│   └── fileM
├── vars/
│   ├── vars1.yml
│   ├── vars2.yml
│   │   ...
│   └── varsN.yml
└── execution-environment.yml
```

Указанные файлы и каталоги используются для следующих целей:

- `playbooks/`.

  Каталог с файлами playbook. Если проект содержит только один файл playbook, допускается разместить его в корневом каталоге проекта.

- `inventory/`.

  Каталог с файлами инвентаря. Рекомендуемый формат файлов – YAML.

  Если инвентарь состоит из одного файла, допускается разместить его в корневом каталоге проекта. Рекомендуемое имя файла в этом случае – `inventory.yml`.

  Примечание: Инвентарь необходимо заполнять только в том случае, когда проект разрабатывается для определенного списка управляемых узлов.

- `files/`.

  Каталог с файлами, используемыми при сборке образа среды исполнения.

- `vars/`.

  Каталог с файлами с переменными, используемыми в playbook.

  Если все переменные хранятся в одном файле, допускается разместить его в корневом каталоге проекта. Рекомендуемое имя файла в этом случае – `vars.yml`.

- `execution-environment.yml`.

  Файл определения среды исполнения, используемой для запуска playbook. Подробности см. в *инструкции по созданию собственных образов среды исполнения*.

11.1.2 Рекомендуемый порядок действий

При работе над проектом рекомендуется следующий порядок действий:

1. Определите список используемых коллекций, изучите особенности их использования.
2. Разработайте код и отладьте его с помощью *Ansible Navigator*.
3. Если проект зависит от коллекций, не входящих в состав используемых вами образов среды исполнения, выполните следующие действия:
   1. Создайте собственный образ среды исполнения, включающий все необходимые коллекции. Примеры создания такого образа представлены в *описании Ansible Builder*.
   2. Опубликуйте созданный образ в Private Automation Hub, следуя *инструкции по загрузке образа с рабочей станции*.
4. Опубликуйте код проекта в системе управления исходным кодом (SCM, Source Code Management).

11.1.3 Применение проекта

Для запуска проекта рекомендуется использовать *Automation Controller* или *Ansible Navigator*.

Automation Controller

Чтобы использовать Automation Controller для запуска проекта, выполните следующие действия:

1. Создайте полномочия нужного типа, следуя *инструкции*.

   Понадобятся как минимум следующие полномочия:

Тип полномочия	Тип ресурса
Реестр контейнеров	Private Automation Hub
Система управления версиями (Source Control)	Система управления кодом, в которой хранится проект
Машина (Machine)	Управляемые узлы

2. Добавьте в Automation Controller образ среды исполнения, следуя *инструкции по загрузке образов*.
3. Создайте проект согласно *инструкции по созданию проектов*.
4. *Создайте инвентарь нужного типа* и добавьте в него сведения об управляемых узлах.
5. Используйте playbook из проекта для *создания шаблона заданий*.
6. Запустите задание, следуя *инструкции по запуску заданий на основе шаблона*.

Ansible Navigator

Чтобы использовать Ansible Navigator для запуска проекта, выполните следующие действия:

1. Загрузите код проекта в рабочий каталог.
2. Создайте в корневом каталоге проекта файл `ansible.cfg`, в котором укажите необходимые настройки, например, путь к инвентарю.
3. Для запуска playbook используйте команду:

   ```
   ansible-navigator run --eei <image> path/to/playbook.yml
   ```

   Здесь:

   - <image> – название образа среды исполнения, используемого для запуска заданий;
   - `/path/to/playbook.yml` – путь к файлу playbook относительно корневого каталога проекта.

11.2 Развертывание веб-сервера NGINX

Использование коллекции `astra.nginx` из реестра Automation Hub[182] позволяет автоматизировать процесс установки и настройки веб-сервера NGINX на необходимом количестве узлов.

[182] https://hub.astra-automation.ru/

11.2.1 Описание сценария

Процесс установки и настройки веб-сервера NGINX состоит из следующих этапов:

1. Создание стенда.

   В качестве стенда рекомендуется использовать управляемые узлы, обладающие следующими характеристиками:

Характеристика	Значение
Кол-во ядер CPU	≥ 2
Кол-во RAM, ГБ	≥ 2
Размер хранилища, ГБ	≥ 20
ОС	Astra Linux Special Edition 1.7.5uu1

   Для подключения к стенду используется протокол SSH.

   Если у вас нет подходящего стенда, вы можете создать его с помощью *Vagrant*, используя приложенный `Vagrantfile`.

2. Подготовка проекта Ansible.

   - `requirements.yml` – зависимости;
   - `ansible.cfg` – настройки Ansible;
   - `inventory.ini` – инвентарь;
   - `playbooks/nginx.yml` – playbook, содержащий сценарий настройки управляемых узлов.

   В демонстрационных целях для NGINX задаются следующие настройки:

   - `worker_connections` – 2048;
   - `keepalive_timeout` – 30.

3. Запуск playbook.
4. Проверка работоспособности развернутой инфраструктуры.

11.2.2 Подготовка к работе

Подготовьте окружение к управлению задачами автоматизации:

1. Настройте управляемые узлы согласно инструкций:
   - *Доступ по SSH с помощью ключей*
   - *Использование sudo без ввода пароля*
2. Согласно *инструкции* настройте Private Automation Hub на синхронизацию с Automation Hub[183].
3. Синхронизируйте Private Automation Hub с Automation Hub. Убедитесь, что коллекция `astra.nginx` загружена в репозиторий, доступный пользователям.
4. Изучите описание[184] коллекции `astra.nginx`.
5. Подготовьте каталог для хранения файлов проекта.

[183] https://hub.astra-automation.ru/
[184] https://hub.astra-automation.ru/ui/repo/validated/astra/nginx/docs/

Automation Controller

Если для управления задачами автоматизации используется Automation Controller, выполните следующие действия:

1. В одном из сервисов хранения репозиториев Git создайте пустой репозиторий для проекта.
2. Клонируйте этот репозиторий на локальный компьютер.

Ansible Navigator

Если для управления задачами автоматизации используется Ansible Navigator, создайте в любом удобном месте пустой каталог, например:

```
mkdir ~/ansible-nginx/
```

11.2.3 Подготовка проекта Ansible

Подготовьте ресурсы, необходимые для использования Ansible.

1. Создайте файл `ansible.cfg` со следующим содержимым:

```
[defaults]
ansible_python_interpreter = /usr/bin/python3
host_key_checking = false
inventory = inventory.ini

[galaxy]
server_list = galaxy, private_automation_hub

[galaxy_server.galaxy]
url = https://galaxy.ansible.com/

[galaxy.private_automation_hub]
url = <private_automation_hub_repository_url>
```

   Согласно *инструкции* добавьте в `ansible.cfg` настройки, обеспечивающие доступ к репозиторию Private Automation Hub, содержащему коллекцию `astra.nginx`.

2. Создайте файл инвентаря `inventory.ini`, например:

```
[all]
node1.example.com   ansible_host=192.168.56.11   ansible_user=vagrant
```

   В этом примере показаны настройки подключения к одному управляемому узлу.

3. Создайте файл зависимостей Ansible `requirements.yml`:

```
---
collections:
  - name: astra.nginx
    version: 1.8.3
```

   В этом файле указаны параметры подключаемой коллекции:

   - `name` – название;
   - `version` – версия.

   Важно: Для получения актуального номера версии коллекции обратитесь к ее описанию в Private Automation Hub.

4. Создайте подкаталог `playbooks/`, а в нем – файл сценария `nginx.yml`:

```
---
- name: Install and configure NGINX server
  hosts: all
  become: true

  roles:
    - role: astra.nginx.nginx
      vars:
        nginx_worker_connections: 2048
        nginx_keepalive_timeout: 30
```

В этом файле настройки NGINX заданы через значения переменных роли `astra.nginx.nginx`.

5. Выполните дополнительные настройки, соответствующие выбранному инструменту для управления задачами автоматизации.

Automation Controller

Если для запуска заданий автоматизации используется Automation Controller, выполните следующие действия:

1. Зафиксируйте сделанные в проекте изменения и опубликуйте их в репозитории Git.
2. Создайте в контроллере полномочие, обеспечивающее доступ к Private Automation Hub. Для этого следуйте *инструкции*.

Ansible Navigator

Если для управления задачами автоматизации используется *Ansible Navigator*, выполните команду установки зависимостей:

```
ansible-navigator exec -- ansible-galaxy collection install -r requirements.yml
```

Если настройки выполнены корректно, в терминал выводятся сообщения следующего вида:

```
Starting galaxy collection install process
Process install dependency map
Starting collection install process
Downloading https://hub.astra-automation.ru/api/galaxy/v3/plugin/ansible/content/
↪validated/collections/artifacts/astra-nginx-1.8.3.tar.gz to /root/.ansible/tmp/
↪ansible-local-157z5isxaa/tmpdjhla2qi/astra-nginx-1.8.3-kzmwnb1x
Installing 'astra.nginx:1.8.3' to '/home/vagrant/nginx/collections/ansible_
↪collections/astra/nginx'
astra.nginx:1.8.3 was installed successfully
```

11.2.4 Запуск playbook

Способ запуска playbook зависит от инструмента, используемого для управления задачами автоматизации.

Automation Controller

Для запуска задания автоматизации в Automation Controller выполните следующие действия:

1. Для доступа к управляемым узлам создайте необходимое количество полномочий типа *Машина*.
2. Если для доступа к репозиторию с кодом проекта требуется авторизация, создайте полномочие типа *Система управления версиями*.
3. Создайте проект со следующими свойствами:
   - **Название**: *NGINX*.
   - **Тип системы управления исходными данными**: **Git**.
   - **URL системы управления исходными данными**: укажите ссылку на репозиторий с кодом проекта.
   - **Полномочия на систему управления исходными данными**: если для доступа к репозиторию с кодом проекта требуется авторизация, выберите созданное ранее полномочие типа «Система управления версиями».
4. Создайте обычный инвентарь и добавьте в него сведения об управляемых узлах.

   Совет: Вы можете использовать файл `inventory.ini` в качестве источника сведений об управляемых узлах.

5. Создайте шаблон задания со следующими свойствами:
   - **Тип задания**: **Выполнение**.
   - **Инвентарь**: выберите инвентарь, созданный на предыдущем шаге.
   - **Проект**: выберите проект *NGINX*.
   - **Playbook**: выберите файл `playbooks/nginx.yml`.
6. Запустите задание на основе созданного шаблона.

Ansible Navigator

При использовании Ansible Navigator для запуска playbook выполните команду:

```
ansible-navigator run playbooks/nginx.yml
```

При этом открывается псевдографический интерфейс, в котором показывается ход выполнения, например:

```
  Play name                           Ok Changed Unreachable  Failed Skipped Ignored In␣
↪progress  Task count   Progress
0|Install and configure NGINX server   5       0           0       0       0       0    ␣
↪      1           6   ██████


^b/PgUp page up       ^f/PgDn page down       ↑↓ scroll       esc back       [0-9] goto ␣
↪     :help help      Running
```

Для просмотра более подробной информации о ходе выполнения нажмите `0`.

Статус задания выводится в правом нижнем углу. Дождитесь перехода задания в статус `Successful`.

11.2.5 Проверка работоспособности развернутой инфраструктуры

Чтобы проверить корректность развертывания веб-сервера NGINX, выполните следующие действия:

1. Подключитесь к любому из управляемых узлов по SSH.
2. Для проверки статуса службы `nginx` выполните команду:

   ```
   systemctl status nginx
   ```

   При успешном развертывании в терминал выводится сообщение вида:

   ```
   ● nginx.service - A high performance web server and a reverse proxy server
      Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset:␣
   ↪enabled)
      Active: active (running) since Mon 2024-10-21 09:28:25 MSK; 2h 22min ago
        Docs: man:nginx(8)
    Main PID: 643 (nginx)
       Tasks: 3 (limit: 2586)
      Memory: 4.6M
         CPU: 14ms
      CGroup: /system.slice/nginx.service
              ├─643 nginx: master process /usr/sbin/nginx -g daemon on; master_
   ↪process on;
              ├─645 nginx: worker process
              └─646 nginx: worker process
   ```

3. Убедитесь, что в файле `/etc/nginx/nginx.conf` значения параметров `worker_connections` и `keepalive_timeout` соответствуют заданным в playbook.

11.2.6 Особенности проекта

Обратите внимание на следующие особенности проекта:

- Порядок подключения коллекции.
- Настройка проекта на использование Private Automation Hub для загрузки коллекций.
- Настройка параметров роли.
- Настройка параметров Ansible.

11.2.7 Заключение

В этом сценарии вы познакомились с основными шагами по развертыванию веб-сервера NGINX с использованием коллекции `astra.nginx`. Из всей последовательности шагов важно выделить следующие действия:

- Подготовка окружения.
- Описание коллекции в файле зависимостей.
- Описание параметров доступа к управляемым узлам в файле инвентаря.
- Связывание управляемых узлов и ролей в playbook.
- Запуск playbook с помощью Ansible Navigator и в Automation Controller.
- Проверка корректности развертывания NGINX.

11.3 Защита информации

В ОС Astra Linux Special Edition встроено множество механизмов безопасности, но некоторые из них по умолчанию выключены с целью повышения быстродействия, удобства администрирования или использования ОС. Однако, при обработке определенных типов информации к информационной системе могут предъявляться различные требования по обеспечению безопасности.

Эти требования могут быть заданы сторонними регуляторами либо собственной службой безопасности. ОС Astra Linux Special Edition позволяет удовлетворить их без использования сторонних (наложенных) *СЗИ*. Для этого необходимо выполнить настройку ОС и входящего в ее состав программного обеспечения.

Коллекция Ansible `astra.hardening` позволяет автоматизировать настройку механизмов безопасности Astra Linux Special Edition и входящего в ее состав программного обеспечения для выполнения необходимых требований. Все роли, входящие в состав коллекции, адаптированы под Astra Linux Special Edition.

11.3.1 CIS Benchmarks

Center for Internet Security (CIS)[185] – некоммерческая организация, специализирующая на повышении общего уровня кибербезопасности как в частном, так и государственном секторах.

CIS разрабатывает:

- наборы стандартных технических контрольных мер (CIS Controls[186]), которые обеспечивают конкретные и актуализированные рекомендации по безопасности ИТ-систем;
- детальные руководства по настройке безопасности (CIS Benchmarks[187]) для обширного спектра информационных технологий и систем.

Указанные меры и руководства широко приняты в отрасли как стандарты безопасности.

Руководство CIS Debian Family Linux Benchmark[188] описывает настройку параметров безопасности операционных систем, основанных на Debian Linux[189]. Это руководство проверено и адаптировано для ОС Astra Linux Special Edition, основанной на Debian Linux.

Роль Ansible `astra.hardening.cis` используется для автоматизации настройки Astra Linux Special Edition согласно рекомендаций и инструкций, приведенных в CIS Debian Family Linux Benchmark.

Особенности роли `astra.hardening.cis`:

- Правило «1.4.1 Ensure AIDE is installed» заменено правилом «1.4.1 Ensure afick is installed».

  В состав Astra Linux Special Edition не входит AIDE (Advanced Intrusion Detection System). Вместо этого используется утилита `afick` (Another File Integrity Checker).

- Правило из раздела «1.7 Mandatory Access Control» переработано таким образом, чтобы вместо системы мандатного контроля доступа AppArmor использовался встроенный в ядро ОС механизм PARSEC.
- Правило «1.9 Ensure GDM is removed or login is configured» заменено правилом «1.9 Ensure Fly DM is removed or login is configured».

  В ОС Astra Linux Special Edition вместо окружения рабочего стола GNOME используется окружение рабочего стола Fly.

[185] https://www.cisecurity.org/
[186] https://www.cisecurity.org/controls
[187] https://www.cisecurity.org/cis-benchmarks
[188] https://www.cisecurity.org/benchmark/debian_family
[189] https://www.debian.org/

- Для выполнения правила «3.1.1 Disable IPv6» рекомендуется отключать протокол IPv6 путем изменения параметров ядра с помощью `sysctl`, а не через настройки загрузчика GRUB2.

  Подробности см. в статье справочного центра IPv6: включение и выключение[190].

- Изначально в CIS Debian Family Linux Benchmark раздел «3.4 TCP Wrappers» отсутствует, однако роль `astra.hardening.cis` реализует все необходимые правила:
  - «3.4.1 Ensure TCP Wrappers is installed»;
  - «3.4.2 Ensure /etc/hosts.allow is configured»;
  - «3.4.3 Ensure /etc/hosts.deny is configured»;
  - «3.4.4 Ensure permissions on /etc/hosts.allow are configured»;
  - «3.4.5 Ensure permissions on /etc/hosts.deny are configured».

- Правило «3.6.3.1.2 Ensure nftables is not installed» переработано в «3.6.3.1.2 Ensure nftables service is disabled».

  В Astra Linux Special Edition пакет `nftables` не удаляется, так как он является зависимостью пакета `iptables`.

- Правило «5.3.4 Ensure password hashing algorithm is SHA-512» заменено правилом «5.3.4 Ensure password hashing algorithm is GOST R 34.10-2012 (512)».

  Для хеширования паролей пользователей в файле `/etc/shadow` в ОС Astra Linux Special Edition вместо алгоритма SHA-512 используется 512-битный вариант алгоритма, описанного в ГОСТ Р 34.11-2012[191].

  Подробности см. в статье справочного центра Генерация хеша пароля по алгоритму ГОСТ Р 34.11-2012[192].

- Использование системы журналирования `rsyslog` в Astra Linux Special Edition x.7 и более новых версиях не рекомендуется. Поддержка пакета `rsyslog` в репозиториях Astra Linux Special Edition не осуществляется, а его установка нарушает работоспособность окружения рабочего стола Fly. Вместо `rsyslog` рекомендуется использовать систему журналирования `syslog-ng`.

  Подробности см. в статье справочного центра Журналы работы системных служб[193].

Названия переменных, управляющих применением правил CIS Debian Family Linux Benchmark, формируются по следующему шаблону:

```
cis_alse_rule_<number>
```

где <number> – номер правила, например, `6.2.1`, `3.6.1.1` или `3.6.3.3.4`.

Пример Ansible playbook, выполняющего настройку управляемых узлов с применением правил 6.2.1, 3.6.1.1 и 3.6.3.3.4:

```
- name: Configure security settings
  hosts: all
  vars:
    cis_alse_rule_6.2.1: true
    cis_alse_rule_3.6.1.1: true
    cis_alse_rule_3.6.3.3.4: true
  roles:
    - role: astra.hardening.cis
```

Полный список переменных роли `astra.hardening.cis` и их значения по умолчанию представлены в файле `hardening/roles/cis/defaults.main`.

Пример использования роли см. в документе *Применение рекомендаций CIS*.

[190] https://wiki.astralinux.ru/pages/viewpage.action?pageId=44892637
[191] https://protect.gost.ru/document.aspx?control=7&id=180209
[192] https://wiki.astralinux.ru/pages/viewpage.action?pageId=216540367
[193] https://wiki.astralinux.ru/pages/viewpage.action?pageId=9011231

11.3.2 DevSec Hardening Framework

DevSec Hardening Framework[194] содержит рекомендации по проверке настроек параметров безопасности ОС и различного ПО с помощью платформы для аудита и автоматического тестирования безопасности и соответствия требованиям Chef InSpec[195], а также по автоматизированной настройке этих параметров с помощью популярных систем управления конфигурацией (Ansible, Chef[196], Puppet[197]).

Роль Ansible `astra.hardening.devsec` реализует настройки безопасности, описанные в следующих разделах DevSec Hardening Framework:

- DevSec Linux Security Baseline[198] v2.3.0;
- DevSec SSH Baseline[199] v2.4.0.

Особенности реализации роли `astra.hardening.devsec`:

- Упомянуто правило «package-04». Это правило пропущено в DevSec Linux Security Baseline.
- Реализовано правило «sysctl-34». При выполнении этого правила устанавливаются безопасные значения для следующих параметров ядра:
  - protected_fifos[200];
  - protected_hardlinks[201];
  - protected_regular[202];
  - protected_symlinks[203].

Названия переменных, управляющих применением правил, формируются по следующему шаблону:

```
devsec_alse_<software>_rule_<name>
```

где:

- <software> – настраиваемое ПО:
  - `linux` – Astra Linux Special Edition;
  - `ssh` – OpenSSH.
- <name> – название правила в соответствующем руководстве, но с заменой символов дефиса `-` на символы подчеркивания `_`.

Пример Ansible playbook, выполняющего настройку управляемых узлов с применением правил «sysctl-02» и «sshd-01»:

```
---
- name: DevSec security settings
  hosts: all
  vars:
    devsec_alse_linux_rule_sysctl_02: true
    devsec_alse_ssh_rule_sshd_01: true
  roles:
    - role: astra.hardening.devsec
```

[194] https://dev-sec.io/
[195] https://community.chef.io/tools/chef-inspec
[196] https://community.chef.io/
[197] https://www.puppet.com/community/open-source/open-source-puppet
[198] https://dev-sec.io/baselines/linux/
[199] https://dev-sec.io/baselines/ssh/
[200] https://docs.kernel.org/admin-guide/sysctl/fs.html#protected-fifos
[201] https://docs.kernel.org/admin-guide/sysctl/fs.html#protected-hardlinks
[202] https://docs.kernel.org/admin-guide/sysctl/fs.html#protected-regular
[203] https://docs.kernel.org/admin-guide/sysctl/fs.html#protected-symlinks

Пример использования роли см. в документе *Применение рекомендаций DevSec*.

11.3.3 Требования и методические рекомендации ФСТЭК России

ОС Astra Linux Special Edition сертифицирована Федеральной службой по техническому и экспортному контролю (ФСТЭК России[204]) как *СЗИ* и реализует функции защиты информации, предусмотренные для 1 класса защиты следующими требованиями:

- Требования по безопасности информации к операционным системам (утв. приказом ФСТЭК России от 19.08.2016 г. № 119. ДСП);
- Требования по безопасности информации к средствам виртуализации (утв. приказом ФСТЭК России от 27.10.2022 г. № 187[205]);
- Требования по безопасности информации к средствам контейнеризации (утв. приказом ФСТЭК России от 04.07.2022 г. № 118[206]).

Кроме того, Astra Linux Special Edition реализует функции системы управления базами данных (СУБД), обеспечивающие выполнение ряда функций безопасности СУБД.

Функции защиты реализованы встроенным комплексом средств защиты информации Astra Linux Special Edition и могут применяться для реализации мер защиты информации ограниченного доступа в соответствии с требованиями следующих приказов ФСТЭК России:

- Приказ № 17 от 11 февраля 2013 г.[207] «Об утверждении требований о защите информации, не составляющей государственную тайну, содержащейся в государственных информационных системах»;
- Приказ № 21 от 18 февраля 2013 г.[208] «Об утверждении состава и содержания организационных и технических мер по обеспечению безопасности персональных данных при их обработке в информационных системах персональных данных»;
- Приказ № 31 от 14 марта 2014 г.[209] «Об утверждении требований к обеспечению защиты информации в автоматизированных системах управления производственными и технологическими процессами на критически важных объектах, потенциально опасных объектах, а также объектах, представляющих повышенную опасность для жизни и здоровья людей и для окружающей природной среды»;
- Приказ № 239 от 25 декабря 2017 г.[210] «Об утверждении требований по обеспечению безопасности значимых объектов критической информационной инфраструктуры российской федерации».

Важно: Требования вышеперечисленных приказов ФСТЭК России не регламентируют настройки средств защиты, а определяют базовый набор мер защиты, который для реализации на объектах информатизации адаптируется и уточняется в зависимости от структурно-функциональных характеристик и модели угроз.

С учетом того, что меры защиты представлены в требованиях приказов без конкретных параметров настроек, наши рекомендации по настройке операционной системы могут оказаться неподходящими для вашей информационной системы или недостаточными для предотвращения несанкционированного доступа. Решение о применении настроек остается за администратором безопасности вашей информационной системы.

В следующих нормативных документах приведены методические рекомендации ФСТЭК России по настройке систем безопасности информационных систем:

[204] https://fstec.ru/
[205] https://fstec.ru/dokumenty/vse-dokumenty/spetsialnye-normativnye-dokumenty/trebovaniya-po-bezopasnosti-informatsii-utverzh
[206] https://fstec.ru/dokumenty/vse-dokumenty/spetsialnye-normativnye-dokumenty/trebovaniya-po-bezopasnosti-informatsii-utverzh
[207] https://fstec.ru/dokumenty/vse-dokumenty/prikazy/prikaz-fstek-rossii-ot-11-fevralya-2013-g-n-17
[208] https://fstec.ru/dokumenty/vse-dokumenty/prikazy/prikaz-fstek-rossii-ot-18-fevralya-2013-g-n-21
[209] https://fstec.ru/dokumenty/vse-dokumenty/prikazy/prikaz-fstek-rossii-ot-14-marta-2014-g-n-31
[210] https://fstec.ru/dokumenty/vse-dokumenty/prikazy/prikaz-fstek-rossii-ot-25-dekabrya-2017-g-n-239

- Методический документ от 11 февраля 2014 г.[211] «Меры защиты информации в государственных информационных системах».
- Методический документ от 25 декабря 2022 г.[212] «Рекомендации по безопасной настройке операционных систем Linux».

В соответствии с требованиями ФСТЭК России в информационных системах, обрабатывающих информацию ограниченного доступа: в государственных информационных системах, информационных системах персональных данных, автоматизированных системах управления, в составе значимых объектов критической информационной инфраструктуры, – должны быть реализованы меры защиты информации, состав которых определяется исходя из класса (уровня, категории) защищенности (значимости) системы и актуальной модели угроз. Соответствие системы защиты информации информационной системы требованиям по безопасности подтверждается на этапе аттестационных испытаний.

Возможности Astra Linux Special Edition для реализации мер защиты информации, предусмотренных указанными выше приказами и описание используемых встроенных средств защиты приведены в статье справочного центра Возможности реализации мер защиты информации в соответствии с приказами ФСТЭК России средствами Astra Linux Special Edition[213].

Для настройки систем безопасности в соответствии с указанными требованиями и рекомендациями можно использовать роль `astra.hardening.fstec`.

11.3.4 Payment Card Industry Data Security Standard

Стандарт Payment Card Industry Data Security Standard (PCI DSS)[214] содержит список требований, предъявляемых к информационным системам по обеспечению безопасности данных платежных карт.

Важно: Эта часть документации находится в стадии разработки.

11.3.5 Методические указания Банка России

Важно: Эта часть документации находится в стадии разработки.

Применение рекомендаций CIS

Использование роли `astra.hardening.cis` из реестра Automation Hub позволяет автоматизировать процесс настройки систем защиты ОС Astra Linux Special Edition в соответствии с рекомендациями, изложенными в руководстве «CIS Debian Family Linux Benchmark».

[211] https://fstec.ru/dokumenty/vse-dokumenty/spetsialnye-normativnye-dokumenty/metodicheskij-dokument-ot-11-fevralya-2014-g
[212] https://fstec.ru/dokumenty/vse-dokumenty/spetsialnye-normativnye-dokumenty/metodicheskij-dokument-ot-25-dekabrya-2022-g
[213] https://wiki.astralinux.ru/pages/viewpage.action?pageId=181666117
[214] https://www.pcisecuritystandards.org/document_library/?document=pci_dss

Описание сценария

Процесс настройки систем безопасности ОС Astra Linux Special Edition с помощью роли `astra.hardening.cis` состоит из следующих этапов:

1. Подготовка проекта Ansible:
   - `requirements.yml` – зависимости;
   - `playbook.yml` – сценарий настройки управляемых узлов.
2. Настройка контроллера на выполнение заданий:
   1. Создание полномочий.
   2. Создание проекта.
   3. Создание инвентаря и записей об управляемых узлах.
   4. Создание шаблона задания.
3. Запуск задания.

С помощью переменных роли к ОС управляемых узлов применяются настройки, описанные в следующих разделах «CIS Debian Family Linux Benchmark»:

- 1.8.5 Ensure remote login warning banner is configured properly

  Содержимое файла `/etc/issue.net` отображается при попытке удаленного подключения к узлу. Эта информация может быть полезна для разработчиков ПО, но также она может быть использована злоумышленником для выбора инструментальных средств, эксплуатирующих уязвимости ОС определенной версии.

  Рекомендуется отключить вывод информации о версии ОС при попытках удаленного подключения к узлу.

- 1.8.6 Ensure local login warning banner is configured properly

  Содержимое файла `/etc/issue` отображается при попытке локального входа в систему. Эта информация может быть полезна для разработчиков ПО, но также может быть использована злоумышленником для выбора инструментальных средств, эксплуатирующих уязвимости ОС определенной версии.

  Рекомендуется отключить вывод информации о версии ОС на экране локального входа в систему.

- 3.2.1 Ensure packet redirect sending is disabled

  Перенаправление ICMP (ICMP Redirects) используется для передачи информации о маршрутизации на другие узлы. Если узел не используется в качестве маршрутизатора, он не должен пересылать пакеты на другие узлы.

Для настройки управляемых узлов используется подключение с применением ключей SSH.

Подготовка к работе

Подготовьте управляемые узлы к настройке с помощью Ansible:

- Обеспечьте доступ к управляемым узлам по SSH.
- *Разрешите* использование sudo без ввода пароля.

Проверка настроек систем защиты по умолчанию

Для проверки действующих значений параметров систем защиты выполните следующие действия:

1. Подключитесь к любому управляемому узлу по SSH.
2. Выполните команды, соответствующие выбранным правилам «CIS Debian Family Linux Benchmark»:
   - 1.8.5 Ensure remote login warning banner is configured properly

     ```
     grep -E -i "(\\v|\\r|\\m|\\s|$(grep '^ID=' /etc/os-release | cut -d= -f2 |␣
     ↪sed -e 's/"//g'))" /etc/issue.net
     ```

     Эта команда выполняет в файле `/etc/issue.net` поиск строки, позволяющей определить версию ОС. При настройках ОС по умолчанию эта команда возвращает строку:

     ```
     Astra Linux 1.7.5
     ```

   - 1.8.6 Ensure local login warning banner is configured properly

     ```
     grep -E -i "(\\v|\\r|\\m|\\s|$(grep '^ID=' /etc/os-release | cut -d= -f2 |␣
     ↪sed -e 's/"//g'))" /etc/issue
     ```

     Эта команда выполняет в файле `/etc/issue` поиск строки, позволяющей определить версию ОС. При настройках ОС по умолчанию эта команда возвращает строку:

     ```
     Astra Linux 1.7.5 \n \l
     ```

   - 3.2.1 Ensure packet redirect sending is disabled

     Для получения действующих значений параметров ядра выполните команды:

     ```
     sudo sysctl net.ipv4.conf.all.send_redirects
     sudo sysctl net.ipv4.conf.default.send_redirects
     ```

     При настройках ОС по умолчанию они возвращают следующий набор строк:

     ```
     net.ipv4.conf.all.send_redirects = 1
     net.ipv4.conf.default.send_redirects = 1
     ```

     Значение 1 означает, что соответствующая функциональность ядра включена.

     Для поиска значений настроек, хранящихся в конфигурационных файлах, выполните команды:

     ```
     grep "net\.ipv4\.conf\.all\.send_redirects" /etc/sysctl.conf /etc/sysctl.d/*
     grep "net\.ipv4\.conf\.default\.send_redirects" /etc/sysctl.conf /etc/sysctl.d/*
     ```

     При настройках ОС по умолчанию они возвращают следующий набор строк:

     ```
     /etc/sysctl.conf:#net.ipv4.conf.all.send_redirects = 0
     /etc/sysctl.d/99-sysctl.conf:#net.ipv4.conf.all.send_redirects = 0
     ```

     В конфигурационных файлах `/etc/sysctl.conf` и `/etc/systctl.d/99-sysctl.conf` присутствуют строки, отвечающие за отключение перенаправления пакетов, но они закомментированы.
3. Для отключения от управляемого узла выполните команду:

   ```
   exit
   ```

Подготовка проекта

Подготовьте файлы проекта Ansible:

1. Создайте файл зависимостей Ansible `requirements.yml`:

```
---
collections:
  - name: astra.hardening
    version: 0.2.0
```

   В этом файле указаны параметры подключаемой коллекции:

   - name – название;
   - `version` – версия.

2. Создайте файл playbook `playbook.yml`:

```
---
- name: Apply CIS requirements
  hosts: all
  become: true

  roles:
    - role: astra.hardening.cis
      vars:
        cis_alse_rule_1_8_5: true
        cis_alse_rule_1_8_6: true
        cis_alse_rule_3_2_1: true
```

   Здесь выбор правил «CID Debian Family Linux Benchmark» выполняется через присваивание переменным роли `astra.hardening.cis` значения `true`.

3. Сделайте код проекта доступным для загрузки в Automation Controller, например, путем публикации проекта на GitFlic[215] или GitHub[216].

Настройка контроллера

Авторизуйтесь в Automation Controller и выполните следующие действия:

1. *Создайте полномочия*:

Тип	Цель
API-токен Ansible Galaxy/Automation Hub	Automation Hub
Система управления версиями	Репозиторий с кодом проекта
Машина	Управляемые узлы

2. *Создайте проект* со следующими характеристиками:

   - **Название** (Name) – *CIS Demo Project*.
   - **Тип системы управления исходными данными** (Source Control Type) – **Git**.
   - **URL системы управления исходными данными** (Source Control URL) – укажите ссылку на репозиторий, например:

     ```
     git@gitflic.ru:example-project/astra-hardening-cis-using.git
     ```

   - **Полномочия на систему управления исходными данными** (Source Control Credential) – выберите созданное ранее полномочие для доступа к репозиторию с кодом проекта.

[215] https://gitflic.ru/
[216] https://github.com/

3. *Создайте обычный инвентарный список* с названием *CIS Demo Inventory*. Добавьте в него сведения об управляемых узлах.
4. *Создайте шаблон задания* со следующими характеристиками:
   - **Название** (Name) – *CIS Demo Job Template*.
   - **Тип задания** – **Выполнение** (Run).
   - **Инвентарь** (Inventory) – *CIS Demo Inventory*.
   - **Проект** (Project) – *CIS Demo Project*.
   - **Playbook** – `playbook.yml`.
   - **Полномочия** – полномочие типа «Машина», используемое для доступа к управляемым узлам.
5. *Запустите задание* на основе созданного шаблона.

   Дождитесь выполнения playbook, это может занять некоторое время.

   Важно: По окончании настройки управляемые узлы перезагружаются автоматически.

   Убедитесь, что выполнение playbook прошло успешно. Об этом свидетельствуют нулевые значения параметров `unreachable` и `failed` в конце вывода:

```
PLAY RECAP *********************************************************************
node1   : ok=32   changed=4   unreachable=0   failed=0   skipped=667   rescued=0   ␣
↪ignored=0
node2   : ok=32   changed=4   unreachable=0   failed=0   skipped=667   rescued=0   ␣
↪ignored=0
node3   : ok=32   changed=4   unreachable=0   failed=0   skipped=667   rescued=0   ␣
↪ignored=0
```

   Параметры `changed` и `skipped` показывают количество выполненных и пропущенных задач Ansible соответственно.

Проверка состояния систем защиты

Чтобы проверить состояние систем защиты, выполните следующие действия:

1. Подключитесь к любому из настроенных узлов по SSH.
2. Выполните команды, соответствующие выбранным правилам «CIS Debian Family Linux Benchmark»:
   - 1.8.5 Ensure remote login warning banner is configured properly

     ```
     grep -E -i "(\\v|\\r|\\m|\\s|$(grep '^ID=' /etc/os-release | cut -d= -f2 |␣
     ↪sed -e 's/"//g'))" /etc/issue.net
     ```

     После настройки эта команда возвращает пустую строку.
   - 1.8.6 Ensure local login warning banner is configured properly

     ```
     grep -E -i "(\\v|\\r|\\m|\\s|$(grep '^ID=' /etc/os-release | cut -d= -f2 |␣
     ↪sed -e 's/"//g'))" /etc/issue
     ```

     После настройки эта команда возвращает пустую строку.
   - 3.2.1 Ensure packet redirect sending is disabled

     Для определения значений действующих параметров ядра выполните команду:

     ```
     sudo sysctl net.ipv4.conf.all.send_redirects
     sudo sysctl net.ipv4.conf.default.send_redirects
     ```

После настройки эти команды возвращают следующий набор строк:

```
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.default.send_redirects = 0
```

Значение параметров, равное 0, свидетельствует о том, что пересылка пакетов выключена путем изменения параметров ядра.

Для поиска значений настроек, хранящихся в конфигурационных файлах, выполните команды:

```
grep "net\.ipv4\.conf\.all\.send_redirects" /etc/sysctl.conf /etc/sysctl.d/*
grep "net\.ipv4\.conf\.default\.send_redirects" /etc/sysctl.conf /etc/sysctl.d/*
```

После настройки эти команды возвращают следующий набор строк:

```
/etc/sysctl.conf:#net.ipv4.conf.all.send_redirects = 0
/etc/sysctl.d/999-astra.conf:net.ipv4.conf.all.send_redirects=0
/etc/sysctl.d/99-sysctl.conf:#net.ipv4.conf.all.send_redirects = 0
/etc/sysctl.d/999-astra.conf:net.ipv4.conf.default.send_redirects=0
```

Отсюда следует, что для хранения необходимых параметров в каталоге `/etc/sysctl.d/` создан конфигурационный файл `999-astra.conf`, в который добавлены следующие строки:

```
net.ipv4.conf.all.send_redirects=0
net.ipv4.conf.default.send_redirects=0
```

Содержимое файлов `/etc/sysctl.conf` и `/etc/sysctl.d/99-sysctl.conf` при этом не изменилось.

3. Для отключения выполните команду:

```
exit
```

Особенности проекта

Обратите внимание на следующие особенности проекта:

- Порядок подключения коллекции.
- Настройка параметров роли `astra.hardening.cis`.
- Способы получения информации о параметрах систем безопасности ОС до и после настройки.

Заключение

В этом сценарии вы познакомились с основными шагами по настройке систем безопасности ОС Astra Linux Special Edition в соответствии с рекомендациями, изложенными в «CIS Debian Family Linux Benchmark». Из всей последовательности шагов важно выделить следующие действия:

- Проверка состояния систем защиты ОС до и после настройки.
- Настройка систем безопасности путем присваивания соответствующих значений переменным роли `astra.hardening.cis`.

Применение рекомендаций DevSec

Важно: Эта часть документации находится в стадии разработки.

Выполнение требований и рекомендаций ФСТЭК России

Важно: Эта часть документации находится в стадии разработки.

Выполнение требований PCI DSS

Важно: Эта часть документации находится в стадии разработки.

Выполнение требований Банка России

Важно: Эта часть документации находится в стадии разработки.

11.4 Пример организации процесса CI/CD

Рассмотрим организацию простейшего процесса непрерывной разработки и развертывания программных продуктов. При необходимости вы можете воспроизвести данный сценарий в собственном окружении.

11.4.1 Описание сценария

Приведен пример планирования и настройки непрерывного процесса разработки программного проекта (CI, continuous integration) и его развертывания (CD, continuous deployment).

Особенности сценария:

- Проект хранится в публично доступном репозитории под управлением Git, используя GitHub.
- Обработку процессов CI/CD выполняет Automation Controller.
- При создании запроса (PR, pull request) на внесение изменений в какую-либо ветку репозитория GitHub вызывает процесс CI, целью которого является проверка корректности разработанного программного продукта. При исполнении контролируемая программа оповещает о корректности путем возврата соответствующего кода:
  - `0` – программа работает корректно;
  - `1` – в программе обнаружена ошибка.

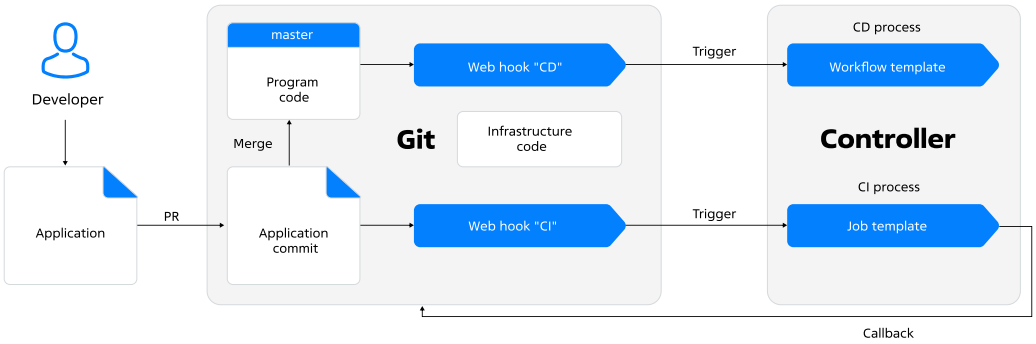
  При обнаружении ошибки в программе разработчик имеет возможность исправить код и повторить процесс CI.
- После слияния рабочей ветки с веткой `master` GitHub вызывает процесс CD, в котором контроллер проверяет, что объектом изменений является ветка `master`. В этом случае он запрашивает разрешение на обновление продукта и после одобрения выполняет обновление продукта.

11.4.2 Структура проекта

Проект реализации сценария включает разработку структуры и инфраструктурного кода.

Взаимодействие компонентов

Структурная схема сценария представляется в следующем виде:



Взаимодействие компонентов в процессе CI происходит следующим образом:

1. В репозитории GitHub разработчик вносит изменения в демонстрационную программу и создает запрос на внесение изменений в ветку `master` (PR).
2. GitHub перехватывает PR с помощью соответствующего объекта webhook и направляет запрос на выполнение процесса CI в контроллере. Запрос PR переходит в состояние ожидания ответа от контроллера.
3. Контроллер получает запрос на запуск задания из шаблона, реализующего процесс CI. В данном сценарии задание выводит содержимое программы в выходной поток и запускает программу для проверки корректности ее выполнения.
4. Контроллер возвращает результат обработки запроса в GitHub.
5. GitHub получает результат запроса и сообщает о готовности внести изменения в `master`.

CD является продолжением предыдущего процесса. Его можно выполнить в случае успешного завершения последнего:

1. Разработчик принимает решение о внесении изменений, представленных PR, в ветку `master` и начинает процесс слияния веток (Git merge).
2. После завершения обновления ветки `master` GitHub перехватывает запрос с помощью соответствующего объекта webhook и направляет запрос на выполнение процесса CD в контроллере.
3. Контроллер получает запрос на запуск заданий из шаблона потока, реализующего процесс CD. Он запускает выполнение цепочки заданий, включая получение одобрения от ответственного лица.

Примечание: В отличие от процесса CI, контроллер не возвращает результат обработки запроса в GitHub. Поэтому результаты процесса CD необходимо анализировать средствами контроллера.

Репозиторий проекта

Для простоты исполняемый код обслуживаемой демонстрационной программы разработчика (`project.sh`) и инфраструктурный код для реализации сценария находятся в одном репозитории в виде следующей структуры каталогов и файлов:

```
./
├── aac
│   ├── cd_check.yml
│   ├── cd_deploy.yml
│   └── ci.yml
└── project.sh
```

Назначение и состав файлов представлены в следующих секциях. Для воспроизведения сценария загрузите эти файлы в свой репозиторий GitHub в соответствии с представленной структурой.

Демонстрационная программа

Демонстрационная программа, к которой применены процессы CI/CD, состоит из одного скрипта shell `project.sh` (download):

Demo program

```
#!/bin/bash

# Use "exit 0" to simulate correct code
# Use "exit 1" to "exit 255" to simulate incorrect code
exit 1
```

CI/CD playbooks

Каждая часть процесса использует отдельные файлы playbook.

Процесс CI состоит из одного задания, которое выполняется с помощью playbook `aac/ci.yml` (download) на локальном узле, то есть на самом контроллере:

CI playbook

```
---
- hosts: localhost
  become: false
  gather_facts: false
  tasks:
    - name: Get code revision "{{ awx_webhook_event_ref }}"
      ansible.builtin.git:
        repo: "{{ awx_webhook_payload.repository.clone_url }}"
        dest: /tmp/project
        version: "{{ awx_webhook_event_ref }}"

    - name: Show main project
      ansible.builtin.shell:
        chdir: /tmp/project
        cmd: cat ./project.sh

    - name: Check main project
      ansible.builtin.shell:
        chdir: /tmp/project
        cmd: ./project.sh
```

Playbook выполняет следующие действия:

1. Создает клон репозитория, адрес которого содержит переменная `awx_webhook_payload.repository.clone_url`, в каталоге `/tmp/project/`.
2. Выводит содержимое файла проекта в выходной поток командой `cat`.
3. Запускает на выполнение скрипт программы для тестирования ее работоспособности. По результатам выполнения контроллер определяет корректность внесенных изменений.

Процесс CD состоит из потока заданий, в который вовлечены следующие файлы playbook:

1. `aac/cd_check.yml` (`download`) – проверка того, что вызов задания связан с обновлением ветки `master` в репозитории GitHub:

   CD check playbook

   ```
   ---
   - hosts: localhost
     become: false
     gather_facts: false
     tasks:
       - name: Check if push is into master branch
         ansible.builtin.fail:
           msg: This push is not into master branch
         when: awx_webhook_payload.ref != "refs/heads/master"
   ```

   Playbook завершает аварийно процесс CD, если обновление было не в ветке `master`. В противном случае процесс продолжается.

2. `aac/cd_deploy.yml` (`download`) – демонстрационное (фиктивное) развертывание продукта:

   CD deployment playbook

   ```
   ---
   - hosts: localhost
     become: false
     gather_facts: false
     tasks:
       - name: Deploy project
         ansible.builtin.debug:
           msg: The project is being deployed.
   ```

   Playbook только выводит сообщение о выполнении развертывания.

11.4.3 Настройка процесса

После создания публичного репозитория необходимо выполнить настройки основных компонентов.

Настройка прав доступа

Для обеспечения взаимодействия компонентов проекта необходимо предоставить контроллеру полномочия на доступ к репозиторию GitHub. Это позволит контроллеру возвращать состояние процесса (callback) в GitHub. Полномочия необходимо предоставить через персональный токен доступа к репозиторию GitHub. Для его создания в GitHub существует следующая последовательность шагов в настройках:

profile > settings > developer settings > personal access tokens > tokens(classic) > generate new token

---

Примечание: Сгенерированный токен показывается только один раз. Для повторного использования необходимо сохранить его в надежном месте.

Контроллер

В контроллере необходимо произвести следующие настройки:

1. Задайте базовый URL сервиса, который для контроллера, установленном на одном узле, совпадает с URL этого контроллера. Шаги настройки:

   Настройки (Settings) > Система (System) > Общие системные настройки (Miscellaneous System settings) > Базовый URL сервиса (Base URL of the service)

   Примечание: Базовый URL будет передан в GitHub для использования его в качестве ссылки на соответствующее задание контроллера.

   ---

2. Создайте организацию. Шаги настройки:

   Access > Organizations > Add

   Параметры настройки:

   - **Название**: *cicd*.

3. Создайте полномочие на доступ к репозиторию GitHub. Шаги настройки:

   Resources > Credentials > Add

   Параметры настройки:

   - **Название** (Name): *cicd*;
   - **Тип полномочия** (Credential Type): **Личный токен доступа к GitHub** (GitHub Personal Access Token);
   - **Сведения о типе** (Type Details) > **Токен** (Token): <содержимое токена, созданного для организации или учетной записи в GitHub>;
   - **Организация** (Organization): *cicd*.

4. Создайте проект на основе репозитория, созданного в GitHub. Шаги настройки:

   Resources > Projects > Add

   Параметры настройки:

   - **Название**: *cicd*;
   - **Организация** (Organization): *cicd*;
   - **Среда исполнения** (Execution Environment): *Default execution environment*;
   - **Тип системы управления исходными данными** (Source Control Type): *Git*;
   - **URL системы управления исходными данными** (Source Control URL): *https://github.com/<GitHub account or organization>/<repo-name>.git*.

5. Создайте шаблоны заданий. Шаги настройки:

   Resources > Templates > Add > Add job template

   Параметры настройки для шаблона CI:

   - **Название** (Name): *ci*;
   - **Тип задания** (Job Type): **Выполнение** (Run);
   - **Инвентарь** (Inventory): *Demo Inventory*;
   - **Проект** (Project): *cicd*;
   - **Playbook**: `aac/ci.yml`.
   - В секции **Опции** (Options) включите следующие флаги:
     - **Включить webhook** (Enable Webhook);
     - **Параллельные задания** (Enable Concurrent Jobs).
   - В секции **Подробности о webhook** (Webhook details) настройте следующие параметры:
     - **Сервис webhook** (Webhook Service): **GitHub**;
     - **Полномочия webhook** (Webhook Credential): *cicd*.

   Сохраните настройки.

   Параметры настройки для шаблона задания на шаге проверки процесса CD:

   - **Название** (Name): *cd_check*;
   - **Тип задания** (Job Type): **Выполнение** (Run);
   - **Инвентарь** (Inventory): *Demo Inventory*;
   - **Проект** (Project): *cicd*;
   - **Playbook**: `aac/cd_check.yml`.

   Сохраните настройки.

   Параметры настройки для шаблона задания на шаге развертывания процесса CD:

   - **Название** (Name): *cd_deploy*;
   - **Тип задания** (Job Type): **Выполнение** (Run);
   - **Инвентарь** (Inventory): *Demo Inventory*;
   - **Проект** (Project): *cicd*;
   - **Playbook**: `aac/cd_deploy.yml`.

   Сохраните настройки.

6. Создайте шаблон потока заданий для процесса CD. Шаги настройки:

   Resources > Templates > Add > Add workflow template

   Параметры настройки:

   - **Название** (Name): *cd*;
   - **Организация** (Organization): *cicd*;
   - **Инвентарь** (Inventory): *Demo Inventory*.
   - В секции **Опции** (Options) выберите следующие опции:
     - **Включить webhook** (Enable Webhook);
     - **Параллельные задания** (Enable Concurrent Jobs).
   - В секции **Подробности о webhook** (Webhook details) настройте следующие параметры:

  - **Сервис webhook** (Webhook Service): **GitHub**;
  - **Полномочия webhook** (Webhook Credential): *cicd*.

Сохраните настройки.

Во всплывающем окне **Пожалуйста, нажмите кнопку «Пуск», чтобы начать** (Please click the Start button to begin) нажмите кнопку **Пуск** (Start) для настройки потока с помощью визуализатора.

- Создайте узел на основе шаблона cd_check со следующими параметрами:
  - **Тип узла** (Node Type): **Шаблон задания** (Job Template);
  - **Название** (Name): *cd_check*.
- Создайте узел согласования со следующими параметрами:
  - **Тип узла** (Node Type): **Согласование** (Approval);
  - **Название** (Name): *Согласование развертывания продукта*.
- Создайте узел на основе шаблона `cd_deploy` со следующими параметрами:
  - **Тип узла** (Node Type): **Job Template**;
  - **Название** (Name): *cd_deploy*.

После сохранения каждого из двух шаблонов контроллер сгенерирует параметры доступа для шаблона в следующем виде:

- **Webhook URL**: <webhook URL>;
- **Ключ webhook** (Webhook key): <hash code>.

Параметры доступа необходимы для настройки объектов webhook в GitHub.

GitHub

В репозитории проекта необходимо настроить перехват событий с помощью объектов webhook.

Шаги настройки:

Settings > Webhooks > Add webhook

Создайте объекты перехвата событий со следующими параметрами:

- Webhook для CI:
  - **Payload URL**: <URL, использованный в настройках шаблона задания CI в контроллере>;
  - **Content type**: **application/json**;
  - **Secret**: <значение Webhook key из настроек шаблона задания CI в контроллере>;
  - **SSL verification**: **Disable**.

    Примечание: Здесь и далее необходимо отключить проверку сертификата, если не установлен корректный сертификат TLS/SSL на сервере, которому вы доверяете.

  - В секции **Which events would you like to trigger this webhook** выберите опцию **Let me select individual events**.
  - В открывшемся списке выберите **Pull requests**.
  - Выберите опцию **Active**.
- Webhook для CD:

- **Payload URL**: <URL, использованный в настройках шаблона потока заданий CD в контроллере>;
- **Content type**: **application/json**;
- **Secret**: <значение Webhook key из настроек шаблона задания CI в контроллере>;
- **SSL verification**: **Disable**.
- В секции **Which events would you like to trigger this webhook** выберите опцию **Let me select individual events**.
- В открывшемся списке выберите **Pushes**.
- Выберите опцию **Active**.

11.4.4 Проверка работоспособности

Проверке подлежат отдельно два процесса.

Проверка CI

Инициируйте процесс CI следующими действиями:

1. В репозитории GitHub внесите изменение в файл `project.sh`, так чтобы в нем оставалась команда `exit 1`, имитирующая некорректный код.
2. Нажмите кнопку **Commit changes...**.
3. Выберите опцию **Create a new branch...** и введите комментарий в поле сообщения.
4. Нажмите кнопку **Propose changes**.
5. Нажмите кнопку **Create pull request**.

Сервис GitHub выведет сообщения о начале обработке события. Эта обработка завершается сообщением об ошибке «All checks have failed, 1 failing check».

Проверьте запуск процесса CI в контроллере:

1. В панели навигации откройте *Views > Jobs*. В верху списка вы видите появление новой записи о запуске задания из шаблона CI.
2. Откройте задание и изучите результат его работы. Поскольку скрипт `project.sh` возвращает код 1, то последняя задача в playbook `ci.yml` возвращает ошибку.

Повторите весь процесс, обеспечив в скрипте `project.sh` выполнение команды `exit 0`, сообщающей об успешном выполнении программы. Отличия от предыдущей попытки:

- Процесс следует начать с редактирования последней ветки, созданной в предыдущей попытке.
- Не надо создавать еще одну ветку, а выберите ту, в которой вы внесли изменения.

В этот раз процесс CI завершается успешно. Для того, чтобы убедиться в этом, откройте в вкладку **Pull requests** и затем откройте новейший PR.

Проверка CD

Процесс CD является продолжением процесса CI. После успешного завершения CI сервис GitHub предоставит возможность начать процесс обновления ветки `master` с последующим обновлением продукта.

После появления зеленой кнопки **Merge pull request** выполните обновление ветки `master`. Для этого нажмите кнопку **Merge pull request** и затем **Confirm merge**.

На этом работа в GitHub завершена.

Проверьте запуск процесса CD в контроллере:

1. В панели навигации откройте *Views* > *Jobs*. В верху списка вы видите появление новой записи о запуске потока заданий из шаблона CD.
2. Откройте задание и изучите результат ее работы.
3. Используйте один из следующих способов для определения узла графа, требующего разрешения на выполнение дальнейших действий:
   - визуализатор открытого потока заданий;
   - иконка уведомлений (в виде колокольчика);
   - список согласований, который можно открыть с помощью панели навигации: *Режимы просмотра* ‣ *Согласование потоков заданий* (*Views* ‣ *Workflow Approvals*).
4. Выберите соответствующий узел и нажмите кнопку **Согласовать** (Approve). Убедитесь, что выполнение потока заданий (процесс CD) завершено успешно.

11.5 Пример настройки GitFlic webhook

Added in version 1.1-upd1.

В Automation Controller есть возможность запускать задания по запросу от внешних систем, в том числе по событиям, обнаруживаемым с помощью специальных программных модулей, называемых webhook.

Важно: Для обработки запросов от webhook Automation Controller должен быть доступен по HTTPS с доверенным сертификатом.

11.5.1 Получение токена

Для получения токена выполните следующие действия:

1. Создайте токен OAuth в настройках GitFlic.
2. Скопируйте созданный токен, поскольку он понадобится позже для настройки контроллера.

11.5.2 Настройка контроллера

Для настройки контроллера выполните следующие действия:

1. В графическом интерфейсе Automation Controller перейдите в раздел *Полномочия*.
2. Нажмите кнопку *Добавить* (Add).
3. Заполните форму **Создать новые учетные данные** (Create New Credential):
   - **Название** (Name) – название полномочия, например, *Доступ к GitFlic*.
   - **Описание** (Description) – дополнительная информация о полномочии, например, *Полномочие создано для настройки webhook GitFlic*.
   - **Организация** (Organization) – организация, которой принадлежит полномочие, например, *Jupiter*.

     Если поле не заполнено, полномочие принадлежит всем организациям контроллера.
   - **Тип полномочия** (Credential Type) – тип полномочия.

     Выберите **Личный токен доступа к GitFlic** (GitFlic Personal Access Token).
   - **Токен** (Token) – введите токен, созданный ранее в GitFlic.
4. Откройте в режиме редактирования шаблон задания, который необходимо связать с webhook, и измените значения следующих настроек:
   1. В секции **Опции** (Options) включите флаг **Включить webhook** (Enable Webhook).
   2. В секции **Подробности о webhook** (Webhook details) настройте следующие параметры:
      - **Сервис webhook** (Webhook Service): **GitFlic**.
      - **Полномочия webhook** (Webhook Credential): *Доступ к GitFlic*.
   3. Сохраните настройки.
5. После сохранения шаблона контроллер сгенерирует параметры доступа для шаблона в следующем виде:
   - **Webhook URL**: <webhook URL>;
   - **Ключ webhook** (Webhook key): <hash code>.
6. Скопируйте сгенерированные параметры, они понадобятся для настройки webhook в GitFlic.

11.5.3 Настройка GitFlic

Для настройки GitFlic выполните следующие действия:

1. В настройках проекта в GitFlic выберите вкладку **Вебхуки**.
2. Нажмите кнопку *Создать*.
3. В полях **URL для отправки данных** и **Секрет** введите значения, полученные ранее. Прочие поля оставьте пустыми.
4. Выберите события, при наступлении которых по указанному URL будут передаваться данные, например, **Новый запрос на слияние**.

11.5.4 Проверка работоспособности

Для проверки корректности настройки webhook выполните следующие действия:

1. Создайте запрос на слияние в GitFlic.
2. Убедитесь, что в Automation Controller стартовало задание.
3. После выполнения задания откройте его и выберите вкладку **Подробности** (Details).
4. Убедитесь, что переменная `awx_webhook_status_api` имеет значение `null`.
5. Убедитесь, что в файле `/var/log/tower/tower.log` нет ошибок.

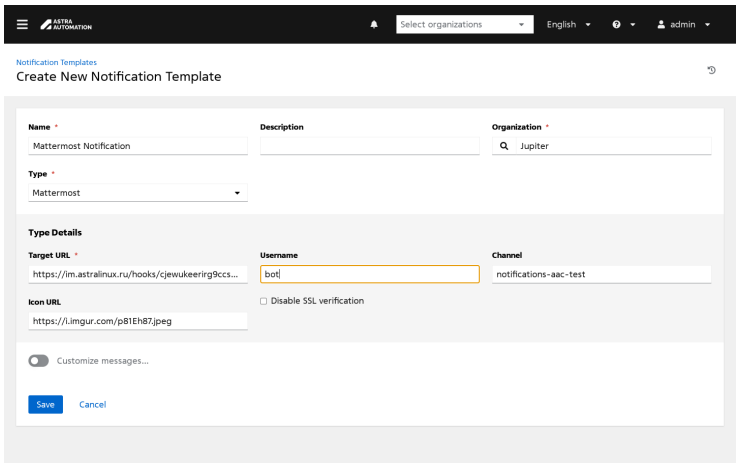
11.6 Отправка уведомлений в Mattermost

В данном документе представлена инструкция по настройке уведомлений из Automation Controller на примере Mattermost.

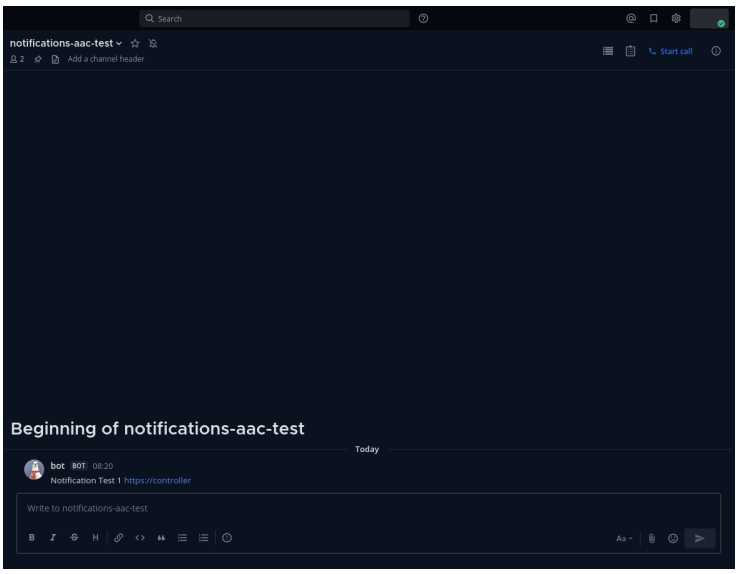
11.6.1 Создание уведомления

Для создания уведомления выполните следующие действия:

1. Перейдите в раздел *Администрирование* ▸ *Уведомления* (*Administration* ▸ *Notifications*) и нажмите кнопку *Добавить* (Add).



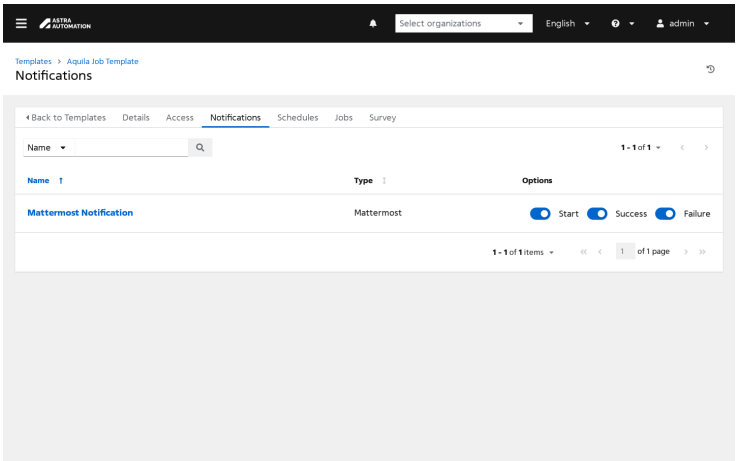
2. *Создайте* новый шаблон уведомления.
3. После сохранения откроется окно с созданным уведомлением, в котором можно проверить его работоспособность, нажав кнопку *Тест*. В канал Mattermost направляется уведомление следующего вида:



11.6.2 Добавление уведомления к шаблонам заданий и потоков заданий

Для добавления уведомлений к шаблонам заданий и потоков заданий выполните следующие действия:

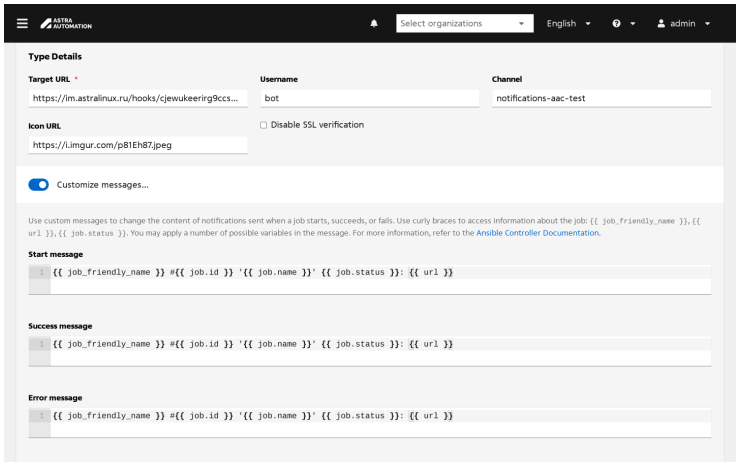
1. Перейдите в раздел *Ресурсы* ▸ *Шаблоны* (*Resources* ▸ *Templates*) и выберите вкладку **Уведомления** (Notifications).
2. Напротив созданного уведомления включите необходимые флаги для отправки уведомления:
   - **Согласование** (Approval) – подтверждение или отмена действия, связанного с заданиями.
   - **Пуск** (Start) – запуск заданий.
   - **Успех** (Success) – успешное выполнение заданий.
   - **Сбой** (Failure)– возникновение ошибок при выполнении заданий.



11.6.3 Изменение текста сообщений

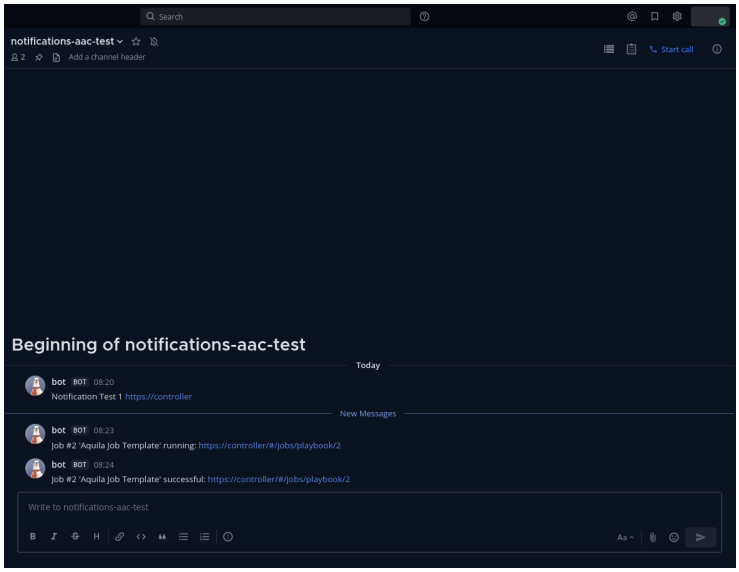
Для изменения текста сообщений в уведомлениях выполните следующие действия:

1. В настройках созданного ранее уведомления переведите переключатель **Настройка сообщений...** во включенное состояние:

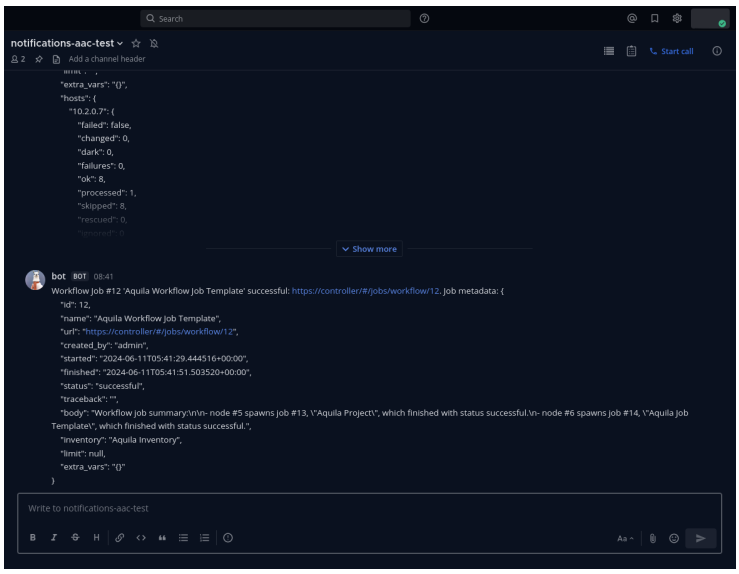


2. Выберите событие, для которого необходимо изменить сообщение.
3. Если необходимо добавить основную информацию о задании или потоке заданий, используйте переменную Ansible `{{ job_metadata }}`. Это словарь, который содержит следующие поля:
   - `inventory`;
   - `project`;
   - `playbook`;
   - `credential`;
   - `limit`;
   - `extra_vars`;
   - `hosts`;
   - `body` (отображается только в потоке заданий).

   Пример вывода для шаблона задания:



   Пример вывода для потока заданий:



11.7 Настройка SSO в Automation Controller с использованием ALD Pro

Automation Controller может использовать контроллер домена ALD Pro в качестве поставщика услуг аутентификации (SSO) с применением метода LDAP.

11.7.1 Описание сценария

Процесс настройки SSO для Automation Controller с использованием ALD Pro состоит из следующих этапов:

1. Настройка домена ALD Pro.

   Параметры домена:

Параметр	Значение
Версия ALD Pro	`2.4.0`
Название домена	`domain.example`
FQDN контроллера домена	`dc.domain.example`

   В домене необходимо создать следующие группы:

   - `network_admins`;
   - `gitflic_admins`;
   - `controller_admins`.

   Пользователи этих групп при авторизации в Automation Controller будут автоматически получать необходимые привилегии.

2. Изменение настроек Automation Controller.

   В организации `Default` необходимо создать команды `Network Admins` и `GitFlic Admins` с необходимыми привилегиями. Пользователи доменных групп `network_admins` и `gitflic_admins` автоматически становятся участниками этих команд.

   Пользователи доменной группы `controller_admins` автоматически получают привилегии администратора Automation Controller.

3. Проверка работоспособности SSO.

11.7.2 Подготовка к работе

Подготовьте окружение к настройке SSO:

1. Авторизуйтесь в панели управления контроллером домена ALD Pro.
2. Чтобы Automation Controller мог получать данные для авторизации из ALD Pro, создайте учетную запись пользователя `acadmin` и добавьте ее в группу `admin`.
3. Создайте группы `network_admins`, `gitflic_admins` и `controller_admins`.
4. Добавьте в созданные группы нужных пользователей.

11.7.3 Настройка Automation Controller

Авторизуйтесь в Automation Controller от имени администратора и выполните следующие действия:

1. В организации `Default` создайте команды `Network Admins` и `GitFlic Admins`, следуя *инструкции*.
2. По своему усмотрению настройте доступ для каждой команды к различным ресурсам. Порядок настройки описан в *инструкции*.
3. На панели навигации выберите *Настройки* ▸ *Аутентификация* (*Settings* ▸ *Authentication*).
4. Нажмите на ссылку *Настройки LDAP* (LDAP Settings).
5. В окне **Подробности** (Details) нажмите кнопку *Редактировать* (Edit).
6. Заполните поля формы **Редактировать детали** (Edit Details):
   - **URI сервера LDAP** (LDAP Server URI):

     ```
     ldap://dc.domain.example:389
     ```

   - **Пароль привязки к LDAP** (LDAP Bind Password): пароль учетной записи, от имени которой Automation Controller будет обращаться к ALD Pro для получения данных авторизации.
   - **Тип группы LDAP** (LDAP Group Type): **NestedMemberDNGroupType**.
   - **Запуск LDAP с TLS** (LDAP Start TLS): если для защиты подключения к контроллеру домена используется сертификат, выданный удостоверяющим центром, переведите этот переключатель во включенное положение.
   - **LDAP Bind DN**:

     ```
     UID=acadmin,CN=users,CN=accounts,DC=domain,DC=example
     ```

   - **Шаблон уникального наименования (DN) пользователя в LDAP** (LDAP User DN Template):

     ```
     UID=%(user)s,CN=users,CN=accounts,DC=domain,DC=example
     ```

   - **Групповой поиск по LDAP** (LDAP Group Search):

     ```
     [
         "cn=groups,cn=accounts,dc=domain,dc=example",
         "SCOPE_SUBTREE",
         "(objectClass=groupOfNames)"
     ]
     ```

   - **Ассоциация пользовательских атрибутов с LDAP** (LDAP User Attribute Map):

```
{
  "email": "email",
  "last_name": "sn",
  "first_name": "givenName"
}
```

- **Параметры типа группы LDAP** (LDAP Group Type Parameters):

```
{
  "name_attr": "cn",
  "member_attr": "member"
}
```

- **Пользовательские флаги LDAP по группам** (LDAP User Flags By Group):

```
{
  "is_superuser": [
    "CN=controller_admins,CN=groups,CN=accounts,DC=domain,DC=example"
  ]
}
```

- **Ассоциация организации с LDAP** (LDAP Organization Map):

```
{
  "Default": {
    "remove_admins": false,
    "remove_users": false,
    "users": [
      "CN=network_admins,CN=groups,CN=accounts,DC=domain,DC=example",
      "CN=gitflic_admins,CN=groups,CN=accounts,DC=domain,DC=example"
    ]
  }
}
```

- **Ассоциация группы с LDAP** (LDAP Team Map):

```
{
  "Network Admins": {
    "organization": "Default",
    "remove": true,
    "users": "CN=network_admins,CN=groups,CN=accounts,DC=domain,DC=example"
  },
  "GitFlic Admins": {
    "organization": "Default",
    "remove": true,
    "users": "CN=gitflic_admins,CN=groups,CN=accounts,DC=domain,DC=example"
  }
}
```

7. Нажмите кнопку *Сохранить* (Save).

11.7.4 Проверка работоспособности SSO

Чтобы проверить корректность настроек SSO, выполните следующие действия:

1. Завершите сессию администратора Automation Controller или откройте страницу авторизации в приватной вкладке.
2. На странице авторизации введите данные доменного пользователя:
   - **Имя пользователя** (Username): краткое название доменной учетной записи.

     Например, если полное название доменной учетной записи `johndow@domain.example`, введите только `johndow`.
   - **Пароль** (Password): пароль доменной учетной записи.

3. Нажмите кнопку *Вход* (Login).
4. Убедитесь, что авторизация прошла успешно и доступны ресурсы, разрешения для которых были настроены на уровне команд.

11.7.5 Особенности проекта

Обратите внимание на следующие особенности проекта:

- Создание отдельного доменного пользователя для настройки SSO.
- Настройка сопоставления доменных групп и команд Automation Controller.
- Особенности заполнения полей при настройке параметров LDAP.

11.7.6 Заключение

В этом сценарии вы познакомились с основными шагами по настройке SSO в Automation Controller с использованием в качестве поставщика услуг единого входа контроллера домена ALD Pro. Из всей последовательности шагов важно выделить следующие действия:

- Подготовка окружения.
- Настройка параметров Automation Controller.
- Проверка корректности работы SSO.

ГЛАВА 12

Обеспечение безопасности

В этом разделе рассматривается обеспечение безопасности Astra Automation.

12.1 Сетевые настройки

Существуют различные способы атак и защиты в сети, которые учитывают в корпоративных инфраструктурах. Некоторые из них следует учитывать при настройке узлов контроллера.

12.1.1 Использование специальных каналов связи

В контурах сети, где накладываются особые ограничения на связь с внешним миром, используют такие каналы связи, которые обеспечивают минимальный размер периметра возможных атак.

Особенности ограничений

Специальный канал связи типично включает следующие ограничения (примеры приведены применительно к Astra Automation):

- Разрешается подключение только к определенным серверам.
- Разрешается подключение только по определенному протоколу, например, по SSH.
- Подключение разрешено определенному пользователю.

Реализация такого канала для контроллера требует учета следующих особенностей:

- Сервисы контроллера работают с привилегиями пользователя awx.
- Синхронизация проектов контроллера с Private Automation Hub или другими реестрами происходит с использованием той системной среды исполнения (EE), которая добавляется к контроллеру в процессе его установки. Системная среда исполнения использует по умолчанию образ `aa-base-ee`, версия которого может изменяться вместе с версией контроллера.

Примечание: Добавление какой-либо другой EE в настройках проекта влияет только на исполнение playbook, но не на синхронизацию проектов с репозиториями.

Пример настройки

Предположим, что выход в интернет по SSH должен происходить через прокси-сервер с адресом 10.111.222.1 по порту 2022. Для установления связи через прокси используется универсальная утилита ncat[217].

Примечание: Это внешняя утилита, которая должна выполняться внутри EE, однако она не установлена там по умолчанию.

Для использования утилиты `ncat` необходимо создать собственный образ EE и использовать его в качестве системного. Соответственно процесс состоит из следующих шагов:

1. *Создайте собственный образ EE*, включающий утилиту `ncat`, с помощью Ansible Builder.
2. Обеспечьте использование собственного образа EE в качестве системного при развертывании платформы. Для этого воспользуйтесь дополнительной переменной Ansible `control_plane_execution_environment`, значением которой должен быть URL требуемого образа EE, например:

```
./aa-setup -- -e control_plane_execution_environment=registry.astralinux.ru/aa/aa-
↪base-ee:<version>
```

Для подключения контроллера к внешнему серверу по SSH выполните следующие настройки в контроллере:

1. В каталоге `/var/lib/awx/.ssh/` создайте файл `config` со следующим содержимым:

```
Host <address>
   ProxyCommand ncat --proxy 10.111.222.1:2022 %h %p
   PreferredAuthentications publickey
   IdentityFile /var/lib/awx/.ssh/id_key
```

   Особенности настройки:

   - Подключение к прокси происходит с помощью утилиты `ncat`.
   - `IdentityFile` ссылается на приватный ключ, публичная часть которого должна быть расположена на сервере, указанном в параметре `Host`.

2. Обеспечьте подключение к EE тех компонентов, которые в нем отсутствуют. Это необходимо выполнить через графический интерфейс контроллера.

   Путь навигации: **Настройки** (Settings) > **Настройки заданий** (Jobs) > **Пути доступа к изолированным заданиям** (Paths to expose to isolated jobs).

   Дополните список монтирования следующими парами:

```
"/var/lib/awx/.ssh/config:/root/.ssh/config:ro",
"/var/lib/awx/.ssh/id_key:/root/.ssh/id_key:ro".
```

   В этом примере в EE попадут следующие недостающие компоненты:

   - файл настроек SSH `config`,
   - приватный ключ `id_key`.

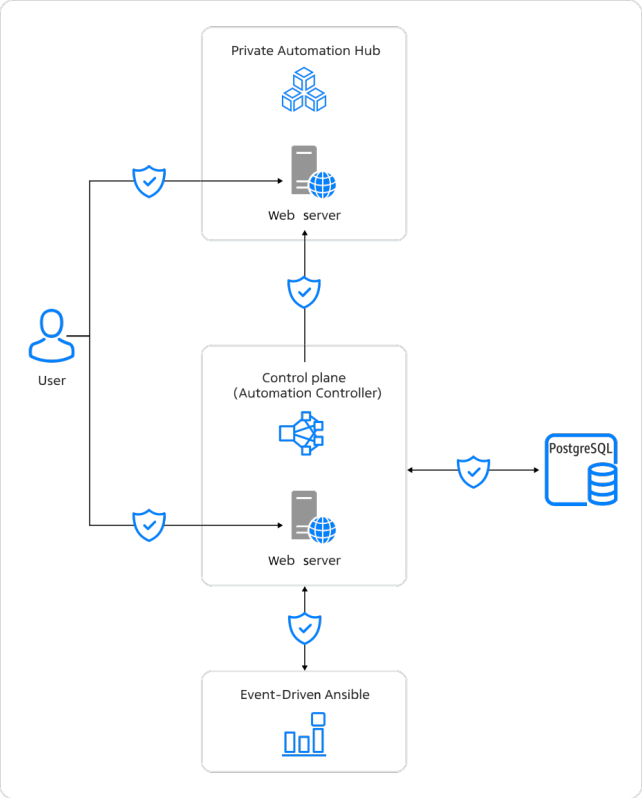
С использованием приведенных настроек контроллер будет иметь доступ к Automation Hub через указанный прокси-сервер для получения необходимого инфраструктурного кода.

[217] https://nmap.org/ncat/

12.2 Сертификаты TLS/SSL

При работе платформы в продуктовой среде для защиты подключения необходимо использовать сертификаты, выданные удостоверяющим центром. Они необходимы для защиты следующих соединений:

- пользователя с веб-сервером Automation Controller;
- пользователя с веб-сервером Private Automation Hub;
- Automation Controller с Private Automation Hub;
- Automation Controller с Event-Driven Ansible Controller;
- Automation Controller с сервером СУБД PostgreSQL, развернутым средствами Astra Automation;
- между узлами кластера Automation Controller (сеть Mesh).



Если Automation Controller или Private Automation Hub развернуты на нескольких узлах, но доступ к ним осуществляется через балансировщики нагрузки, в поле сертификата Common Name (CN) должно быть указано *FQDN* соответствующего балансировщика, а не узлов Automation Controller или Private Automation Hub.

Допускается использовать Wildcard-сертификат для узлов, находящихся в одном домене.

Если политики организации требуют использования уникального сертификата для каждого компонента Astra Automation, необходимо соблюдение следующих условий:

- Если используется балансировщик нагрузки, в данных каждого сертификата должно присутствовать поле Subject Alt Name (SAN), указывающее на FQDN балансировщика.
- На каждый узел следует установить соответствующий сертификат.

Пошаговые инструкции по управлению сертификатами приведены в документе *Управление сертификатами TLS/SSL*.

ГЛАВА 13

Производительность

Важно: Эта часть документации находится в стадии разработки.

13.1 Концепции

Важно: Эта часть документации находится в стадии разработки.

13.2 Мониторинг

Важно: Эта часть документации находится в стадии разработки.

13.3 Управление

Важно: Эта часть документации находится в стадии разработки.

ГЛАВА 14

Техническое обслуживание

Astra Automation представляет собой сложную платформу со значительным количеством различных компонентов. В связи с большой значимостью процессов автоматизации вопросы технической поддержки платформы являются одними из наиболее важных.

14.1 Использование журналов контроллера

Для хранения служебных сообщений в виде журналов в Automation Controller используется syslog-ng[218].

14.1.1 Общие сведения

`syslog-ng` позволяет передавать данные журналов в различные *приемники*: файлы на локальных и сетевых дисках, серверы журналирования, а также во внешние *агрегаторы журналов*.

Если сетевой приемник становится недоступным, `syslog-ng` сохраняет сообщения на локальном диске. При восстановлении соединения сообщения автоматически передаются из локальной копии в приемник в том же порядке, в котором они были получены.

При каждом изменении настроек журналирования происходит следующее:

- обновляется содержимое конфигурационного файла `/var/lib/awx/syslogng/syslog-ng.conf`;
- для применения новых настроек перезапускается служба `awx-syslogng`.

Совет: Настройки журналирования рекомендуется изменять через *графический интерфейс*.

`syslog-ng` поддерживает два режима дисковой буферизации:

- Надежная (reliable):
  - Сообщения из буфера удаляются только при получении от приемника подтверждения записи.

[218] https://github.com/syslog-ng/syslog-ng

  - Низкая производительность (по сравнению с нормальной дисковой буферизацией).
  - Обеспечивается сохранность сообщений при перезагрузке и перезапуске службы, потере подключения к приемнику или сбоя в работе самого `syslog-ng`.
- Обычная (normal):
  - Высокая производительность.
  - Возможна потеря сообщений в случае сбоя `syslog-ng`.

Если Automation Controller настроен на передачу журналов во внешний агрегатор, для `syslog-ng` используются настройки, реализующие следующее поведение:

- Исходящие сообщения помещаются непосредственно в очередь вывода, если она не заполнена.
- Если очередь вывода заполнена (количество сообщений в очереди достигло значения `mem-buf-length`), а параметру `reliable` присвоено значение `no`, сообщения помещаются в обычный дисковый буфер.

Если контроллер настроен на передачу журналов во внешний агрегатор с использованием HTTP/TCP/UDP, по умолчанию используются следующие настройки дисковой буферизации:

```
disk-buffer(
  dir("/var/lib/awx")
  disk-buf-size(1073741824)
  qout-size(1000)
  mem-buf-length(131072)
  reliable(no)
)
```

Некоторые из указанных настроек можно изменить только через API (точка доступа – `/api/v2/settings/logging/`):

- `dir` – полный путь к локальному каталогу для хранения журналов.

  Параметр API – `LOG_AGGREGATOR_MAX_DISK_USAGE_PATH`.
- `disk-buf-size` – размер дискового буфера в байтах.

  Параметр API – `LOG_AGGREGATOR_ACTION_MAX_DISK_USAGE_GB`.

  Важно: При изменении через API значение этого параметра следует указывать в гигабайтах.

- `mem-buf-length` – максимальное количество сообщений в очереди.

  Параметр API – `LOG_AGGREGATOR_ACTION_QUEUE_SIZE`.

14.1.2 Агрегаторы

Automation Controller может быть настроен на передачу журналов в следующие внешние агрегаторы:

- Logstash[219];
- Splunk[220];
- Loggly[221];
- Sumo Logic[222].

[219] https://www.elastic.co/guide/en/logstash/current/index.html
[220] https://docs.splunk.com/Documentation
[221] https://documentation.solarwinds.com/en/success_center/loggly/content/loggly_documentation.htm
[222] https://help.sumologic.com/docs/get-started/

В агрегатор передаются только сообщения, важность которых выше или равна указанной (в порядке возрастания важности):

- **DEBUG** – все сообщения журнала;
- **INFO** – информационные сообщения;
- **WARNING** – предупреждения;
- **ERROR** – информация об ошибках;
- **CRITICAL** – информация о критических событиях.

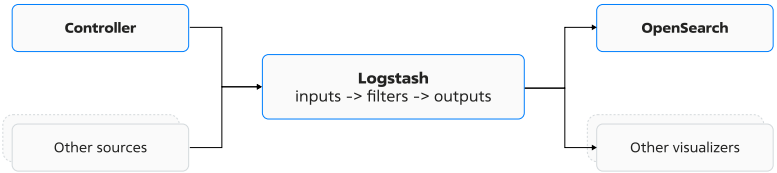
По умолчанию в агрегатор передаются записи, полученные от всех внутренних регистраторов Automation Controller:

- awx – общие журналы сервера;
- `activity_stream` – запись изменений объектов в Automation Controller;
- `job_events` – данные, возвращаемые модулем обратного вызова Ansible;
- `system_tracking` – данные о фактах, собранные модулем `setup`;
- `broadcast_websocket` – данные об отправленных сообщениях по WebSocket.

14.1.3 Журналирование с помощью Logstash и OpenSearch

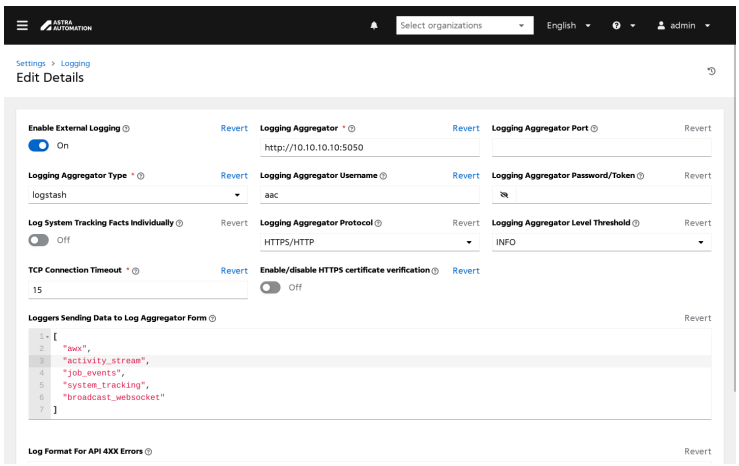
Программное обеспечение Logstash используется для хранения записей журналов различных форматов. Поддержка форматов реализуется расширениями. Для визуализации хранящихся данных могут быть использованы различные внешние системы, например, OpenSearch, Elasticsearch и так далее.

Logstash может одновременно обрабатывать данные, полученные от разных поставщиков.



Чтобы настроить Automation Controller на передачу журналов в систему OpenSearch через агрегатор Logstash, выполните следующие действия:

1. В *настройках журналирования* задайте значения следующих параметров:
   - **Агрегатор протоколирования**: `http://10.10.10.10:5050`.
   - **Тип агрегатора ведения журнала**: **logstash**.
   - **Имя пользователя агрегатора ведения журнала**: aac.
   - **Пароль/токен агрегатора ведения журнала**: пароль для подключения к Logstash.
   - **Протокол ведения журналов**: **HTTPS/HTTP**.
   - **Включить/отключить проверку сертификата HTTPS**: выкл.
   - **Порог уровня агрегатора ведения журнала**: **INFO**.
   - **Включить внешнее ведение журнала**: вкл.



2. Установите Logstash согласно инструкции[223].
3. Для работы с Automation Controller отредактируйте файл конфигурации агрегатора `logstash.conf`:

Logstash 7

```
input {
  http {
    port => <listen_port>
    user => <username>
    password => "<password>"
    codec => json {
      charset => "UTF-8"
    }
  }
}

output {
  opensearch {
    hosts => ["https://<opensearch_host>:<opensearch_port>"]
    index => "<aac-index-name>-%{+YYYY.MM.dd}"
    user => "<opensearch_user>"
    password => "<opensearch_password>"
    ssl => true
    ssl_certificate_verification => false
  }
}
```

Logstash 8

```
input {
  http {
    port => <listen_port>
    user => <username>
    password => "<password>"
    codec => json {
      charset => "UTF-8"
    }
  }
}

filter {
  json {
```

(continues on next page)

[223] https://opensearch.org/docs/latest/tools/logstash/index/

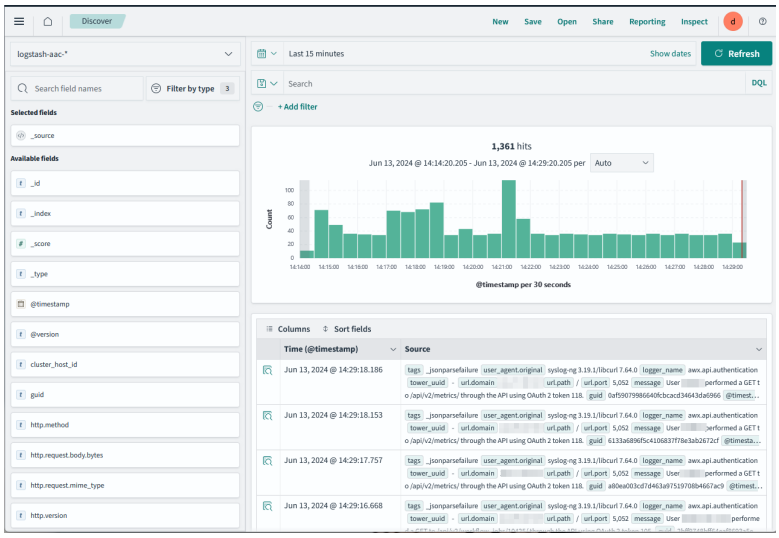
(продолжение с предыдущей страницы)

```
    source => "message"
  }
}

output {
  opensearch {
    hosts => ["https://<opensearch_host>:<opensearch_port>"]
    index => "<aac-index-name>-%{+YYYY.MM.dd}"
    user => "<opensearch_user>"
    password => "<opensearch_password>"
    ssl => true
    ssl_certificate_verification => false
  }
}
```

где:

- `listen_port` – порт, на котором Logstash будет ожидать соединение;
- `username` – название учетной записи для подключения к серверу Logstash;
- `password` – пароль для подключения к серверу Logstash;
- `opensearch_host` – IP-адрес или FQDN сервера OpenSearch;
- `opensearch_port` – порт сервера OpenSearch;
- `opensearch_user` – название учетной записи пользователя для подключения к серверу OpenSearch;
- `opensearch_password` – пароль для подключения к серверу OpenSearch.

Пример визуализации журналов Automation Controller в режиме Discover[224]:



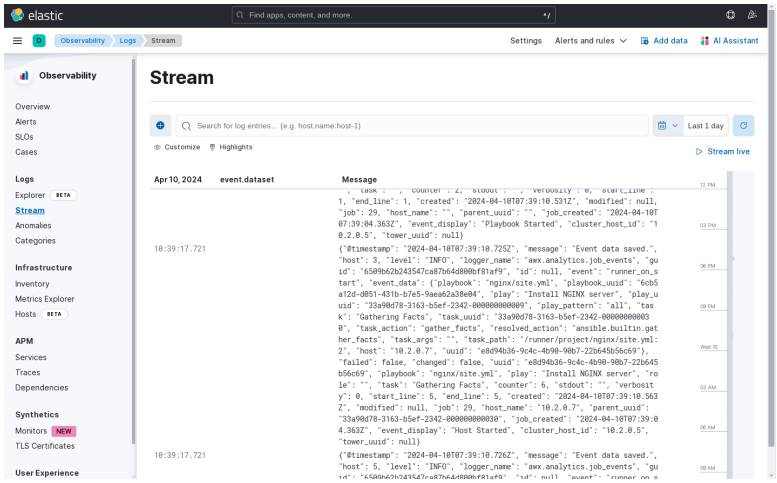
[224] https://opensearch.org/docs/2.4/dashboards/discover/index-discover/

14.1.4 Журналирование с помощью Logstash и Elasticsearch

Чтобы настроить Automation Controller на передачу журналов в систему Elasticsearch через агрегатор Logstash, в *настройках журналирования* укажите значения следующих параметров:

- **Агрегатор протоколирования** – FQDN сервера Elasticsearch, например, `test-elastic.zapto.org`.
- **Порт агрегатора протоколирования** – порт сервера Elasticsearch, например, `50000`.
- **Тип агрегатора ведения журнала** – **logstash**.
- **Протокол ведения журналов** – **TCP**.
- **Порог уровня агрегатора ведения журнала** – **INFO**.
- **Включить внешнее ведение журнала** – вкл.

Пример визуализации журналов контроллера в режиме **Stream**:



14.2 Поиск причин неисправностей

Важно: Эта часть документации находится в стадии разработки.

14.3 Примеры сценариев

Раздел содержит примеры сценариев технического обслуживания Astra Automation.

14.3.1 Решение проблем с помощью awx-manage

При работе контроллера возможен выход из строя как отдельных его узлов, так и групп узлов. Повторный запуск сценария настройки не приводит к автоматическому исключению из топологии таких узлов и групп узлов, поскольку у утилиты `aa-setup` нет возможности определить причину выхода узла или группы узлов из строя. Для восстановления работоспособности контроллера можно использовать утилиту `awx-manage`.

Удаление одного узла

Для удаления одного узла выполните следующие действия:

1. Подключитесь по SSH к любому управляющему узлу контроллера.
2. Запустите утилиту `awx-manage` с параметром `deprovision_instance`:

   ```
   sudo awx-manage deprovision_instance --hostname=<hostname>
   ```

   Здесь <hostname> – название удаляемого узла в инвентаре установщика платформы.

Примечание: Узел необходимо удалить из инвентаря установщика платформы, если он не должен появиться в кластере при повторном запуске сценария настройки.

Удаление группы узлов

Для удаления группы узлов выполните следующие действия:

1. Удалите каждый узел отдельно как описано ранее.
2. Удалите группу узлов:

   ```
   sudo awx-manage unregister_queue --queuename=<queue_name>
   ```

   Здесь <queue_name> – название группы узлов.

14.4 Использование базы знаний

Важно: Эта часть документации находится в стадии разработки.

14.5 Техническая поддержка

Положение о технической поддержке Astra Automation доступно на сайте ПАО Группа Астра по ссылке: https://astra.ru/support/support-docs/polozhenie-o-tekhnicheskoy-podderzhke-astra-automation/

Перед созданием обращения в службу технической поддержки убедитесь, что подходящего решения нет в документации Astra Automation и Базе знаний[225]. При создании обращения через Личный кабинет[226] решения из Базы знаний предлагаются автоматически.

Инструкции по использованию Личного кабинета доступны в справочном центре в разделе Личный кабинет пользователя Astra Linux[227].

[225] https://wiki.astralinux.ru/kb/
[226] https://lk-new.astralinux.ru/
[227] https://wiki.astralinux.ru/pages/viewpage.action?pageId=128615802

14.6 Резервное копирование и восстановление

Резервное копирование – это процесс создания резервной копии, то есть данных, которые необходимы для восстановления платформы Astra Automation.

Резервная копия представляет собой единый файл архива, включающий в себя следующие данные:

- информация о настройках системы;
- ключи шифрования паролей и секретов базы данных;
- каталог с локальными копиями проектов;
- резервная копия базы данных;
- сертификаты.

Архив состоит из файлов:

- `<node>.tar.gz` – настройки системы и каталог локальных проектов (`<node>` – название первого узла в группе `[automationcontroller]`);
- `ca.tar.gz` – сертификаты;
- `common.tar.gz` – ключи шифрования паролей и секретов базы данных;
- `postgres.tar.gz` – резервная копия базы данных.

14.6.1 Требования к узлу

Рекомендуется выполнять резервное копирование и восстановление на установочном узле или на узле, удовлетворяющем следующим требованиям:

- установленный пакет `astra-automation-setup`;
- корректно заполненный файл `/opt/rbta/aa/astra-automation-setup/inventory`.

Утилита назначает владельцем создаваемого файла пользователя, указанного в переменной `ansible_user` раздела `[all:vars]` инвентаря. Если эта переменная отсутствует, то этим пользователем будет `admin`.

Пример настройки:

```
[all:vars]
ansible_user = astra
```

> **Предупреждение:** Восстанавливать кластер следует в той же конфигурации, которая была на момент создания резервной копии.

При установке с помощью коллекции эти требования соблюдаются на узле с примененной ролью `astra.aa_controller.aac`.

14.6.2 Резервное копирование

Для выполнения резервного копирования выполните следующие действия:

1. Убедитесь, что в файле `inventory` в разделе `[all:vars]` значением параметра `ansible_user` является название учетной записи пользователя ОС установочного узла, например:

   ```
   [all:vars]
   ansible_user = astra
   ```

2. Для выполнения резервного копирования выполните команду:

```
sudo ./aa-setup -b
```

По умолчанию резервная копия сохраняется в каталог `/tmp/backups/astra-automation/`. Чтобы сохранить резервную копию в другой каталог, укажите полный путь нему в качестве позиционного аргумента после аргумента `-b`, например:

```
sudo ./aa-setup -b /opt/backups/
```

В каталоге для хранения резервных копий также создается символическая ссылка `astra-automation-backups-latest.tar.gz`, которая всегда указывает на самую новую резервную копию в каталоге.

14.6.3 Восстановление из резервной копии

Для восстановления Astra Automation из резервной копии выполните следующие действия:

1. Скопируйте архив с резервной копией в каталог `/tmp/backups/astra-automation/`.
2. Перейдите в каталог `/opt/rbta/aa/astra-automation-setup/` и запустите утилиту `aa-setup` с аргументом `-r`:

   ```
   sudo ./aa-setup -r
   ```

   По умолчанию утилита `aa-setup` восстанавливает резервную копию из самого нового архива, размещенного в каталоге `/tmp/backups/astra-automation/`. Чтобы использовать для восстановления другой архив, укажите полный путь к нему в качестве позиционного аргумента после аргумента `-r`, например:

   ```
   sudo ./aa-setup -r /opt/backups/astra-automation-backup-16-06-2024-155236.tar.gz
   ```

3. Если резервная копия была создана в версии платформы отличающейся от той, в которой она восстанавливается, утилиту `aa-setup` обязательно нужно запустить еще раз:

   ```
   sudo ./aa-setup
   ```

   Это обусловлено несоответствием структуры таблиц базы данных в разных версиях платформы.

4. Для проверки корректности восстановления выполните вход в веб-интерфейс контроллера от имени администратора и проверьте содержимое разделов.

14.7 Дополнительные настройки контроллера

Большинство настроек контроллера можно изменить через графический интерфейс, CLI или API. При этом изменения автоматически применяются на всех узлах управляющей плоскости.

Значения по умолчанию для многих настроек хранятся в конфигурационном файле `/var/lib/awx/venv/awx/lib/python3.9/site-packages/awx/settings/defaults.py`. Также в этом файле задаются значения для нескольких настроек, которые нельзя изменить указанными выше способами.

Если для некоторых настроек необходимо задать значения, отличающиеся от значений по умолчанию, на всех узлах управляющей плоскости выполните следующие действия:

1. В каталоге `/etc/tower/conf.d/` создайте файл `custom.py`, в котором укажите нужные значения соответствующих настроек.
2. Перезапустите службы контроллера:

```
sudo automation-controller-service restart
```

Предупреждение: Хотя в некоторых случаях управление настройками указанным способом может быть оправданным, описанный способ не рекомендуется по следующим причинам:

- Поведение многих параметров не документировано.
- Контроллер не проверяет корректность настроек в `/etc/tower/conf.d/custom.py` на этапе заполнения.
- Контроллер не обеспечивает синхронизацию содержимого файла `/etc/tower/conf.d/custom.py` между узлами. Любые изменения в этом файле нужно будет распространять на узлах из плоскости управления вручную, в том числе при добавлении новых узлов.
- Настройки, заданные в `/etc/tower/conf.d/custom.py`, могут быть переопределены значениями, хранящимися в базе данных.

Ниже приводятся примеры изменения значений настроек контроллера, которые невозможно изменить через графический интерфейс, CLI или API.

- Период проверки статуса узлов в секундах:

```
CLUSTER_NODE_HEARTBEAT_PERIOD = 30
```

- Количество неуспешных проверок статуса узла, при котором он считается вышедшим из строя:

```
CLUSTER_NODE_MISSED_HEARTBEAT_TOLERANCE = 1
```

- Изменение пути к каталогу для хранения проектов:

```
PROJECT_ROOT = '/var/share/projects/'
```

ГЛАВА 15

Дополнительные материалы

Раздел содержит дополнительные материалы по продуктам, сервисам и инструментам для работы с Astra Automation.

15.1 Приложение 1. Общие настройки

Для применения Astra Automation необходимо подготовить управляемые узлы к работе.

15.1.1 Доступ по SSH с помощью ключей

Рекомендуемым способом доступа к *управляемым узлам* является подключение по протоколу SSH.

Для создания пары ключей SSH выполните команду:

```
ssh-keygen -C "<comment>" -f ~/.ssh/<file_name>
```

Здесь:

- <comment> – комментарий к ключу. Например, здесь можно указать свой адрес электронной почты или предназначение ключа.
- <file_name> – имя файла, под которым будет сохранена приватная часть ключа. Публичная часть ключа будет сохранена в том же каталоге в файле с тем же именем, но с расширением `.pub`.

Подробности о команде `ssh-keygen` доступны во встроенной справке:

```
man ssh-keygen
```

Пример

Для создания пары ключей можно использовать следующую команду:

```
ssh-keygen -C "Deployment key" -f ~/.ssh/deployment
```

В каталоге `~/.ssh/` будут созданы два файла:

- `deployment` – приватная часть ключа;
- `deployment.pub` – публичная часть ключа.

15.1.2 Использование sudo без ввода пароля

Для настройки управляемых узлов с помощью Ansible необходимо разрешить на них использование команды sudo без ввода пароля.

Выполните настройки, соответствующие используемому дистрибутиву Linux:

- Astra Linux Special Edition

  Выполните команду:

  ```
  sudo astra-sudo-control disable
  ```

  Примечание: В универсальных базовых образах Astra Linux[228] использование sudo без ввода пароля включено по умолчанию.

- Другие дистрибутивы

  Создайте в каталоге `/etc/sudoers.d/` файл `sudo-without-password` со следующим содержимым:

  ```
  %sudo ALL=(ALL) NOPASSWD: ALL
  ```

15.2 Приложение 2. Устанавливаемые продукты

Важно: Эта часть документации находится в стадии разработки.

В данном разделе вы найдете общее описание продуктов, поддерживаемых платформой Astra Automation.

15.2.1 ALD Pro

ALD Pro – это набор сетевых служб сервера Astra Linux для организации централизованного управления ИТ-инфраструктурой. Развертывание домена ALD Pro с использованием коллекции `astra.ald_pro` из реестра Automation Hub позволяет автоматизировать большую часть рутинных операций.

[228] https://registry.astralinux.ru/

Структура

Домен ALD Pro может состоять из контроллера домена, его реплик, узлов с дополнительными службами домена и клиентов домена. Для настройки каждого компонента существует отдельная роль. Также в состав коллекции входит служебная роль `astra.ald_pro.common`, которая используется другими ролями.

Далее приводится пример использования следующих ролей:

- `astra.ald_pro.controller`;
- `astra.ald_pro.replica`;
- `astra.ald_pro.client`.

Параметры настройки

Для настройки параметров домена необходимо присвоить соответствующие значения переменным роли.

Роль `astra.ald_pro.controller`

Роль `astra.ald_pro.controller` используется для настройки контроллера домена ALD Pro.

Переменные роли:

- `aldpro_global_forwarders` – IP-адреса DNS-серверов, используемых для разрешения имен за пределами домена.

  Значение по умолчанию:

  - `8.8.8.8`;
  - `8.8.4.4`.

- `aldpro_server_minimal_memory_mb` – минимальное количество мегабайт оперативной памяти, необходимое для развертывания служб контроллера домена ALD Pro.

  Развертывание контроллера на узле запускается только в том случае, когда на нем установлено количество оперативной памяти большее или равное указанному.

  Значение по умолчанию – `4096` (4 ГБ).

- `aldpro_enable_syncer` – установка модуля синхронизации `aldpro-syncer`.

  Этот модуль необходим для использования расширенных функций интеграции с доменом Microsoft Active Directory.

  Значение по умолчанию – `true`.

- `aldpro_enable_gc` – установка модуля глобального каталога `aldpro-gc`.

  Этот модуль необходим, если используется топология из контроллера домена и нескольких реплик. Службы, предоставляемые этим модулем, выполняют синхронизацию данных пользователей между контроллером домена и его репликами.

  Значение по умолчанию – `true`.

- `aldpro_start_upgrade` – флаг, указывающий, что выполняется обновление версии ALD Pro, а не развертывание контроллера домена.

  Значение по умолчанию – `false`.

Роль astra.ald_pro.replica

Роль astra.ald_pro.replica используется для настройки реплик контроллера домена ALD Pro.

Переменные роли:

- aldpro_pdc_ip – IP-адрес контроллера домена, например, 192.168.56.10.
- aldpro_pdc_name – краткое имя контроллера домена.

  Например, если FQDN контроллера домена – dc1.moscow.example.com, то краткое имя – dc1.
- aldpro_dc_site_name – название сайта ALD Pro.

  Значение по умолчанию – Головной офис.

Развертывание реплики происходит в два этапа:

1. развертывание служб клиента домена;
2. преобразование клиента домена в реплику.

Если службы клиента домена уже настроены, этот этап развертывания реплики можно пропустить. Для этого необходимо выполнить следующие действия:

1. Добавить в playbook для соответствующего задания тег replica, например:

```
---
- name: Enrolling ALD Pro replica
  hosts: replicas
  become: true
  tags: replica
  roles:
    - astra.ald_pro.replica
```

2. Запустить playbook с тегом replica.

```
ansible-navigator run playbook.yml -i inventory.yml -t replica -m stdout
```

Роль astra.ald_pro.client

Роль astra.ald_pro.client используется для настройки клиентов домена ALD Pro.

Переменные роли:

- aldpro_pdc_ip – IP-адрес контроллера домена.

  Значение по умолчанию – 192.168.56.10.
- aldpro_pdc_name – имя контроллера домена.

  Значение по умолчанию – dc01.

Роль astra.ald_pro.common

Переменные роли astra.ald_pro.common:

- aldpro_domain – название домена.
- aldpro_admin_password – пароль администратора домена.

  Длина пароля – не менее 8 символов.
- aldpro_version – используемая версия ALD Pro.

  По умолчанию используется самая новая из доступных версий ALD Pro.

- `aldpro_repo` – список ссылок на репозитории, которые следует использовать для установки пакетов ALD Pro.

  Например, для развертывания ALD Pro версии 2.4.1 эта ссылка может иметь следующий вид:

  ```
  deb https://dl.astralinux.ru/aldpro/frozen/01/2.4.1 1.7_x86-64 main base
  ```

- `aldpro_start_upgrade` – если значение этой переменной равно `true`, производится обновление ALD Pro до более новой версии. При значении `false` происходит развертывание контроллера и реплик, ввод клиентов в домен.
- `aldpro_main_repo_upgrade` – список ссылок на репозитории, которые следует использовать для обновления пакетов ALD Pro.

  Например, для обновления ALD Pro до версии 2.4.1 эта ссылка может иметь следующий вид:

  ```
  deb https://dl.astralinux.ru/aldpro/frozen/01/2.4.1 1.7_x86-64 main base
  ```

- `aldpro_upgrade_hosts_list` – список узлов, на которых необходимо обновить версию ALD Pro.

Обязательные переменные

При использовании ролей обязательны для заполнения следующие переменные:

- `astra.ald_pro.controller`:
  - `aldpro_admin_password`;
  - `aldpro_domain`;
  - `aldpro_main_repo_upgrade`;
  - `aldpro_repo`;
- `astra.ald_pro.replica`:
  - `aldpro_admin_password`;
  - `aldpro_dc_site_name`;
  - `aldpro_domain`;
  - `aldpro_main_repo_upgrade`;
  - `aldpro_pdc_ip`;
  - `aldpro_pdc_name`;
  - `aldpro_repo`;
- `astra.ald_pro.client`:
  - `aldpro_admin_password`;
  - `aldpro_domain`;
  - `aldpro_main_repo_upgrade`;
  - `aldpro_pdc_ip`;
  - `aldpro_pdc_name`;
  - `aldpro_repo`.

Последовательность применения

Для применения коллекции `astra.ald_pro` выполните следующие действия:

1. Подготовьте узлы с необходимыми техническими характеристиками.

   Требования к техническим характеристикам узлов и методики расчета приведены в документе РДЦП.10101-01 95 01 «Программный комплекс «ALD Pro». Руководство администратора. Инструкция по развертыванию и обновлению»[229], п. 1.4 «Рекомендации по планированию ресурсов службы каталога».

2. Настройте управляемые узлы согласно инструкций:

   - *Доступ по SSH с помощью ключей*
   - *Использование sudo без ввода пароля*

3. Создайте каталог проекта, например:

```
mkdir ald-pro-enrolling
```

   Все упомянутые ниже файлы следует создавать в каталоге проекта, если только явно не указано иное.

4. Создайте файл с настройками Ansible `ansible.cfg`:

```
[defaults]
ansible_ssh_private_key_file = /path/to/private/ssh/key
host_key_checking = false
interpreter_python = /usr/bin/python3
inventory = inventory.ini
```

5. Создайте файл инвентаря `inventory.ini` и добавьте в него сведения об управляемых узлах, например:

```
[all]
dc1       ansible_host=192.168.56.12
dc2       ansible_host=192.168.56.13
client1   ansible_host=192.168.56.14
client2   ansible_host=192.168.56.15

[all:vars]
ansible_user=vagrant

[clients]
client1
client2
```

6. Создайте файл с переменными ролей `variables.yml`, например:

```
---
aldpro_admin_password: "aldpropass123456"
aldpro_dc_site_name: "Центральный офис"
aldpro_domain: "domain.example.com"
aldpro_enable_gc: true
aldpro_enable_syncer: true
aldpro_global_forwarders:
  - "8.8.8.8"
  - "1.1.1.1"
aldpro_pdc_ip: "192.168.56.12"
aldpro_pdc_name: "dc1"
aldpro_repo:
  - "deb https://dl.astralinux.ru/aldpro/frozen/01/2.4.1 1.7_x86-64 main base"
aldpro_start_upgrade: false
```

[229] https://www.aldpro.ru/docs/manual_admin.pdf

7. Создайте файл playbook и добавьте в него соответствующие роли для узлов домена, например:

```
---
- name: Enrolling ALD Pro domain controller
  hosts: dc1
  become: true
  vars_files:
    - variables.yml
  roles:
    - astra.ald_pro.controller

- name: Enrolling ALD Pro domain replica
  hosts: dc2
  become: true
  vars_files:
    - variables.yml
  roles:
    - astra.ald_pro.replica

- name: Enrolling ALD Pro clients
  hosts: clients
  become: true
  vars_files:
    - variables.yml
  roles:
    - astra.ald_pro.client
```

8. Запустите выполнение playbook с использованием образа среды исполнения `aa-full-ee`:

```
ansible-navigator run playbook.yml \
  --inventory inventory.ini \
  --eei hub.astra-automation.ru/aa-1.2/aa-full-ee:latest \
  --inventory inventory.ini
```

15.2.2 RuBackup

RuBackup – клиент-серверная система резервного копирования. Развертывание RuBackup с использованием коллекции `astra.rubackup` из реестра Automation Hub позволяет автоматизировать большую часть рутинных операций.

Структура

Система резервного копирования RuBackup может состоять из основного сервера резервного копирования, резервного сервера, медиа-серверов и клиентов.

Существуют роли для настройки следующих компонентов RuBackup:

- `astra.rubackup.rubackup_server` – основной сервер резервного копирования;
- `astra.rubackup.rubackup_media_server` – медиасерверы;
- `astra.rubackup.rubackup_client` – клиенты.

Также в состав коллекции входит роль `astra.rubackup.rubackup_common`, однако, ее непосредственное использование не допускается. Это служебная роль, которая используется другими ролями.

Для работы RuBackup необходим сервер PostgreSQL. Для его развертывания и настройки можно использовать коллекцию `astra.postgresql` из реестра Automation Hub.

Параметры настройки

Для настройки параметров RuBackup необходимо присвоить соответствующие значения переменным роли.

Роль astra.rubackup.rubackup_server

Роль astra.rubackup.rubackup_server используется для настройки основного сервера резервного копирования.

Переменные роли:

- rubackup_def_pool – путь к пулу хранилища по умолчанию.
- rubackup_server_pgsql_user – название учетной записи пользователя-администратора PostgreSQL.
- rubackup_server_psql_fqdn – *FQDN* сервера PostgreSQL.
- rubackup_server_psql_password – пароль учетной записи пользователя-администратора PostgreSQL.
- rubackup_server_psql_password_rubackup – пароль учетной записи пользователя PostgreSQL, от имени которого RuBackup подключается к PostgreSQL.

  Важно: Требования к паролю:

  - длина не менее 12 символов;
  - содержит хотя бы по одному символу следующих типов:
    * символ верхнего регистра;
    * цифра;
    * специальный символ.

- rubackup_server_users – учетные данные пользователей PostgreSQL, от имени которых RuBackup подключается к СУБД.

Роль astra.rubackup.rubackup_media_server

Роль astra.rubackup.rubackup_media_server используется для настройки медиа-серверов.

Переменные роли:

- rubackup_media_server_db_host – FQDN сервера PostgreSQL.
- rubackup_media_server_db_pass – пароль пользователя-администратора сервера PostgreSQL.
- rubackup_pri_fqdn – FQDN основного сервера резервного копирования.

Роль astra.rubackup.rubackup_client

Роль astra.rubackup.rubackup_client используется для настройки клиентов.

Переменные роли:

- rubackup_pri_fqdn – FQDN основного сервера резервного копирования.
- rubackup_servers_group_name – название группы, узлы которой необходимо зарегистрировать как клиенты.
- rubackup_users – список названий учетных записей пользователей PostgreSQL.

Обязательные переменные

При использовании ролей обязательные для заполнения следующие переменные:

- astra.rubackup.rubackup_server:
  - rubackup_packages_version;
  - rubackup_ram;
  - rubackup_server_psql_fqdn;
  - rubackup_server_psql_password_rubackup;
- astra.rubackup.rubackup_media_server:
  - rubackup_media_server_db_pass;
  - rubackup_packages_version;
  - rubackup_pri_fqdn;
  - rubackup_ram;
- astra.rubackup.rubackup_client:
  - rubackup_packages_version;
  - rubackup_pri_fqdn;
  - rubackup_ram;
  - rubackup_servers_group_name.

Последовательность применения

Для применения указанной коллекции выполните следующие действия:

1. Разверните сервер PostgreSQL любым удобным способом.
2. Подготовьте узлы с необходимыми техническими характеристиками.
3. Подготовьте узлы к настройке с помощью Ansible:
   - *обеспечьте доступ к узлам по SSH* с использованием ключей;
   - *разрешите на всех узлах использование sudo без ввода пароля*;
4. Создайте каталог проекта, например:

   ```
   mkdir rubackup-enrolling
   ```

   Все упомянутые ниже файлы следует создавать в каталоге проекта, если только явно не указано иное.
5. Создайте файл с настройками Ansible ansible.cfg:

```
[defaults]
collections_path = ./collections
host_key_checking = false
interpreter_python = /usr/bin/python3
inventory = inventory.ini
gathering = smart
timeout = 60

[ssh_connection]
pipelining = True
retries = 10
```

6. Создайте файл инвентаря `inventory.ini` и добавьте в него сведения об управляемых узлах, например:

```
[all]
server.example.com         ansible_host=192.168.56.101
media-server.example.com   ansible_host=192.168.56.102

[all:vars]
ansible_user=vagrant
ansible_ssh_private_key=./ssh/key
```

7. Создайте файл с переменными `variables.yml`, например:

```
---
# Settings for astra.rubackup roles
rubackup_user: "rubackup"
rubackup_user_password: "p@ssword!1"
rubackup_default_repository: >-
  deb https://dl.astralinux.ru/rubackup/repository-deb-main/ astra_1.7 stable
rubackup_pri_fqdn: "server.example.com"
rubackup_users:
  - rubackup
rubackup_media_server_db_pass: "SecretPassw@rd1"
rubackup_server_psql_password_rubackup: "SecretPassw@rd1"
rubackup_server_psql_fqdn: "server.example.com"
rubackup_server: "server.example.com"

postgresql_users:
  - name: "{{ rubackup_server_pgsql_user }}"
    password: "{{ rubackup_server_psql_password }}"
    state: present
postgresql_hba_entries:
  - type: local
    database: all
    user: "{{ rubackup_server_pgsql_user }}"
    auth_method: peer
  - type: local
    database: all
    user: all
    auth_method: peer
  - type: host
    database: all
    user: all
    address: "127.0.0.1/32"
    auth_method: md5
  - type: host
    database: all
    user: all
    address: "192.168.56.31/32"
    auth_method: md5
  - type: host
    database: all
    user: all
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    address: "192.168.56.32/32"
    auth_method: md5
  - type: host
    database: all
    user: all
    address: "192.168.56.33/32"
    auth_method: md5
rubackup_server_users:
  - rubackup
```

8. Создайте файл зависимостей Ansible `requirements.yml`, в который добавьте ссылку на загрузку коллекции `astra.ald_pro`, например:

```
---
collections:
  - name: astra.postgresql
    version: 2.2.3

  - name: astra.rubackup
    version: 0.6.0
```

9. Установите зависимости:

```
ansible-navigator exec -- ansible-galaxy install -r requirements.yml
```

10. Создайте файл playbook, в который добавьте роли для всех узлов домена, например:

```
---
# Create rubackup user and join it to the rubackup group
- hosts: server.example.com
  become: true
  tasks:
    - name: Create RuBackup system user (on Server)
      ansible.builtin.user:
        name: "{{ rubackup_user }}"
        password: "{{ rubackup_user_password | password_hash('sha512') }}"
        state: present

  vars_files:
    - variables.yml

# SERVER
- hosts: server.example.com
  become: true
  gather_facts: true
  roles:
    - astra.postgresql.postgresql
    - astra.rubackup.rubackup_server

  vars_files:
    - variables.yml

# MEDIA SERVER
- hosts: media-server.example.com
  become: true # may be not necessary
  roles:
    - astra.rubackup.rubackup_media_server
  vars_files:
    - variables.yml
```

11. Запустите выполнение playbook.

15.3 Приложение 3. Облачные сервисы

15.3.1 ПК СВ «Брест»

Astra Automation предоставляет множество коллекций и модулей для автоматизации установки и настройки продуктов ПАО Группа Астра в ПК СВ «Брест»[230]. Перечень ПО, которое можно установить в этом комплексе средств виртуализации, представлен в разделе *Приложение 2. Устанавливаемые продукты*.

Ресурсная модель

В рамках Astra Automation чаще всего используются следующие ресурсы ПК СВ «Брест»:

- виртуальные сети;
- образы дисков;
- шаблоны виртуальных машин;
- экземпляры виртуальных машин.

Все создаваемые ресурсы размещаются внутри кластера ПК СВ «Брест».

Способы управления

В Astra Automation используются следующие способы управления ресурсами ПК СВ «Брест»:

- Веб-интерфейс.

  Этот способ удобен для получения информации о существующих ресурсах ПК СВ «Брест» и управления ими вручную.

- CLI.

  Этот способ удобен для создания сценариев автоматизации, однако, требует доступа к фронтальной машине от имени учетной записи администратора в сессии с высоким уровнем целостности.

Необходимые привилегии

Для использования ПК СВ «Брест» при работе с Astra Automation необходимо иметь привилегии, позволяющие управлять следующими объектами:

- Виртуальные сети.
- Образы дисков.
- Шаблоны виртуальных машин.
- Экземпляры виртуальных машин.

[230] https://astra.ru/software-services/application-software-astra-group/brest/

Необходимые ресурсы

Для использования Astra Automation совместно с ПК СВ «Брест» необходимы:

- URI точки доступа (endpoint) к API;
- образы Astra Linux Special Edition;
- шаблоны ВМ;
- идентификатор виртуальной сети;
- идентификатор группы ВМ;
- токен входа;
- ключи SSH для доступа к стендам.

Пошаговые инструкции по получению данных и созданию необходимых ресурсов приведены далее.

Настройка ПК СВ «Брест»

Для работы с ресурсами ПК СВ «Брест» необходимо подготовить следующие данные:

1. URI точки доступа к API.
2. Образы Astra Linux Special Edition.
3. Шаблоны ВМ.
4. Идентификатор виртуальной сети.
5. Токен входа.
6. Ключи SSH для доступа к создаваемым стендам.

Все описанные ниже действия следует выполнять в веб-интерфейсе ПК СВ «Брест».

Предполагается, что комплекс средств виртуализации настроен согласно инструкций:

- РДЦП.10001-02 95 01-1 Программный комплекс «Средства виртуализации «БРЕСТ». Руководство администратора. Часть 1»
- РДЦП.10001-02 95 01-2 Программный комплекс «Средства виртуализации «БРЕСТ». Руководство администратора. Часть 2»

Определение точки доступа к API

URI точки доступа к API формируется следующим образом:

```
https://<brest>:8443/RPC2/
```

где <brest> – FQDN или IP-адрес фронтальной машины ПК СВ «Брест».

Загрузка установочных образов в хранилище ПК СВ «Брест»

Если у вас нет подготовленных образов Astra Linux Special Edition, выполните следующие действия для загрузки в ПК СВ «Брест» установочных образов Astra Linux Special Edition:

1. Из Личного кабинета[231] загрузите на фронтальную машину установочные образы Astra Linux Special Edition в формате ISO.

   Примечание: Образы доступны для загрузки при наличии действующей лицензии на продукт.

[231] https://lk-new.astralinux.ru

2. На панели навигации ПК СВ «Брест» выберите *Хранилище* ‣ *Образы*.
3. Нажмите кнопку + и в выпадающем меню выберите **Создать**.
4. Заполните форму **Образ**:
   - **Название**: произвольное название образа.

     Допускается использование букв латинского алфавита, цифр и дефиса.
   - **Тип**: **CD-ROM только для чтения**.
   - **Расположение образа**:
     a. Выберите значение **Закачать**.
     b. Нажмите кнопку *Browse* и укажите путь к файлу образа.
5. Нажмите кнопку *Создать*.
6. Дождитесь загрузки образа в хранилище. Загрузка считается выполненной, когда статус образа принимает значение ГОТОВО.

Создание образа диска

Образ диска необходим для создания шаблона ВМ. Чтобы создать образ диска, выполните следующие действия:

1. На панели навигации выберите *Хранилище* ‣ *Образы*.
2. Нажмите кнопку + и в выпадающем меню выберите **Создать**.
3. Заполните форму **Образ**:
   a. **Название**: произвольное название образа.

      Допускается использование букв латинского алфавита, цифр и дефиса.
   b. **Тип**: **Общий блок данных хранилища**.
   c. **Этот образ является постоянным**: **Да**.
   d. **Расположение образа**: **Пустой образ диска**.
   e. **Размер**: размер диска по умолчанию.

      Рекомендуемое значение – не менее 20 ГБ.
4. Нажмите кнопку *Создать*.

Дождитесь создания образа. Создание считается завершенным, когда статус образа принимает значение ГОТОВО.

Создание шаблона ВМ

Чтобы создать шаблон ВМ, выполните следующие действия:

1. На панели навигации выберите *Шаблоны* ‣ *ВМ*.
2. Нажмите кнопку + и в выпадающем меню выберите **Создать**.
3. Заполните форму **Создать шаблон ВМ**:
   a. Заполните вкладку **Общие**:
      - **Название**: произвольное название шаблона ВМ.

        Допускается использование букв латинского алфавита, цифр и дефиса.
      - **Гипервизор**: KVM.
      - **Память**: объем памяти, доступный создаваемым ВМ.

        Рекомендуемое значение – не менее 2 ГБ.

- **Физический CPU**: укажите значение не ниже `0.5`.
- **Виртуальный CPU**: количество ядер, доступных ВМ.

  Рекомендуемое значение – не менее 2.

b. Заполните вкладку **Хранилище**:

- **ДИСК0**:
  1. Выберите значение **Образ**.
  2. Из списка образов выберите созданный ранее образ диска типа «Блок данных».
- В списке дисков нажмите кнопку +.
- **ДИСК1**:
  1. Выберите значение **Образ**.
  2. Из списка образов выберите образ установочного диска Astra Linux Special Edition.

4. Нажмите кнопку *Создать*.

Определение идентификатора виртуальной сети

По умолчанию при развертывании ПК СВ «Брест» создается виртуальная сеть `virtnetwork` с идентификатором `0`. Если для Astra Automation планируется использовать другую виртуальную сеть, для получения ее идентификатора на панели навигации выберите *Сеть* ▸ *Вирт. сети*. Идентификаторы виртуальных сетей указаны в колонке **ID**.

Создание токена входа

Чтобы сформировать токен входа, выполните следующие действия:

1. Нажмите кнопку с именем активного пользователя и в выпадающем меню выберите **Настройки**.
2. Выберите вкладку **Аутентификация**.
3. Нажмите кнопку *Управление токенами входа*.
4. В форме **Токен входа** заполните поле **Истечение, в секундах**, в котором укажите срок действия токена.
5. Нажмите кнопку *Получить новый токен*.
6. Значение из колонки **Токен** сохраните в надежном месте.
7. Закройте форму **Токен входа**.

Настройка SSH

ПК СВ «Брест» позволяет подключаться к созданным ВМ по протоколам RDP и VNC, однако, для использования Astra Automation требуется доступ к ВМ по SSH. Чтобы настроить его, выполните следующие действия:

1. Нажмите кнопку с именем активного пользователя и в выпадающем меню выберите **Настройки**.
2. Выберите вкладку **Аутентификация**.
3. В разделе **Публичный SSH ключ** нажмите кнопку редактирования.
4. В открывшемся поле введите содержимое публичного ключа, *используемого для доступа к стендам*.

5. Нажмите кнопку редактирования еще раз.

15.3.2 Cloud.ru

Astra Automation позволяет развертывать продукты ПАО Группа Астра в Cloud.ru. Перечень ПО, поддерживающего развертывание в Cloud.ru, представлен в документе *Приложение 2. Устанавливаемые продукты*.

Ресурсная модель

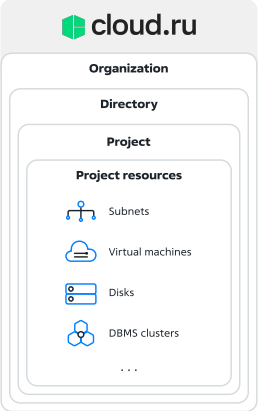
В рамках Astra Automation чаще всего используются следующие ресурсы Cloud.ru:

- виртуальные машины;
- сети;
- подсети.

Все создаваемые ресурсы размещаются внутри проектов. Каждый проект размещается в определенном каталоге. Каждый каталог принадлежит определенной организации.

Примечание: При необходимости можно создать дополнительные проекты.

Упрощенная структура Cloud.ru показана на схеме:



Способы управления

В Astra Automation для управления ресурсами Cloud.ru следует использовать личный кабинет. Он доступен по адресу https://console.cloud.ru/ и предоставляет веб-интерфейс для управления ресурсами облака. Этот способ удобен для получения информации о существующих ресурсах Cloud.ru и управления ими вручную.

Необходимые привилегии

Для использования Cloud.ru при работе с Astra Automation необходимо иметь роли, назначенные на организацию или проект и обеспечивающие выполнение следующих операций:

- Создание сервисных аккаунтов и назначение им ролей.
- Создание ключей доступа.

  Ключи доступа используются для управления ресурсами Cloud.ru от имени сервисного аккаунта.
- Управление сетями и подсетями.
- Управление виртуальными машинами.

- Создание бакетов в Объектном хранилище, загрузка и выгрузка объектов из бакетов.
- Импорт ключей SSH.

Предупреждение: Если в организации или проекте не существует сервисный аккаунт с необходимыми ролями, для его создания и назначения ему ролей потребуется учетная запись администратора или пользователя, входящего в группу `admin`.

Необходимые ресурсы

Для использования Astra Automation совместно с Cloud.ru необходимы:

- Права на управление ресурсами Cloud.ru от имени любой из указанных учетных записей:
  - собственная учетная запись;
  - учетная запись организации;
  - сервисный аккаунт[232].
- Организация и ее идентификатор.
- Каталог и его идентификатор.
- Проект и его идентификатор.

  В этом проекте размещаются ВМ, используемые для развертывания необходимого ПО.

Пошаговые инструкции по получению данных и созданию необходимых ресурсов приведены далее.

Настройка сервисного аккаунта

Для работы с ресурсами Cloud.ru можно использовать учетную запись обычного пользователя, однако, для средств автоматизации лучше подходит сервисный аккаунт.

Для подготовки сервисного аккаунта Cloud.ru к работе с Astra Automation выполните следующие действия:

1. Убедитесь, что сервисный аккаунт существует и ему назначены необходимые роли.
2. Назначьте созданному сервисному аккаунту необходимые роли.
3. Создайте ключ доступа к ресурсам проекта.

Проверка наличия необходимых ролей

Для управления ресурсами сервисному аккаунту должны быть назначены роли, дающие необходимые привилегии.

Примечание: Роль `Администратор` дает привилегии на создание, изменение и удаление любых типов ресурсов в проекте, однако, в большинстве случае столь широкие полномочия избыточны. Для более тонкой настройки ознакомьтесь со списком стандартных ролей[233], предоставляемых Cloud.ru, и назначьте сервисному аккаунту только те роли, которые действительно необходимы.

Если сервисный аккаунт уже существует, для проверки наличия у него необходимых ролей используйте Личный кабинет пользователя Cloud.ru.

[232] https://cloud.ru/docs/console/ug/topics/guides__service_accounts.html
[233] https://cloud.ru/docs/iam/ug/topics/overview__roles-and-policies-list.html

Создание сервисного аккаунта

Если сервисный аккаунт не существует, создайте его согласно инструкции[234].

Назначение ролей сервисному аккаунту

Чтобы назначить сервисному аккаунту роль на проект, воспользуйтесь инструкцией[235].

Создание ключа доступа

Чтобы создать для сервисного аккаунта ключ доступа, воспользуйтесь инструкцией[236].

15.3.3 Yandex Cloud

Astra Automation позволяет развертывать продукты ПАО Группа Астра в Yandex Cloud. Перечень ПО, поддерживающего развертывание в Yandex Cloud, представлен в разделе *Приложение 2. Устанавливаемые продукты*.

Ресурсная модель

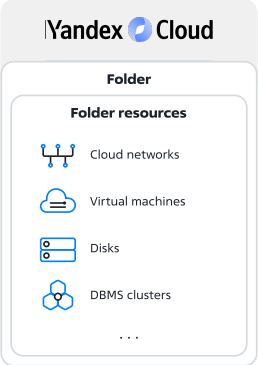
В рамках Astra Automation чаще всего используются следующие ресурсы Yandex Cloud:

- облачные сети и подсети[237];
- виртуальные машины Yandex Compute Cloud[238];
- бакеты Yandex Object Storage[239].

Все создаваемые ресурсы размещаются внутри облачных каталогов. Каждый облачный каталог является частью облака. При создании облака в нем автоматически создается облачный каталог с именем `default`.

Примечание: При необходимости можно создать дополнительные облачные каталоги.

Упрощенная структура Yandex Cloud показана на схеме:



[234] https://cloud.ru/docs/console/ug/topics/guides__service_accounts_create.html
[235] https://cloud.ru/docs/console/ug/topics/guides__employees_edit.html
[236] https://cloud.ru/docs/console/ug/topics/guides__service_accounts_key.html
[237] https://yandex.cloud/ru/docs/vpc/concepts/network/
[238] https://yandex.cloud/ru/docs/compute/concepts/vm/
[239] https://yandex.cloud/ru/docs/storage/

Способы управления

В Astra Automation используются следующие способы управления ресурсами Yandex Cloud:

- Консоль управления.

  Консоль управления доступна по адресу https://console.yandex.cloud/ru/ и предоставляет веб-интерфейс для управления ресурсами облака. Этот способ удобен для получения информации о существующих ресурсах Yandex Cloud и управления ими вручную.

- Клиент командной строки.

  YC CLI позволяет управлять облачными ресурсами с помощью утилиты командной строки. Этот способ удобен при использовании сценариев оболочки командной строки.

  Все дальнейшие инструкции предполагают использование YC CLI для выполнения необходимых настроек.

Необходимые привилегии

Для использования Yandex Cloud при работе с Astra Automation необходимо иметь роль editor[240] на облачный каталог, либо роли, обеспечивающие выполнение следующих операций:

- Создание сервисных аккаунтов и назначение им ролей.

  Эта привилегия необходима, если сервисные аккаунты с нужными ролями на каталог не существуют.

- Создание авторизованных и AWS-совместимых статических ключей доступа.

  Эта привилегия необходима для настройки доступа сервисного аккаунта к ресурсам Yandex Cloud и бакетам Yandex Object Storage.

- Управление облачными сетями и подсетями.
- Управление виртуальными машинами Yandex Compute Cloud.
- Создание бакетов Yandex Object Storage, загрузка и выгрузка объектов из них.

Предупреждение: Если в облачном каталоге не существует сервисный аккаунт с необходимыми ролями, для его создания и назначения ролей потребуется роль admin[241].

Необходимые ресурсы

Для использования Astra Automation совместно с Yandex Cloud необходимы:

- Права на управление ресурсами Yandex Cloud от имени любой из указанных учетных записей:
  - собственная учетная запись;
  - учетная запись организации;
  - сервисный аккаунт[242].
- Облако и его идентификатор.
- Облачный каталог и его идентификатор. В этом каталоге размещаются ВМ, используемые для развертывания необходимого ПО.

[240] https://yandex.cloud/ru/docs/iam/concepts/access-control/roles#editor
[241] https://yandex.cloud/ru/docs/iam/concepts/access-control/roles#admin
[242] https://yandex.cloud/ru/docs/iam/concepts/users/service-accounts

- Сервисный аккаунт Yandex Cloud. Этот аккаунт позволяет средствам Astra Automation иметь доступ к Yandex Cloud.
- Файл формата JSON, содержащий авторизованные ключи. Используется при управлении облачными ресурсами с использованием сервисного аккаунта.

Пошаговые инструкции по получению данных и созданию необходимых ресурсов приведены далее.

Настройка интерфейса командной строки (YC CLI)

Платформа Yandex Cloud предоставляет возможность управления облачными ресурсами с помощью *YC CLI*. Основу клиентской части составляет утилита yc. Все дальнейшие инструкции по использованию Yandex Cloud совместно с Astra Automation предполагают использование этого инструмента для работы с облаком.

Установка и настройка интерфейса командной строки YC CLI

Детальная информация о начале работы в командной строке приведена в документации Yandex Cloud[243].

Чтобы начать пользоваться YC CLI, выполните следующие действия:

1. Установите клиентскую часть:

   ```
   curl -sSL https://storage.yandexcloud.net/yandexcloud-yc/install.sh | bash
   ```

   Эта команда выполняет следующие действия:

   - создает каталог `yandex-cloud/` в домашнем каталоге пользователя;
   - дополняет файл инициализации `.bashrc` вызовом скриптов настройки клиента – `~/yandex-cloud/path.bash.inc` и `~/yandex-cloud/completion.bash.inc`.

2. Перезапустите терминал или выполните команду:

   ```
   source ~/.bashrc
   ```

3. Создайте профиль пользователя для работы с облачным каталогом.

   Примечание: Профиль определяет набор параметров окружения, в котором пользователь работает в каждый конкретный момент.

   При использовании федеративного метода идентификации на базе SAML выполните команду:

   ```
   yc init --federation-id=<federation_ID>
   ```

   где <federation_ID> – идентификатор федерации[244] для вашей организации или подразделения.

   Необходимо выбрать – использовать существующий каталог или создать новый.

4. (Опционально) Настройте зону доступности по умолчанию.
5. Проверьте полученные настройки с помощью команды:

   ```
   yc config list
   ```

   В терминал выводятся идентификаторы федерации, облака и каталога, используемых в текущий момент.

[243] https://yandex.cloud/ru/docs/cli/
[244] https://yandex.cloud/ru/docs/organization/concepts/add-federation#saml-authentication

6. Проверьте список профилей:

```
yc config profile list
```

В терминал выводится список существующих профилей YC CLI. Активный профиль будет отмечен меткой ACTIVE.

Управление профилями

Дополнительные профили YC CLI могут понадобиться, например:

- для работы с разными облаками и каталогами Yandex Cloud;
- для управления ресурсами одного и того же каталога от имени разных пользователей, в том числе сервисных аккаунтов.

Чтобы создать дополнительный профиль YC CLI, выполните следующие действия:

1. Получите идентификаторы нужного облака и облачного каталога любым удобным способом. Чтобы получить сведения об облаке и каталоге, к которым привязан активный профиль YC CLI, выполните команду:

```
yc config profile get <profile>
```

В терминал выводится следующая информация о профиле:

- `cloud-id` – идентификатор облака;
- `folder-id` – идентификатор облачного каталога;
- `compute-default-zone` – зона доступности по умолчанию (если была указана при настройке профиля).

2. Выполните команду создания нового профиля:

```
yc config profile create <profile_name>
```

Профиль автоматически активируется после создания.

3. Привяжите профиль к нужному облаку и каталогу:

```
yc config set cloud-id <cloud_ID>
yc config set folder-id <folder_ID>
```

где <cloud_ID> и <folder_ID> – полученные ранее идентификаторы облака и облачного каталога соответственно.

Совет: Полный перечень команд YC CLI приведен в документации Yandex Cloud[245].

Примеры использования

Получение списка доступных профилей:

```
yc config profile list
```

Переключение на другой профиль:

```
yc config profile activate <profile>
```

Назначение зоны доступности по умолчанию:

[245] https://yandex.cloud/ru/docs/cli/cli-ref/

```
yc config set compute-default-zone <zone>
```

где <zone> – название зоны доступности, например:

```
yc config set compute-default-zone ru-central1-a
```

Подробности о зонах доступности см. в документации Yandex Cloud[246].

Настройка сервисного аккаунта

Для работы с ресурсами Yandex Cloud можно использовать учетную запись обычного пользователя, однако, для средств автоматизации лучше подходит сервисный аккаунт.

Для подготовки сервисного аккаунта Yandex Cloud к работе с Astra Automation выполните следующие действия:

1. Если клиент командной строки Yandex Cloud не настроен, следуйте инструкциям из секции *Установка и настройка интерфейса командной строки YC CLI*.
2. Убедитесь, что сервисный аккаунт существует и ему назначены необходимые роли.
3. Создайте ключ доступа к ресурсам каталога.

Проверка наличия необходимых ролей

Для управления ресурсами сервисному аккаунту должны быть назначены роли, дающие необходимые привилегии.

Примечание: Роль `editor` дает привилегии на создание, изменение и удаление любых типов ресурсов в каталоге, однако, в большинстве случаев столь широкие полномочия избыточны. Для более тонкой настройки ознакомьтесь со списком всех ролей[247], предоставляемых Yandex Cloud, и назначьте сервисному аккаунту только те роли, которые действительно необходимы.

Если сервисный аккаунт уже существует, убедитесь, что ему назначены роли, позволяющие управлять необходимыми типами ресурсов:

```
yc resource-manager folder list-access-bindings <folder_name|folder_ID> | grep <sa_ID>
```

В терминал выводится таблица, отображающая список ролей, назначенных сервисному аккаунту с указанным идентификатором <sa_ID>.

```
+------------------+----------------+----------------------+
|     ROLE ID      |  SUBJECT TYPE  |      SUBJECT ID      |
+------------------+----------------+----------------------+
| storage.uploader | serviceAccount | a..................1 |
| editor           | serviceAccount | a..................1 |
+------------------+----------------+----------------------+
```

[246] https://yandex.cloud/ru/docs/overview/concepts/geo-scope
[247] https://yandex.cloud/ru/docs/iam/concepts/access-control/roles

Создание сервисного аккаунта

Если сервисный аккаунт не существует, создайте его:

```
yc iam service-account create \
   --name=<sa_name> \
   --description="<description>"
```

где

- <sa_name> – имя сервисного аккаунта. Должно удовлетворять следующим требованиям:
  - длина от 3 до 63 символов;
  - может содержать только строчные буквы латинского алфавита, цифры и дефисы;
  - первый символ – буква;
  - последний символ – не дефис;
  - имя должно быть уникальным в рамках облака.
- <description> – опциональное описание сервисного аккаунта.

Подробности о создании сервисных аккаунтов см. в документации Yandex Cloud[248].

Назначение ролей сервисному аккаунту

Чтобы назначить сервисному аккаунту роль на каталог, выполните команду:

```
yc resource-manager folder add-access-binding \
   --name <folder_name> \
   --role <role_name> \
   --service-account-name <sa_name>
```

где

- <folder_name> – имя облачного каталога;
- <role_name> – имя роли;
- <sa_name> – имя сервисного аккаунта.

Подробности о назначении ролей сервисным аккаунтам см. в документации Yandex Cloud[249].

Создание пары авторизованных ключей доступа

Чтобы создать пару авторизованных ключей доступа к облачному каталогу и сохранить ее в файл формата JSON, выполните команду:

```
yc iam key create \
   --service-account-name <sa_name> \
   --output key.json
```

где

- <sa_name> – имя сервисного аккаунта;
- `key.json` – имя файла для сохранения пары ключей.

При успешном выполнении команды в терминал выводится информация о созданной паре ключей, например:

[248] https://yandex.cloud/ru/docs/iam/quickstart-sa#create-sa
[249] https://yandex.cloud/ru/docs/iam/operations/sa/assign-role-for-sa

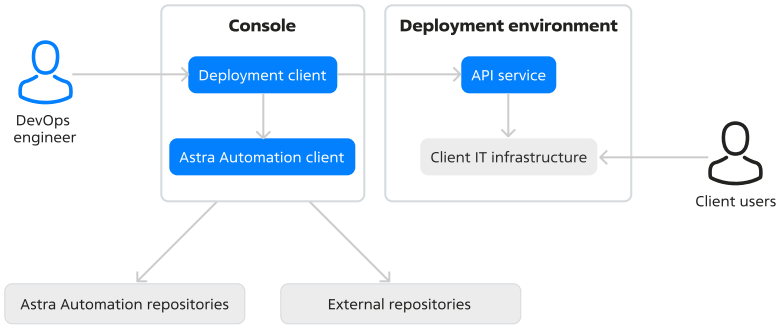
```
id: ajonetwothreefoursp63
service_account_id: aj**************o3
created_at: "2023-03-21T08:13:19.002945823Z"
key_algorithm: RSA_2048
```

Пример содержимого файла key.json (ключи представлены в сокращенном виде):

```
{
   "id": "ajonetwothreefoursp63",
   "service_account_id": "aj**************o3",
   "created_at": "2023-03-21T08:13:19.002945823Z",
   "key_algorithm": "RSA_2048",
   "public_key": "-----BEGIN PUBLIC KEY-----\nMIIBI...DAQAB\n-----END PUBLIC KEY-----\n",
   "private_key": "-----BEGIN...END PRIVATE KEY-----\n"
}
```

Подробности о создании авторизованных ключей доступа см. в документации Yandex Cloud[250].

Типовая схема развертывания облачной инфраструктуры:



15.4 Приложение 4. Инструментарий

Для реализации своих функций Astra Automation использует различные методы, инструменты и системы, описания которых приводятся в следующих разделах.

15.4.1 Ansible

Astra Automation использует Ansible как основной инструмент для установки и настройки систем различной сложности.

Назначение

Ansible[251] – это система управления развертыванием и настройкой программного обеспечения. Система выполняет заданные директивы на указанных серверах, подключаясь к ним по протоколу SSH. Для этого необходимы только ключи SSH. Установка агентов на управляемые узлы не требуется.

Базовая функциональность Ansible обеспечивается большим количеством встроенных модулей. При этом возможности Ansible можно расширять с помощью следующих компонентов:

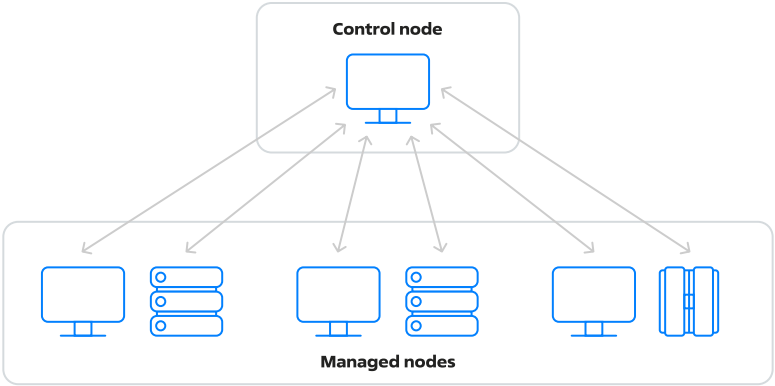
- расширения (plugins);
- модули (modules);
- роли (roles);

[250] https://yandex.cloud/ru/docs/iam/operations/authorized-key/create
[251] https://docs.ansible.com/

- коллекции (collections).

Структура управления

При использовании Ansible управляющий узел подключается к управляемым узлам и запускает на них команды настройки.



Управляющий узел

Управляющим узлом (control node) называется компьютер, на который устанавливают компоненты Ansible и который используют для запуска заданий по настройке управляемых узлов.

Подробности об управляющем узле см. в документации Ansible[252].

Управляемые узлы

Управляемым узлом (managed node) называется компьютер, настраиваемый с помощью Ansible.

Подробности об управляемых узлах см. в документации Ansible[253].

Инвентарь

Инвентарь – это файл формата `INI` или `YAML`, в котором хранятся данные об управляемых узлах. Подробности о файлах инвентаря см. в документации Ansible[254].

В самом простом случае достаточно указать для каждого управляемого узла только его IP-адрес:

```
192.168.56.11
192.168.56.12
192.168.56.13
```

Также допускается использовать доменные имена, например:

```
dc01.example.com
dc02.example.com
dc03.example.com
```

В этом случае разрешение доменного имени каждого узла в IP-адрес происходит путем обращения к системной службе разрешения имен. В Astra Linux настройки этой службы

[252] https://docs.ansible.com/ansible/latest/getting_started/basic_concepts.html#control-node
[253] https://docs.ansible.com/ansible/latest/getting_started/basic_concepts.html#managed-nodes
[254] https://docs.ansible.com/ansible/latest/getting_started/basic_concepts.html#inventory

хранятся в файле /etc/nsswitch.conf. По умолчанию в нем указан следующий порядок разрешения имен:

1. Проверка записей, хранящихся в файле /etc/hosts.
2. Обращение к серверу DNS, указанному в настройках сети.

Чтобы сделать инвентарь не зависящим от службы разрешения имен, для каждого узла всегда явно указывайте его IP-адрес в поле ansible_host, например:

```
dc01.example.com   ansible_host=192.168.56.11
dc02.example.com   ansible_host=192.168.56.12
dc03.example.com   ansible_host=192.168.56.13
```

Использование переменных при заполнении инвентаря

Использование переменных Ansible позволяет упростить заполнение инвентаря.

Пусть имеется инвентарь следующего вида:

```
[all]
node1  ansible_host=192.168.56.1  ansible_ssh_private_key=./node1-ssh-key
node2  ansible_host=192.168.56.2  ansible_ssh_private_key=./node2-ssh-key
node3  ansible_host=192.168.56.3  ansible_ssh_private_key=./node3-ssh-key
node4  ansible_host=192.168.56.4  ansible_ssh_private_key=./node4-ssh-key
node5  ansible_host=192.168.56.5  ansible_ssh_private_key=./node5-ssh-key
```

Поскольку имена файлов приватных ключей SSH содержат названия соответствующих узлов, эту запись можно упростить следующим образом:

```
[all]
node1  ansible_host=192.168.56.1
node2  ansible_host=192.168.56.2
node3  ansible_host=192.168.56.3
node4  ansible_host=192.168.56.4
node5  ansible_host=192.168.56.5

[all:vars]
ansible_ssh_private_key=./{{ inventory_hostname }}-ssh-key
```

Иерархия компонентов playbook

Иерархия исполняемых компонентов в скриптах Ansible:

- *Playbook* (набор сценариев) состоит из plays (сценариев), использующих инвентарный список, который указывается при запуске playbook с помощью параметра -i или в конфигурационном файле ansible.cfg. Утилита ansible-playbook начинает выполнение playbook.
- *Play* (сценарий) состоит из множества задач, выполняемых для указанных узлов и групп узлов из определенного инвентарного списка.
- *Task* (задача) использует один из модулей Ansible для выполнения отдельной операции по настройке управляемого узла.
- *Module* (модуль) реализует требуемую функциональность. Например, существуют модули для копирования файлов, управления пакетами и службами и т. д. Код модулей выполняется на управляемых узлах.
- *Plugin* (расширение) расширяет функциональность Ansible. В отличие от модулей, код расширений выполняется на управляющем узле.

Задача

Задача (task) – это минимальный шаг по настройке управляемого узла.

Подробности о задачах см. в документации Ansible[255].

Сценарий

Сценарий (play) – это одна или несколько задач, выполняемых на указанных управляемых узлах.

Подробности о сценариях см. в документации Ansible[256].

Playbook

Playbook – это файл формата YAML, содержащий перечень сценариев по настройке управляемых узлов.

Подробности о playbook см. в документации Ansible[257].

Роль

Роли используют для группирования задач при развертывании и настройке сложных систем. Роль объединяет несколько задач и может быть использована в различных playbook.

Пример создания структуры каталогов для роли:

```
ansible-galaxy init httpd
```

где `httpd` – название каталога, который будет реализовывать требуемую роль.

Файлы playbook содержат список подключаемых ролей как в следующем примере:

```
- hosts: all
  user: root
  port: 22
  gather_facts: True
  roles:
    - { role: selinux, tags: selinux }
    - { role: httpd, tags: httpd }
    - { role: resolver, tags: resolver }
```

Если необходимо выполнить какую-то одну роль, то можно выполнить этот playbook, указав конкретную роль с помощью аргумента `-t`, например `-t selinux`.

Подробности о ролях см. в документации Ansible[258].

[255] https://docs.ansible.com/ansible/latest/getting_started/basic_concepts.html#tasks
[256] https://docs.ansible.com/ansible/latest/getting_started/basic_concepts.html#plays
[257] https://docs.ansible.com/ansible/latest/getting_started/basic_concepts.html#playbooks
[258] https://docs.ansible.com/ansible/latest/getting_started/basic_concepts.html#roles

Коллекция

Коллекцией (collection) называется распространяемый как единое целое набор компонентов, расширяющих возможности Ansible.

Подробности о коллекциях см. в документации Ansible[259].

Особенности коллекций, доступных на портале Automation Hub, описаны в разделе *Automation Hub*.

Факты

Факты (facts) – это набор сведений об управляемом узле, например, архитектура и версия установленной ОС, версия BIOS, наличие в системе определенных устройств и тому подобное.

Для просмотра всех доступных фактов об узле необходимо выполнить команду:

```
- name: Show all available facts
  ansible_builtin.debug:
    var: ansible_facts
```

Пример списка фактов об управляемом узле


```
{
    "ansible_facts": {
        "all_ipv4_addresses": [
            "10.0.2.15",
            "192.168.56.101"
        ],
        "all_ipv6_addresses": [
            "fe80::a00:27ff:fe0d:82ad",
            "fe80::a00:27ff:feac:2f5c"
        ],
        "ansible_local": {},
        "apparmor": {
            "status": "disabled"
        },
        "architecture": "x86_64",
        "bios_date": "12/01/2006",
        "bios_vendor": "innotek GmbH",
        "bios_version": "VirtualBox",
        "board_asset_tag": "NA",
        "board_name": "VirtualBox",
        "board_serial": "NA",
        "board_vendor": "Oracle Corporation",
        "board_version": "1.2",
        "chassis_asset_tag": "NA",
        "chassis_serial": "NA",
        "chassis_vendor": "Oracle Corporation",
        "chassis_version": "NA",
        "cmdline": {
            "BOOT_IMAGE": "/boot/vmlinuz-6.1.50-1-generic",
            "net.ifnames": "0",
            "parsec.max_ilev": "63",
            "quiet": true,
            "ro": true,
            "root": "UUID=17007b23-97d0-46fd-90b6-2c359be9c2b0"
        },
        "date_time": {
            "date": "2024-01-29",
```


(continues on next page)

[259] https://docs.ansible.com/ansible/latest/getting_started/basic_concepts.html#collections

(продолжение с предыдущей страницы)


```
            "day": "29",
            "epoch": "1706532112",
            "epoch_int": "1706532112",
            "hour": "15",
            "iso8601": "2024-01-29T12:41:52Z",
            "iso8601_basic": "20240129T154152867152",
            "iso8601_basic_short": "20240129T154152",
            "iso8601_micro": "2024-01-29T12:41:52.867152Z",
            "minute": "41",
            "month": "01",
            "second": "52",
            "time": "15:41:52",
            "tz": "MSK",
            "tz_dst": "MSK",
            "tz_offset": "+0300",
            "weekday": "Понедельник",
            "weekday_number": "1",
            "weeknumber": "05",
            "year": "2024"
        },
        "default_ipv4": {
            "address": "10.0.2.15",
            "alias": "eth0",
            "broadcast": "10.0.2.255",
            "gateway": "10.0.2.2",
            "interface": "eth0",
            "macaddress": "08:00:27:0d:82:ad",
            "mtu": 1500,
            "netmask": "255.255.255.0",
            "network": "10.0.2.0",
            "prefix": "24",
            "type": "ether"
        },
        "default_ipv6": {},
        "device_links": {
            "ids": {
                "sda": [
                    "ata-VBOX_HARDDISK_VBcf71a1d6-26264a12"
                ],
                "sda1": [
                    "ata-VBOX_HARDDISK_VBcf71a1d6-26264a12-part1"
                ]
            },
            "labels": {},
            "masters": {},
            "uuids": {
                "sda1": [
                    "17007b23-97d0-46fd-90b6-2c359be9c2b0"
                ]
            }
        },
        "devices": {
            "loop0": {
                "holders": [],
                "host": "",
                "links": {
                    "ids": [],
                    "labels": [],
                    "masters": [],
                    "uuids": []
                },
                "model": null,
                "partitions": {},
                "removable": "0",
```


(continues on next page)

(продолжение с предыдущей страницы)


```
                "rotational": "1",
                "sas_address": null,
                "sas_device_handle": null,
                "scheduler_mode": "none",
                "sectors": "0",
                "sectorsize": "512",
                "size": "0.00 Bytes",
                "support_discard": "0",
                "vendor": null,
                "virtual": 1
            },
            "loop1": {
                "holders": [],
                "host": "",
                "links": {
                    "ids": [],
                    "labels": [],
                    "masters": [],
                    "uuids": []
                },
                "model": null,
                "partitions": {},
                "removable": "0",
                "rotational": "1",
                "sas_address": null,
                "sas_device_handle": null,
                "scheduler_mode": "none",
                "sectors": "0",
                "sectorsize": "512",
                "size": "0.00 Bytes",
                "support_discard": "0",
                "vendor": null,
                "virtual": 1
            },
            "loop2": {
                "holders": [],
                "host": "",
                "links": {
                    "ids": [],
                    "labels": [],
                    "masters": [],
                    "uuids": []
                },
                "model": null,
                "partitions": {},
                "removable": "0",
                "rotational": "1",
                "sas_address": null,
                "sas_device_handle": null,
                "scheduler_mode": "none",
                "sectors": "0",
                "sectorsize": "512",
                "size": "0.00 Bytes",
                "support_discard": "0",
                "vendor": null,
                "virtual": 1
            },
            "loop3": {
                "holders": [],
                "host": "",
                "links": {
                    "ids": [],
                    "labels": [],
                    "masters": [],
```


(continues on next page)

(продолжение с предыдущей страницы)


```
                "uuids": []
            },
            "model": null,
            "partitions": {},
            "removable": "0",
            "rotational": "1",
            "sas_address": null,
            "sas_device_handle": null,
            "scheduler_mode": "none",
            "sectors": "0",
            "sectorsize": "512",
            "size": "0.00 Bytes",
            "support_discard": "0",
            "vendor": null,
            "virtual": 1
        },
        "loop4": {
            "holders": [],
            "host": "",
            "links": {
                "ids": [],
                "labels": [],
                "masters": [],
                "uuids": []
            },
            "model": null,
            "partitions": {},
            "removable": "0",
            "rotational": "1",
            "sas_address": null,
            "sas_device_handle": null,
            "scheduler_mode": "none",
            "sectors": "0",
            "sectorsize": "512",
            "size": "0.00 Bytes",
            "support_discard": "0",
            "vendor": null,
            "virtual": 1
        },
        "loop5": {
            "holders": [],
            "host": "",
            "links": {
                "ids": [],
                "labels": [],
                "masters": [],
                "uuids": []
            },
            "model": null,
            "partitions": {},
            "removable": "0",
            "rotational": "1",
            "sas_address": null,
            "sas_device_handle": null,
            "scheduler_mode": "none",
            "sectors": "0",
            "sectorsize": "512",
            "size": "0.00 Bytes",
            "support_discard": "0",
            "vendor": null,
            "virtual": 1
        },
        "loop6": {
            "holders": [],
```


(continues on next page)

(продолжение с предыдущей страницы)


```
                "host": "",
                "links": {
                    "ids": [],
                    "labels": [],
                    "masters": [],
                    "uuids": []
                },
                "model": null,
                "partitions": {},
                "removable": "0",
                "rotational": "1",
                "sas_address": null,
                "sas_device_handle": null,
                "scheduler_mode": "none",
                "sectors": "0",
                "sectorsize": "512",
                "size": "0.00 Bytes",
                "support_discard": "0",
                "vendor": null,
                "virtual": 1
            },
            "loop7": {
                "holders": [],
                "host": "",
                "links": {
                    "ids": [],
                    "labels": [],
                    "masters": [],
                    "uuids": []
                },
                "model": null,
                "partitions": {},
                "removable": "0",
                "rotational": "1",
                "sas_address": null,
                "sas_device_handle": null,
                "scheduler_mode": "none",
                "sectors": "0",
                "sectorsize": "512",
                "size": "0.00 Bytes",
                "support_discard": "0",
                "vendor": null,
                "virtual": 1
            },
            "sda": {
                "holders": [],
                "host": "SATA controller: Intel Corporation 82801HM/HEM (ICH8M/ICH8M-E)␣
↪SATA Controller [AHCI mode] (rev 02)",
                "links": {
                    "ids": [
                        "ata-VBOX_HARDDISK_VBcf71a1d6-26264a12"
                    ],
                    "labels": [],
                    "masters": [],
                    "uuids": []
                },
                "model": "VBOX HARDDISK",
                "partitions": {
                    "sda1": {
                        "holders": [],
                        "links": {
                            "ids": [
                                "ata-VBOX_HARDDISK_VBcf71a1d6-26264a12-part1"
                            ],
```


(continues on next page)

(продолжение с предыдущей страницы)


```
                        "labels": [],
                        "masters": [],
                        "uuids": [
                            "17007b23-97d0-46fd-90b6-2c359be9c2b0"
                        ]
                    },
                    "sectors": "61435904",
                    "sectorsize": 512,
                    "size": "29.29 GB",
                    "start": "2048",
                    "uuid": "17007b23-97d0-46fd-90b6-2c359be9c2b0"
                }
            },
            "removable": "0",
            "rotational": "1",
            "sas_address": null,
            "sas_device_handle": null,
            "scheduler_mode": "mq-deadline",
            "sectors": "61440000",
            "sectorsize": "512",
            "size": "29.30 GB",
            "support_discard": "0",
            "vendor": "ATA",
            "virtual": 1
        }
    },
    "distribution": "Astra Linux",
    "distribution_file_parsed": true,
    "distribution_file_path": "/etc/os-release",
    "distribution_file_variety": "NA",
    "distribution_major_version": "1",
    "distribution_release": "1.7_x86-64",
    "distribution_version": "1.7_x86-64",
    "dns": {
        "domain": "astralinux.ru",
        "nameservers": [
            "10.0.2.3"
        ],
        "search": [
            "astralinux.ru"
        ]
    },
    "domain": "",
    "effective_group_id": 1000,
    "effective_user_id": 1000,
    "env": {
        "HOME": "/home/vagrant",
        "LANG": "ru_RU.UTF-8",
        "LC_CTYPE": "C.UTF-8",
        "LOGNAME": "vagrant",
        "PATH": "/usr/local/bin:/usr/bin:/bin:/usr/games",
        "PWD": "/home/vagrant",
        "SHELL": "/bin/bash",
        "SHLVL": "1",
        "SSH_CLIENT": "192.168.56.11 37284 22",
        "SSH_CONNECTION": "192.168.56.11 37284 192.168.56.101 22",
        "SSH_TTY": "/dev/pts/0",
        "TERM": "xterm",
        "USER": "vagrant",
        "XDG_RUNTIME_DIR": "/run/user/1000",
        "XDG_SESSION_CLASS": "user",
        "XDG_SESSION_ID": "31",
        "XDG_SESSION_TYPE": "tty",
        "_": "/usr/bin/python3"
```


(continues on next page)

(продолжение с предыдущей страницы)


```
        },
        "eth0": {
            "active": true,
            "device": "eth0",
            "ipv4": {
                "address": "10.0.2.15",
                "broadcast": "10.0.2.255",
                "netmask": "255.255.255.0",
                "network": "10.0.2.0",
                "prefix": "24"
            },
            "ipv6": [
                {
                    "address": "fe80::a00:27ff:fe0d:82ad",
                    "prefix": "64",
                    "scope": "link"
                }
            ],
            "macaddress": "08:00:27:0d:82:ad",
            "module": "e1000",
            "mtu": 1500,
            "pciid": "0000:00:03.0",
            "promisc": false,
            "speed": 1000,
            "type": "ether"
        },
        "eth1": {
            "active": true,
            "device": "eth1",
            "ipv4": {
                "address": "192.168.56.101",
                "broadcast": "192.168.56.255",
                "netmask": "255.255.255.0",
                "network": "192.168.56.0",
                "prefix": "24"
            },
            "ipv6": [
                {
                    "address": "fe80::a00:27ff:feac:2f5c",
                    "prefix": "64",
                    "scope": "link"
                }
            ],
            "macaddress": "08:00:27:ac:2f:5c",
            "module": "e1000",
            "mtu": 1500,
            "pciid": "0000:00:08.0",
            "promisc": false,
            "speed": 1000,
            "type": "ether"
        },
        "fibre_channel_wwn": [],
        "fips": false,
        "form_factor": "Other",
        "fqdn": "node-1",
        "gather_subset": [
            "all"
        ],
        "hostname": "node-1",
        "hostnqn": "",
        "interfaces": [
            "eth0",
            "eth1",
            "lo"
```


(continues on next page)

(продолжение с предыдущей страницы)


```
        ],
        "is_chroot": false,
        "iscsi_iqn": "",
        "kernel": "6.1.50-1-generic",
        "kernel_version": "#astra2+ci6 SMP PREEMPT_DYNAMIC Fri Oct  6 14:38:42 UTC 2023",
        "lo": {
            "active": true,
            "device": "lo",
            "ipv4": {
                "address": "127.0.0.1",
                "broadcast": "",
                "netmask": "255.0.0.0",
                "network": "127.0.0.0",
                "prefix": "8"
            },
            "ipv6": [
                {
                    "address": "::1",
                    "prefix": "128",
                    "scope": "host"
                }
            ],
            "mtu": 65536,
            "promisc": false,
            "type": "loopback"
        },
        "lsb": {
            "codename": "1.7_x86-64",
            "description": "Astra Linux 1.7 x86-64",
            "id": "AstraLinux",
            "major_release": "1",
            "release": "1.7_x86-64"
        },
        "machine": "x86_64",
        "machine_id": "d383adfed45648cdbd75a02273b61314",
        "memfree_mb": 1639,
        "memory_mb": {
            "nocache": {
                "free": 1805,
                "used": 160
            },
            "real": {
                "free": 1639,
                "total": 1965,
                "used": 326
            },
            "swap": {
                "cached": 0,
                "free": 0,
                "total": 0,
                "used": 0
            }
        },
        "memtotal_mb": 1965,
        "module_setup": true,
        "mounts": [
            {
                "block_available": 6732778,
                "block_size": 4096,
                "block_total": 7514846,
                "block_used": 782068,
                "device": "/dev/sda1",
                "fstype": "ext4",
                "inode_available": 1884545,
```


(continues on next page)

(продолжение с предыдущей страницы)


```
                "inode_total": 1921360,
                "inode_used": 36815,
                "mount": "/",
                "options": "rw,relatime,errors=remount-ro",
                "size_available": 27577458688,
                "size_total": 30780809216,
                "uuid": "17007b23-97d0-46fd-90b6-2c359be9c2b0"
            }
        ],
        "nodename": "node-1",
        "os_family": "Astra Linux",
        "pkg_mgr": "apt",
        "proc_cmdline": {
            "BOOT_IMAGE": "/boot/vmlinuz-6.1.50-1-generic",
            "net.ifnames": "0",
            "parsec.max_ilev": "63",
            "quiet": true,
            "ro": true,
            "root": "UUID=17007b23-97d0-46fd-90b6-2c359be9c2b0"
        },
        "processor": [
            "0",
            "GenuineIntel",
            "Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz",
            "1",
            "GenuineIntel",
            "Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz"
        ],
        "processor_cores": 2,
        "processor_count": 1,
        "processor_nproc": 2,
        "processor_threads_per_core": 1,
        "processor_vcpus": 2,
        "product_name": "VirtualBox",
        "product_serial": "NA",
        "product_uuid": "NA",
        "product_version": "1.2",
        "python": {
            "executable": "/usr/bin/python3",
            "has_sslcontext": true,
            "type": "cpython",
            "version": {
                "major": 3,
                "micro": 3,
                "minor": 7,
                "releaselevel": "final",
                "serial": 0
            },
            "version_info": [
                3,
                7,
                3,
                "final",
                0
            ]
        },
        "python_version": "3.7.3",
        "real_group_id": 1000,
        "real_user_id": 1000,
        "selinux": {
            "status": "disabled"
        },
        "selinux_python_present": true,
        "service_mgr": "systemd",
```


(continues on next page)

(продолжение с предыдущей страницы)


```
        "ssh_host_key_ecdsa_public":
↪"AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBJe9D5fI+xDzdcOfpQ/
↪JzxooISS7wskk7RzuX1Qe2eB+yIitXmNBkzRUTvTGsokhQCGPzl8rRlPEOcWCkwA37O4=",
        "ssh_host_key_ecdsa_public_keytype": "ecdsa-sha2-nistp256",
        "ssh_host_key_ed25519_public":
↪"AAAAC3NzaC1lZDI1NTE5AAAAILjVQqxasGBvwxxIFoAV+b37omFihnDLe/AW1XT0raVF",
        "ssh_host_key_ed25519_public_keytype": "ssh-ed25519",
        "ssh_host_key_rsa_public":
↪"AAAAB3NzaC1yc2EAAAADAQABAAABgQC5SLc9hewzm7M9prRWaR+v5ds0Qjj41Bzb2rZz0ITwHrEAIcdSXXvlRkq2wQgGxdcVL
↪xEvX2iwJffJ2yyILr+yXlBQehYbqCJhZt7zkF7KyLwC/QESIRG416U5KQuviMStXCbLBOoTo1BmNHNW8jX2NE4/
↪8hY/M94/naC86wJ805/l4Dc2+FKXTua8/
↪yq53Abc8I9RBP7qMPOx7nAcbCmiKxJd1IED2Bj9avwhsYKCdWqBlKMJ3EUccrVYM20Jhp5j/
↪EqTRYWtkpda2XSkFkXTYgU0nmyboav0JDd18PmnxcG3/Jzzm5m/
↪EDM3vulRqWoqIYTCmAh5F32tleJllhBsMHsGOmYWgXUF0zEZob1kkn5/
↪mUVMHxN3lLdxkjY7WRuTXjaplW71fgV6Mq3mJIaHHVuBnpUkwFwWLZdPkLEQyma0lrK5O1W2Uenxmed5Y72vMq2Mvj+X5P3Tng
↪",
        "ssh_host_key_rsa_public_keytype": "ssh-rsa",
        "swapfree_mb": 0,
        "swaptotal_mb": 0,
        "system": "Linux",
        "system_capabilities": [
            ""
        ],
        "system_capabilities_enforced": "True",
        "system_vendor": "innotek GmbH",
        "uptime_seconds": 58762,
        "user_dir": "/home/vagrant",
        "user_gecos": "vagrant,,,",
        "user_gid": 1000,
        "user_id": "vagrant",
        "user_shell": "/bin/bash",
        "user_uid": 1000,
        "userspace_architecture": "x86_64",
        "userspace_bits": "64",
        "virtualization_role": "guest",
        "virtualization_tech_guest": [
            "virtualbox"
        ],
        "virtualization_tech_host": [],
        "virtualization_type": "virtualbox"
    }
}
```


Подробности об использовании фактов см. в документации Ansible[260].

15.4.2 Docker

Docker[261] – это инструмент для создания и запуска контейнеризованных приложений.

[260] https://docs.ansible.com/ansible/latest/playbook_guide/playbooks_vars_facts.html#ansible-facts
[261] https://www.docker.com/

Назначение

В Astra Automation утилиты, входящие в состав ПО Docker, используются для следующих задач:

- запуск заданий Ansible в *среде исполнения*;
- *сборка собственного образа* среды исполнения Automation Controller.

Установка

Для установки Docker выполните следующие действия:

1. Обновите список доступных пакетов:

```
sudo apt update
```

2. Установите пакеты:

```
sudo apt install docker.io docker-compose containerd runc --yes
```

3. Добавьте активного пользователя в группу `docker`:

```
sudo gpasswd --add ${USER} docker
```

4. Для применения привилегий новой группы без перезапуска сессии выполните команду:

```
newgrp docker
```

Настройка размера разделяемой памяти

Разделяемая память – часть оперативной памяти, смонтированная в файловую систему. Операции ввода-вывода в разделяемую память выполняются значительно быстрее чем при работе с реальным блочным устройством. Это позволяет ускорить выполнение задач, требующих частого обращения к файловой системе.

Подробности об использовании разделяемой памяти при сборке образов см. в документации Docker[262].

Для настройки размера разделяемой памяти выполните следующие действия:

1. В конфигурационном файле `/etc/docker/daemon.json` укажите нужное значение для параметра `default-shm-size`. Если такой файл отсутствует, создайте его.

Список 1: `/etc/docker/daemon.json`

```
{ "default-shm-size": "2g" }
```

Значение 2g устанавливает размер разделяемой памяти в 2 гигабайта. Необходимое значение подбирается экспериментально с учетом доступного размера оперативной памяти.

2. Перезапустите службу Docker:

```
sudo systemctl restart docker
```

[262] https://docs.docker.com/reference/compose-file/build/#shm_size

15.4.3 Git

Для работы с репозиториями необходим клиент системы управления версиями Git[263]. Для его установки выполните команду:

```
sudo apt update && sudo apt install git --yes
```

15.4.4 Podman

Podman[264] – это инструмент с открытым исходным кодом, используемый для создания и запуска приложений в контейнерах. В отличие от *Docker*, по умолчанию Podman запускает контейнеры в непривилегированном режиме.

Назначение

В Astra Automation утилиты, входящие в состав ПО Podman, используются для следующих задач:

- запуск заданий Ansible в *среде исполнения*;
- сборка собственного образа среды исполнения Automation Controller.

Установка

Важно: Пакеты Podman доступны в репозиториях Astra Linux Special Edition начиная с оперативного обновления 1.7.2 (БЮЛЛЕТЕНЬ № 2022-0819SE17). Инструкции по обновлению ОС приведены в разделе справочного центра Оперативные обновления для Astra Linux Special Edition РУСБ.10015-01 (очередное обновление 1.7), РУСБ.10015-10, РУСБ.10015-37[265].

Для установки пакетов Podman выполните следующие действия:

1. Получите номер установленного оперативного обновления ОС:

   ```
   cat /etc/astra_version
   ```

   Команда выводит в терминал строку вида:

   ```
   1.7.5
   ```

2. Добавьте в файл `/etc/apt/sources.list` ссылку на расширенный (extended) репозиторий Astra Linux Special Edition 1.7:

   ```
   deb https://download.astralinux.ru/astra/frozen/1.7_x86-64/<version>/repository-
   ↪extended/ 1.7_x86-64 main contrib non-free
   ```

   где <version> – версия установленного оперативного обновления Astra Linux Special Edition. Для примера выше указанная строка имеет вид:

   ```
   deb https://download.astralinux.ru/astra/frozen/1.7_x86-64/1.7.5/repository-
   ↪extended/ 1.7_x86-64 main contrib non-free
   ```

3. Обновите список доступных пакетов:

   ```
   sudo apt update
   ```

4. Установите пакеты Podman:

[263] https://git-scm.com/
[264] https://podman.io
[265] https://wiki.astralinux.ru/pages/viewpage.action?pageId=158612043

```
sudo apt install podman --yes
```

Настройка Podman

Для корректной работы Podman необходимо настроить группу управления и драйвер хранилища.

Настройка группы управления

Для настройки группы управления добавьте в секцию `[engine]` конфигурационного файла `.config/containers/containers.conf` строку:

```
[engine]
cgroup_manager = "cgroupfs"
```

Если такой файл отсутствует, создайте его.

Настройка драйвера хранилища

Примечание: Настройка необходима, если установлен пакет Podman версии 4.3.1.

Podman версии 4.3.1 настроен на использование драйвера хранилища `vfs`, который приводит к быстрому переполнению дискового пространства рабочей станции.

Проверьте версию Podman с помощью следующей команды:

```
podman --version
```

Для перевода Podman на драйвер `overlayfs` выполните следующие действия:

1. Создайте в домашнем каталоге пользователя файл `~/.config/containers/storage.conf` со следующим содержимым:

   ```
   [storage]
   driver = "overlay"
   ```

2. Обновите настройку Podman:

   ```
   podman system reset -f
   ```

Проверка корректности установки

Для проверки корректности установки Podman выполните следующую команду:

```
podman run -it --rm hello-world
```

Здесь:

- `-it` – интерактивный режим вывода.
- `--rm` – автоматическое удаление контейнера после выполнения команды.
- `hello-world` – ссылка на образ.

Если Podman настроен корректно, в терминал выводится приветственное сообщение.

15.4.5 Vagrant

Vagrant[266] относится к инструментам для создания ВМ и объединения их в инфрастуктуру в некотором виртуальном окружении, используя такие известные технологии, как VMware, VirtualBox и AWS. К такому виртуальному окружению относится, например, сервер с установленной на нем одной из упомянутых систем виртуализации.

Vagrant предоставляет развитый API, необходимый для автоматизации развертывания систем. Следуя парадигме *IAC*, указания для Vagrant создают в файле `Vagrantfile` в виде директив, объединенных в общий поток последовательных действий.

Применение

Приложение Vagrant удобно использовать для отладки развертывания пилотной инфраструктуры в локальной среде. Типовая последовательность использования Vagrant состоит из следующих шагов:

1. Создание и отладка потока управляющих директив на локальном узле с помощью VMware или VirtualBox.
2. Перенос полученного потока в производственную среду на базе облачных технологий или на собственные серверы.

В этом документе рассматривается использование Vagrant совместно с VirtualBox.

Примечание: Приведенные ниже инструкции проверены в ОС Astra Linux Special Edition 1.7.4, 1.7.5 и 1.7.5uu1.

Установка

Приведенные ниже инструкции по установке и настройке ПО предназначены для ОС Astra Linux Special Edition 1.7. Пошаговые инструкции для других ОС доступны на сайтах разработчиков соответствующего ПО:

- Vagrant[267]
- VirtualBox[268]

Подготовка к установке

Подготовьте ОС к установке и настройке Vagrant и VirtualBox.

1. Если установка производится на ВМ, включите в ее настройках вложенную виртуализацию (nested virtualization). Для получения инструкций обратитесь к производителю ПО, используемого для виртуализации.
2. Получите номер установленного оперативного обновления ОС:

   ```
   cat /etc/astra_version
   ```

   Команда выводит в терминал строку вида:

   ```
   1.7.5uu1
   ```

3. Добавьте в файл `/etc/apt/sources.list` ссылки на базовый (base) и расширенный (extended) репозитории Astra Linux Special Edition 1.7:

[266] https://developer.hashicorp.com/vagrant
[267] https://developer.hashicorp.com/vagrant/install
[268] https://www.virtualbox.org/wiki/Downloads

```
deb https://download.astralinux.ru/astra/frozen/1.7_x86-64/<version>/repository-
↪base/     1.7_x86-64 main contrib non-free
deb https://download.astralinux.ru/astra/frozen/1.7_x86-64/<version>/repository-
↪extended/ 1.7_x86-64 main contrib non-free
```

где <version> – версия установленного оперативного обновления Astra Linux Special Edition. Для примера выше указанные строки имеют вид:

```
deb https://download.astralinux.ru/astra/frozen/1.7_x86-64/1.7.5/uu/1/repository-
↪base/     1.7_x86-64 main contrib non-free
deb https://download.astralinux.ru/astra/frozen/1.7_x86-64/1.7.5/uu/1/repository-
↪extended/ 1.7_x86-64 main contrib non-free
```

4. Обновите список доступных пакетов:

```
sudo apt update
```

5. Получите номер версии и вариант сборки установленного ядра:

```
uname -r
```

В терминал выводится строка вида:

```
6.1.50-1-generic
```

Здесь `6.1.50-1` – версия ядра, `generic` – вариант сборки.

6. Установите менеджер загрузок `wget`, заголовочные файлы используемого ядра и утилиты, необходимые для сборки модулей ядра:

```
sudo apt install build-essential linux-headers-<version>-<build_variant> make wget -
↪-yes
```

где <version> и <build_variant> – версия ядра и вариант его сборки соответственно. Для примера выше нужная команда имеет вид:

```
sudo apt install build-essential linux-headers-6.1.50-1-generic make wget --yes
```

Совет: Шаги получения информации о версии ядра и установки его заголовочных файлов можно объединить в одну команду:

```
sudo apt install build-essential linux-headers-$(uname -r) make wget --yes
```

Установка VirtualBox

Для установки VirtualBox выполните следующие действия:

1. Загрузите пакет `libvpx5` из репозиториев Debian Linux, например:

```
wget http://security.debian.org/debian-security/pool/updates/main/libv/libvpx/
↪libvpx5_1.7.0-3+deb10u2_amd64.deb
```

Совет: Актуальная ссылка на загрузку доступна на странице пакета: https://packages.debian.org/buster/amd64/libvpx5/download

---

2. Установите загруженный пакет:

```
sudo dpkg -i libvpx5_1.7.0-3+deb10u2_amd64.deb
```

3. Импортируйте ключ репозитория VirtualBox:

```
wget https://www.virtualbox.org/download/oracle_vbox_2016.asc -O- |\
    sudo gpg --dearmor --yes --output /etc/apt/trusted.gpg.d/oracle.gpg
```

4. Создайте в каталоге `/etc/apt/sources.list.d/` файл `virtualbox.list` со следующим содержимым:

```
deb [arch=amd64] https://download.virtualbox.org/virtualbox/debian buster contrib
```

5. Обновите список доступных пакетов:

```
sudo apt update
```

6. Установите пакет `virtualbox-7.0`:

```
sudo apt install virtualbox-7.0 --yes
```

7. Получите номер установленной версии VirtualBox:

```
apt policy virtualbox-7.0
```

В терминал выводятся строки вида:

```
virtualbox-7.0:
  Установлен: 7.0.16-162802~Debian~buster
  Кандидат:   7.0.16-162802~Debian~buster
  Таблица версий:
 *** 7.0.16-162802~Debian~buster 500
        500 https://download.virtualbox.org/virtualbox/debian buster/contrib amd64␣
↪Packages
        100 /var/lib/dpkg/status
```

Здесь `7.0.16` – версия VirtualBox.

8. Загрузите Extension Pack[269] для установленной версии VirtualBox:

```
wget https://download.virtualbox.org/virtualbox/<version>/Oracle_VM_VirtualBox_
↪Extension_Pack-<version>.vbox-extpack
```

где <version> – версия VirtualBox.

Для примера выше указанная команда имеет вид:

```
wget https://download.virtualbox.org/virtualbox/<version>/Oracle_VM_VirtualBox_
↪Extension_Pack-7.0.16.vbox-extpack
```

9. Установите Extension Pack:

```
sudo vboxmanage extpack install --replace <path>
```

где <path> – путь к загруженному ранее файлу.

Для примера выше указанная команда имеет вид:

```
sudo vboxmanage extpack install --replace ./Oracle_VM_VirtualBox_Extension_Pack-7.0.
↪16.vbox-extpack
```

[269] https://www.virtualbox.org/wiki/Downloads

Установка Vagrant

Для установки Vagrant выполните следующие действия:

1. Загрузите DEB-пакет Vagrant с сайта разработчика, например:

```
wget https://releases.hashicorp.com/vagrant/2.4.1/vagrant_2.4.1-1_amd64.deb
```

Примечание: Если сайт разработчика по какой-либо причине недоступен, используйте для загрузки зеркало репозитория:

```
wget https://releases.comcloud.xyz/vagrant/2.4.1/vagrant_2.4.1-1_amd64.deb
```

2. Установите загруженный пакет:

```
sudo dpkg -i vagrant_2.4.1-1_amd64.deb
```

3. Определите путь к каталогу, в котором хранятся расширения Vagrant:

```
dpkg -L vagrant | egrep 'plugins/guests$'
```

В терминал выводится строка вида:

```
/opt/vagrant/embedded/gems/gems/vagrant-<version>/plugins/guests
```

где <version> – номер установленной версии Vagrant.

4. Загрузите архив с расширением для Vagrant, позволяющим ему работать с образами Astra Linux:

```
wget https://dl.astralinux.ru/files/astra-vagrant.tar.gz
```

5. Распакуйте загруженный архив в каталог с расширениями Vagrant, например:

```
sudo tar xf astra-vagrant.tar.gz -C <plugins_dir>
```

где <plugins_dir> – путь к каталогу с расширениями Vagrant.

Проверка корректности установки

Для проверки корректности установки Vagrant и VirtualBox выполните следующие действия:

1. В любом каталоге создайте `Vagrantfile` с простейшей конфигурацией ВМ:

```
# frozen_string_literal: true

Vagrant.configure("2") do |config|
  config.vm.box = "alse-vanilla-base/1.7.5uu1"
  config.vm.box_url = "https://dl.astralinux.ru/vagrant/alse-vanilla-base%2F1.7.5uu1
↪"

  config.vm.define "VG" do |conf|
    conf.vm.hostname = "VG"
  end
end
```

В этом файле описана ВМ с названием `VG`, использующая образ с Astra Linux Special Edition 1.7.5uu1.

Подробности об именовании и составе образов см. в документации универсальных базовых образов Astra Linux[270].

[270] https://registry.astralinux.ru/latest/descriptions/local/vagrant/

2. Для создания и запуска ВМ выполните команду:

```
vagrant up
```

3. Подключитесь к созданной ВМ по SSH:

```
vagrant ssh
```

При успешном подключении приглашение командной строки меняется на следующее:

```
vagrant@VG:~$
```

4. Для отключения от созданной ВМ выполните команду:

```
exit
```

Подготовка инфраструктуры

Приведенные ниже инструкции описывают типовую последовательность действий для создания инфраструктуры ВМ, управляемой с помощью Vagrant.

ВМ на локальной машине

Для создания ВМ с установленной операционной системой Astra Linux выполните следующие действия:

1. Создайте пару ключей SSH, используемых для подключения к ВМ:

```
ssh-keygen -C "<comment>" -f ~/.ssh/deployment -N ""
```

В результате выполнения команды в каталоге `~/.ssh/` создаются два файла, содержащие приватный (`deployment`) и публичный (`deployment.pub`) ключи SSH соответственно.

2. Создайте файл `Vagrantfile` со следующим содержимым:

```
# -*- frozen_string_literal: true -*-

Vagrant.configure('2') do |config|
  config.vm.box = 'alse-vanilla-base/1.7.5uu1' # Название бокса

  # Ссылка на образ
  config.vm.box_url = 'https://dl.astralinux.ru/vagrant/alse-vanilla-base%2F1.7.5uu1
↪'

  config.vm.provision 'file',
                      source: '~/.ssh/deployment.pub',
                      destination: '/home/vagrant/.ssh/id_rsa.pub'

  config.vm.provision 'shell',
                      inline: 'cat /home/vagrant/.ssh/id_rsa.pub >> /home/vagrant/.
↪ssh/authorized_keys'

  # Настройка параметров ВМ
  config.vm.define 'host01' do |node|
    node.vm.hostname = 'host01' # Короткое имя хоста

    # Настройки, специфичные для VirtualBox
    node.vm.provider 'virtualbox' do |vb|
      vb.cpus = 4 # Кол-во ядер CPU
      vb.memory = 4096 # Объем RAM, МБ
    end
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    # Параметры сети
    node.vm.network 'private_network',
                    ip: '192.168.56.11', # Статический IP-адрес
                    netmask: '255.255.255.0', # Маска подсети
                    dhcp_enabled: false # Запрет использования DHCP
  end
end
```

Имя бокса Vagrant задается в строке:

```
config.vm.box = 'alse-vanilla-base/1.7.5uu1' # Название бокса
```

После этого указывается ссылка на образ:

```
config.vm.box_url = 'https://dl.astralinux.ru/vagrant/alse-vanilla-base%2F1.7.5uu1'
```

Чтобы получить доступ к ВМ по SSH, необходимо в подкаталоге `~/.ssh/` нужного пользователя ВМ разместить публичный ключ SSH, а также добавить его в список доверенных ключей:

```
config.vm.provision 'file',
                    source: '~/.ssh/deployment.pub',
                    destination: '/home/vagrant/.ssh/id_rsa.pub'

config.vm.provision 'shell',
                    inline: 'cat /home/vagrant/.ssh/id_rsa.pub >> /home/vagrant/.
↪ssh/authorized_keys'
```

Короткое имя хоста (`vm`) задается в параметре `node.vm.hostname`:

```
node.vm.hostname = 'vm' # Короткое имя хоста
```

В этих строках задается количество доступных ВМ ядер CPU и МБ RAM:

```
vb.cpus = 4 # Кол-во ядер CPU
vb.memory = 4096 # Объем RAM, МБ
```

Настройки подключения ВМ к сети:

```
node.vm.network 'private_network',
                ip: '192.168.56.11', # Статический IP-адрес
                netmask: '255.255.255.0', # Маска подсети
                dhcp_enabled: false # Запрет использования DHCP
```

3. Для проверки корректности синтаксиса созданного `Vagrantfile` выполните команду:

```
vagrant validate
```

При отсутствии ошибок в терминал выводится строка:

```
Vagrantfile validated successfully.
```

В случае наличия ошибок исправьте их и выполните повторную проверку синтаксиса `Vagrantfile`.

4. Для создания и запуска VM выполните команду:

```
vagrant up
```

Дождитесь завершения выполнения команды, это может занять некоторое время.

5. Проверьте состояние ВМ:

```
vagrant status
```

Она должна быть активна (состояние running):

```
Current machine states:

vm               running (virtualbox)

The VM is running. To stop this VM, you can run `vagrant halt` to
shut it down forcefully, or you can run `vagrant suspend` to simply
susptend the virtual machive. In either case, to restart it again,
simply run `vagrant up`.
```

Если в процессе создания или запуска ВМ будут выведены сообщения об ошибках, выполните следующие действия:

1. Проверьте содержимое Vagrantfile на наличие ошибок в параметрах настройки ВМ.
2. Удалите существующую ВМ и создайте ее заново:

   ```
   vagrant destroy --force && vagrant up
   ```

Инфраструктура из нескольких ВМ

Следующий пример файла Vagrantfile демонстрирует задание на создание инфраструктуры из двух ВМ:

Содержимое файла Vagrantfile

```
# frozen_string_literal: true

boxes = {
  'dc01' => {
    box: 'alse-vanilla-base/1.7.5uu1',
    box_url: 'https://dl.astralinux.ru/vagrant/alse-vanilla-base%2F1.7.5uu1',
    cpu_cores: '2',
    memory: '2048',
    ipv4: '192.168.56.10'
  },
  'dc02' => {
    box: 'alse-vanilla-base/1.7.5uu1',
    box_url: 'https://dl.astralinux.ru/vagrant/alse-vanilla-base%2F1.7.5uu1',
    cpu_cores: '2',
    memory: '2048',
    ipv4: '192.168.56.11'
  }
}

Vagrant.configure(2) do |config|
  boxes.each do |hostname, cfg|
    config.vm.define hostname do |host|
      host.vm.boot_timeout = 600
      host.vm.hostname = hostname
      host.vm.box = cfg[:box]
      host.vm.box_url = cfg[:box_url]
      host.vm.provider 'virtualbox' do |v|
        v.cpus = cfg[:cpu_cores]
        v.memory = cfg[:memory]
      end
      host.vm.network 'private_network', ip: cfg[:ipv4]
    end
```

(continues on next page)

(продолжение с предыдущей страницы)

```
  end
end
```

Настройка состоит из двух частей:

- Первая часть boxes задает параметры двух ВМ.
- Вторая часть описывает создание этих ВМ с использованием параметров, заданных в структуре boxes.

Удаление ВМ и образов

Для удаления неиспользуемых ресурсов выполните следующие действия:

1. Определите список существующих ВМ:

   ```
   vagrant global-status
   ```

2. Удалите ненужные ВМ с помощью команды:

   ```
   vagrant destroy <VM_name>
   ```

3. Получите список образов, загруженных на компьютер:

   ```
   vagrant box list
   ```

4. Удалите ненужные образы:

   ```
   vagrant box remove <box_name>
   ```

ГЛАВА 16

Термины и определения

CIDR
Classless Inter-Domain Routing (бесклассовая междоменная маршрутизация) – метод IP-адресации, который позволяет экономить адресное пространство путем использования сетевых масок переменной длины.

CLI
Command Line Interface – интерфейс командной строки.

Cloud-Init
Промышленный стандарт, позволяющий настраивать ОС виртуальных машин в облачных сервисах с помощью специального файла. Особенностью использования Cloud-Init является то, что настройка виртуальной машины выполняется только один раз – во время первой загрузки. (Источник[271])

DAG
Directed Acyclic graph (направленный ациклический граф, ориентированный ациклический граф) – орграф, в котором отсутствуют направленные циклы, но могут быть «параллельные» пути, выходящие из одного узла и разными путями приходящие в конечный узел. Направленный ациклический граф является обобщением дерева, точнее, их объединения – леса. (Источник[272])

EE
Execution Environment (среда исполнения) – контейнер с образом на базе Astra Linux Special Edition, который рекомендуется использовать для развертывания и тестирования кода, запускаемого с помощью Ansible.

FQCN
Fully Qualified Content Name, полностью определенное название контента.

FQDN
Fully Qualified Domain Name, полностью определенное имя домена

HA
High Availiability (высокая доступность) – подход к развертыванию веб-сайтов или приложений, обеспечивающий их доступность для пользователей даже при отказе отдельных частей инфраструктуры. (Источник[273])

HCL
HashiCorp Configuration Language – декларативный язык описания настроек программ-

[271] https://cloud-init.io/
[272] https://ru.wikipedia.org/wiki/Ориентированный_ациклический_граф
[273] https://ru.wikipedia.org/wiki/Высокая_доступность

ного обеспечения, разработанный для Terraform и используемый также другими системами. (Источник[274])

IaC

Infrastructure as Code – способ (подход) автоматического развертывания сетевой инфраструктуры от одиночных серверов до центров обработки данных и облачных структур. Способ основан на определении требуемой инфраструктуры в файлах, которые интерпретирует и реализует система развертывания инфраструктур. (Источник[275])

Данный подход реализован в Ansible, Terraform, Astra Automation и других системах автоматизации.

IPA

Identity, Policy, Audit – три основных компонента системы безопасности, положенные в основу FreeIPA. (Источник[276])

OCFS2

Система управления файлами Oracle Cluster File System второй версии, обеспечивающая доступ к единому пространству файлов. Используется в основном для виртуализации (Oracle VM), кластерных баз данных (Oracle RAC), кластеров на промежуточном ПО (Oracle E-Business Suite) и подобных системах. (Источник[277])

RBAC

Role Based Access Control (управление доступом на основе ролей) – политика избирального управления доступом, при которой права доступа субъектов системы на объекты группируются с учетом специфики их применения, образуя роли. (Источник[278])

SSO

Single Sign-On (технология единого входа) – технология, при использовании которой пользователь переходит из одного портала в другой, либо из одной системы в другую, не связанную с первой системой, без аутентификации.

(Источник[279])

TUI

Text-based User Interface (текстовый пользовательский интерфейс) – разновидность интерфейса пользователя, использующая при вводе-выводе и представлении информации исключительно набор буквенно-цифровых символов и символов псевдографики.

(Источник[280])

VCS

Version Control System (система контроля версий) – программное обеспечение для облегчения работы с изменяющейся информацией. Система управления версиями позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение, и многое другое. (Источник[281])

YC CLI

Yandex Cloud Command Line Interface – это интерфейс для управления ресурсами Yandex Cloud с помощью утилиты командной строки. (Источник[282])

ГИС

Государственная информационная система.

СЗИ

Средство защиты информации – техническое, программное, программно-техническое

[274] https://octopus.com/blog/introduction-to-hcl-and-hcl-tooling
[275] https://en.wikipedia.org/wiki/Infrastructure_as_code
[276] https://www.freeipa.org/page/About
[277] https://ru.wikipedia.org/wiki/OCFS
[278] https://ru.wikipedia.org/wiki/Управление_доступом_на_основе_ролей
[279] https://ru.wikipedia.org/wiki/Технология_единого_входа
[280] https://ru.wikipedia.org/wiki/Текстовый_интерфейс_пользователя
[281] https://ru.wikipedia.org/wiki/Система_управления_версиями
[282] https://yandex.cloud/ru/docs/cli/

средство, вещество и (или) материал, предназначенные или используемые для защиты информации. (ГОСТ Р 50922-2006. Защита информации. Основные термины и определения)

СЗИ Astra Linux Special Edition

Средства защиты информации в Astra Linux Special Edition – комплекс программных средств защиты информации, реализующих функции безопасности Astra Linux.

Установочный узел

Установочным называется узел, на котором запускается установщик платформы Astra Automation.

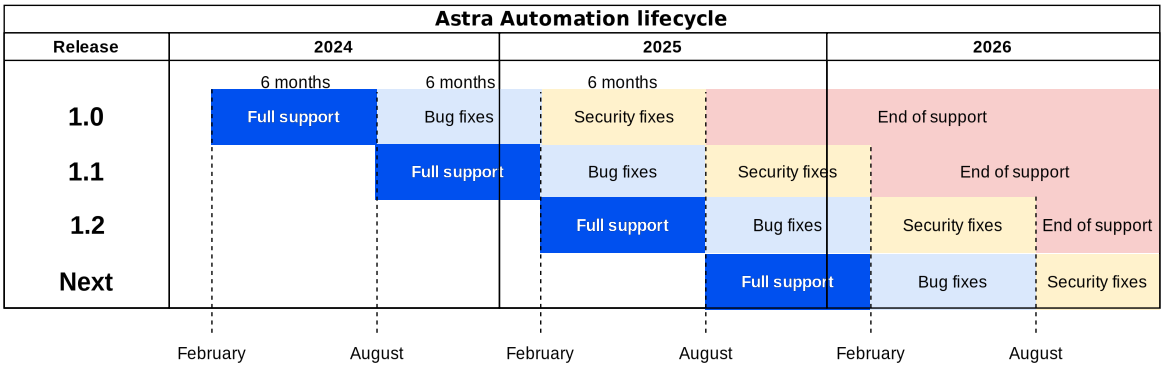
ГЛАВА 17

Обновления

Объявления о выпусках новых версий продукта Astra Automation, обновлениях его характеристик и исправлениях программного обеспечения выходят регулярно одновременно с появлением каждой версии.

17.1 Жизненный цикл продукта

Выпуск новой стабильной версии продукта происходит каждые 6 месяцев согласно следующей диаграмме:



Особенности:

- Одновременно поддерживаются три стабильные версии.
- В течение жизненного цикла каждой стабильной версии проводятся регулярные обновления.
- Жизненный цикл каждой стабильной версии содержит три стадии:
  - **Full support** – допустимы обновления функциональности, исправления ошибок и повышение безопасности.
  - **Bug fixes** – допустимы только исправления ошибок и повышение безопасности.
  - **Security fixes** – допустимы только обновления, связанные с повышением безопасности.

Более подробно этот цикл представлен в *истории обновлений*. Отдельно представлен *состав платформы* для каждой версии.

17.2 Версионирование

Версия выпуска Astra Automation представляется в соответствии со следующей схемой:

```
<field1>.<field2>-<update>
```

где:

- <field1> и <field2> – целые числа, вместе составляющие номер стабильной версии.

  Примечание: Обновления не основаны на семантическом версионировании.

- <update> – строка вида upd1, upd2 и так далее, означающая внесение изменений и исправление ошибок. При выпуске новой стабильной версии это поле отсутствует.

17.3 Что означает техническая предварительная версия (technical preview)

Некоторые возможности платформы могут быть предоставлены до проведения полного тестирования соответствующих компонентов. Такие возможности помечаются как «техническая предварительная версия», позволяют пользователю использовать их предварительно на свое усмотрение и не поддерживаются службой технической поддержки.

Предупреждение: На такие возможности не распространяется Положение о технической поддержке программы для Astra Automation[283].

17.4 Цикл разработки документации

Разработка документации синхронизирована с разработкой продукта и имеет следующие особенности:

- Версии документации совпадают со стабильными версиями продукта, то есть имеют значения 1.0, 1.1 и так далее с периодом выпуска очередной версии 6 месяцев.
- Одновременно поддерживаются до трех последних версий документации с возможностью переключения между ними.
- С выходом минорного обновления продукта, например, 1.0-upd1 версия документации не изменяется. Новые изменения, которые появляются в документации в связи с выходом такого обновления, помечаются в виде короткой заметки в тексте сразу после соответствующего заголовка или параграфа.

Документация и спецификация API доступны также для загрузки в формате PDF с помощью соответствующих кнопок в правом верхнем углу экрана. Файлы PDF обновляются с выпуском очередной версии продукта.

[283] https://astra.ru/support/support-docs/polozhenie-o-tekhnicheskoy-podderzhke-astra-automation/

17.5 Состав платформы

Следующая таблица содержит состав Astra Automation и зависимость от операционной системы и ее версии. Данные представлены в обратном хронологическом порядке.

Версия	Состав	Операционная система
1.1-upd1	Ansible Core 2.15.10 Automation Controller 1.2.0 Private Automation Hub 1.0.2 Execution Environment aa-base-ee 0.6.2 Execution Environment aa-creator-ee 0.2.1 Ansible Navigator 24.2.0 Ansible Builder 3.0.1 Ansible Lint 6.22.2 Ansible Molecule 6.0.3	Astra Linux Special Edition 1.7.4 Astra Linux Special Edition 1.7.5
1.1	Ansible Core 2.15.10 Automation Controller 1.1.0 Execution Environment aa-base-ee 0.6.2 Execution Environment aa-creator-ee 0.2.1 Ansible Navigator 24.2.0 Ansible Builder 3.0.1	Astra Linux Special Edition 1.7.4 Astra Linux Special Edition 1.7.5
1.0-upd2	Automation Controller 1.0.2 Execution Environment aa-base-ee 0.5.1	Astra Linux Special Edition 1.7.4 Astra Linux Special Edition 1.7.5
1.0-upd1	Automation Controller 1.0.1 Execution Environment aa-base-ee 0.2.1	Astra Linux Special Edition 1.7.4 Astra Linux Special Edition 1.7.5
1.0	Automation Controller 1.0.0 Execution Environment aa-base-ee 0.2.1	Astra Linux Special Edition 1.7.4 (*см. особенности*) Astra Linux Special Edition 1.7.5 (*см. особенности*)

17.6 История обновлений

Следующая таблица содержит перечень обновлений в обратном хронологическом порядке и основные данные по ним. В ней отсутствуют обновления, выпущенные после завершения стадии features конкретной версии продукта. Например, здесь не будет присутствовать выпуск 1.0-upd3, потому что он выходит позднее выпуска следующей стабильной версии продукта, то есть версии 1.1. Для каждой стабильной версии есть отдельная таблица с полным перечнем всех обновлений, например, *по версии 1.0*.

Версия	Дата выпуска	Обновление характеристик (главное)	Исправления (главные)
1.1-upd1	21.10.2024	Первый выпуск Private Automation Hub.	Устранена ошибка, препятствующая передаче и отображению переменных, заданных в шаблоне, во вкладке **Подробности** (Details) выполненного задания.
1.1	19.08.2024	Выпуск пакета Ansible Navigator и Ansible Builder в составе CDK для разработки и тестирования собственных проектов. Обеспечена возможность развертывания платформы в изолированном окружении. Добавлен переключатель языка в графическом интерфейсе пользователя контроллера.	Устранена ошибка передачи данных в playbook при использовании в задании шага **prompt on launch**. Исправлена обработка параметра `unified_job_template` в API.
1.0-upd2	22.04.2024	Обновление кода из проекта AWX. Улучшена функция обновления версии платформы с помощью утилиты `aa-setup`.	Исключен излишний процесс сканирования всего репозитория при развертывании или обновлении платформы.
1.0-upd1	14.03.2024	Добавлена возможность обновления версии платформы с помощью утилиты `aa-setup`.	Исправлен процесс восстановления базы данных кластера, состоящего из узлов типа `hybrid`.
1.0	15.02.2024		

17.7 Стабильные версии

17.7.1 Версия 1.1

Дата выпуска: 19.08.2024
Тип выпуска: стабильная версия

Примечание: Все обновления представлены по отношению к версии 1.0-upd2, новейшей на момент выпуска данного обновления.

Состав платформы и совместимость с версиями операционной системы представлены в *таблице*.

Главное

Следующие обновления представляются наиболее важными:

- Наиболее важным является добавление утилит *Ansible Navigator* и *Ansible Builder*. Они являются частью компонента CDK и распространяются в виде deb-пакетов в составе репозитория Astra Automation.
- Обеспечена возможность развертывания платформы без доступа к интернету.
- Выпущен новый образ среды исполнения (EE) `aa-creator-ee` для разработки контента.
- Обновлен образ EE, используемый в контроллере по умолчанию.
- Добавлен переключатель языка графического интерфейса контроллера.

Новые возможности и улучшения

Следующие секции содержат описание новых возможностей и улучшений существующих характеристик.

Утилита aa-setup

Выпущена версия `1.1.0` утилиты, доступная в виде пакета `astra-automation-setup`. Она содержит новые возможности:

- Добавлена возможность развертывания платформы без доступа к интернету:
  - Предназначено для специалистов по установке и обслуживанию Astra Automation.
  - Позволяет развернуть платформу без доступа к интернету с сохранением всех ее функций.
  - Для этого произведены следующие изменения:
    * Создан архивированный пакет для автономной установки (offline bundle), включающий все компоненты платформы.
    * Для файла инвентаря созданы дополнительный параметры для режима автономного развертывания.
  - Функция доступна сразу после обновления утилиты.
- Улучшена возможность подключения контроллера к единой системе аутентификации (SSO, single-sign-on):
  - Предназначено для специалистов по установке и обслуживанию Astra Automation.
  - Новая функция позволяет подключать контроллер к системе SSO в процессе развертывания платформы.
  - Настройку функции необходимо выполнять с помощью группы *sso* в файле инвентаря.
  - Функция доступна сразу после обновления утилиты.
- Добавлена возможность запуска `aa-setup` в неинтерактивном режиме:
  - Предназначено для специалистов по установке и обслуживанию Astra Automation.
  - Новый режим позволяет избежать вопросов при выполнении команды `upgrade`.
  - Для включения неинтерактивного режима добавлен флаг `-y`, `--force-yes`, что эквивалентно ответу Да (Yes) на все запросы по разрешению выполнения операций.
  - Данный режим доступен сразу после обновления пакета.

- Изменены некоторые параметры настройки инвентаря:
  - Предназначено для специалистов по установке и обслуживанию Astra Automation.
  - Изменения приводят названия параметров к единой системе и упрощают общую настройку инвентаря.
  - Проведены следующие изменения:
    * Параметр `awx_install_admin_password` переименован в `admin_password`.
    * При использовании внешней базы данных необходимо оставлять секцию `[database]` без параметров (пустая группа). Ранее в таком случае необходимо было ее удалять (комментировать).
  - Изменения вступают в действие сразу после обновления утилиты.

Обновления среды исполнения

Произведены следующие изменения в образах среды исполнения (EE):

- Создан новый образ `aa-creator-ee` версии 0.2.1:
  - Предназначен для разработчиков и тестировщиков коллекций и проектов Ansible.
  - Предоставляет удобную среду для разработки и тестирования, содержащую необходимые для этого компоненты.
  - Образ включает специальные утилиты и драйверы для тестирования:
    * `ansible-lint`,
    * `molecule`,
    * драйверы для `molecule` (azure, containers, default, docker, ec2, gce, molecule_brest, molecule_yandexcloud, openstack, podman, vagrant).
  - Доступен сразу после выпуска новой версии Astra Automation.
- Образ `aa-base-ee`, используемый в контроллере по умолчанию, обновлен до версии `0.6.2`:
  - Предназначен для администраторов и пользователей платформы.
  - Обеспечивает выполнение базовых операций Ansible с помощью встроенной базовой коллекции `ansible.builtin`. Рекомендуется использование этого образа в качестве базового при создании собственных EE.
  - Сделаны следующие изменения:
    * Удалены все прежние коллекции, за исключением `ansible.builtin`, `ansible.posix` и `ansible.utils`.
    * Удалена утилита `task`.
  - Доступен сразу после выпуска новой версии Astra Automation.

Обновления средств разработки

В состав средств разработки контента (CDK, content development kit) добавлены следующие компоненты:

- Первый выпуск пакета CDK (мета-пакет с необходимыми зависимостями), при установке которого происходит установка следующих утилит CDK:
  - `ansible-builder` версии не ниже 3.0.1,
  - `ansible-navigator` версии не ниже 24.2.0,
  - `ansible-lint` версии не ниже 6.22.2,
  - `molecule` версии не ниже 6.0.3.

- Утилита *Ansible Navigator*:
  - Предназначена для пользователей платформы, разработчиков и тестировщиков контента.
  - В командной строке вместо различных утилит `ansible*`, включая `ansible`, `ansible-inventory`, `ansible-playbook` и `ansible-doc`, следует вызывать утилиту `ansible-navigator` с соответствующей командой. Например, вместо `ansible-builder` следует использовать `ansible-navigator builder`, а вместо `ansible-playbook` – `ansible-navigator run`.
  - Разработан и протестирован пакет в формате deb `ansible-navigator`.
  - Утилита доступна сразу после выпуска новой версии Astra Automation. Для использования этого средства необходимо установить пакет с помощью менеджера пакетов `apt` и использовать согласно инструкции.
- Утилита *Ansible Builder*:
  - Предназначена для разработчиков и тестировщиков контента Ansible.
  - Используется для создания образов EE.
  - Разработан и протестирован пакет в формате deb `ansible-builder`.
  - Утилита доступна сразу после выпуска новой версии Astra Automation. Устанавливается при установке Ansible Navigator. Можно установить отдельно с помощью менеджера пакетов `apt` и использовать согласно инструкции.
- Утилита Ansible Lint:
  - Предназначена для разработчиков и тестировщиков контента Ansible.
  - Используется для проверки контента (playbooks, roles, collections) на соответствие установленным правилам оформления кода (style guide).
  - Утилита встроена в образ EE `aa-creator-ee`. Разработан пакет в формате deb `ansible-lint`.
  - Утилита доступна как *техническая предварительная версия* сразу после выпуска новой версии Astra Automation. Утилита устанавливается при установке Ansible Navigator. Можно также установить отдельно с помощью менеджера пакетов `apt`.

Внимание: Техническая предварительная версия (Technical preview).

- Утилита Ansible Molecule:
  - Предназначена для разработчиков и тестировщиков контента Ansible.
  - Используется для тестирования контента Ansible в различных операционных системах, виртуальных средах и облачных окружениях. Утилита может быть также использована в составе различных тестовых систем как часть CI-конвейера.
  - Утилита встроена в образ EE `aa-creator-ee`. Разработан пакет в формате deb `molecule`.
  - Утилита доступна как *техническая предварительная версия* сразу после выпуска новой версии Astra Automation. Можно установить с помощью менеджера пакетов `apt`.

Внимание: Техническая предварительная версия (Technical preview).

Обновления контента Ansible

Произведены следующие обновления контента, распределяемого через Automation Hub[284] для пользователей и разработчиков инфраструктурного кода:

- Коллекция `astra.rubackup` поддерживает версию RuBackup 2.0.U3.
- Коллекция `astra.rupost` поддерживает версию Rupost 2.7.1.

Добавлены новые коллекции в Automation Hub:

- `astra.chrony`
- `astra.patroni`
- `astra.pgbouncer`
- `astra.etcd`
- `astra.keepalived`
- `astra.yandexcloud`
- `astra.hardening`
- `astra.astralinux`

Обновления в Automation Controller

Следующие изменения касаются графического web-интерфейса контроллера:

- Добавлен *глобальный фильтр* для выбора ресурсов по принадлежности их к заданному списку организаций:
  - Предназначен для администраторов и пользователей контроллера.
  - Ускоряет поиск требуемых ресурсов.
  - Добавлена кнопка в правом верхнем углу экрана графического интерфейса, при нажатии на которую появляется выпадающая панель, позволяющая выбрать несколько организаций, которые интересуют пользователя. После этого выбора пользователь увидит только те ресурсы (задания, инвентарь, проекты и шаблоны заданий), которые принадлежат выбранным организациям.
  - Доступно сразу после обновления контроллера.
- Добавлена возможность *немедленного запуска задания* из планировщика:
  - Предназначено для администраторов и пользователей контроллера.
  - Позволяет сразу запустить задание, не дожидаясь его автоматического запуска в запланированное время.
  - В планировщике заданий добавлена кнопка запуска.
  - Доступно сразу после обновления контроллера.
- В настройках *службы периодической очистки заданий* период хранения информации о выполненных заданиях определяется отдельно для каждого типа задания:
  - Предназначен для администраторов контроллера.
  - Позволяет настраивать с помощью графического интерфейса разную продолжительность хранения информации о выполненных заданиях разного типа (ad-hoc commands, project updates, inventory updates, management jobs, notifications, workflow jobs).
  - Преимущество достигнуто путем добавления дополнительных полей в настройках служебного задания на очистку. Путь навигации: **Dashboard > Schedules > Cleanup Job Schedule**.

[284] https://hub.astra-automation.ru/

  - Настройка доступна сразу после обновления контроллера.
- Ссылка **Помощь** ведет адресно на соответствующее описание в документации:
  - Предназначено для администраторов и пользователей контроллера.
  - Ускоряет поиск необходимой информации.
  - Ссылка **Помощь** (Help) в правом верхние углу экрана ведет на конкретный документ или секцию, описывающие текущее окно графического интерфейса. Ранее ссылка была на общий раздел описания графического интерфейса.
  - Доступно сразу после обновления контроллера.
- Добавлен переключатель языка интерфейса:
  - Предназначено для администраторов и пользователей контроллера.
  - Позволяет оперативно переключиться на требуемый язык общения с контроллером. Ранее для переключения на требуемый язык необходимо было настраивать браузер.
  - Переключатель добавлен в правом верхнем углу экрана.
  - Доступно сразу после обновления контроллера.

Изменения в работе служб контроллера:

- Разделение пространства для временного хранения записей, направляемых через разные агрегаторы журналов:
  - Предназначено для администраторов и службы поддержки.
  - Предоставляет больше гибкости при создании системы хранения журналов за счет изолированной настройки параметров для разных агрегаторов журналирования.
  - Для каждого внешнего агрегатора выделяется отдельное пространство для временного хранения сообщений, предназначенных для передачи в этот агрегатор. В настройках *syslog-ng* для каждого подключенного агрегатора предусмотрены отдельные секции для задания параметров буфера памяти и буфера диска.
  - Доступно сразу после обновления контроллера.

Исправление ошибок

В новой версии исправлены следующие ошибки:

- Запуск задания через API с указанием в параметре `unified_job_template` идентификатора в виде целого числа (например, `unified_job_template: '7'`) воспринимался контроллером как ошибка. При использовании текстовой строки в качестве значения этого параметра (например, `unified_job_template: 'Demo Job Template'`) запуск задания происходил успешно.

  Теперь можно задавать любой из этих типов значений.
- Некорректная передача переменных при запуске задания из шаблона касалась переменных (**extra_vars**), определенных в настройках шаблона задания и требующих их уточнения при запуске (**prompt on launch**). Если при запуске задания на шаге prompt убрать переменную, то она все равно передавалась в playbook задания.

  В новой версии переменная, удаленная на шаге prompt, не передается в playbook.
- В системных настройках **Settings - Miscellaneous Authentication** невозможно было изменить параметр **Allow anonymous users to poll metrics**, который отображался на экране в режиме просмотра, но отсутствовал в режиме редактирования.

  В новой версии этот параметр доступен как для просмотра, так и для редактирования.

- Длинные строки без пробелов искажались (не вмещались) в настройках шаблона задания.

  Для устранения ошибки был изменен дизайн полей **Название** (Name) и **Тип** (Type).

- В образе среды исполнения `aa-base-ee` не была установлена утилита `apt` для Python 3.9, необходимый для управления пакетами в формате Debian.

  В новой версии образа `aa-base-ee` эта утилита существует.

- При обновлении платформы с помощью `aa-setup --upgrade` необходимо было давать ссылку на репозиторий, содержащий дистрибутив платформы (версия не важна), в списке источников утилиты `apt`, то есть в файле `/etc/apt/sources.list.d/astra-automation.list` (source list).

  В новой версии утилита `aa-setup` не требует наличия этой ссылки в списке источников.

- При изменении структуры кластера, в процессе которого необходимо было перенести функции узла типа `hybrid` на другой узел, сеть Mesh становилась неработоспособной из-за несогласованности сертификата рецептора на новом узле типа `hybrid` с сертификатом рецептора на старом узле типа `execution`.

  В новой версии платформы такой сценарий стал работоспособен.

- При использовании параметров `nginx_tls_protocols` (для явного задания версии протокола TLS) и `nginx_user_headers` (для задания специального заголовка `X-Custom-Header`) в файле инвентаря установщика сервис NGINX не мог стартовать в процессе установки контроллера.

  В новой версии утилиты развертывания платформы данный сценарий работоспособен.

- При отключении одного или нескольких узлов, выполняющих роль execution node в кластере, в журнале создавалось большое количество однотипных записей.

  В новой версии платформы эта неисправность устранена.

- В процессе работы утилита `aa-setup` может вносить некоторые изменения в исходный файл инвентаря. Это приводило к удалению всех комментариев.

  В новой версии платформы комментарии сохраняются.

- Ссылка на инструкцию по развертыванию платформы, которая появляется в окне терминала после установки пакета `astra-automation-setup`, была некорректной.

  Ссылка исправлена на URL, ведущий на *инструкцию по быстрому развертыванию платформы*.

- Ответ на запрос API `/api` в окне браузера содержал некорректную ссылку на документацию API (иконка в виде знака вопроса).

  Теперь она содержит правильный URL, указывающий на *описание API контроллера Astra Automation*.

- Исправлены ошибки в коллекциях `astra.ald_pro`, `astra.freeipa` и `astra.postgresql`, негативно влиявшие на установку соответствующих продуктов с использованием этих коллекций.

Сценарии применения Astra Automation

Полностью обновлен демонстрационный проект aac-samples[285] путем замены сценариев на новые, описывающие следующие возможности:

- настройка и обновление ОС Astra Linux;
- настройка сетевого оборудования;
- создание и удаление виртуальных машин в Yandex Cloud;
- развертывание продуктов ПАО Группа Астра.

[285] https://source.astragroup.ru/aa-gca/AA/aac-samples

Обновление платформы до версии 1.1

Переход от одной из предыдущих версий к версии `1.1` необходимо производить, используя возможности предыдущей версии:

- Для перехода с версии `1.0-upd2` и более поздних версий 1.0 на `1.1` выполните следующую команду на *установочном узле*:

```
sudo ./aa-setup --upgrade
```

- Для перехода с версии `1.0-upd1` на `1.1` выполните следующую команду на установочном узле:

```
sudo ./aa-setup --upgrade --product-version 1.1
```

- Переход с версии `1.0` на `1.1` необходимо выполнять так же, как и при *переходе с 1.0 на 1.0-upd1*.

История обновлений стабильной версии

Следующая таблица содержит перечень обновлений в обратном хронологическом порядке и основные данные по ним.

Версия	Дата выпуска	Обновление характеристик (главное)	Исправления (главные)
1.1-upd1	21.10.2024	Первый выпуск Private Automation Hub.	Устранена ошибка, препятствующая передаче и отображению переменных, заданных в шаблоне, во вкладке **Подробности** (Details) выполненного задания.
1.1	19.08.2024	Выпуск утилит Ansible Navigator и Ansible Builder в составе CDK. Добавлен переключатель языка в графическом интерфейсе пользователя контроллера.	Устранена ошибка передачи данных в playbook при использовании в задании шага **prompt on launch**. Исправлена обработка параметра `unified_job_template` в API.

1.1-upd1

Дата выпуска: 21.10.2024
Тип выпуска: обновление стабильной версии

Состав платформы и совместимость с версиями операционной системы представлены в *таблице*.

Главное

Следующие обновления представляются наиболее важными:

- К платформе добавлен важный компонент – Private Automation Hub.
- Запуск заданий на основе событий с использованием механизма WebHooks теперь также поддерживает *GitFlic*.

Новые возможности и улучшения

Новая версия содержит несколько новых возможностей и улучшений существующих характеристик.

Private Automation Hub

Выпущен новый компонент платформы – Private Automation Hub – на основе свободно распространяемого кода Pulp Project[286].

Этот компонент существенно расширяет сферу применения платформы, благодаря следующим возможностям:

- Управление через *графический web-интерфейс* и *API*.
- Распространение важнейших компонентов инфраструктурного кода, которые могут быть загружены из других реестров, включая Automation Hub, Ansible Galaxy, или созданы самостоятельно:
  - Коллекции Ansible – управляющий контент для использования в различных сценариях автоматизации.
  - *Образы среды исполнения*.

Обновления коллекций Ansible

Произошли следующие изменения в составе коллекций, распространяемых через Automation Hub:

- Добавлена новая коллекция `astra.repo_mirror`, предназначенная для следующих целей:
  - развертывание deb-репозиториев из файлов ISO;
  - создание зеркалированных репозиториев.
- Обновлена коллекция `astra.aa-controller`:
  - добавлена возможность развертывания Private Automation Hub;
  - добавлены модули управления Automation Controller.
- Обновлена коллекция `astra.keepalived`:
  - улучшено масштабирование сервиса;
  - оптимизирована установка коллекции.
- Обновлена коллекция `astra.nfs`:
  - добавлены возможности тонкой настройки;
  - добавлена возможность управления правами на каталоги.
- Обновлена коллекция `astra.nginx`:
  - добавлены опции SSL;

[286] https://pulpproject.org/

  - удалены устаревшие директивы.
- Обновлена коллекция `astra.rupost`:
  - добавлена поддержка новой версии RuPost 3.0.1.
- Обновлена коллекция `astra.termidesk`:
  - добавлена поддержка новой версии Termidesk 5.0.0;
  - добавлена роль шлюза;
  - добавлены модули для управления продуктом.
- Обновлена коллекция `astra.ald_pro`:
  - переработана роль `replica`;
  - коллекция переведена в статус сертифицированной.

Обновления CDK

Произведены следующие изменения:

- Установка утилит Ansible (Ansible Navigator, Ansible Builder, Ansible Molecule, and Ansible Lint) теперь возможна в условиях отсутствия связи с интернетом:
  - Эта возможность предназначена для разработчиков контента и пользователей утилит командной строки.
  - Позволяет устанавливать утилиты в закрытом контуре, когда нет доступа к репозиторию Astra Automation, размещенному в интернете.
  - Создан пакет (offline bundle) для установки утилит локально без доступа в интернет.
  - Пакет доступен в личном кабинете клиента.
- Утилиты Ansible Lint и Ansible Molecule переведены из состояния *technical preview* в состояние полноценной технической поддержки.

Обновления в Automation Controller

В контроллере произведены следующие изменения:

- Добавлена возможность обрабатывать входные запросы от перехватчиков событий webhook системы контроля версий GitFlic:
  - Предназначена для команд devops, интегрирующих различные системы.
  - Позволяет автоматизировать обработку событий в GitFlic средствами Astra Automation.
  - Добавлена функция обработки запросов от GitFlic в приемнике запросов (receiver) контроллера.
  - Новая возможность доступна сразу после обновления контроллера.

Исправление ошибок

В новой версии исправлены следующие ошибки:

- При удалении компонентов платформы с помощью утилиты `aa-setup` кроме компонентов платформы удалялись также некоторые другие компоненты программного обеспечения. Это поведение исправлено, так что команда `aa-setup -u` удаляет только компоненты платформы на узлах, отмеченных в инвентаре.
- Если при обновлении версии платформы в инвентаре был указан узел, который по каким-то причинам оказался недоступен, утилита `aa-setup` ошибочно интерпретировала это как отсутствие требуемого компонента платформы на этом узле и предлагала выполнить установку компонента. В новой версии утилита корректно определяет отсутствие узла и завершает обновление аварийно.
- В контроллере после выполнения задания по шаблону переменные, заданные в шаблоне, не отображались во вкладке **Подробности** (Details) выполненного задания. В новой версии они отображаются корректно.
- В контроллере невозможно было запланировать задания по расписанию. В новой версии расписания (schedules) создаются корректно.
- Исправлено несколько ошибок в коллекциях Ansible, распространяемых через Automation Hub:
  - В коллекциях `astra.aldpro`, `astra.ceph` и `astra.hardening` исправлена ошибка совместимости со средой исполнения `aa-base-ee`.
  - В коллекции `astra.grafana` и `astra.freeipa` исправлены ошибки в документации.
  - В коллекции `astra.yandexcloud` исправлены ошибки в обеспечении безопасности и в документации.
  - В коллекции `astra.iscsi` исправлены ошибки в работе инициатора.
  - В коллекции `astra.postgresql` исправлены ошибки при работе с базой данных PARSEC.
  - В коллекции `astra.rupost` исправлена ошибка установки сертификатов.

Обновление платформы до версии 1.1-upd1

Переход от одной из предыдущих версий к версии `1.1` необходимо производить, используя возможности предыдущей версии:

- Для перехода с версии `1.0-upd2` и более поздних версий на `1.1-upd1` выполните следующую команду на *установочном узле*:

```
sudo ./aa-setup --upgrade
```

- Для перехода с версии `1.0-upd1` на `1.1-upd1` выполните следующую команду на установочном узле:

```
sudo ./aa-setup --upgrade --product-version 1.1-upd1
```

- Переход с версии `1.0` на `1.1-upd1` необходимо выполнять так же, как и при *переходе с 1.0 на 1.0-upd1*.

17.7.2 Версия 1.0

Дата выпуска: 15.02.2024
Тип выпуска: стабильная версия

Примечание: Первый выпуск продукта.

Состав платформы и совместимость с версиями операционной системы представлены в *таблице*.

Основные характеристики

Astra Automation является комплексной платформой для автоматизации задач по развертыванию и управлению инфраструктур различной сложности.

Состав

Платформа Astra Automation состоит из следующих основных компонентов:

- Automation Hub – облачный сервис, предоставляющий доступ к коллекциям Ansible и модулям Terraform, составляющим основу инфраструктурного кода для выполнения задач Astra Automation.
- Automation Controller – контроллер (плоскость управления), предоставляющий удобный графический интерфейс и API для управления выполнением задач Astra Automation. Этот компонент наследует характеристики проекта AWX[287], адаптирует и развивает их для удовлетворения нужд заказчиков ПАО Группа Астра.
- Execution plane – сеть исполнительных узлов (плоскость исполнения), расширяющая возможности Automation Controller по масштабированию и повышению уровня защиты структуры управления. Этот компонент позволяет создавать распределенную систему управления большими и сверхбольшими инфраструктурами, разделенными по географическому или другим принципам.

Функциональные возможности

Платформа позволяет развертывать инфраструктуру заказчика, устанавливать и настраивать программное обеспечение согласно *инструкции*.

Развертывание платформы

Развертыванию на стороне заказчика подлежат следующие компоненты:

- Automation Controller можно установить в различных конфигурациях:
  - на одном узле для ознакомления и тестирования или для управления небольшой инфраструктурой;
  - в кластере, как основной рекомендуемый вариант.
- (Дополнительно) Компонент Execution plane необходим для создания распределенной структуры управления. Его устанавливают в едином процессе развертывания вместе с контроллером или при последующем обновлении.
- СУБД PostgreSQL необходимо установить на каком либо сервере в процессе развертывания платформы или использовать готовую базу данных.

[287] https://github.com/ansible/awx

Особенности установки и ограничения:

1. Установка первой версии платформы протестирована в Astra Linux Special Edition с использованием следующих версий:

Версия операционной системы	Версия ядра
1.7.4	5.15.0-generic
1.7.5	6.1.50-generic

2. При использовании Astra Linux Special Edition 1.7.5 необходимо исключить установку графической системы Fly и средств работы с графикой. Для развертывании кластера на виртуальных машинах рекомендуется использовать универсальные базовые образы Astra Linux от производителя[288], в названии которых отсутствует поле `gui`, например образ alse-vanilla-1.7.5-virtualbox-adv-mg12.1.2.ova[289] для VirtualBox.

   Примечание: Исправлено в обновлении *1.0-upd1*.

Преимущества

Astra Automation предоставляет ряд преимуществ по сравнению с решениями, доступными от свободно распространяемых продуктов:

- Безопасный проверенный код инструментария.
- Использование надежной операционной системы, оснащенной средствами защиты информации в соответствии с требованиями официальных регуляторов.
- Доступ к уникальному проверенному инфраструктурному коду, предназначенному для установки на операционной системе Astra Linux.
- Регулярные плановые обновления стабильных версий и внеплановые обновления текущих версий по запросам при обнаружении ошибок или уязвимостей.
- Резервное копирование и восстановление данных и настроек.
- Использование более совершенной системы журналирования, основанной на syslog-ng.
- Использование защищенного протокола SSL/TLS при работе с PostgreSQL.
- Документация на русском языке.
- Техническая поддержка.
- Обучение.

История обновлений стабильной версии

Следующая таблица содержит перечень обновлений в обратном хронологическом порядке и основные данные по ним.

[288] https://registry.astralinux.ru/latest/
[289] https://registry.astralinux.ru/images/alse/virtualbox/alse-vanilla-1.7.5-virtualbox-adv-mg12.1.2.ova

Версия	Дата выпуска	Обновление характеристик (главное)	Исправления (главные)
1.0-upd2	22.04.2024	Обновление кода из проекта AWX. Улучшена функция обновления версии платформы с помощью утилиты `aa-setup`.	Исключен излишний процесс сканирования всего репозитория при развертывании или обновлении платформы.
1.0-upd1	14.03.2024	Добавлена возможность обновления версии платформы с помощью утилиты `aa-setup`.	Исправлен процесс восстановления базы данных кластера, состоящего из узлов типа `hybrid`.
1.0	15.02.2024		

1.0-upd2

Дата выпуска: 22.04.2024
Тип выпуска: обновление стабильной версии

Состав платформы и совместимость с версиями операционной системы представлены в *таблице*.

Главное

Следующие обновления представляются наиболее важными:

- Automation Controller теперь базируется на AWX версии `23.7.0`.
- Улучшена функция обновления версии платформы с помощью утилиты `aa-setup`.
- Утилита `aa-setup` позволяет указать образ среды исполнения при развертывании или обновлении платформы.
- Устранено ограничение на количество узлов в кластере Astra Automation.
- Обновлена версия образа среды исполнения.

Новые возможности и улучшения

Следующие секции содержат описание новых возможностей и улучшений существующих характеристик.

Переход на новую версию проекта AWX

Automation Controller теперь базируется на новой версии upstream-проекта AWX. Произведен переход с версии 22.1.0[290] на версию 23.7.0[291]. Из большого количества обновлений наиболее заметными и значимыми являются следующие:

[290] https://github.com/ansible/awx/releases/tag/22.1.0
[291] https://github.com/ansible/awx/releases/tag/23.7.0

- В панели навигации пункт просмотра топологии доступен только для системных администраторов и аудиторов[292]. Ранее он был виден также для администраторов организаций, но, поскольку сама топология им недоступна, это приводило к появлению сообщения об ошибке.
- Обновлено расширение (plugin) credential_plugin[293] так, что теперь при работе с хранилищем vault поддерживается аутентификация через LDAP.
- Поддержка webhooks от Bitbucket Data Center[294] с отправлением обратно состояния соответствующего запроса.

Выполнен перевод на русский язык новых и обновленных строк в графическом интерфейсе контроллера.

Утилита aa-setup

Выпущена версия утилиты `1.0.2` в составе пакета `astra-automation-setup`, содержащая новые возможности:

- Улучшена функция обновления версии платформы:
  - Предназначено для специалистов по установке и обслуживанию Astra Automation.
  - Новая функция автоматически определяет текущую версию платформы на всех компонентах кластера с помощью файла `/etc/astra_automation.version`, присутствующего в каждом узле кластера.
  - Пользователь, у которого есть административные полномочия на узлы кластера с доступом по SSH (используя приватный и публичный ключи SSH), может запускать процесс обновления, указав аргумент `--upgrade`. Утилиту можно запускать с узла, прошедшего *процесс подготовки*. Ранее процесс обновления необходимо было запускать с того узла, который использовался для первоначального развертывания платформы.
  - Функция доступна сразу после обновления версии утилиты, но применить ее можно только с выпуском следующей версии платформы, то есть при переходе с версии `1.0-upd2` на `1.0-upd3`. Обновление на рассматриваемую версию подробно описано в секции *Обновление платформы до версии 1.0-upd2*.
- Утилита `aa-setup` позволяет указать образ среды исполнения, используемой по умолчанию для выполнения заданий:
  - Предназначено для специалистов по установке и обслуживанию Astra Automation.
  - Добавлен аргумент `--default-job-ee`, указывающий адрес требуемого образа.
  - Данная функция позволяет использовать требуемый образ среды исполнения вместо поставляемого в составе пакета.
  - Опция доступна сразу после установки или обновления утилиты.

  Пример запуска команды:

  ```
  sudo ./aa-setup --default-job-ee registry.astralinux.ru/aa/aa-base-ee:0.5.1
  ```

- Устранено ограничение на количество узлов в кластере Astra Automation:
  - Предназначено для специалистов по установке и обслуживанию Astra Automation.
  - Устранено ограничение на количество устанавливаемых узлов.
  - Позволяет создавать более сложную топологию платформы.
  - Доступно сразу после установки или обновления утилиты.
- Добавлена проверка корректности установки параметра `pg_host`:

[292] https://github.com/ansible/awx/pull/13905
[293] https://github.com/ansible/awx/pull/14654
[294] https://github.com/ansible/awx/pull/14674

  - Предназначено для специалистов по установке и обслуживанию Astra Automation.
  - Проверяет, что при отсутствии в описании инвентаря группы database (установка базы данных на локальном узле), параметру pg_host не назначено какое-либо значение. Допускается назначать адрес локального узла, то есть, 127.0.0.1. При нарушении этого условия утилита завершается аварийно с выводом сообщения об ошибке вида:

```
Параметр pg_host должен содержать пустое значение или '127.0.0.1' при␣
↪выполнении установки с внутренней базой данных (незаполненной секцией␣
↪[database]).
```

  - Эта функция повышает надежность развертывания платформы и сокращает время на выяснение возможных проблем.
  - Доступно сразу после установки или обновления утилиты.

- Утилита aa-setup позволяет удалить ранее установленные компоненты платформы:
  - Предназначено для специалистов по установке и обслуживанию Astra Automation.
  - Добавлена функция удаления, вызываемая с помощью аргумента -u или --uninstall. Она удаляет все данные, установленные в процессе развертывания Astra Automation на узлах, входящих в состав файла инвентаризации.
  - Новая функция упрощает процесс удаления компонентов платформы.
  - Доступна сразу после установки или обновления утилиты.

  Пример полного удаления всех компонентов Astra Automation из узлов, перечисленных в файле inventory:

```
aa-setup -i inventory -u --force-postgres-removal
```

Обновление спецификации API

Произведено обновление спецификации API на версию 1.0.2+23.7.0.

Образ среды исполнения

Образ среды исполнения (EE, Execution Environment) aa-base-ee, используемый в платформе по умолчанию, обновлен до 0.5.1:

- Предназначено для пользователей Astra Automation и специалистов по установке и настройке.
- Выполнены следующие обновления:
  - В основу положен образ Astra Linux Special Edition 1.7.5uu1 (registry.astralinux.ru/library/alse:1.7.5uu1-mg12.5.0).
  - Python и необходимые пакеты установлены из публичного репозитория Astra Automation.
  - Обновлены пакеты ansible и ansible-core.
  - Terraform заменен на OpenTofu[295].
- Новая версия расширяет возможности EE по использованию необходимого программного обеспечения и новых возможностей Ansible. Она также позволяет отказаться от Terraform, на использование которого наложены ограничения от производителя.
- Образ доступен с момента выпуска данной версии Astra Automation.

[295] https://opentofu.org/

Обновления коллекций Ansible

Произведено обновление следующих коллекций:

- Коллекция `astra.ceph` обновлена до версии `2.1.5` для обеспечения совместимости с EE версии `0.5.x`.
- Коллекция `astra.ald_pro` обновлена до версии `0.8.0` для поддержания ALD Pro версии `2.3.0`.
- Коллекция `astra.postgresql` обновлена до версии `2.2.1` для совместимости с операционной системой Astra Linux Special Edition версии `1.8`.

Исправление ошибок

В новой версии исправлены следующие ошибки:

- Исключен излишний процесс сканирования версий платформы в репозитории, когда версия платформы явно указана в качестве аргумента при запуске утилиты `aa-setup`.
- Устранена неисправность резервного копирования базы данных (dump), выражавшаяся в том, что прерывание процесса dump по какой-либо причине приводило к невозможности повторного запуска этого процесса.
- В структуре LOG_AGGREGATOR_SYSLOGNG_DEBUG, используемой для записи строк в журнал, исправлено название сервиса с `rsyslogd` на `syslog-ng`.

Обновление платформы до версии 1.0-upd2

Обновление с помощью утилиты `aa-setup`, описанное ранее, доступно уже в этой версии, но применить новые возможности можно будет только при переходе на будущую версию, то есть на `1.0-upd3` и последующие.

Поэтому переход от одной из предыдущих версий необходимо производить, используя возможности той версии:

- Переход с версии `1.0` на `1.0-upd2` необходимо выполнять так же, как и при *переходе с 1.0 на 1.0-upd1*.
- Переход с версии `1.0-upd1` на `1.0-upd2` позволяет использовать утилиту `aa-setup` из установочного пакета для версии `1.0-upd1`:

```
sudo ./aa-setup --upgrade --product-version 1.0-upd2
```

Примечание: Есть временные ограничения:

- Вместе с аргументом `--upgrade` требуется использовать `--product-version` для явного указания версии Astra Automation.
- Обновление необходимо производить с того установочного узла, который использовался при первоначальном развертывании платформы. Вместо такого узла можно воспользоваться другим, на котором установлен пакет для развертывания Astra Automation и восстановлены необходимые настройки установочного узла.

При любом из перечисленных вариантов перехода необходимо использовать файл, описывающий действующий инвентарь платформы. По умолчанию он находится по адресу `/opt/rbta/aa/astra-automation-setup/inventory`.

1.0-upd1

Дата выпуска: 14.03.2024
Тип выпуска: обновление стабильной версии

Состав платформы и совместимость с версиями операционной системы представлены в *таблице*.

Главное

Следующие обновления представляются наиболее важными:

- Добавлена функция автоматического обновления версии платформы с помощью утилиты `aa-setup`.
- Исправлен процесс восстановления базы данных кластера, состоящего из узлов типа `hybrid`.
- Исправлен процесс записи сообщений в журнал в формате JSON.

Новые возможности и улучшения

Следующие секции содержат описание новых возможностей и улучшений существующих характеристик.

Утилита aa-setup

Выпущена версия утилиты 1.0.1 в составе пакета astra-automation-setup, которая содержит новые возможности:

- Утилиту `aa-setup` теперь можно использовать для *обновления версии платформы*:
  - Предназначено для специалистов по установке и обслуживанию Astra Automation.
  - Добавлена опция `--upgrade` для обновления версии кластера.
  - Новая функция автоматически определяет текущую версию и обновляет ее до новейшей версии, доступной в репозитории Astra Automation.
  - Функция доступна сразу после обновления утилиты, но применить ее можно только с выпуском следующей версии платформы, то есть `1.0-upd2`. Обновление на рассматриваемую версию подробно описано в секции *Обновление платформы до версии 1.0-upd1*.
- С помощью утилиты `aa-setup` можно узнать список версий платформы, доступных в репозиториях APT:
  - Предназначено для специалистов по установке и обслуживанию Astra Automation.
  - Добавлена опция `--check-releases`.
  - Данная опция позволяет инженерам узнать список версий платформы, доступных в главном репозитории (dl.astralinux.ru) или в репозитории, указанном с помощью аргумента `--repo-url`. Таким образом, они могут выбрать конкретную версию продукта для первичного развертывания или обновления.
  - Доступно сразу после установки или обновления утилиты`.

  Пример запуска команды:

  ```
  sudo ./aa-setup --check-releases
  ```

  Пример вывода:

```
INFO: Astra Automation Setup has been started
INFO:

Following Astra Automation releases available:
mvp, mvp-upd1, mvp-upd2, 1.0, 1.0-upd1

To upgrade to the latest version, use:
./aa-setup --upgrade

Information about Astra Automation releases:
https://docs.astra-automation.ru/latest/releases/

INFO: Available releases checked
```

Исправление ошибок

В новой версии исправлены следующие ошибки:

- Исправлен процесс восстановления базы данных кластера, состоящего из узлов типа `hybrid`. Ранее при восстановлении данные не передавались из архива в базу данных.
- Исправлен процесс записи сообщений в журнал в формате JSON. Ранее формат записи JSON был искажен. Это, в частности, приводило к ошибкам интерпретации записей с помощью внешних обработчиков журналов, например, `logstash`.
- Исправлен счетчик затраченного времени при обработке задания. Ранее, при определенных условиях, он мог начинать отсчет от некоторого отрицательного значения.
- Устранено ограничение на наличие графических средств в узлах, на которых устанавливаются компоненты кластера платформы.

Обновление платформы до версии 1.0-upd1

Обновление с помощью утилиты `aa-setup` доступно уже в этой версии, но применить эту функцию можно будет только при переходе на будущую версию, то есть на `1.0-upd2` и последующие.

Поэтому обновление с версии `1.0` до `1.0-upd1` требует особой подготовки и последовательности.

Подготовка

Необходимо обеспечить выполнение следующих условий:

- Для обновления используйте тот же узел, с помощью которого происходило развертывание платформы версии 1.0, без внесения каких-либо изменений на нем.
- Убедитесь, что на этом узле остался в неизменном виде каталог `/opt/rbta/aa/astra-automation-setup/`, в частности файл инвентаря, описывающий состав платформы:

  `/opt/rbta/aa/astra-automation-setup/inventory` – описание структуры платформы.

Обновление

Обновление с версии `1.0` до `1.0-upd1` выполняйте в следующем порядке:

1. Добавьте ссылку на репозиторий, содержащий файлы Astra Automation, в файл `/etc/apt/sources.list` или же в любой файл в каталоге `/etc/apt/sources.d/`:

```
deb https://dl.astralinux.ru/aa/aa-debs-for-alse-1.7 1.0-upd1 main
```

Примечание: Убедитесь, что нет конфликтов, то есть, что этот репозиторий не объявлен в каком-либо другом из перечисленных файлов.

2. Обновите индекс пакетов с помощью следующей команды:

```
sudo apt update
```

3. Обновите пакет `astra-automation-setup` с помощью следующей команды:

```
sudo apt install --only-upgrade astra-automation-setup
```

4. Запустите процесс обновления таким же образом, как вы изначально развертывали платформу, то есть одним из следующих способов:
   - Запуск с использованием привилегий пользователя root:

   ```
   sudo ./aa-setup
   ```

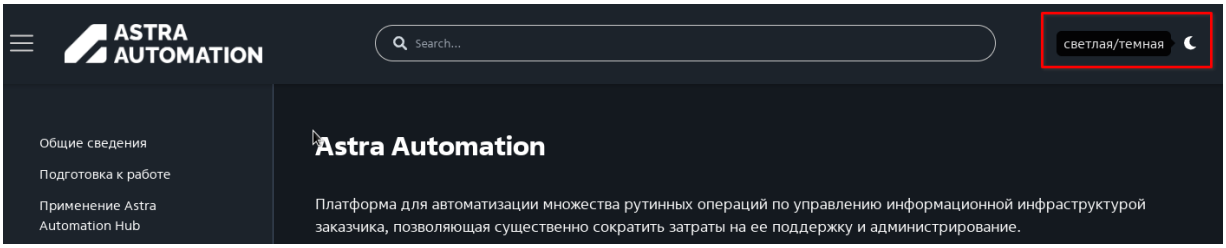
   - Запуск с привилегиями обычного пользователя:

   ```
   ./aa-setup --log-path=<log_path>
   ```

Изменения в документации

Добавлены новые возможности в документации:

- Добавлена поддержка темной темы:



- Изменен стиль для более заметного разграничения заголовков второго и третьего уровней.

Алфавитный указатель

Non-alphabetical

<command>
    command line option, 583
<playbook>
    command line option, 585
<playbook_artifact>
    command line option, 585
<plugin>
    command line option, 582
[<lintables>]
    command line option, 584

Г

ГИС, **742**

П

Предложения об улучшениях Python
    PEP 8, 599
    PEP 508, 553, 631, 632

С

СЗИ, **742**
СЗИ ALSE, **743**

У

Установочный узел, **743**

A

--all
    command line option, 610
ansible_connection
    command line option, 611
ansible_connection_options
    command line option, 611
ANSIBLE_LINT_CUSTOM_RULESDIR
    command line option, 601
ANSIBLE_LINT_IGNORE_FILE
    command line option, 601
ANSIBLE_LINT_NODEPS
    command line option, 601
ANSIBLE_LINT_SKIP_SCHEMA_UPDATE
    command line option, 601
ANSIBLE_LINT_WRITE_TMP
    command line option, 601
--ansible-runner-artifact-dir
    command line option, 578
--ansible-runner-rotate-artifacts-count
    command line option, 578
--ansible-runner-timeout
    command line option, 578
--ansible-runner-write-job-events
    command line option, 578

B

-b
    command line option, 53
--backup
    command line option, 53
--base-config
    command line option, 606
--build-arg
    command line option, 596
--bwd
    command line option, 581

C

-c
    command line option, 582, 597, 601
--c
    command line option, 606
--cdcp
    command line option, 578
--ce
    command line option, 578
--check-releases
    command line option, 55
CIDR, **741**
CLI, **741**
Cloud-Init, **741**
--co
    command line option, 579
--collection-doc-cache-path
    command line option, 578
command
    command line option, 612
command line option
    <command>, 583
    <playbook>, 585
    <playbook_artifact>, 585
    <plugin>, 582
    [<lintables>], 584
    --all, 610

ansible_connection, 611
ansible_connection_options, 611
ANSIBLE_LINT_CUSTOM_RULESDIR, 601
ANSIBLE_LINT_IGNORE_FILE, 601
ANSIBLE_LINT_NODEPS, 601
ANSIBLE_LINT_SKIP_SCHEMA_UPDATE, 601
ANSIBLE_LINT_WRITE_TMP, 601
--ansible-runner-artifact-dir, 578
--ansible-runner-rotate-artifacts-count, 578
--ansible-runner-timeout, 578
--ansible-runner-write-job-events, 578
-b, 53
--backup, 53
--base-config, 606
--build-arg, 596
--bwd, 581
-c, 582, 597, 601
--c, 606
--cdcp, 578
--ce, 578
--check-releases, 55
--co, 579
--collection-doc-cache-path, 578
command, 612
--component, 54
--conf.format, 395
--conf.host, 395
--config, 582
--config-file, 601
--conf.insecure, 395
--conf.password, 395
--conf.token, 395
--conf.username, 395
--container-engine, 578
--container-keyring, 597
--container-options, 579
--container-policy, 597
--container-runtime, 596
--context, 597
-d, 54, 583, 610
--dc, 579
--debug, 54, 606
--default-job-ee, 54
--dependency-name, 610
--destroy, 610
--details, 583
directory, 613
--display-color, 579
--driver-name, 610
-e, 606
--ecmd, 579
--econ, 579
--editor-command, 579
--editor-console, 579
--ee, 579
--eei, 579
--eev, 579
--effective, 586
enable_list, 603
enabled, 612, 613
--enable-list, 601
--enable-prompts, 585
--env-file, 606
--ep, 585
--exclude, 601
exclude_paths, 602
--exec-shell, 583
--execution-environment, 579
--execution-environment-image, 579
--execution-environment-volume-mounts, 579
--exshell, 583
extra_vars, 604
-f, 395, 597, 599, 608, 609
--file, 597
---fix, 600
--fmt, 581, 583
folder, 598
--force, 609
--force-color, 601
--force-postgres-removal, 54
--force-yes, 54
--format, 581, 583, 599, 608
--galaxy-ignore-signature-status-codes, 597
--galaxy-keyring, 597
--galaxy-required-valid-signature-count, 597
--generate-ignore, 600
--generate-key, 53
groups, 613
--gs, 586
-h, 53, 394, 578, 596, 598, 599, 610
--hb, 581
--hc, 582
--hd, 582
--help, 53, 394, 578, 596, 598, 599, 606
--help-builder, 581
--help-config, 582
--help-doc, 582
--help-inventory, 584
--help-playbook, 585
--hi, 584
--host, 610
--hp, 585
-i, 53, 584, 585, 601
--ic, 584
ignore-certs, 612
ignore-errors, 612
--ignore-file, 601
image, 613
--inventory, 53, 584, 585
--inventory-column, 584
-k, 53, 395
kinds, 605
-L, 599
--la, 579
--lf, 580
--lic, 584
lintables, 599

--lint-config, 584
--list-profiles, 599
--list-rules, 599
--list-tags, 599
--ll, 580
--log-append, 579
--log-file, 580
login_cmd_template, 611
--log-level, 580
--log-path, 53
loop_var_prefix, 603
-m, 580
managed, 611
max_block_depth, 605
mock_modules, 602
mock_roles, 603
--mode, 580
name, 611613
--no-cache, 596
--nocolor, 601
--no-debug, 606
--no-force, 609
offline, 604
--offline, 601
only_builtins_allow_collections, 605
only_builtins_allow_modules, 605
options, 611, 612
--osc4, 580
--output-filename, 597
-P, 599
-p, 53, 600, 610
--pa, 580
--pae, 585
--parallel, 610
parseable, 602
--parseable, 600
--pas, 585
--pass-environment-variable, 580
--penv, 580
--plain, 54
--platform-name, 610
--playbook-artifact-enable, 585
--playbook-artifact-save-as, 585
--pp, 580
privileged, 613
--product-version, 54
profile, 602
--profile, 600
--project-dir, 600
--provisioner-name, 610
--prune-images, 596
--pull-arguments, 580
--pull-policy, 580
-q, 600
quiet, 602
-R, 600
-r, 54, 600
--rac, 578
--rad, 578
--repo-url, 54
requirements-file, 612
--restore, 54
role-file, 612
--rt, 578
rulesdir, 603
--rules-dir, 600
--rwje, 578
-s, 600, 610
--sample, 586
--sanitize, 598
sarif_file, 602
--sarif-file, 600
scenario, 610
--scenario-name, 610
--schema, 586
--se, 586
--senv, 581
--set-environment-variable, 581
--show-relpath, 600
skip_action_validation, 604
skip_list, 603
--skip-list, 600
--so, 586
--sources, 586
--squash, 597
--ss, 586
strict, 602
--strict, 600
supported_ansible_also, 605
-T, 599
-t, 582, 596, 600
--tag, 596
tags, 604
--tags, 600
task_name_prefix, 605
--time-zone, 581
--type, 582
--tz, 581
-u, 54
--uninstall, 54
--upgrade, 54
use_default_rules, 603
--user-bindep, 598
--user-pip, 598
-v, 54, 395, 597, 598, 600, 606
var_naming_pattern, 603
--verbose, 395, 606
verbosity, 602
--verbosity, 597, 598
--version, 54, 578, 601, 606
-w, 600
warn_list, 604
--warn-list, 600
--workdir, 581
write_list, 604
--write-bindep, 598
--write-pip, 598
-x, 600
-y, 54
--component
command line option, 54
--conf.format

command line option, 395
--conf.host
command line option, 395
--config
command line option, 582
--config-file
command line option, 601
--conf.insecure
command line option, 395
--conf.password
command line option, 395
--conf.token
command line option, 395
--conf.username
command line option, 395
--container-engine
command line option, 578
--container-keyring
command line option, 597
--container-options
command line option, 579
--container-policy
command line option, 597
--container-runtime
command line option, 596
--context
command line option, 597

D

-d
command line option, 54, 583, 610
DAG, **741**
--dc
command line option, 579
--debug
command line option, 54, 606
--default-job-ee
command line option, 54
--dependency-name
command line option, 610
--destroy
command line option, 610
--details
command line option, 583
directory
command line option, 613
--display-color
command line option, 579
--driver-name
command line option, 610

E

-e
command line option, 606
--ecmd
command line option, 579
--econ
command line option, 579
--editor-command
command line option, 579
--editor-console
command line option, 579
EE, **741**
--ee
command line option, 579
--eei
command line option, 579
--eev
command line option, 579
--effective
command line option, 586
enable_list
command line option, 603
enabled
command line option, 612, 613
--enable-list
command line option, 601
--enable-prompts
command line option, 585
--env-file
command line option, 606
--ep
command line option, 585
--exclude
command line option, 601
exclude_paths
command line option, 602
--exec-shell
command line option, 583
--execution-environment
command line option, 579
--execution-environment-image
command line option, 579
--execution-environment-volume-mounts
command line option, 579
--exshell
command line option, 583
extra_vars
command line option, 604

F

-f
command line option, 395, 597, 599, 608, 609
--file
command line option, 597
---fix
command line option, 600
--fmt
command line option, 581, 583
folder
command line option, 598
--force
command line option, 609
--force-color
command line option, 601
--force-postgres-removal
command line option, 54
--force-yes
command line option, 54
--format

command line option, 581, 583, 599, 608
FQCN, **741**
FQDN, **741**

G

--galaxy-ignore-signature-status-codes
command line option, 597
--galaxy-keyring
command line option, 597
--galaxy-required-valid-signature-count
command line option, 597
--generate-ignore
command line option, 600
--generate-key
command line option, 53
groups
command line option, 613
--gs
command line option, 586

H

-h
command line option, 53, 394, 578, 596, 598, 599, 610
HA, **741**
--hb
command line option, 581
--hc
command line option, 582
HCL, **741**
--hd
command line option, 582
--help
command line option, 53, 394, 578, 596, 598, 599, 606
--help-builder
command line option, 581
--help-config
command line option, 582
--help-doc
command line option, 582
--help-inventory
command line option, 584
--help-playbook
command line option, 585
--hi
command line option, 584
--host
command line option, 610
--hp
command line option, 585

I

-i
command line option, 53, 584, 585, 601
IaC, **742**
--ic
command line option, 584
ignore-certs
command line option, 612
ignore-errors
command line option, 612
--ignore-file
command line option, 601
image
command line option, 613
--inventory
command line option, 53, 584, 585
--inventory-column
command line option, 584
IPA, **742**

K

-k
command line option, 53, 395
kinds
command line option, 605

L

-L
command line option, 599
--la
command line option, 579
--lf
command line option, 580
--lic
command line option, 584
lintables
command line option, 599
--lint-config
command line option, 584
--list-profiles
command line option, 599
--list-rules
command line option, 599
--list-tags
command line option, 599
--ll
command line option, 580
--log-append
command line option, 579
--log-file
command line option, 580
login_cmd_template
command line option, 611
--log-level
command line option, 580
--log-path
command line option, 53
loop_var_prefix
command line option, 603

M

-m
command line option, 580
managed
command line option, 611
max_block_depth
command line option, 605
mock_modules

command line option, 602
mock_roles
command line option, 603
--mode
command line option, 580

N

name
command line option, 611613
--no-cache
command line option, 596
--nocolor
command line option, 601
--no-debug
command line option, 606
--no-force
command line option, 609

O

OCFS2, **742**
offline
command line option, 604
--offline
command line option, 601
only_builtins_allow_collections
command line option, 605
only_builtins_allow_modules
command line option, 605
options
command line option, 611, 612
--osc4
command line option, 580
--output-filename
command line option, 597

P

-P
command line option, 599
-p
command line option, 53, 600, 610
--pa
command line option, 580
--pae
command line option, 585
--parallel
command line option, 610
parseable
command line option, 602
--parseable
command line option, 600
--pas
command line option, 585
--pass-environment-variable
command line option, 580
--penv
command line option, 580
--plain
command line option, 54
--platform-name
command line option, 610
--playbook-artifact-enable
command line option, 585
--playbook-artifact-save-as
command line option, 585
--pp
command line option, 580
privileged
command line option, 613
--product-version
command line option, 54
profile
command line option, 602
--profile
command line option, 600
--project-dir
command line option, 600
--provisioner-name
command line option, 610
--prune-images
command line option, 596
--pull-arguments
command line option, 580
--pull-policy
command line option, 580

Q

-q
command line option, 600
quiet
command line option, 602

R

-R
command line option, 600
-r
command line option, 54, 600
--rac
command line option, 578
--rad
command line option, 578
RBAC, **742**
--repo-url
command line option, 54
requirements-file
command line option, 612
--restore
command line option, 54
RFC
RFC 4517, 300
RFC 7617, 314
role-file
command line option, 612
--rt
command line option, 578
rulesdir
command line option, 603
--rules-dir
command line option, 600
--rwje
command line option, 578

S

-s
    command line option, 600, 610
--sample
    command line option, 586
--sanitize
    command line option, 598
sarif_file
    command line option, 602
--sarif-file
    command line option, 600
scenario
    command line option, 610
--scenario-name
    command line option, 610
--schema
    command line option, 586
--se
    command line option, 586
--senv
    command line option, 581
--set-environment-variable
    command line option, 581
--show-relpath
    command line option, 600
skip_action_validation
    command line option, 604
skip_list
    command line option, 603
--skip-list
    command line option, 600
--so
    command line option, 586
--sources
    command line option, 586
--squash
    command line option, 597
--ss
    command line option, 586
SSO, **742**
strict
    command line option, 602
--strict
    command line option, 600
supported_ansible_also
    command line option, 605

T

-T
    command line option, 599
-t
    command line option, 582, 596, 600
--tag
    command line option, 596
tags
    command line option, 604
--tags
    command line option, 600
task_name_prefix
    command line option, 605
--time-zone
    command line option, 581
TUI, **742**
--type
    command line option, 582
--tz
    command line option, 581

U

-u
    command line option, 54
--uninstall
    command line option, 54
--upgrade
    command line option, 54
use_default_rules
    command line option, 603
--user-bindep
    command line option, 598
--user-pip
    command line option, 598

V

-v
    command line option, 54, 395, 597, 598, 600, 606
var_naming_pattern
    command line option, 603
VCS, **742**
--verbose
    command line option, 395, 606
verbosity
    command line option, 602
--verbosity
    command line option, 597, 598
--version
    command line option, 54, 578, 601, 606

W

-w
    command line option, 600
warn_list
    command line option, 604
--warn-list
    command line option, 600
--workdir
    command line option, 581
write_list
    command line option, 604
--write-bindep
    command line option, 598
--write-pip
    command line option, 598

X

-x
    command line option, 600

Y

-y

command line option, 54
YC CLI, **742**