

?????? ? ??????????

????????????? ansible ??????????????

### Запуск одного действия

ansible [host/group/all] [action parameters]

#### Действия

-a "shell command" shell command

-m exec\_name installed module

-i <filename> inventory file

--become все команды + sudo

--become-user user1 обязательно с become, переключается на выбранного пользователя

-f 1 использование одного потока. По умолчанию процессы параллельно для всех хостов.

#### Пример:

```
ansible multi -a "hostname"
```

### Запуск через ansible playbook

Список действий сохраняется в файле \*.yaml (playbook)

```
ansible-playbook filename.yaml
```

### Структура playbook файла

```
---
- hosts: [host ip | host group | all]
  remote_user: []
  become: true
  tasks:
    - [task 1]
    - [task n]
---
- hosts: [host ip | host group | all]
  remote_user: []
  become: true
  tasks:
    - [task 1]
    - [task n]
```

### Параметры задачи

gather\_facts: False Проводить сбор данных по умолчанию  
ignore\_errors: True Прекращать или нет работу при ошибке

## Переменные из модулей

```
---
- name: Check if host is up and running
  hosts: all
  become: false
  gather_facts: false

  tasks:
    - name: Ping the remote server
      ping:
        register: result
        until: result is succeeded
        retries: 3
        delay: 10

    - name: Get system uptime
      command: uptime
      when: result is succeeded
      register: uptime_result

    - name: Show system uptime
      debug:
        var: uptime_result.stdout
      when: result is succeeded
```

10: полученное значение из модуля ping сохраняется в переменной result

11: данное задание выполняется до тех пор, пока result не будет succeeded, но с ограничением кол-ва попыток (12) и задержкой между попытками (13)

17: задание 15 выполняется если result succeeded

18: полученное значение заносится в переменную uptime\_result

21: вывод значения uptime\_result

## Условия (handlers)

```
when: ansible_os_family == "Debian"
```

## Получить список фактов:

```
- name: Show facts available on the system
  ansible.builtin.debug:
  var: ansible_facts
```

В папке `/etc/ansible/facts.d/*.fact` файлы с локальными фактами о машине. Обращение к фактам в yaml:

```
ansible_local.<fact file name>.<fact group inside file>.<varname>
```

Создание новой переменной:

```
set_fact:<valname>=<...>
```

## Циклы (loops)

Цикл с объявленной переменной:

```
- name: use apt to install multiple apps
  apt:
    name: '{{ app }}'
    state: latest
    update_cache: yes
    vars:
      app:
        - htop
        - mc
        - nload
    become: yes
```

Цикл с необъявленной переменной:

```
- name: use apt to install multiple apps
  apt:
    name: '{{ item }}'
    state: latest
    update_cache: yes
  loop:
    - htop
    - mc
```

```
- nload
become: yes
```

## Цикл с переменной-словарем

```
- name: install python packages
  pip: name={{item.name}} version={{item.version}}
  become: True
  with_items:
    - {name=mezzanine, version=1.2.1}
    - {name=guincorn, version=3.2.9}
```

## Проверка ошибок выполнения

Проверяет условия и останавливает выполнение при нарушении

```
- name: Проверка значения переменной
  assert:
    that:
      - my_variable == "expected_value"
```

## Примеры из документации:

```
- name: A single condition can be supplied as string instead of list
  ansible.builtin.assert:
    that: "ansible_os_family != 'RedHat'"

- name: Use yaml multiline strings to ease escaping
  ansible.builtin.assert:
    that:
      - "'foo' in some_command_result.stdout"
      - number_of_the_counting == 3
      - >
        "reject" not in some_command_result.stderr

- name: After version 2.7 both 'msg' and 'fail_msg' can customize failing assertion
  message
  ansible.builtin.assert:
    that:
      - my_param <= 100
```

```
- my_param >= 0
fail_msg: "'my_param' must be between 0 and 100"
success_msg: "'my_param' is between 0 and 100"

- name: Please use 'msg' when ansible version is smaller than 2.7
ansible.builtin.assert:
  that:
    - my_param <= 100
    - my_param >= 0
  msg: "'my_param' must be between 0 and 100"

- name: Use quiet to avoid verbose output
ansible.builtin.assert:
  that:
    - my_param <= 100
    - my_param >= 0
  quiet: true
```

---

Revision #5

Created 23 June 2024 05:07:37 by Admin

Updated 25 June 2024 02:22:01 by Admin