

# Модули

- [Общая информация](#)
- [Системные модули и скрипты](#)
- [Git и pip](#)
- [Docker](#)

# Общая информация

Модули - созданный на python скрипт, упрощающий конкретную задачу.

Модули возвращают значения, название модуля должно быть под name. Есть общие переменные и специфичные для модулей

Справка по модулю

```
ansible-doc apt
```

Список модулей

```
ansible-doc -l
```

Пример проверки необходимости перезагрузки

```
- name: check if reboot is required
  become: yes
  become_method: sudo
  shell: "[ -f /var/run/reboot-required ]"
  failed_when: False
  register: reboot_required
  changed_when: reboot_required.rc == 0
  notify: reboot

handlers:
- name: reboot
  command: shutdown -r now "Ansible triggered reboot after system updated"
  async: 0
  poll: 0
  ignore_errors: true
```

Ссылки

[Информация на русском языке о модулях](#)

# Системные модули и скрипты

## Установка и/или проверка установки apt пакета

**Название модуля:** apt

### Переменные:

- name: ntp #имя проверяемого и устанавливаемого модуля
- state: present #состояние после завершения
- update\_cache: yes # обновлять ли кэш

```
- name: Install module
apt:
  name: ntp
  state: present
  update_cache: yes
```

## Добавить ключ стороннего репозитория

```
- name: Add Docker GPG key
apt_key:
  url: https://download.docker.com/linux/ubuntu/gpg
```

## Добавить сторонний репозиторий

```
- name: Add Docker repository
apt_repository:
  repo: deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable
```

## Обновить apt кэш

```
- name: Update apt cache
  apt: update_cache=yes
```

## Копирование файлов

### Копирование с локального на удаленный

```
- name: Copy from server to client
  copy:
    src: /home/user/file.txt
    dest: /home/setup/file.txt
    owner: foo
    group: foo
    mode: '0644'
```

### Копирование с удаленного на локальный

```
- name: Copy from client to server
  fetch:
    src: /var/log/access.log
    dest: /var/log/fetched
    flat: true удалить структуру родительских папок для файла
```

## Создание пользователя и группы

### Создание пользователя

```
- name: Create and/or check presence user
  user:
    name: install
    state: present
    shell: /bin/bash
    group: sudo
    system: yes
```

hidden: yes  
ssh\_key\_file: .ssh/id\_rsa  
expires: -1

## Создание группы

- name: Create check group  
group:  
  name: clustergroup  
  state: present  
  gid: 1040

## Управление сервисами (daemon)

- name: Update sysctl  
sysctl:  
  name: net.ipv4.ip\_forward  
  value: 1  
  sysctl\_set: yes  
  state: present  
  reload: yes

- name: Set or check service  
service:  
  name: ntp  
  state: started  
  enabled: yes

- name: Set check daemon starting  
systemd:  
  name: ntp  
  state: started  
  enabled: yes  
  masked: no  
  daemon\_reload: yes  
  register: systemd

cron:

## Скрипты и консольные команды

- name: Raw command

raw: echo "this was written by a raw Ansible module!!" >> ~/raw.txt

- name: Executing script

shell: ./shell\_script.sh >> ~/shell.txt

args:

chdir: /usr/local/

creates: ~/shell.txt

executable: /bin/csh

- name: Executing python script

script: ./shell\_script.py --some-arguments "42"

args:

creates: ~/shell.txt

executable: python

Для исполнения expect скриптов нужно сначала проверить и установить пакет expect

- name: Expect module

expect:

command: passwd user1

responses:

(?! )password: "Ju5tAn07herP@55w0rd":

# Git и pip

## Git

```
- name: Clone update repo
git:
  repo: https://github.com/ansible/ansible.git
  dest: /usr/local/ansible
  clone: yes
  update: yes
```

## Pip

Из официального репозитория:

```
- name: Install python package
pip:
  name: numpy
  version: 0.3
```

Из внешнего источника

```
- name: install a python library from a github
pip:
  name: https://github.com/jakubroztocil/httpie
```

# Docker

## Необходимые модули:

pip install 'docker-py>=1.7.0'

pip install 'docker-compose>=1.7.0'

ansible-container позволяет работать с docker без dockerfile

### Создание контейнера

- name: create a container

docker\_container:

name: debianlinux

image: debian:9

pull: yes

state: present

### Запуск контейнера

- name: start a container

docker\_container:

name: debianlinux

state: started

devices:

- "/dev/sda:/dev/xvda:rwm"

### Остановка контейнера

- name: stop a container

docker\_container:

name: debianlinux

state: stopped

### Удаление образа из локального хранилища



- name: remove a container image

docker\_image:

name: labimages/ubuntu

state: absent

tag: lab16

## Авторизация на docker hub

- name: login to DockerHub

docker\_login:

username: labuser1

password: "L@bp@55w0rd"

email: user1@lab.edu

## Скачать образ из docker hub

- name: pull a container image

docker\_image:

name: ubuntu:18.04

pull: yes

## Сохранить образ в docker hub

- name: push a container image to docker hub

docker\_image:

name: labimages/ubuntu

repository: labimages/ubuntu

tag: lab18

push: yes