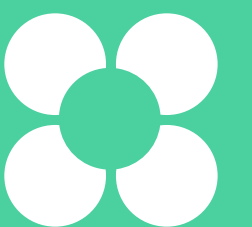


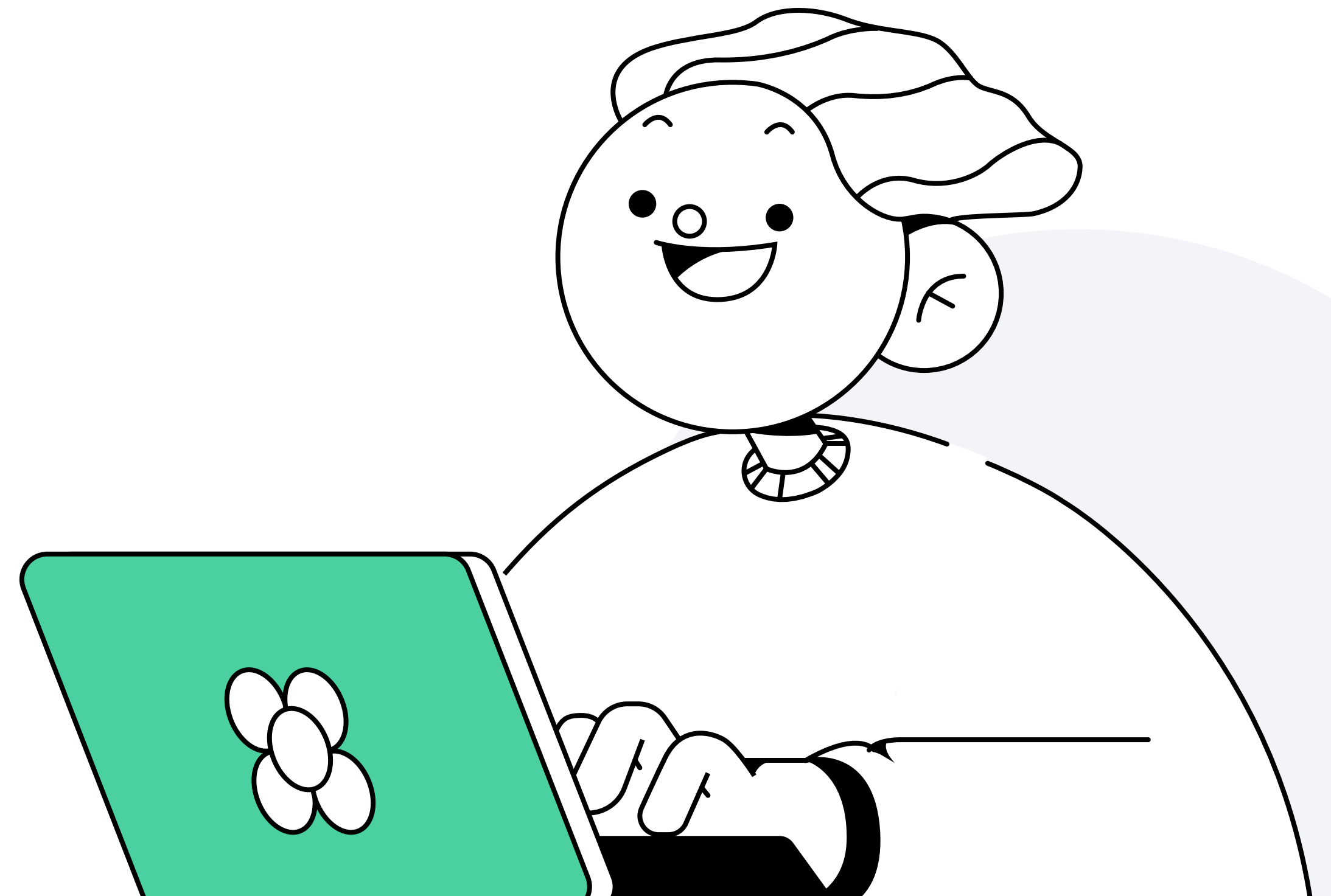
Введение в Ansible

Алексей Метляков
Tech lead DevOps engineer в OpenWay



План занятия

- 1 Ansible
- 2 Playbook
- 3 Inventory
- 4 Vars & Facts
- 5 Vault & Templates
- 6 Collections



Ansible

Алексей Метляков

Tech lead DevOps engineer в OpenWay

Что такое Ansible

Ansible – комплекс ППО для управления инфраструктурой составляющей ваших систем и развёртыванием приложений:

- прост в использовании
- не требует установки агентов
- идиempотентен
- легко расширяем



Ansible

Основной концепт **Ansible**:

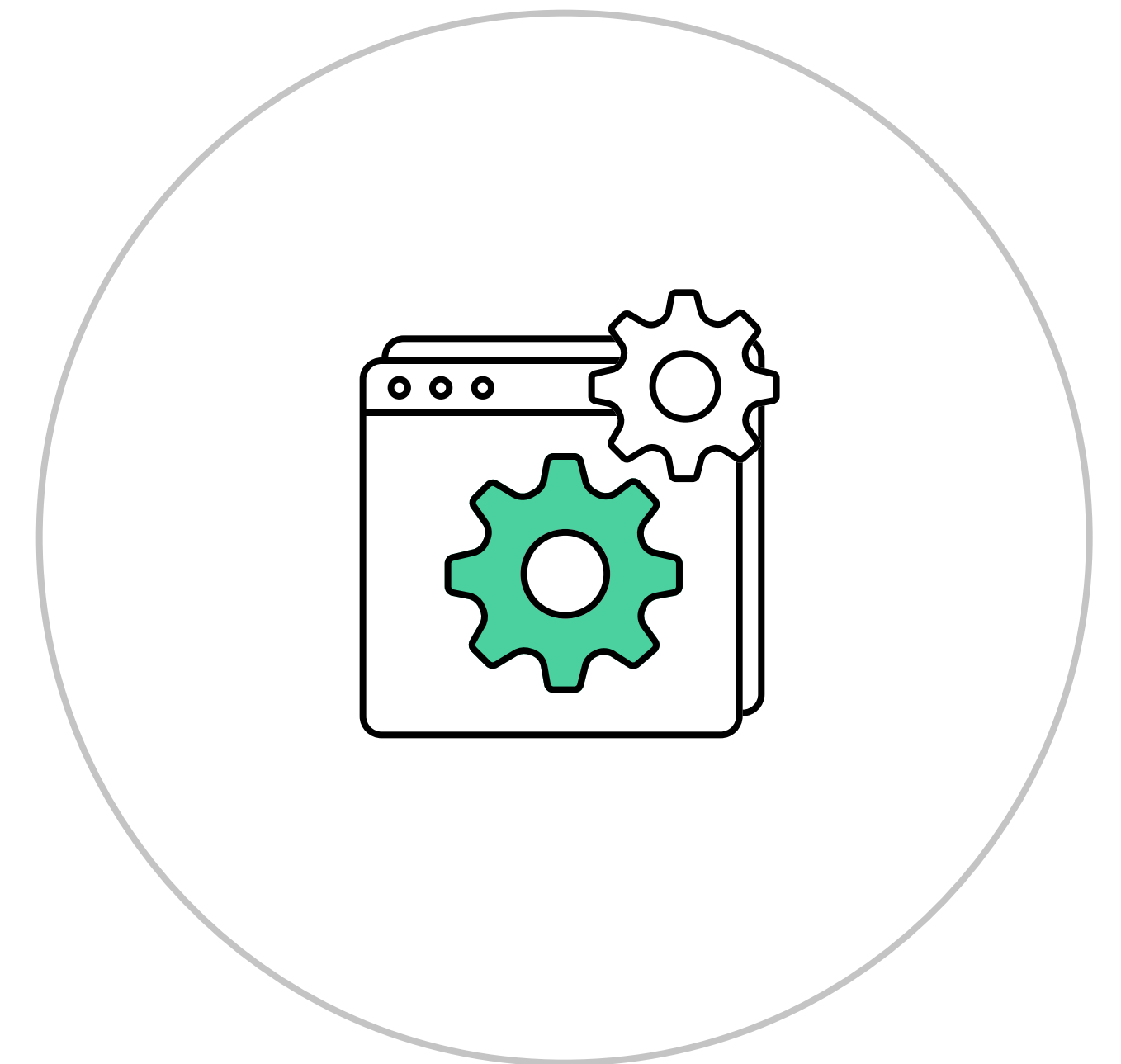
- Существует некоторая **control node** — хост с предустановленным **Ansible**
- **Managed node** — хосты, на которых нужно получить результат исполнения инструкций
- **Inventory** — описание **managed node**



Ansible

Внутри **Ansible** существуют следующие понятия:

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



Подготовка к запуску

Чтобы скачать, нужно воспользоваться пакетными менеджерами:

- `yum install ansible`
- `apt install ansible`
- `pip3 install ansible --user`

Инструкции по установке в разных версиях ОС.

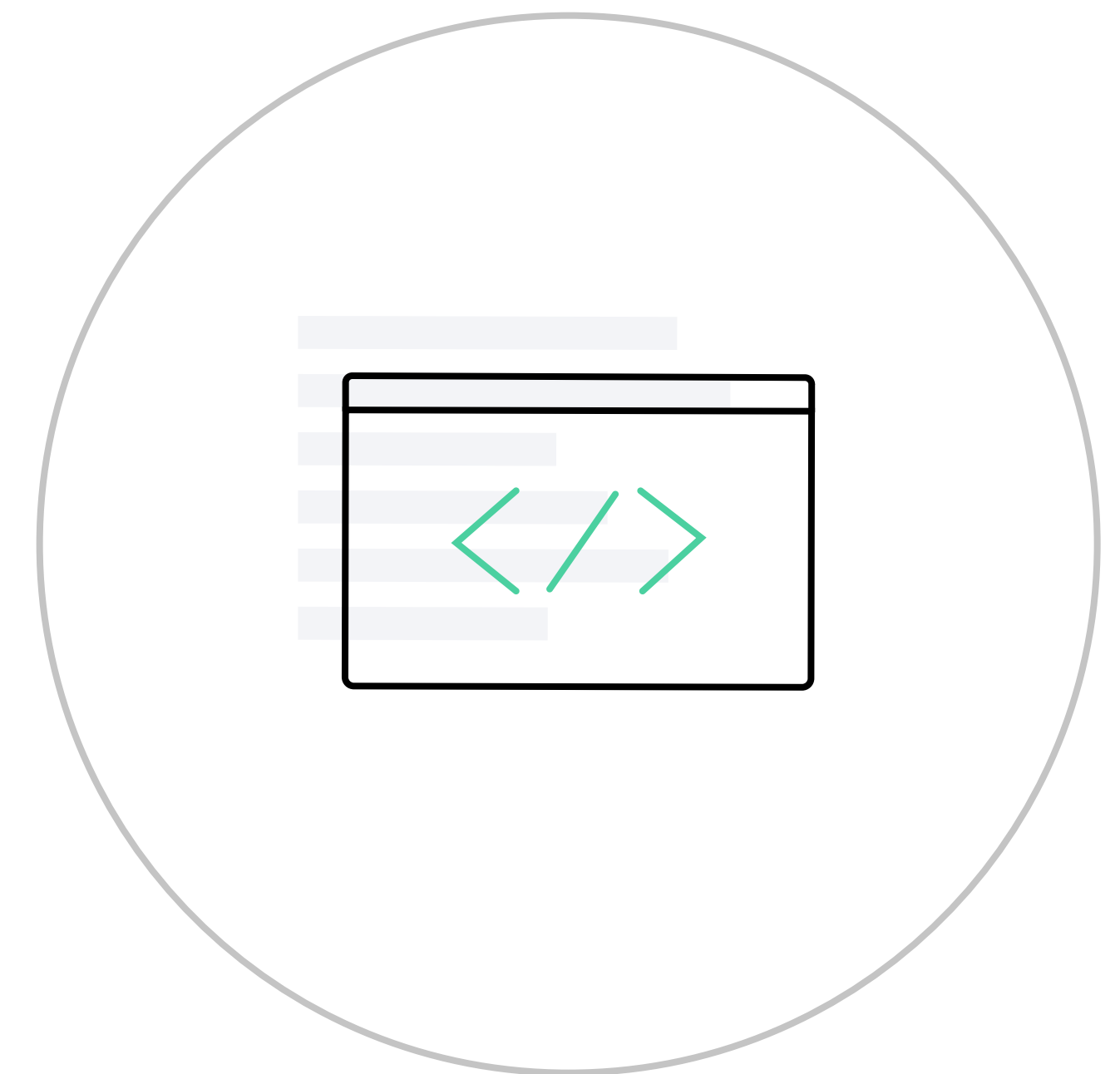
Если уже установлен Ansible, перед установкой версии выше 2.10, нужно удалить старую



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 **plugins**
- **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 localhost
# 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 зашифрованный 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
```

Playbook

Алексей Метляков

Tech lead DevOps engineer в OpenWay

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

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**, нацеленная 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
```

# Role

В рамках использования role в playbook достаточно знать несколько пунктов:

- **Role** можно скачивать через **ansible-galaxy**
- Какие **role** скачивать, лучше указать в **requirements.yml**
- **Role** может быть написана самостоятельно или скопирована из galaxy при помощи команды **ansible-galaxy**
- Чтобы использовать **role** в **playbook**, нужно применять следующий синтаксис:

```
roles: # Объявление списка roles
 - vector
 - java
```

# Inventory

Алексей Метляков

Tech lead DevOps engineer в OpenWay

# Inventory

**Inventory** — директория с файлом или группой файлов, в которых описано, на каких хостах нужно выполнять действия

- **Inventory** может быть описан в виде стандартного файла **host.ini** или при помощи **yaml**-структуры
- Лучшая практика — 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
```



# Vars & Facts

Алексей Метляков

Tech lead DevOps engineer в OpenWay

# 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 важны для пользователей

# 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: false**

# Приоритеты переменных

**Переменные** могут определяться и переопределяться на многих уровнях в **Ansible**.

Уровни приоритизации от меньшего к большему:

- 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**
- экстрааргументы из командной строки (**-e “user=myuser”**)

Полный перечень приоритетов можно увидеть в официальной [документации](#)

# Vault & Templates

Алексей Метляков

Tech lead DevOps engineer в OpenWay

# Vault

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

# Vault

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 используют **Jinja**.

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

Шаблонизация напоминает форматирование строк:

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



# Collections

Алексей Метляков

Tech lead DevOps engineer в OpenWay

# 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 **namespace**
- Под **collections** понимается само название коллекции
- Создаются и публикуются при помощи **ansible-galaxy**