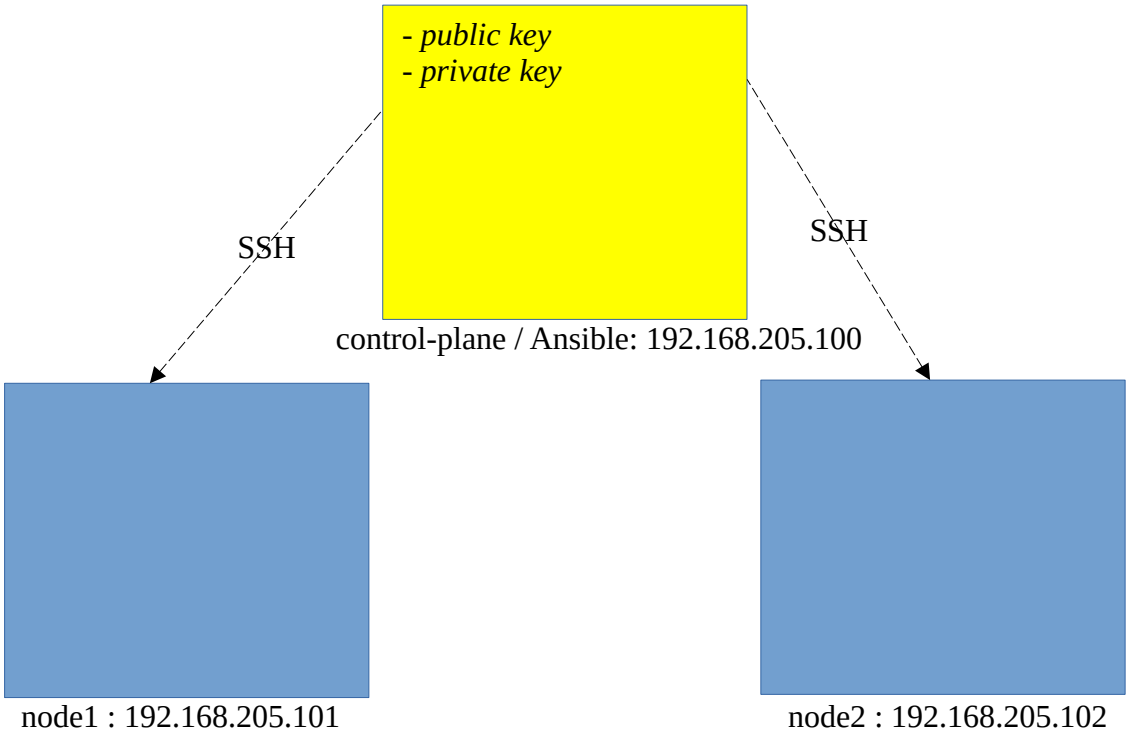
Lab8 - Ansible

Lab environment



Ansible install and ssh prepare

1. Install ansible on the master VM, using the following commands :

   sudo apt-add-repository ppa:ansible/ansible

   sudo apt update

   sudo apt install ansible

```
brahim@Training:~/k8s-lab$ vagrant ssh kube-control-plane
Last login: Sun Apr 28 07:47:11 2024 from 10.0.2.2
vagrant@kube-control-plane:~$ sudo apt-add-repository ppa:ansible/ansible
Repository: 'deb https://ppa.launchpadcontent.net/ansible/ansible/ubuntu/ jammy main'
Description:
Ansible is a radically simple IT automation platform that makes your applications and syster
m code to deploy and update your applications— automate in a language that approaches plain
```

...

- Check ansible version.

```
vagrant@kube-control-plane:~$ ansible --version
ansible [core 2.16.5]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/vagrant/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /home/vagrant/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.10.6 (main, Mar 10 2023, 10:55:28) [GCC 11.3.0] (/usr/bin/python3)
  jinja version = 3.0.3
  libyaml = True
```

2. Generate ssh keys with defaults parameters.

```
vagrant@kube-control-plane:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/vagrant/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/vagrant/.ssh/id_rsa
Your public key has been saved in /home/vagrant/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:wRQ9LIrflMhqazB1DIshdK8TKntvtzdHNtFdUbCtuBM vagrant@kube-control-plane
The key's randomart image is:
+---[RSA 3072]----+
|.. .    o+    .o+|
|. o o  o. +    o.|
| . + B ooo o ....|
|  o * * o.. .... |
|.. + + oS  .E .  |
|..o + . . +  o   |
|. .+ .   o .o    |
| . .+ . o .  .   |
|   o...o o       |
+----[SHA256]-----+
vagrant@kube-control-plane:~$
vagrant@kube-control-plane:~$ ls /home/vagrant/.ssh/
authorized_keys  id_rsa  id_rsa.pub  known_hosts
vagrant@kube-control-plane:~$
```

- Add the public key to the *authorized_keys* file on all ansible hosts (node1 and node2) using following commands :

  ssh-copy-id -i /home/vagrant/.ssh/id_rsa.pub vagrant@192.168.205.101

  ssh-copy-id -i /home/vagrant/.ssh/id_rsa.pub vagrant@192.168.205.102

```
vagrant@kube-control-plane:~$ ssh-copy-id -i /home/vagrant/.ssh/id_rsa.pub vagrant@192.168.205.101
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/vagrant/.ssh/id_rsa.pub"
/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
vagrant@192.168.205.101's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'vagrant@192.168.205.101'"
and check to make sure that only the key(s) you wanted were added.

vagrant@kube-control-plane:~$
vagrant@kube-control-plane:~$ ssh-copy-id -i /home/vagrant/.ssh/id_rsa.pub vagrant@192.168.205.102
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/vagrant/.ssh/id_rsa.pub"
The authenticity of host '192.168.205.102 (192.168.205.102)' can't be established.
ED25519 key fingerprint is SHA256:ugTrfKtqV0Q+JGa7a4Ffmtg4ypTgdf8GVkGHi3FWjfU.
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
vagrant@192.168.205.102's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'vagrant@192.168.205.102'"
and check to make sure that only the key(s) you wanted were added.
```

3. Using following commands, try ssh connection with keys from ansible management node to ansible hosts.

ssh 'vagrant@192.168.205.101'

ssh 'vagrant@192.168.205.102'

```
vagrant@kube-control-plane:~$ ssh 'vagrant@192.168.205.101'
Last login: Sun Apr 21 12:11:34 2024 from 10.0.2.2
vagrant@kube-node1:~$
vagrant@kube-node1:~$ exit
logout
Connection to 192.168.205.101 closed.
vagrant@kube-control-plane:~$
vagrant@kube-control-plane:~$ ssh 'vagrant@192.168.205.102'
Last login: Sun Apr 21 12:12:48 2024 from 10.0.2.2
vagrant@kube-node2:~$
vagrant@kube-node2:~$ exit
logout
Connection to 192.168.205.102 closed.
vagrant@kube-control-plane:~$
```

Now you can connect from Ansible to target nodes using public key-based authentication.

4. Add all kubernetes nodes to default inventory file */etc/ansible/hosts*, put ***node1*** and ***node2*** in the ***nodes*** group (you can copy the following content)

localhost *ansible_connection=local*

[nodes]

192.168.205.101 *ansible_user=vagrant*

192.168.205.102 *ansible_user=vagrant*

```
vagrant@kube-control-plane:~$ sudo vim /etc/ansible/hosts
vagrant@kube-control-plane:~$ cat /etc/ansible/hosts
[control]
localhost               ansible_connection=local

[nodes]
192.168.205.101         ansible_user=vagrant
192.168.205.102         ansible_user=vagrant
vagrant@kube-control-plane:~$
```

Ansible ad hoc commands

5. Validate the connection between ansible management and target hosts using ansible ***ping*** module.

```
vagrant@kube-control-plane:~$ ansible all -m ping
localhost | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
192.168.205.102 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
192.168.205.101 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
vagrant@kube-control-plane:~$
```

6. Gather facts about all hosts with the following command :

   ansible all -m setup

7. Get the *uptime* of remote hosts using any ansible ad hoc command.

8. Check the memory usage of all node hosts.

Ansible playbook

9. Create a playbook which install *apache2* package on hosts

```
- name: play1
  hosts: all
  gather_facts: false
  become : true
  tasks:
  - name: Installing Apache2
    apt:
      name: apache2
      state: latest
```

```
vagrant@kube-control-plane:~$ vim playbook1.yaml
vagrant@kube-control-plane:~$ cat playbook1.yaml
- name: play1
  hosts: all
  gather_facts: false
  become: true
  tasks:
  - name: Installing Apache2
    apt:
      name: apache2
      state: latest
vagrant@kube-control-plane:~$ _
```

   - Apply the playbook.

```
vagrant@kube-control-plane:~$ ansible-playbook playbook1.yaml

PLAY [play1] ***************************************************************************************************

TASK [Installing Apache2] **************************************************************************************
changed: [192.168.205.102]
changed: [192.168.205.101]

PLAY RECAP *****************************************************************************************************
192.168.205.101            : ok=1    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
192.168.205.102            : ok=1    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

vagrant@kube-control-plane:~$
```

10. Add a condition to the playbook to provision only hosts of the Debian family (you must enable gathering facts), then apply the playbook.

```
vagrant@kube-control-plane:~$ vim playbook1.yaml
vagrant@kube-control-plane:~$ cat playbook1.yaml
- name: play1
  hosts: all
#  gather_facts: false
  become: true
  tasks:
  - name: Installing Apache2
    apt:
      name: apache2
      state: latest
    when: ansible_os_family=="Debian"
vagrant@kube-control-plane:~$
vagrant@kube-control-plane:~$ ansible-playbook playbook1.yaml

PLAY [play1] ***************************************************************************************************

TASK [Gathering Facts] *****************************************************************************************
ok: [192.168.205.101]
ok: [192.168.205.102]

TASK [Installing Apache2] **************************************************************************************
ok: [192.168.205.101]
ok: [192.168.205.102]

PLAY RECAP *****************************************************************************************************
192.168.205.101            : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
192.168.205.102            : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

vagrant@kube-control-plane:~$
```

11. Modify the playbook to provision only hosts of the Redhat family, then apply the playbook.