

Homework_Lesson 19

Цель: получить практический опыт работы с системой управления конфигурацией Ansible

! перед началом работы с Ansible, разверните еще одну виртуальную машину, которая будет управляться с помощью него.

Задание 1: установить Ansible

Обновите список пакетов:

Установите пакет software-properties-common, который позволит добавлять репозитории:

```
sudo apt install software-properties-common
```

Добавьте репозиторий Ansible PPA:

```
sudo apt-add-repository ppa:ansible/ansible
```

Обновите список пакетов:

```
sudo apt update
```

Установите Ansible:

```
sudo apt install ansible
```

```
sttewie@sttewie:~$ ansible --version
ansible [core 2.16.3]
  config file = None
  configured module search path = ['/home/sttewie/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /home/sttewie/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.12.7 (main, Nov  6 2024, 18:29:01) [GCC 14.2.0] (/usr/bin/python3)
  jinja version = 3.1.3
  libyaml = True
```

Задание 2: сгенерировать SSH-ключ

- Откройте терминал на вашем контролирующем хосте.
- Запустите команду для генерации ключа:

```
ssh-keygen -t rsa -b 4096
```

При выполнении этой команды вам будет предложено ввести путь к файлу, в котором будет сохранен ключ, а также парольная фраза

```
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\Adminbh\.ssh/id_rsa):
C:\Users\Adminbh\.ssh/id_rsa already exists.
Overwrite (y/n)?
PS C:\WINDOWS\system32>
PS C:\WINDOWS\system32>
PS C:\WINDOWS\system32>
PS C:\WINDOWS\system32> ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\Adminbh\.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\Adminbh\.ssh/id_rsa
Your public key has been saved in C:\Users\Adminbh\.ssh/id_rsa.pub
The key fingerprint is:
SHA256:2v3Sq0kXKk5SztBSw2WNFOY9KhpXQi00OGEXWK2aLrg sttewie@BHL020
The key's randomart image is:
+---[RSA 4096]-----+
|  o*B=*+          |
|  .+. *+. .       |
|    .oo o         |
|  o.+ . .         |
|  +o= S           |
|  oX + . .        |
|  . .o = ...+     |
|  . . .o. ..+..   |
|E. . ..o+o ..    |
+---[SHA256]-----+
PS C:\WINDOWS\system32>
```

После того, как ключ будет сгенерирован, скопируйте его на управляемый хост:
`ssh-copy-id user@managed_host`

В этой команде "user" - это имя пользователя на управляемом хосте, а "managed_host" - имя или IP-адрес управляемого хоста.

- При выполнении этой команды вам будет предложено ввести пароль пользователя на управляемом хосте.
- После успешной передачи ключа на управляемый хост вы можете использовать Ansible для управления им.

```
PS C:\WINDOWS\system32> ssh -i C:\Users\Adminbh\Documents\mykey sttewie@192.168.20.202
Welcome to Ubuntu 24.10 (GNU/Linux 6.11.0-13-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Tue Jan 21 01:53:14 PM UTC 2025

System load:  0.08               Processes:            279
Usage of /:   61.4% of 18.01GB   Users logged in:     1
Memory usage: 62%               IPv4 address for eth0: 192.168.20.202
Swap usage:   25%

=> There is 1 zombie process.

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

10 updates can be applied immediately.
6 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Last login: Wed Jan 15 09:14:38 2025 from 192.168.20.5
sttewie@sttewie:~$
```

Задание 3: просмотрите и повторите все манипуляции, указанные в видеоуроке по ссылке

Цель: настроить Ansible окружение на вашей машине, чтобы вы могли использовать его для автоматизации управления конфигурацией и развертывания приложений на удаленных серверах.

Задание: Настройка Ansible окружения

Шаги:

1. Установите Ansible на вашу локальную машину. В зависимости от операционной системы, которую вы используете, установка может отличаться. Инструкции по установке Ansible можно найти на официальном сайте: https://docs.ansible.com/ansible/latest/installation_guide/index.html
2. Создайте инвентарный файл, который содержит информацию о серверах, которые вы планируете управлять с помощью Ansible. Инвентарный файл может быть в формате ini или YAML.

Например:

```
[servers]
```

```
server1 ansible_host=192.168.1.100 ansible_user=myuser
ansible_ssh_private_key_file=/path/to/key.pem
server2 ansible_host=192.168.1.101 ansible_user=myuser
ansible_ssh_private_key_file=/path/to/key.pem
```

В этом примере мы определяем два сервера с IP-адресами 192.168.1.100 и 192.168.1.101. Мы также указываем имя пользователя и путь к файлу с приватным ключом SSH, который Ansible будет использовать для аутентификации.

3. Создайте файл конфигурации Ansible, который будет содержать глобальные настройки Ansible. Файл конфигурации Ansible может быть

расположен в нескольких местах, включая каталоги /etc/ansible и ~/.ansible/.

Например:

```
[defaults]
```

```
inventory = /path/to/inventory_file
```

```
remote_user = myuser
```

```
private_key_file = /path/to/key.pem
```

В этом примере мы указываем путь к инвентарному файлу, имя пользователя

SSH и путь к файлу с приватным ключом SSH, который Ansible будет

использовать для аутентификации.

Создаем директорию и файл hosts.txt в котором будет список серверов для подключения. Проверяем

```
sttewie@sttewie:~$ pwd
/home/sttewie
sttewie@sttewie:~$ mkdir adsible
sttewie@sttewie:~$ cd adsible/
sttewie@sttewie:~/adsible$ nano hosts.txt
sttewie@sttewie:~/adsible$ cat hosts.txt
[staging servers]
linux1 ansible_host= 192.168.20.204 ansible_user=sttewie ansible_ssh_private_key_file=/home/sttewie/.ssh/mykey
```

```
sttewie@sttewie:~/adsible$ ansible -i hosts.txt all -m ping
The authenticity of host '192.168.20.204 (192.168.20.204)' can't be established.
ED25519 key fingerprint is SHA256:c6IehTfmCNYfsYH5H0dWjFUTWJpYZUj+O3pEdHmjStY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? █
```

```
sttewie@sttewie:~/adsible$ ansible -i hosts.txt all -m ping
linux1 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
```

Создадим конфиг файл с указанием инвентарного файла. Проверяем

```
sttewie@sttewie:~/adsible$ nano ansible.cfg
sttewie@sttewie:~/adsible$ ^
sttewie@sttewie:~/adsible$ cat ansible.cfg
[defaults]
host_key_checking = false
inventory          = ./hosts.txt
```

```
sttewie@sttewie:~/adsible$ ansible all -m ping
linux1 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
```

4. Создайте простой плейбук Ansible, который будет выполнять простую задачу на одном из ваших серверов. Например, вы можете создать плейбук, который устанавливает пакеты на сервере. Пример

- name: Install packages

hosts: servers

become: true

tasks:

- name: Install Nginx

apt:

name: nginx

state: present

В этом примере мы создаем плейбук, который устанавливает пакет Nginx на всех серверах, определенных в инвентарном файле. Мы также указываем, что Ansible должен использовать привилегии root для выполнения

Создадим плейбук , который будет ставить nginx на убунту и добавлять в автозагрузку и проверим

```
GNU nano 8.1 install nginx.yml
--
- name: Установка и настройка Nginx
  hosts: all
  become: yes # Для выполнения команд с правами root
  tasks:
    - name: Обновить список пакетов (для Debian/Ubuntu)
      ansible.builtin.apt:
        update_cache: yes
      when: ansible_os_family == "Debian"

    - name: Установить Nginx (для Debian/Ubuntu)
      ansible.builtin.apt:
        name: nginx
        state: present
      when: ansible_os_family == "Debian"

    - name: Установить EPEL-репозиторий (для CentOS/RHEL)
      ansible.builtin.yum:
        name: epel-release
        state: present
      when: ansible_os_family == "RedHat"

    - name: Установить Nginx (для CentOS/RHEL)
      ansible.builtin.yum:
        name: nginx
        state: present
      when: ansible_os_family == "RedHat"

    - name: Включить Nginx в автозагрузку
      ansible.builtin.service:
        name: nginx
        enabled: yes

    - name: Убедиться, что Nginx запущен
      ansible.builtin.service:
        name: nginx
        state: started
```

```
sttewie@sttewie:~/adsible$ ansible-playbook install_nginx.yml
```

```
PLAY [Установка и настройка Nginx] *****

TASK [Gathering Facts] *****
fatal: [linux1]: FAILED! => {"msg": "Missing sudo password"}

PLAY RECAP *****
linux1                : ok=0    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0
```

```
● sttewie@sttewie:~/adsible$ cat hosts.txt
```

```
[staging_servers]
linux1 ansible host=192.168.20.204 ansible user=sttewie ansible ssh private key file=/home/sttewie/.ssh/mykey ansible become password=5550
```

```
● sttewie@sttewie:~/adsible$ ansible-playbook install_nginx.yml
```

```
PLAY [Установка и настройка Nginx] *****

TASK [Gathering Facts] *****
ok: [linux1]

TASK [Обновить список пакетов (для Debian/Ubuntu)] *****
changed: [linux1]

TASK [Установить Nginx (для Debian/Ubuntu)] *****
changed: [linux1]

TASK [Установить EPEL-репозиторий (для CentOS/RHEL)] *****
skipping: [linux1]

TASK [Установить Nginx (для CentOS/RHEL)] *****
skipping: [linux1]

TASK [Включить Nginx в автозагрузку] *****
ok: [linux1]

TASK [Убедиться, что Nginx запущен] *****
ok: [linux1]

PLAY RECAP *****
linux1                : ok=5    changed=2    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```