



Методическое пособие по выполнению домашнего задания по курсу «Администратор Linux. Professional»

Первые шаги с Ansible

Цель домашнего задания

Написать первые шаги с Ansible.

Описание домашнего задания

Подготовить стенд на Vagrant как минимум с одним сервером. На этом сервере, используя Ansible необходимо развернуть nginx со следующими условиями:

- необходимо использовать модуль yum/apt
- конфигурационные файлы должны быть взяты из шаблона jinja2 с переменными
- после установки nginx должен быть в режиме enabled в systemd
- должен быть использован notify для старта nginx после установки
- сайт должен слушать на нестандартном порту - **8080**, для этого использовать переменные в Ansible

* Сделать все это с использованием Ansible роли

Установка Ansible

Версия Ansible =>2.4 требует для своей работы Python 2.6 или выше

!!! Убедитесь что у Вас установлена нужная версия:

Далее произведите установку для Вашей ОС [по инструкции](#) и убедитесь что Ansible установлен корректно:

```
dz@DZs-MacBook-Pro ansible-vagrant % ansible --version
ansible [core 2.16.0]
  config file = None
  configured module search path = ['/Users/dz/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/local/Cellar/ansible/9.0.1/libexec/lib/python3.12/site-packages/ansible
  ansible collection location = /Users/dz/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/local/bin/ansible
  python version = 3.12.0 (main, Oct 5 2023, 15:48:48) [Clang 14.0.0 (clang-1400.0.29.202)] (/usr/local/Cellar/ansible/9.0.1/libexec/bin/python)
  jinja version = 3.1.2
```

- Для управления хостами Ansible использует SSH соединение. Поэтому перед стартом необходимо убедиться что у Вас есть доступ до управляемых хостов.
- Также на управляемых хостах должен быть установлен Python 2.X

Подготовка окружения

Vagrantfile

https://drive.google.com/file/d/17MEtg20TFSjKil6ih7PvPez7jmCvo6fb/view?usp=share_link

- Создайте каталог [Ansible](#) и положите в него этот Vagrantfile
- Поднимите управляемый хост командой [vagrant up](#) и убедитесь, что все прошло успешно и есть доступ по [ssh](#)
- Для подключения к хосту [nginx](#) нам необходимо будет передать множество параметров - это особенность Vagrant. Узнать эти параметры можно с помощью команды [vagrant ssh-config](#). Вот основные необходимые нам:

<code>Host nginx</code>	имя хоста
<code>HostName 127.0.0.1</code>	IP адрес
<code>User vagrant</code>	имя пользователя под которым подключаемся
<code>Port 2222</code>	порт, который проброшен на 127.0.0.1
<code>IdentityFile .vagrant/machines/nginx/virtualbox/private_key</code>	путь до приватного ключа

Ansible

Создадим свой первый inventory файл `./staging/hosts`

Со следующим содержимым:

```
[web]nginx ansible_host=127.0.0.1 ansible_port=2222 ansible_user=vagrant
ansible_private_key_file=.vagrant/machines/nginx/virtualbox/private_key
```

И наконец убедимся, что Ansible может управлять нашим хостом. Сделать это можно с помощью команды:

```
[dz@DZs-MacBook-Pro ansible-vagrant % ansible nginx -i staging/hosts -m ping
nginx | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
```

Как видно, нам придется каждый раз явно указывать наш инвентори файл и вписывать в него много информации. Это можно обойти используя [ansible.cfg](#) файл - прописав конфигурацию в нем.

- Для этого в текущем каталоге создадим файл [ansible.cfg](#) со следующим содержанием:

```
[defaults]
inventory = staging/hosts
remote_user = vagrant
host_key_checking = False
retry_files_enabled = False
```

- Теперь из инвентори можно убрать информацию о пользователе:

```
[web]
nginx ansible_host=127.0.0.1 ansible_port=2222
ansible_private_key_file=.vagrant/machines/nginx/virtualbox/private_key
```

- Еще раз убедимся, что управляемый хост доступен, только теперь без явного указания inventory файла:

```
dz@DZs-MacBook-Pro ansible-vagrant % ansible nginx -m ping
nginx | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
```

- Теперь, когда мы убедились, что у нас все подготовлено - установлен Ansible, поднят хост для теста и Ansible имеет к нему доступ, мы можем конфигурировать наш хост.

- Для начала воспользуемся [Ad-Hoc](#) командами и выполним некоторые удаленные команды на нашем хосте.

- Посмотрим какое ядро установлено на хосте:

```
[dz@DZs-MacBook-Pro ansible-vagrant % ansible nginx -m command -a "uname -r"
nginx | CHANGED | rc=0 >>
5.15.0-88-generic
```

- Проверим статус сервиса firewalld

```
dz@DZs-MacBook-Pro ansible-vagrant % ansible nginx -m systemd -a name=firewalld
nginx | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "name": "firewalld",
  "status": {
    "ActiveEnterTimestamp": "n/a",
    "ActiveEnterTimestampMonotonic": "0",
    "ActiveExitTimestamp": "n/a",
    "ActiveExitTimestampMonotonic": "0",
    "ActiveState": "inactive",
```

- Теперь собственно приступим к выполнению домашнего задания и написания Playbook-а для установки **NGINX**. Будем писать его постепенно,

```
dz@DZs-MacBook-Pro ansible-vagrant % cat nginx.yml
---
- name: NGINX | Install and configure NGINX
  hosts: nginx
  become: true

  tasks:
    - name: update
      apt:
        update_cache=yes

    - name: NGINX | Install NGINX
      apt:
        name: nginx
        state: latest
```

```
dz@DZs-MacBook-Pro ansible-vagrant % ansible-playbook nginx.yml

PLAY [NGINX | Install and configure NGINX] *****
*****

TASK [Gathering Facts] *****
*****
ok: [nginx]

TASK [update] *****
*****
changed: [nginx]

TASK [NGINX | Install NGINX] *****
*****
changed: [nginx]

PLAY RECAP *****
nginx                : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

- Далее добавим шаблон для конфига NGINX и модуль, который будет

копировать этот шаблон на хост:

```
- name: NGINX | Create NGINX config file from template
```

```
  template:
```

```
    src: templates/nginx.conf.j2
```

```
    dest: /tmp/nginx.conf
```

```
  tags:
```

```
- nginx-configuration
```

- Сразу же пропишем в Playbook необходимую нам переменную. Нам нужно чтобы NGINX слушал на порту 8080:

```
- name: NGINX | Install and configure NGINX
```

```
hosts: nginx
```

```
become: true
```

```
vars:                                Добавлена только секция vars
```

```
nginx_listen_port: 8080
```

Также добавим теги, на данном этапе наш файл будет выглядеть следующим образом

```
---
- name: NGINX | Install and configure NGINX
  hosts: nginx
  become: true
  vars:
    nginx_listen_port: 8080

  tasks:
    - name: update
      apt:
        update_cache=yes
      tags:
        - update apt

    - name: NGINX | Install NGINX
      apt:
        name: nginx
        state: latest
      tags:
        - nginx-package

    - name: NGINX | Create NGINX config file from template
      template:
        src: templates/nginx.conf.j2
        dest: /etc/nginx/nginx.conf
      tags:
        - nginx-configuration
```

- Сам шаблон будет выглядеть так:

```
events {
```

```

worker_connections 1024;
}

http {
    server {
        listen {{ nginx_listen_port }} default_server;
        server_name default_server;
        root /usr/share/nginx/html;

        location / {
        }
    }
}

```

[Ссылка](#)

Теперь создадим [handler](#) и добавим [notify](#) к копирования шаблона. Теперь каждый раз когда конфиг будет изменяться - сервис перезагрузиться. Секция с handlers будет выглядеть следующим образом:

handlers:	Так же создадим handler для
- name: restart nginx	рестарта и включения сервиса
systemd:	при загрузке
name: nginx	
state: restarted	
enabled: yes	
- name: reload nginx	Перечитываем конфиг
systemd:	
name: nginx	
state: reloaded	

- Notify будут выглядеть так:

```

- name: NGINX | Install NGINX
  apt:
    name: nginx
    state: latest
  notify:
    - restart nginx
  tags:
    - nginx-package

- name: NGINX | Create NGINX config file from template
  template:
    src: templates/nginx.conf.j2
    dest: /etc/nginx/nginx.conf
  notify:
    - reload nginx
  tags:
    - nginx-configuration

```

Результирующий файл [nginx.yml](#). Теперь можно его запустить

```

1  ---
2  - name: NGINX | Install and configure NGINX
3    hosts: nginx
4    become: true
5    vars:
6      nginx_listen_port: 8080
7
8    tasks:
9      - name: update
10        apt:
11          update_cache=yes
12        tags:
13          - update apt
14
15      - name: NGINX | Install NGINX
16        apt:
17          name: nginx
18          state: latest
19        notify:
20          - restart nginx
21        tags:
22          - nginx-package
23

```

```

23
24     - name: NGINX | Create NGINX config file from template
25       template:
26         src: templates/nginx.conf.j2
27         dest: /etc/nginx/nginx.conf
28       notify:
29         - reload nginx
30       tags:
31         - nginx-configuration
32
33     handlers:
34       - name: restart nginx
35         systemd:
36           name: nginx
37           state: restarted
38           enabled: yes
39
40       - name: reload nginx
41         systemd:
42           name: nginx
43           state: reloaded

```

```

dz@DZs-MacBook-Pro ansible-vagrant % ansible-playbook nginx.yml
PLAY [NGINX | Install and configure NGINX] *****
TASK [Gathering Facts] *****
ok: [nginx]
TASK [update] *****
changed: [nginx]
TASK [NGINX | Install NGINX] *****
ok: [nginx]
TASK [NGINX | Create NGINX config file from template] *****
changed: [nginx]
RUNNING HANDLER [reload nginx] *****
changed: [nginx]
PLAY RECAP *****
nginx                : ok=5    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

```

Теперь можно перейти в браузере по адресу <http://192.168.11.150:8080> и убедиться, что сайт доступен.

Или из консоли выполнить команду: `curl http://192.168.11.150:8080`

```

[dz@DZs-MacBook-Pro ansible-vagrant % vagrant ssh
Last login: Thu Nov 30 21:22:44 2023 from 10.0.2.2
[vagrant@nginx:~$ curl http://192.168.11.150:8080
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
  body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
  }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

```