



Онлайн образование

otus.ru

• REC

Проверить, идет ли запись

**Меня хорошо видно
&& слышно?**



Тема вебинара

Ansible



Дроздецкий Владимир

DevOps TeamLead at Magnit

Правила вебинара



Активно
участвуем



Off-topic обсуждаем
в Slack #канал группы
или #general



Задаем вопрос
в чат или ГОЛОСОМ



Вопросы вижу в чате,
могу ответить не сразу

Условные обозначения



Индивидуально



Время, необходимое
на активность



Пишем в чат



Говорим голосом



Документ



Ответьте себе или
задайте вопрос

IaC - infrastructure as code

Ansible



Ansible

Configuration Management

- Общий фреймворк
- Повторное использование кода
- Версионирование
- Совместная работа
- Самодокументирование
- Повторяемость (контроль за изменением состояния)

Ansible

Что еще есть?

- Ansible (AWX, Ansible Tower)
- Chef
- Puppet
- SaltStack

Ansible

| | Chef | Puppet | Ansible | SaltStack |
|-----------------------|---------------|---------------|----------------|------------------|
| Code | Open-source | Open-source | Open-source | Open-source |
| Cloud | All | All | All | All |
| Infrastructure | Mutable | Mutable | Mutable | Mutable |
| Language | Procedural | Declarative | Procedural | Declarative |
| Architecture | Client/Server | Client/Server | Client-only | Client/Server |

Ansible

- Текущая версия: 2.15.3
- Готовые модули (> 1000)
- Готовые роли на (> 11000 ролей)
- Быстрый старт
- Отсутствие агента и минимальные требования к хостам
- Идемпотентность

Ansible

```
→ practice-1 ansible-playbook site.yml

PLAY [Install default packages] *****

TASK [setup] *****
ok: [common]

TASK [default_packages : Install GnuPG v.2] *****
changed: [common]

PLAY RECAP *****
common          : ok=2    changed=1    unreachable=0    failed=0

→ practice-1 ansible-playbook site.yml

PLAY [Install default packages] *****

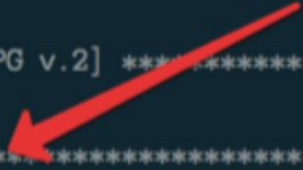
TASK [setup] *****
ok: [common]

TASK [default_packages : Install GnuPG v.2] *****
ok: [common]

PLAY RECAP *****
common          : ok=2    changed=0    unreachable=0    failed=0

→ practice-1 □
```

1 запуск



2 запуск



Ansible

- Декларативный язык (YAML)
- Поддержка ОС Windows, помимо *nix
- Управление внешними и внутренними инфраструктурными сервисами (облака, системы виртуализации, СХД)
- Особое внимание уделяется управлению сетевым оборудованием (Cisco, Arista, Juniper, F5 и другие)

Ansible Yaml

- Проще читать и редактировать
- Отступы для уровней вложенности
- Массивы и словари (hash, dictionary)
- Поддержка примитивов

Ansible

Для начала работы нам нужны:

- ОС Linux/Unix/MacOS
- Python 3
- SSH

Ansible особенности

- Нельзя определять зависимости между задачами (только handlers или "черная магия")
- Не хранит состояние между запусками сценариев
- Сетевые задержки (RTT) очень влияют на время выполнения сценариев
- Нет реагирования на события - соответственно нет автоматического применения сценариев, самовосстановления и т.п.
- Когда узлов много (несколько сотен) - надо вкладываться в оптимизацию Playbook или переходить на AWX

Ansible ansible.cfg

В файле конфигурации содержатся следующие настройки:

- Управление плагинами
- Переопределяемые параметры и указатели
- Параметры по умолчанию
- Расположение файла inventory или скрипта dynamic inventory
- Способ подключения

Ansible inventory

- Группировка и разделение хостов
- Вложенные группы Inventory файлов может быть несколько
- Inventory файл может быть динамическим и формироваться на основе каких-то внешних данных
- Предоставляет возможность формировать алиасы нашим хостам
- Предоставляет возможность переопределять параметры подключения, описанные в `ansible.cfg`

Ansible Modules

- Небольшие библиотеки для выполнения и отслеживания состояния задач
- Операции ОС
- Команды по управлению программной частью физических устройств
- Управление ресурсами внешних сервисов
- Основа для выполнения действий в Ansible

Ansible Modules

```
# Данный кусочек кода устанавливает пакет Git с помощью APT, выполнив apt update
перед этим
apt:
  name: git
  state: present
  update_cache: yes
```

Ansible Playbooks

Playbooks - это список сценариев для достижения целевого состояния системы с использованием модулей Ansible

Используются для:

- Описания процесса установки и настройки ПО
- Развертывания приложений на группах хостов и настройки Base Layer
- Управления внешними сервисами

Ansible Playbooks

- Один сценарий называется play
- В одном playbook может использоваться несколько play сценариев с различными условиями запуска (группы хостов, стратегия выполнения, набор стартовых переменных)

Ansible Variables

- Нужны для переиспользования в различных частях описания сценариев и определения отличий
- Дополняют циклы и операторы с условиями, для определения отличий на основе переменных
- Могут использоваться почти везде, в пределах инфраструктурного репозитория

Ansible Variables

YAML поддерживают словари и списки.

Их можно использовать для переменных (key:value параметры)

Пример использования переменной в playbook

```
- hosts: reddit_app
  vars:
    active: yes
    app_path: "{{ base_dir }}/reddit_app"
```

Вопросы? =)