нетология

Введение в Ansible

Алексей
Метляков

Алексей Метляков

DevOps Engineer

OpenWay

Алексей Метляков

План занятия

1. Ansible
2. Playbook
3. Role
4. Inventory
5. Group Vars
6. Vault
7. Templates
8. Facts
9. Collections
10. Первый запуск
11. Итоги
12. Домашнее задание

Что такое Ansible?

Что такое Ansible?

Ansible - комплекс ППО для управления инфраструктурной составляющей ваших систем и развёртыванием приложений.

- **Прост в использовании** - написан на python и YAML
- **Не требует установки агентов** - для подключения к удалённому хосту используется SSH
- **Идемпотентен** - независимо от того, сколько раз вы совершите запуск, результат будет идентичным
- **Легко расширяем** - любой дополнительный функционал можно реализовать на bash/python

Ansible

Основной концепт **ansible** заключается в следующем:

- Существует некоторая **control node** - хост с предустановленным **ansible**. С этой node мы будем исполнять инструкции на нужных нам хостах
- **Managed node -** хосты, на которых мы хотим получить результат исполнения инструкций
- **Inventory** - описание **managed node**

Ansible

Внутри **Ansible** существуют следующие понятия:

- Playbook
- Play
- Role
- Task
- Handlers
- Inventory
- Group vars
- Facts
- Templates
- Collections

Playbook

Ansible Playbook - набор plays, содержащих в себе roles и\или tasks, которые выполняются на указанных в inventory хостах с определёнными параметрами для каждого из них или для их групп.

Playbook описывается на языке **YAML.**

Пример содержимого одного **play** в **Playbook**:

```
---
      - name: Try run Vector  # Произвольное название play
        hosts: all # Перечисление хостов
        tasks: # Объявление списка tasks
            - name: Get Vector version # Произвольное имя для task
              ansible.builtin.command: vector --version # Что и как необходимо сделать
              register: is_installed # Запись результата в переменную is_installed
            - name: Get RPM # Произвольное имя для второй task
              ansible.builtin.get_url: # Объявление использования модуля get_url, ниже указание его
      параметров
                url: “https://package.timber.io/vector/{{ vector_version }}/vector.rpm”
                dest: “{{ ansible_user_dir }}/vector.rpm”
                mode: 0755
              when: # Условия при которых task будет выполняться
                - is_installed is failed
                - ansible_distribution == “CentOS”
```

Role

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

- **Role** - выполняет список действий
- Список может состоять из одного действия
- Role может быть написана самостоятельно или скопирована из galaxy при помощи команды **ansible-galaxy**
- **Role** хранят по умолчанию в директории **roles**, у каждой **role** своя директория внутри
- Пример использования **role** в рамках **play**:

```
---
      - name: Try run Vector  # Произвольное название play
        hosts: all # Перечисление хостов
        roles: # Объявление списка roles
            - vector # Вызов роли vector из директории с roles
```

Inventory

Inventory - директория с файлом или группой файлов, в которых описано на каких хостах необходимо выполнять действия.

- **Inventory** может быть описан в виде стандартного **host.ini** файла или при помощи **yaml** структуры
- Лучшей практикой является использование **yaml inventory**.
- Пример **inventory** файла:

```
---
      prod: # Группа серверов
        children:
          nginx:
            hosts:
              prod-ff-74669-02:
                ansible_host: 255.245.12.32
                ansible_user: prod
        children:
          application:
            hosts:
              174.96.45.23:
      test:
        children:
          nginx:
            hosts:
              localhost:
                ansible_connection: local
```

Group vars

Group vars - в общем понимании, файлы с переменными для групп хостов или для всех хостов, указанных в **inventory**.

- По умолчанию, хранятся в директории **group_vars**
- Определение переменных для всех хостов происходит в директории **all**
- Определение переменных для групп из **inventory** происходит в соответствующих им директориях
- Файлы с переменными могут называться, основываясь на внутренней логике **playbook**, сами имена имеют большую важность для пользователей

Приоритеты переменных

Переменные могут определяться и переопределяться на многих уровнях в **ansible**. Уровень приоритезации (от меньшего к большему) указаны ниже:

- Значения из командной строки **(-u username)**
- Значения по умолчанию из **roles**
- Значения из файла **inventory**
- Значения из файлов **group_vars/all**
- Значения из файлов **group_vars/{groupname}**
- Переменные из **play**
- Значения переменных **role** из **vars**
- Экстра-аргументы из командной строки **(-e “user=myuser”)**

Полный перечень приоритетов можно увидеть в официальной документации

Vault

Ansible Vault - инструмент, позволяющий зашифровать переменные (AES256), скрыв чувствительные данные от общего использования

- Удобно хранить параметры пользователей (логины, пароли)
- Можно шифровать как отдельные файлы, так и отдельные переменные
- Для использования шифрованных данных необходимо предоставлять пароль прямым вводом в консоль или в виде файла

Основные команды для управления **vault**:

- ansible-vault create <filename>
- ansible-vault view <filename>
- ansible vault edit <filename>

Templates

Templates - инструмент, позволяющий создать кастомизированный конфигурационный файл, на основе шаблона. Для шаблонизации используется **Jinja**

- Любой конфигурационный файл, даже без переменных внутри, может быть использован
- Шаблон должен иметь расширение **j2**

Шаблонизация напоминает использование форматирования строк:

```
‘Привет, {name}!’.format(name=’Мир’)
>>> Привет, Мир!
```

Facts

Facts - сбор информации об удалённом хосте, включая сетевую информацию, информацию о системе, информацию о пользователе, и прочее.

- Можно собирать данные об одном хосте и использовать эти данные для настройки другого хоста
- Факты собираются автоматически в начале проигрывания **play**
- **ansible <hostname> -m setup** - получить **facts** с **hostname**
- **Facts** хранятся в переменной **ansible_facts**
- Сбор **facts** можно принудительно выключить, вписав в **play** **gather_facts: no**

Collections

Collections - способ распространения контента **Ansible**. Включает в себя набор **roles**, **modules**, **playbooks**.

- Наименование состоит из **namespace.collections**
- Под **namespace** понимается, например, название компании или нечто объединяющее все **collections** для вашего **namespace**
- Под **collections** понимается само название коллекции
- Создаются и публикуются при помощи **ansible-galaxy**

Краткий итог

- **Ansible** - занимается автоматизацией рутины
- Весь процесс автоматизации описывается в **playbook**
- **Playbook** содержит информацию о том **что** и **где** необходимо сделать
- То, **что** необходимо сделать описывается в блоке **play**
- **Где** необходимо выполнять **play** написано в **inventory**
- **Play** состоит из перечислений **task** и **role**
- **Task** - атомарное действие над **host** из **inventory**
- **Role** - набор **tasks** вне **playbook**, которые выполняются для получение одного общего результата
- Абсолютно все сущности кастомизируются при помощи переменных
- Переменные **playbook** лучше всего хранить в **group vars**
- Переменные можно хранить и в других местах, существует приоритезация переменных
- Переменные можно шифровать при помощи **vault**
- Переменные можно подставлять в **templates** для создания конфигурационных файлов

Первый запуск

Подготовка к запуску

Для скачивания необходимо воспользоваться пакетными менеджерами:

- yum install ansbile
- apt install ansible
- pip3 install ansible --user

Инструкции по установке в разных версиях ОС

На текущий момент, стабильная версия - 2.10

Если уже установлен ansible, то перед установкой текущей версии, нужно удалить старую

Подготовка к запуску

В пакет входят:

- **ansible** - определение и запуск **playbook** из одного **task** на наборе **hosts**
- **ansible-playbook** - запуск полноценного **playbook**
- **ansible-vault** - шифрование хранилища методом AES256
- **ansible-galaxy** - скачивание **roles** и **collections**
- **ansible-lint** - используется для проверки синтаксиса
- **ansible-console** - консоль для запуска **tasks**
- **ansible-config** - просмотр и управление конфигурацией **ansible**
- **ansible-doc** - просмотр документации **modules**
- **ansible-inventory** - просмотр информации о **hosts** из **inventory**
- **ansible-pull** - скачивание **playbook** и запуск на **localhost**
- **ansible-test** - тестирование **collections**

Запуск команд

```
# ansible -m ping localhost #Сделаем ping на locahost
# ansible -m ping -i inventory.yml all #Сделаем ping на всех
хостах из inventory
# ansible -m ping -i inventory.yml <group_name> #Сделаем ping на
всех хостах группы <group_name>
# ansible-playbook site.yml -i inventory/test.yml #Запуск site на
хостах из test
# ansible-inventory -i inventory.yml --graph <group_name>
#Показать хосты группы
# ansible-inventory -i inventory.yml --list #Показать все
переменные всех хостов из inventory
# ansible-inventory -i inventory.yml --list <hostname> #Показать
все переменные хоста из inventory
# ansible-doc <plugin_name> #Показать документацию по плагину
# ansible-vault create <filename> #Создать новый зашифрованные
файл
# ansible-vault edit <filename> #Отредактировать зашифрованный
файл
# ansible-vault view <filename> #Просмотреть зашифрованный файл
# ansible-vault rekey <filename> #Поменять пароль у файла
# ansible-vault decrypt <filename> #Расшифровать файл
```

Структура директории с Playbook

```
group_vars/
    all/
        some_variable.yml
    <group_name>/
inventory/
    prod.yml
    test.yml
roles/
    <role_fodlers>/
site.yml
requirement.yml
```

Итоги

Итоги

- **Ansible** - занимается автоматизацией рутины
- Весь процесс автоматизации описывается в **playbook**
- **Playbook** содержит информацию о том **что** и **где** необходимо сделать
- **Role** - набор **tasks** вне **playbook**, которые выполняются для получение одного общего результата
- Переменные **playbook** лучше всего хранить в **group vars**
- Переменные можно хранить и в других местах, существует приоритезация переменных
- Переменные можно подставлять в **templates** для создания конфигурационных файлов
- **ansible** - определение и запуск **playbook** из одного **task** на наборе **hosts**
- **ansible-playbook** - запуск полноценного **playbook**
- **ansible-vault** - шифрование хранилища методом AES256

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**.

нетология

Задавайте вопросы и пишите отзыв о лекции!

Алексей Метляков

Алексей Метляков