

Homework_Lesson19_Ansible_1_2

Цель 1: 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: `sudo apt update`. 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`

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 (пароль не обязателен, но рекомендуется). 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.

Задание 3: просмотрите и повторите все манипуляции, указанные в видеоуроке по ссылке https://www.youtube.com/watch?v=O5R6EBdaZZg&list=PLg5SS_4L6LYufspdPupdynbMQTBnZd31N&index=4. В данном ролике рассматриваются шаги подключения к серверам Linux.

Цель 2: 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.

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

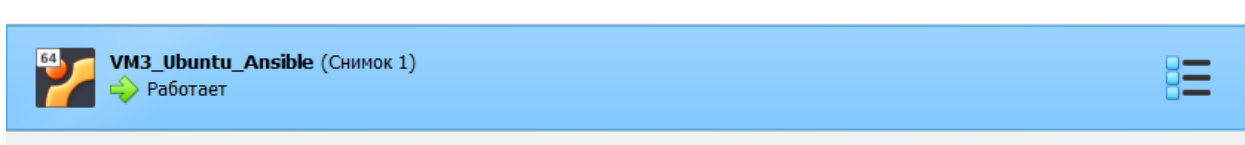
state: present

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.

Цель 1-ая: получить практический опыт работы с системой управления конфигурацией Ansible.

Выполнение 1-ого задания:

Создана отдельная ВМ на VirtualBox с которой будет менеджиться Ansible. ВМ VM_Ubuntu_Ansible: 4 ГБ ОЗУ, 1 CPU, 25 ГБ жесткий диск, развернута на ней ОС Ubuntu 22.04 и настроено две адаптера сети NAT и Сетевой мост (IP-192.168.100.14/24).



На WSL (наш контролирующий хост) выполняем следующие действия:

```
$ sudo apt update
```

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

 \\ уст. пакета для доб. репозиториев

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

 \\ добавление репозитория Ansible

```
$ sudo apt update
```

```
$ sudo apt install ansible
```

 \\ установка Ansible

```
$ ansible --version
```

 \\ проверка версии Ansible

```
makarov@DESKTOP-UG6J7T7:~$ ansible --version
ansible [core 2.14.18]
  config file = None
  configured module search path = ['/home/makarov/.ansible/plugins/modules',
  '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /home/makarov/.ansible/collections:/usr/shar
  e/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.11.2 (main, Nov 30 2024, 21:22:50) [GCC 12.2.0] (/usr/b
  in/python3)
  jinja version = 3.1.2
  libyaml = True
makarov@DESKTOP-UG6J7T7:~$ |
```

Версия установленного Ansible 2.17.7

Выполнение 2-ого задания:

На WSL выполняем следующие действия:

```
$ ssh-keygen -t rsa -b 4096           \ \ создание ключа
$ ssh-copy-id user@192.168.100.14     \ \ копируем наш ключ на ВМ
```

где, user - это пользователь на ВМ (VM_Ubuntu_Ansible) и его IP-адрес 192.168.100.14.

```
makarov@DESKTOP-UG6J7T7:~$ ssh-copy-id user@192.168.100.14
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/makarov/.ssh/id_rsa.pub"
The authenticity of host '192.168.100.14 (192.168.100.14)' can't be established.
ED25519 key fingerprint is SHA256:LDt1pxiMRWuYj6riQsBewRJx8+WxIIA4mh6C+Db306g.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
user@192.168.100.14's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'user@192.168.100.14'"
and check to make sure that only the key(s) you wanted were added.
```

Копирование ssh-ключа на ВМ 192.168.100.14

Выполнение 3-его задания:

```
$ mkdir ansible
$ cd ./ansible
$ touch hosts.txt
```

Вносим данные в файл hosts.txt:

Создаем группу [staging_servers] и дальше в строчку прописываем его alias name (user-VirtualBox), ansible_host=IP-адрес (192.168.100.14), ansible_user=user (пользователь которым подключаешься к ВМ (user - на стороне ВМ 192.168.100.14) + ansible_pass=password, но он нам не понадобится, т.к. мы уже прокинули ключи по SSH. Также можно использовать ключи “.pem” и вместо ansible_pass прописывается ansible_ssh_private_key_file=/*//*.pem (путь, где лежат *.pem ключи).

```
[staging_servers]
user-VirtualBox      ansible_host=192.168.100.14      ansible_user=user
```

```
$ ansible -i hosts.txt all -m ping
```

где, -i - использование инвентори файла (в нашей ситуации, это hosts.txt)

all - указывает на все группы

-m - указания модуля

```
makarov@DESKTOP-UG6J7T7:~/ansible$ ansible -i hosts.txt all -m ping
user-VirtualBox | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
makarov@DESKTOP-UG6J7T7:~/ansible$ |
```

Успешное подключение к клиенту

Создаем файл конфигурационный файл ansible.cfg. В нем можно конфигурировать и настраивать под себя настройки.

\$ ansible --version

```
makarov@DESKTOP-UG6J7T7:~/ansible$ ansible --version
ansible [core 2.14.18]
  config file = None
  configured module search path = ['/home/makarov/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /home/makarov/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.11.2 (main, Nov 30 2024, 21:22:50) [GCC 12.2.0] (/usr/bin/python3)
  jinja version = 3.1.2
  libyaml = True
makarov@DESKTOP-UG6J7T7:~/ansible$ |
```

Здесь мы видим, что конфигурационного настроенного файла нету

Создаем конфигурационный файл .cfg:

\$ nano ansible.cfg

```
makarov@DESKTOP-UG6J7T7:~/ansible$ cat ansible.cfg

[defaults]
host_key_checking = false
inventory          = ./hosts.txt
```

host_key_checking = false \\
inventory = ./hosts.txt \\ указывает какой файл инвентори используем

Пробуем запустить ansible:

\$ ansible all -m ping

```
makarov@DESKTOP-UG6J7T7:~/ansible$ ansible all -m ping
user-VirtualBox | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
makarov@DESKTOP-UG6J7T7:~/ansible$ |
```

Теперь не нужно указывать inventory файл, т.к. мы его уже прописали в конфигурационном файле ansible.cfg

Цель 2-ая: настроить Ansible окружение на вашей машине, чтобы вы могли использовать его для автоматизации управления конфигурацией и развертывания приложений на удаленных серверах.

Выполнение 1-ого задания:

На WSL установлена ОС Debian 12. Установка Ansible:

\$ sudo apt update
\$ sudo apt install software-properties-common \\ уст. пакета для доб. репозитория
\$ sudo apt-add-repository ppa:ansible/ansible \\ добавление репозитория Ansible
\$ sudo apt update

```
$ sudo apt install ansible
```

\\ установка Ansible

```
$ ansible --version
```

\\ проверка версии Ansible

На оф. сайте указано, что в репозиториях Debian могут быть не самые актуальные версии Ansible и есть такой путь установки:

```
$ UBUNTU_CODENAME=jammy
```

```
$ wget -O-
```

```
"https://keyserver.ubuntu.com/pks/lookup?fingerprint=on&op=get&search=0x6125E2A8C77F2818FB7BD15B93C4A3FD7BB9C367" | sudo gpg --dearmor -o /usr/share/keyrings/ansible-archive-keyring.gpg
```

```
$ echo "deb [signed-by=/usr/share/keyrings/ansible-archive-keyring.gpg]
```

```
http://ppa.launchpad.net/ansible/ansible/ubuntu $UBUNTU_CODENAME main" | sudo tee /etc/apt/sources.list.d/ansible.list
```

```
$ sudo apt update && sudo apt install ansible
```

Выполнение 2-ого задания:

На ВМ создаем приватный ключ и перекидываем его на хостовую машину (WSL).
Создаем инвентарный файл, который содержит информацию о серверах.

```
$ cd /home/makarov/ansible
```

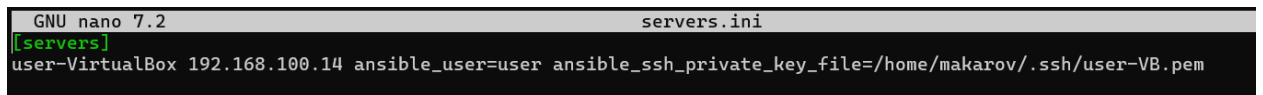
```
$ nano servers.ini
```

В файл servers.ini добавляем строку:

```
[servers]
```

```
user-VirtualBox ansible_host=192.168.100.14 ansible_user=user
```

```
ansible_ssh_private_key_file=/home/makarov/.ssh/user-VB.pem
```



```
GNU nano 7.2 servers.ini
[servers]
user-VirtualBox 192.168.100.14 ansible_user=user ansible_ssh_private_key_file=/home/makarov/.ssh/user-VB.pem
```

Выполнение 3-его задания:

Создание и настройка конфигурационного файла:

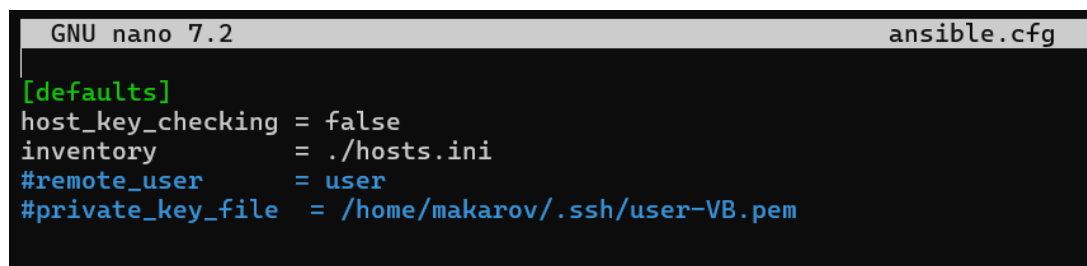
```
$ touch ansible.cfg
```

```
$ nano ansible.cfg
```

```
[defaults]
```

```
inventory = ./hosts.ini
```

```
remote_user = user
```



```
GNU nano 7.2 ansible.cfg
[defaults]
host_key_checking = false
inventory          = ./hosts.ini
#remote_user       = user
#private_key_file  = /home/makarov/.ssh/user-VB.pem
```

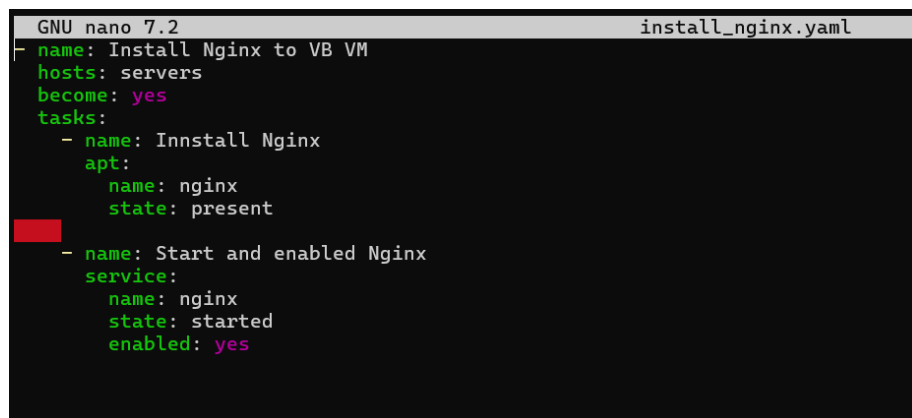
Выполнение 4-ого задания:

Создаем `playbook.yaml`, который будет выполнять установку пакета Nginx и заодно его сразу запустит через `systemctl`:

```
$ touch install_nginx.yaml
$ nano install_nginx.yaml
```

В файл `install_nginx.yaml` добавляем следующие данные:

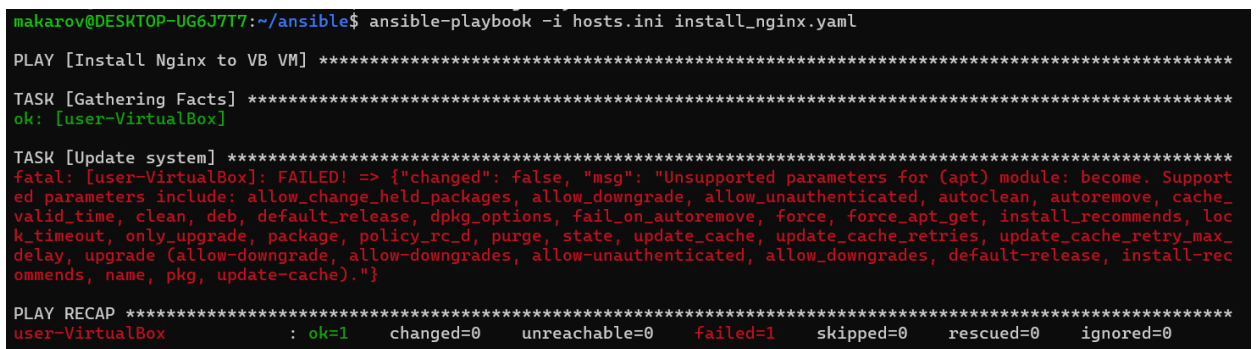
```
- name: Install Nginx to VB VM
  hosts: servers
  become: yes
  tasks:
### Указываем установку Nginx
  - name: Install Nginx
    apt:
      name: nginx
      state: present
### Старт Nginx после установки
  - name: Start and enabled Nginx
    service:
      name: nginx
      state: started
      enabled: yes
```



```
GNU nano 7.2 install_nginx.yaml
- name: Install Nginx to VB VM
  hosts: servers
  become: yes
  tasks:
  - name: Install Nginx
    apt:
      name: nginx
      state: present
  - name: Start and enabled Nginx
    service:
      name: nginx
      state: started
      enabled: yes
```

После запускаем ansible playbook:

```
$ ansible-playbook install_nginx.yaml
```



```
makarov@DESKTOP-UG6J7T7:~/ansible$ ansible-playbook -i hosts.ini install_nginx.yaml

PLAY [Install Nginx to VB VM] *****

TASK [Gathering Facts] *****
ok: [user-VirtualBox]

TASK [Update system] *****
fatal: [user-VirtualBox]: FAILED! => {"changed": false, "msg": "Unsupported parameters for (apt) module: become. Support
ed parameters include: allow_change_held_packages, allow_downgrade, allow_unauthenticated, autoclean, autoremove, cache_
valid_time, clean, deb, default_release, dpkg_options, fail_on_autoremove, force, force_apt_get, install_recommends, loc
k_timeout, only_upgrade, package, policy_rc_d, purge, state, update_cache, update_cache_retries, update_cache_retry_max_
delay, upgrade (allow-downgrade, allow-downgrades, allow-unauthenticated, allow-downgrades, default-release, install-rec
ommends, name, pkg, update-cache)."}

PLAY RECAP *****
user-VirtualBox      : ok=1    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0
```

При первом запуске, валятся ошибки, т.к. нехватает прав `sudo/root`

По SSH заходим на наш целевой хост (VM_Ubuntu_Ansible):

```
$ ssh user@192.168.100.14 \\коннектимся по SSH на VM
```

```
$ sudo visudo \\данная команда запускает конфигурационный файл /etc/sudoers.tmp
```

В конфигурационном файле sudoers.tmp добавляем строку:

user ALL=(ALL) NOPASSWD: ALL \\данная строка нам позволяет разрешить пользователю user выполнять любую команду без запроса пароля. И данный пользователь используется у нас в playbook Ansible.

```
GNU nano 6.2 /etc/sudoers.tmp
# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin   ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "@include" directives:

@include /etc/sudoers.d

user     ALL=(ALL) NOPASSWD: ALL
|
```

Конфигурационный файл /etc/sudoers.tmp

Возвращаемся обратно к серверу Master Ansible т.е. к нашей WSL. И вводим заново нашу команду ansible-playbook install_nginx.yaml:

```
$ ansible-playbook install_nginx.yaml
```

```
makarov@DESKTOP-UG6J7T7:~/ansible$ ansible-playbook install_nginx.yaml
PLAY [Install Nginx to VB VM] *****
TASK [Gathering Facts] *****
ok: [user-VirtualBox]
TASK [Install Nginx] *****
changed: [user-VirtualBox]
TASK [Start and enabled Nginx] *****
ok: [user-VirtualBox]
PLAY RECAP *****
user-VirtualBox : ok=3  changed=1  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
makarov@DESKTOP-UG6J7T7:~/ansible$ |
```

Успешная установка веб-сервера Nginx на VM с помощью Ansible