


Проверить, идет ли запись!






Меня хорошо видно и слышно?

Ставьте , если все хорошо
Напишите в чат, если есть проблемы

Ansible: Экосистема и реестр



Константин Брюханов

 vk.com/deusops

 [deus_ops](https://t.me/deus_ops)

 [deusops](https://www.instagram.com/deusops)

Маршрут вебинара

Разбираем устройство Ansible



Знакомство с базовыми модулями



Написание плейбука



Рефлексия

Смысл | Зачем вам это уметь

1

Экономия времени благодаря переиспользованию кода

2

Получение ожидаемого результата от деплоя

3

Системный подход к решению инженерных задач

4

Глобальная автоматизация

The background of the slide features an aerial view of a city skyline, likely New York City, with numerous skyscrapers. The image is overlaid with a semi-transparent blue layer. A network diagram, consisting of interconnected nodes and lines, is visible across the blue layer, particularly concentrated around the central text.

Устройство Ansible

Устройство Ansible

Что такое

- Инструмент управления конфигурациями
- **Push-based** модель, отсутствие агентов
- Низкий порог входа
- Инфраструктурный код, который можно версионировать и тестировать
- Идемпотентные сценарии
- Язык YAML, шаблоны на Jinja
- Написан на Python



Устройство Ansible

Структура



- **ansible.cfg**
настройки для ansible
- **<playbookname>.yaml**
ansible-playbook, в котором описаны задачи
- **requirements.yaml**
файл перечнем ansible-ролей - отдельно вынесенных задач
- **inventories**
директория для инвентарей ansible
- **hosts**
файл с реестром хостов для управления

Устройство Ansible

Реестр Hosts



- Группировка и разделение хостов
- Вложенные группы
- Inventory файлов может быть несколько
- Inventory файл может быть динамическим
- Предоставляет возможность формировать алиасы нашим хостам
- Предоставляет возможность переопределять параметры подключения, описанные в **ansible.cfg**

Устройство Ansible

Реестр Hosts



```
[reddit_app]  
app01.myredditclone.internal
```

```
# <- Названия секций –  
#     это названия групп
```

```
[reddit_front]  
web01.myredditclone.internal  
web02.myredditclone.internal
```

```
# <- Хосты принадлежащие группе
```

```
[vote_sharding_db]  
mongo_db[01:20].myredditclone.internal
```

```
# <- Хосты можно задавать паттернами  
#     (и да, они могут повторяться)
```

```
[eu1_region:children]  
reddit_front  
reddit_app  
vote_sharding_db
```

```
# <- Группы могут состоять из групп
```


Устройство Ansible

Примеры задач



```
- hosts: app
  become: yes
  tasks:
    - name: Install the LibreOffice
      apt:
        name: libreoffice-core
        install_recommends: no
        state: latest
```

```
- name: Add systemd services
  template:
    src: "{{ inventory_dir }}/templates/systemd_service.j2"
    dest: "/etc/systemd/system/{{ service.name }}.service"
  with_items:
    - name: "celery"
      start: "/bin/sh -c '{{ django_app_user_home }}/local/celery_start.sh'"
      stop: "/bin/sh -c '{{ django_app_user_home }}/local/celery_stop.sh'"
  loop_control:
    loop_var: service

- name: Start systemd services
  service:
    name: "{{ item }}"
    state: started
  with_items:
    - celery
  changed_when: False
```

Устройство Ansible

Структура плейбука



```
- hosts: app
  become: yes
  roles:
    - redis
    - nodejs
    - djangoapp
  tasks:
    - name: Add configs
      copy:
        src: "{{ inventory_dir }}/files/"
        dest: "{{ django_app_user_home }}/local"
        owner: "{{ project_user }}"
        group: "{{ project_user }}"
        mode: 0750
```

- задаем список хостов для работы
- указываем дополнительные параметры
- перечисляем роли
- перечисляем отдельные задачи

Устройство Ansible

Условия выполнения

- Несколько Play сценариев выполняются в том порядке, в котором они описаны
- Play сценарии могут включать в себе несколько задач
- Задачи выполняются последовательно, по одной на всех хостах, попадающих под host условия



Устройство Ansible

Модули



- Небольшие библиотеки для выполнения и отслеживания состояния задач
- Основа для выполнения действий в Ansible
- Готовый модуль для работы с пакетным менеджером APT:

```
- name: Ensure nginx is installed.  
  apt:  
    name: nginx  
    state: present
```


The background of the slide features an aerial view of a city skyline, likely New York City, with numerous skyscrapers. The image is overlaid with a semi-transparent blue layer. A network diagram, consisting of interconnected nodes and lines, is visible across the blue layer, particularly concentrated around the central text.

Переменные Ansible

Переменные Ansible

Переменные



- 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

Переменные Ansible

Переменные



YAML поддерживают словари и списки. Их можно использовать для переменных (key:value параметры)

Пример использования переменной в playbook:

```
- hosts: app
  become: yes
  roles:
    - redis
    - nodejs
    - djangoapp
  tasks:
    - name: Add configs
      copy:
        src: "{{ inventory_dir }}/files/"
        dest: "{{ django_app_user_home }}/local"
        owner: "{{ project_user }}"
        group: "{{ project_user }}"
        mode: 0750
```

```
- name: Start systemd services
  service:
    name: "{{ item }}"
    state: started
  with_items:
    - celery
  changed_when: False

- name: Clean patch directories
  file:
    state: absent
    path: "{{ django_app_user_home }}/{{ project_user }}/{{ item }}"
    force: yes
  with_items:
    - media
    - staticfiles
```

Переменные Ansible

Registered переменные



В некоторых случаях, можно создавать переменную на основе выполненной команды или работы модуля.

Эти переменные будут доступны в пределах исполняемого хоста и могут отличаться между хостами

```
- name: Remove default nginx vhost config file (if configured).
  file:
    path: "{{ nginx_default_vhost_path }}"
    state: absent
  register: nginx_flag_reload
```

```
- name: Ensure nginx is started and enabled to start at boot.
  service:
    name: nginx
    state: reloaded
    enabled: yes
  when: nginx_flag_reload is defined and nginx_flag_reload
```

Переменные Ansible

Факты



В Ansible, помимо явно определенных разработчиком переменных, существуют read-only системные переменные, которые также известны как факты. За это отвечает модуль `setup`.

Посмотреть все факты, которые может собрать `setup` модуль из системы можно командой:

```
ansible localhost -m setup
```


Переменные Ansible

Факты



Пример использования:

```
- include: install.deb.yml
  when: ansible_os_family == 'Debian'      # <- Переменная из фактов

- name: Install Systemd files
  include: inst_systemd.yml
  when: ansible_service_mgr == 'systemd'   # <- Переменная из фактов
```

Переменные Ansible

Особые переменные



- Зарезервированные переменные самого Ansible: `hostvars`, `group_vars`, `inventory_hostname`, `playbook_dir` и тд
- **`hostvars`** и **`group_vars`** - особые переменная, позволяющая получить доступ к значениям переменных другого хоста или группы хостов



LIVE



Рефлексия



С какими основными мыслями и инсайтами уходите с вебинара?



Достигли ли вы цели вебинара?

Цели вебинара | Проверка достижения целей

1


Научились писать Ansible-плейбуки

2

Научились составлять шаблоны

3

Понимание базовых аспектов Ansible

The background of the image is an aerial photograph of a dense city skyline, likely New York City, with numerous skyscrapers. The image is overlaid with a semi-transparent blue layer that features a white network pattern of interconnected dots and lines, resembling a digital or social network. The text is centered within this blue layer.

Заполните, пожалуйста,
опрос о занятии по ссылке в чате

Спасибо за внимание!



Константин Брюханов



vk.com/deusops



[deus_ops](https://t.me/deus_ops)



[deusops](https://www.instagram.com/deusops)