Системы управления конфигурациями

Урок 5. Практика на проекте

Курсовой проект - установка Wordpress в прод кофигурации

Должны быть сервисы:

Сервер реверс-прокси
Два сервера приложения
Один сервер БД + мониторинг (Prometeus)

Практическое задание к пятому уроку:

Создать роль Ansible установки Vagrant и запуска 4-х ВМ
В качестве отчёта приложить ссылку на публичный репозиторий с кодом

Тк на RedOs и в РФ есть проблемы с Vagrant (установкой и получением пакетов), а в качестве хостинга используется форк oVirt - для деплоя VM будем использовать Ansible с коллекцией для oVirt hosting.

```
Requirements
The below requirements are needed on the host that executes this module.

python >= 2.7
ovirt-engine-sdk-python >= 4.4.0
```

Установка:

```
sudo yum -y install ovirt-engine-sdk-python #ставим зависимости, питон 3.8 уже в системе.
ansible-galaxy collection install ovirt.ovirt
```

Создаем роль для деплоя VM:

```
cd ~/ansible/roles && ansible-galaxy init deploy_vm_ovirt
```

Создаем файл тасков **~/ansible/roles/deploy_vm_ovirt/tasks/main.yml**:

```
---
- name: Obtain SSO token with using username/password credentials
  ovirt.ovirt.ovirt_auth:
    url: https://eleanora.local/ovirt-engine/api
    username: admin@internal
    ca_file: /etc/pki/ca-trust/extracted/pem/eleanora.pem
    password: "{{ ovirtvm_password }}"

- name: Deploy VM "{{ vm_name }}"
  ovirt.ovirt.ovirt_vm:
    auth: "{{ ovirt_auth }}"
    name: "{{ vm_name }}"
```

```
    template: "{{ vm_template }}"
    cluster: "Default"
    state: "running"
    cloud_init:
      host_name: "{{ vm_host_name }}"
      user_name: "{{ vm_user_name }}"
      root_password: "{{ vm_root_password }}"
      timezone: "Europe/Moscow"
      authorized_ssh_keys: "{{ ssh_keys }}"
      nic_name: "enp1s0"
      dns_search: ".local"
      dns_servers: "10.78.0.1 212.1.224.6"
      nic_boot_protocol: "dhcp"
      nic_boot_protocol_v6: "none"
    cloud_init_persist: true
    wait: true

- name: Revoke the SSO token
  ovirt_auth:
    state: absent
    ovirt_auth: "{{ ovirt_auth }}"
```

Создаем файл с переменными **~/ansible/roles/deploy_vm_ovirt/vars/ovirtvm_vars.yml**:

```
ssh_keys: /home/sa/.ssh/id_rsa.pub
vm_template: "TMPL_Ubuntu20_04"
vm_host_name: "{{ vm_name }}"
vm_user_name: "sa"
```

Создаем зашифрованное хранилище паролей **~/ansible/roles/deploy_vm_ovirt/vars/password**:

```
---
ovirtvm_password: Ahtufn}1
vm_root_password: Ahtufn}1
```

Создаем плейбук для создание 4х ВМ для проекта **~/ansible/roles/deploy_vm_ovirt/deploy_vm_ovirt.yml**:

```
---
- hosts: localhost
  connection: local

  vars_files:
  - vars/password.yml
  - vars/ovirtvm_vars.yml

  roles:
    - role: deploy_vm_ovirt
      vars:
        vm_name: "ha_proxy.local"
      tags: proxy
    - role: deploy_vm_ovirt
      vars:
        vm_name: "app01.local"
```

```
      tags: app
    - role: deploy_vm_ovirt
      vars:
        vm_name: "app02.local"
      tags: app
    - role: deploy_vm_ovirt
      vars:
        vm_name: "mysql_db.local"
      tags: db
```

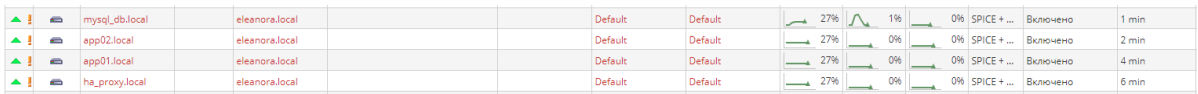
Запускаем плейбук:

```
cd ~/ansible && ansible-playbook roles/deploy_vm_ovirt/deploy_vm_ovirt.yml  --vault-password-file ./vault.pass
```

Задание успешно выполнено:

```
[sa@ansible ansible]$ ansible-playbook roles/deploy_vm_ovirt/deploy_vm_ovirt.yml  --vault-password-file ./vault.pass

PLAY [localhost] ***************************************************************************************************

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

TASK [deploy_vm_ovirt : Obtain SSO token with using username/password credentials] *********************************
ok: [localhost]

TASK [deploy_vm_ovirt : Deploy VM "ha_proxy.local"] ****************************************************************
changed: [localhost]

TASK [deploy_vm_ovirt : Revoke the SSO token] **********************************************************************
ok: [localhost]

TASK [deploy_vm_ovirt : Obtain SSO token with using username/password credentials] *********************************
ok: [localhost]

TASK [deploy_vm_ovirt : Deploy VM "app01.local"] *******************************************************************
changed: [localhost]

TASK [deploy_vm_ovirt : Revoke the SSO token] **********************************************************************
ok: [localhost]

TASK [deploy_vm_ovirt : Obtain SSO token with using username/password credentials] *********************************
ok: [localhost]

TASK [deploy_vm_ovirt : Deploy VM "app02.local"] *******************************************************************
changed: [localhost]

TASK [deploy_vm_ovirt : Revoke the SSO token] **********************************************************************
ok: [localhost]

TASK [deploy_vm_ovirt : Obtain SSO token with using username/password credentials] *********************************
ok: [localhost]

TASK [deploy_vm_ovirt : Deploy VM "mysql_db.local"] ****************************************************************
changed: [localhost]

TASK [deploy_vm_ovirt : Revoke the SSO token] **********************************************************************
ok: [localhost]

PLAY RECAP *********************************************************************************************************
localhost                  : ok=13   changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

На виртуализации:



Добавим хосты в список:

```
[sa@ansible ansible]$ cat inventory/hosts.yml
all:
  children:
    webservers:
      hosts:
        geekbrains:
          ansible_host: geekbrains.local
        app01:
          ansible_host: app01.local
        app02:
          ansible_host: app02.local
        haproxy:
          ansible_host: haproxy.local
      vars:
        ansible_user: sa
        ansible_python_interpreter: /usr/bin/python3
    databases:
      hosts:
        mysqldb:
          ansible_host: geekbrains.local
      vars:
        ansible_user: sa
        ansible_python_interpreter: /usr/bin/python3
```

Добавим ключи и проверим работу:

```
ssh-copy-id sa@app01.local
ssh-copy-id sa@app02.local
ssh-copy-id sa@mysqldb.local
ssh-copy-id sa@haproxy.local
```

```
[sa@ansible ansible]$ ansible all -m ping
app01 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
app02 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
haproxy | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
mysqldb | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
geekbrains | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```