

Работа с Roles



Алексей
Метляков



Алексей Метляков

DevOps Engineer

OpenWay



Алексей Метляков



План занятия

1. [Что такое Role?](#)
2. [Для чего нужна Role?](#)
3. [Из чего состоит Role?](#)
4. [Где хранить Role?](#)
5. [Как создать Role?](#)
6. [Хранение и использование Role](#)
7. [Итоги](#)
8. [Домашнее задание](#)

Что такое Role?

Role – группа **tasks**, которая нацелена на выполнение действий, приводящих к единому результату.

- **Role** – выполняет список действий;
- Список может состоять из одного действия;
- Действия должны приводить к одному общему результату (например, установка и настройка nginx);
- Если в рамках действий **role** необходимо получить отдельный результат, стоит написать для этого отдельную **role**;
- Если ваш **playbook** использует **roles**, то использование **tasks** нежелательно, при этом можно использовать **pre_tasks** и **post_tasks**;
- Если ваш **playbook** использует **roles**, то для расширения функционала стоит писать **role**, даже если она состоит из атомарного действия.

Для чего нужна Role?

Использование **role** помогает достигнуть таких целей, как:

- Повышение читаемости кода;
- Создание верхнеуровневых **playbook**;
- Решение вопроса версионирования;
- Повышение универсализации.

Из чего состоит Role?

Стандартная общая структура директории с **role** состоит из:

- **Defaults** – директория, которая содержит в себе **yaml** файл с переменными по умолчанию, необходимыми для выполнения **role**;
- **Handlers** – директория с **yaml**, в котором содержатся все **handlers** от **role**;
- **Meta** – содержит информацию о самой роли для обеспечения работоспособности с **ansible-galaxy**;
- **Tasks** – директория содержит в себе **yaml** файлы с **plays** и **tasks**;
- **Templates** – директория с шаблонами конфигурационных файлов (в случае, если они нужны);
- **Tests** – тестовый **inventory** и **playbook**, которые обычно позволяют запустить роль на **localhost**;
- **Vars** – директория, которая содержит **yaml** файлы с остальными переменными.

Из чего состоит Role?

- В каждой из этих директорий, **ansible** ищет **main.yml**;
- В некоторых случаях можно добавлять свои **.yaml** файлы с наполнением и подключать их в **main.yml**;
- Роль можно дополнить своим собственным **module** или другим **plugin**, для этого его нужно добавить в директорию **library**.

Как создать Role?

Стартовый набор директорий можно получить при помощи команды: `ansible-galaxy role init <rolename>`

Команда создаст директорию **<rolename>** с базовым набором директорий и файлов, которые необходимо наполнить своими действиями

Как создать Role?

Так как мы свою **role** делаем из **playbook**, то нам нужно:

- Скопировать все **tasks** из **playbook** в файл **main.yml** директории **tasks**;
- Перенести все нужные для **tasks handlers** в файл **main.yml** директории **handlers**;
- Перенести все **templates** в соответствующую директорию **role**;
- Все переменные, необходимые для **tasks**, перенести в файлы **main.yml** директорий **defaults** и **vars**, в зависимости от внутренней логики.

Как создать Role?

Так как мы свою **role** делаем из **playbook**, то нам нужно:

- Расширить функционал **tasks**, возможно, использовать несколько уровней вложенности tasks-файлов;
- Заполнить **meta** информацию, она нужна для работы **ansible-galaxy**;
- Заполнить **README.md** файл, он нужен пользователям для использования.

Где хранить Role?

- Директория **roles** в плейбуке
- По пути **/etc/ansible/roles**
- По пути **/usr/share/ansible/roles**
- Кастомный путь

Для кастомного пути нужно:

- Переопределять путь в **ansible.cfg** в параметре **DEFAULT_ROLE_PATH**
- Создать переменную окружения **ANSIBLE_ROLES_PATH**

Хранение и использование role

Общедоступные роли хранятся по адресу galaxy.ansible.com

- Покрывают **~90%** необходимого функционала;
- Большинство распространяются по лицензии **MIT**;
- Есть возможность как стать **соавтором** роли, так и написать **собственную**.

Хранение и использование **role**

Традиционно, для версионирования и хранения **role** используется **git**. Для определения текущей версии используют семантический подход:

- **major** – добавление feature ломает совместимость с предыдущей версией;
- **minor** – добавление feature не ломает совместимость;
- **patch** – исправление багов, рефакторинг и прочие схожие.

Номер версии необходимо выставлять при помощи **git tag** или интерфейсов вашего инструмента для **git**.

Хранение и использование role

Чтобы использовать **role** в **playbook**, необходимо объявить её в файле **requirements.yml** и использовать команду **ansible-galaxy** для последующего скачивания нужной версии роли. Список команд:

- `ansible-galaxy install -r -requirements.yml`
- `ansible-galaxy install -r -requirements.yml`
- `ansible-galaxy install -r -requirements.yml --force`
- `ansible-galaxy install <rolename>`



Итоги

Итоги

- **Role** необходимо использовать в том случае, если снижается текущая читаемость **playbook**.
- Разделение **playbook** на отдельные **role** позволяет жить каждой логической единице независимыми жизненными циклами и обрастать дополнительным функционалом.
- Дополнительный функционал можно принудительно не использовать, оставаясь на старой версии **role**.
- Множество готовых **role** находятся в открытом доступе с лицензиями **MIT**.
- Собственные **role** можно хранить в частных хранилищах **git** и использовать напрямую оттуда.

Домашнее задание

Давайте посмотрим ваше [домашнее задание](#).

- Вопросы по домашней работе задавайте **в чате** мессенджера Slack.
- Задачи можно сдавать **по частям**.
- Зачёт по домашней работе проставляется после того, как **приняты все задачи**.

**Задавайте вопросы и
пишите отзыв о лекции!**

Алексей Метляков