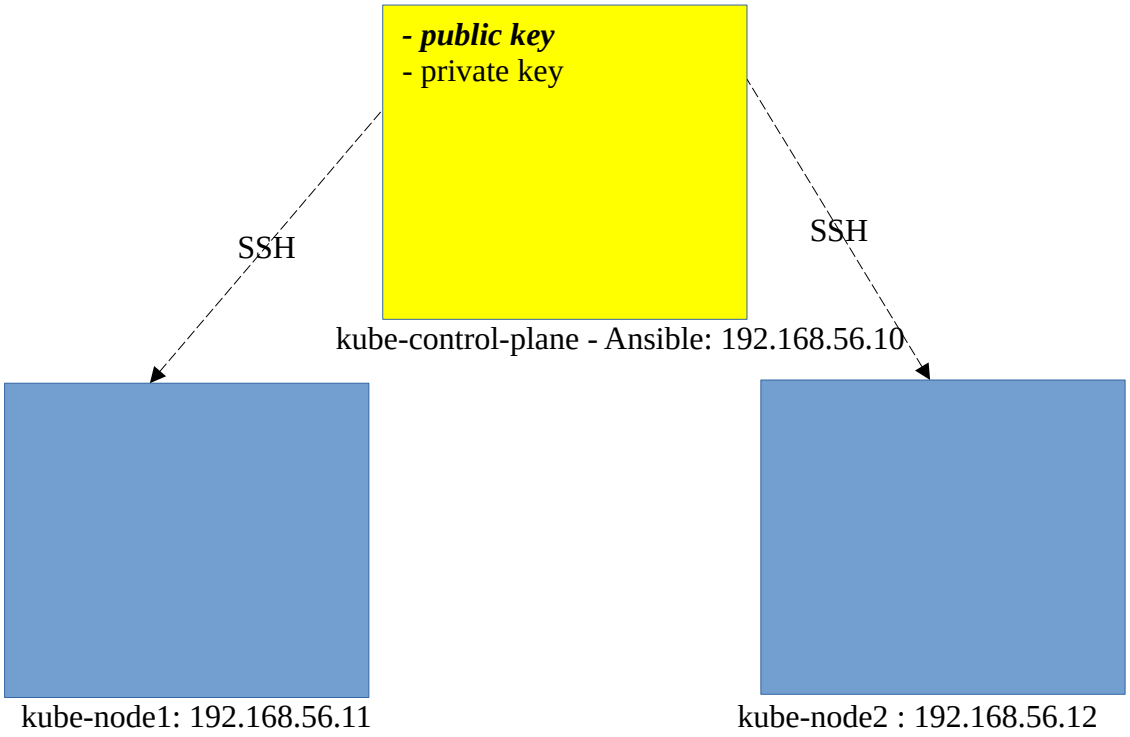
Lab8 - Ansible

Lab environment



Ansible ssh prepare

On ansible management node (Master) :

1. Check ansible version.

```
vagrant@k8s-master:~/Lab8_ansible$ ansible --version
ansible 2.9.6
  config file = /home/vagrant/Lab8_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
  executable location = /usr/bin/ansible
  python version = 3.8.10 (default, Nov 14 2022, 12:59:47) [GCC 9.4.0]
vagrant@k8s-master:~/Lab8_ansible$
```

2. Generate ssh keys with defaults parameters.

```
vagrant@kube-control-plane:~/Lab8_ansible$ 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:O+Dlst8gaa/889AayYlo94DYztMIcwMkHmRh0fXNEKg vagrant@kube-control-plane
The key's randomart image is:
+---[RSA 3072]----+
|.*+ ...o.        |
|+... .. +        |
|.o. .  . o       |
| ..E             |
|   .  . S        |
|  ooo+ B =       |
|  .+=+X @ .      |
|   +o+.B.B       |
|    o.++*oo      |
+----[SHA256]-----+
vagrant@kube-control-plane:~/Lab8_ansible$
vagrant@kube-control-plane:~/Lab8_ansible$
vagrant@kube-control-plane:~/Lab8_ansible$
vagrant@kube-control-plane:~/Lab8_ansible$ ls -l /home/vagrant/.ssh/
total 16
-rw------- 1 vagrant vagrant  389 Aug  8 10:14 authorized_keys
-rw------- 1 vagrant vagrant 2610 Aug  9 10:08 id_rsa
-rw-r--r-- 1 vagrant vagrant  580 Aug  9 10:08 id_rsa.pub
-rw-r--r-- 1 vagrant vagrant  444 Aug  9 09:19 known_hosts
vagrant@kube-control-plane:~/Lab8_ansible$
```

- Add the public key to the *authorized_keys* file on all ansible hosts.

```
vagrant@kube-control-plane:~/Lab8_ansible$ ssh-copy-id -i /home/vagrant/.ssh/id_rsa.pub vagrant@192.168.56.11
/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@192.168.56.11's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'vagrant@192.168.56.11'"
and check to make sure that only the key(s) you wanted were added.

vagrant@kube-control-plane:~/Lab8_ansible$
vagrant@kube-control-plane:~/Lab8_ansible$ ssh-copy-id -i /home/vagrant/.ssh/id_rsa.pub vagrant@192.168.56.12
/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@192.168.56.12's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'vagrant@192.168.56.12'"
and check to make sure that only the key(s) you wanted were added.

vagrant@kube-control-plane:~/Lab8_ansible$
```

3. Try ssh connection with keys, from ansible management node to ansible hosts.

```
vagrant@kube-control-plane:~/Lab8_ansible$ ssh 'vagrant@192.168.56.11'
Last login: Wed Aug  9 10:11:38 2023 from 10.0.2.2
vagrant@kube-node1:~$ logout
Connection to 192.168.56.11 closed.
vagrant@kube-control-plane:~/Lab8_ansible$ ssh 'vagrant@192.168.56.12'
Last login: Wed Aug  9 09:30:08 2023 from 192.168.56.10
vagrant@kube-node2:~$
vagrant@kube-node2:~$ logout
Connection to 192.168.56.12 closed.
vagrant@kube-control-plane:~/Lab8_ansible$
vagrant@kube-control-plane:~/Lab8_ansible$
```

4. Create inventory file to manager worker1 and worker2 as nodes group.

```
vagrant@kube-control-plane:~/Lab8_ansible$ cat inventaire
control-plane   ansible_connection=local        ansible_host=192.168.56.10
[nodes]
node1           ansible_host=192.168.56.11      ansible_user=vagrant
node2           ansible_host=192.168.56.12      ansible_user=vagrant
vagrant@kube-control-plane:~/Lab8_ansible$
```

Ansible ad hoc commands

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

```
vagrant@kube-control-plane:~/Lab8_ansible$ ansible all -m ping
control-plane | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
node1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
node2 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
vagrant@kube-control-plane:~/Lab8_ansible$
vagrant@kube-control-plane:~/Lab8_ansible$
```

6. Gathering facts about all hosts

```
vagrant@kube-control-plane:~/Lab8_ansible$ ansible nodes -m setup
node2 | SUCCESS => {
    "ansible_facts": {
        "ansible_all_ipv4_addresses": [
            "10.0.2.15",
            "10.244.2.0",
            "192.168.56.12"
        ],
        "ansible_all_ipv6_addresses": [
            "fe80::a00:27ff:fe8e:a077",
            "fe80::a00:27ff:fed6:1476"
        ],
        "ansible_apparmor": {
            "status": "enabled"
        },
        "ansible_architecture": "x86_64",
        "ansible_bios_date": "12/01/2006",
        "ansible_bios_version": "VirtualBox",
        "ansible_cmdline": {
            "BOOT_IMAGE": "/vmlinuz-5.4.0-139-generic",
            "biosdevname": "0",
            "net.ifnames": "0",
            "quiet": true,
```

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

```
vagrant@kube-control-plane:~/Lab8_ansible$ ansible nodes -m command -a "uptime"
node2 | CHANGED | rc=0 >>
 10:22:53 up  2:16,  1 user,  load average: 0.02, 0.04, 0.00
node1 | CHANGED | rc=0 >>
 10:22:51 up  2:18,  1 user,  load average: 0.00, 0.00, 0.00
vagrant@kube-control-plane:~/Lab8_ansible$
vagrant@kube-control-plane:~/Lab8_ansible$ ansible nodes -m shell -a "uptime"
node1 | CHANGED | rc=0 >>
 10:23:22 up  2:18,  1 user,  load average: 0.00, 0.00, 0.00
node2 | CHANGED | rc=0 >>
 10:23:24 up  2:17,  1 user,  load average: 0.01, 0.03, 0.00
vagrant@kube-control-plane:~/Lab8_ansible$
vagrant@kube-control-plane:~/Lab8_ansible$
```

8. Check the memory usage of hosts.

```
vagrant@kube-control-plane:~/Lab8_ansible$ ansible all -m setup -a 'filter=ansible_*_mb'
node1 | SUCCESS => {
    "ansible_facts": {
        "ansible_memfree_mb": 163,
        "ansible_memory_mb": {
            "nocache": {
                "free": 697,
                "used": 279
            },
            "real": {
                "free": 163,
                "total": 976,
                "used": 813
            },
            "swap": {
                "cached": 0,
                "free": 0,
                "total": 0,
                "used": 0
            }
        },
        "ansible_memtotal_mb": 976,
        "ansible_swapfree_mb": 0,
        "ansible_swaptotal_mb": 0,
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false
}
node2 | SUCCESS => {
    "ansible_facts": {
        "ansible_memfree_mb": 135,
        "ansible_memory_mb": {
            "nocache": {
                "free": 692,
                "used": 284
```

Ansible playbook

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

```
- name: play1
  hosts: all
  become: true
#  gather_facts: false
  tasks:
  - name: Installation d'apache2
    apt:
      name: apache2
      state: latest
```

- Apply the playbook.

```
vagrant@k8s-master:~/Lab8_ansible$ ansible-playbook playbook1.yaml

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

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

TASK [Installation d'apache2] **********************************************************************************
changed: [192.168.205.102]
changed: [192.168.205.101]
changed: [ansible]

PLAY RECAP *****************************************************************************************************
192.168.205.101            : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
192.168.205.102            : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ansible                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

vagrant@k8s-master:~/Lab8_ansible$
```

10. Add a condition to the playbook to provision only hosts of the Debian family.

```yaml
- name: play1
  hosts: all
  become: true
#  gather_facts: false
  tasks:
  - name: Installation d'apache2
    apt:
      name: apache2
      state: latest
    when: ansible_os_family=="Debian"
```

- Apply the playbook.

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

```yaml
- name: play1
  hosts: all
  become: true
#  gather_facts: false
  tasks:
  - name: Installation d'apache2 sur Debian/Ubuntu
    apt:
      name: apache2
      state: latest
    when: ansible_os_family=="Debian"
  - name: Installation d'apache2 sur RedHat/CentOS
    yum:
      name: httpd
      state: latest
    when: ansible_os_family=="RedHat"
```

```
vagrant@k8s-master:~/Lab8_ansible$ ansible-playbook playbook1.yaml

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

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

TASK [Installation d'apache2 sur Debian/Ubuntu] ****************************************************************
ok: [ansible]
ok: [192.168.205.102]
ok: [192.168.205.101]

TASK [Installation d'apache2 sur RedHat/CentOS] ****************************************************************
skipping: [ansible]
skipping: [192.168.205.101]
skipping: [192.168.205.102]

PLAY RECAP *****************************************************************************************************
192.168.205.101            : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.205.102            : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
ansible                    : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0

vagrant@k8s-master:~/Lab8_ansible$
```

Deploy Dockercoins to k8s

In this scenario we will use k8s module to deploy Dockercoins application on k8s cluster.

12. Create an inventory file called “inventory”.

- Add master, worker1 and worker2 nodes to inventory file.

```
ansible         ansible_host=192.168.205.100            ansible_connection=local
[nodes]
192.168.205.101         ansible_user=vagrant
192.168.205.102         ansible_user=vagrant
~
```

- Test ssh connection to every target node.

```
vagrant@k8s-master:~/Lab8_ansible$ ansible all -m ping
ansible | 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"
}
192.168.205.102 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
vagrant@k8s-master:~/Lab8_ansible$
```

13. Create *playbook_all.yml* that contain following task:

- Install k8s binaries and Disable swap on all target nodes

(See attached file *playbook_all.yml*)

- Apply the playbook and check if swap is disabled.

```
vagrant@k8s-master:~/Lab8_ansible$ ansible-playbook playbook_all.yml

PLAY [all] *********************************************************************************************************************************************

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

TASK [Installer des packages qui permettent à apt d'être utilisé sur HTTPS] ****************************************************************************
changed: [ansible]
changed: [192.168.205.101]
changed: [192.168.205.102]

TASK [Ajouter une clé de signature apt pour Docker] *****************************************************************************************************
ok: [ansible]
ok: [192.168.205.101]
ok: [192.168.205.102]

TASK [Ajouter un repo apt docker] **********************************************************************************************************************
changed: [192.168.205.102]
changed: [ansible]
changed: [192.168.205.101]

TASK [Supprimer config de containerd] ******************************************************************************************************************
changed: [ansible]
changed: [192.168.205.101]
changed: [192.168.205.102]

TASK [Installer containerd] ****************************************************************************************************************************
ok: [ansible]
ok: [192.168.205.101]
ok: [192.168.205.102]

TASK [Générer la config de containerd] *****************************************************************************************************************
changed: [ansible]
changed: [192.168.205.102]
changed: [192.168.205.101]

TASK [Install crictl et activer ip_forwarding] *********************************************************************************************************
[WARNING]: Consider using the get_url or uri module rather than running 'wget'.  If you need to use command because get_url or uri is
insufficient you can add 'warn: false' to this command task or set 'command_warnings=False' in ansible.cfg to get rid of this message.
changed: [ansible]
changed: [192.168.205.101]
changed: [192.168.205.102]

TASK [Supprimer le swap de /etc/fstab] *****************************************************************************************************************
ok: [ansible] => (item=swap)
ok: [192.168.205.102] => (item=swap)
ok: [192.168.205.101] => (item=swap)
```

14. Create *playbook_master.yml* that contain all following tagged tasks:

- Copy dockercoins.yml to master
- Sleep 30 seconds
- Deploy dockercoins to k8s cluster and register deployment result to deploy_result variable
- Display deployment result on screen
- Delete dockercoins from k8s cluster.

(See attached files *playbook_master.yml and playbook_workers*)

```
vagrant@k8s-master:~/Lab8_ansible$ ansible-playbook playbook_master.yml --tags reset_cluster

PLAY [ansible] *****************************************************************************************************************

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

TASK [Reset cluster] ***********************************************************************************************************
changed: [ansible]

PLAY RECAP *********************************************************************************************************************
ansible                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

vagrant@k8s-master:~/Lab8_ansible$
```

```
vagrant@k8s-master:~/Lab8_ansible$
vagrant@k8s-master:~/Lab8_ansible$
vagrant@k8s-master:~/Lab8_ansible$ ansible-playbook playbook_master.yml --skip-tags reset_cluster

PLAY [ansible] *****************************************************************************************************************

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

TASK [Configurer ip du node] ***************************************************************************************************
ok: [ansible]

TASK [Initialiser le cluster Kubernetes à l'aide de kubeadm] *******************************************************************
changed: [ansible]

TASK [Résultat d'init cluster] *************************************************************************************************
ok: [ansible] => {
    "msg": [
        "[init] Using Kubernetes version: v1.27.3",
        "[preflight] Running pre-flight checks",
        "[preflight] Pulling images required for setting up a Kubernetes cluster",
        "[preflight] This might take a minute or two, depending on the speed of your internet connection",
        "[preflight] You can also perform this action in beforehand using 'kubeadm config images pull'",
        "[certs] Using certificateDir folder \"/etc/kubernetes/pki\"",
        "[certs] Generating \"ca\" certificate and key",
        "[certs] Generating \"apiserver\" certificate and key",
        "[certs] apiserver serving cert is signed for DNS names [k8s-master kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster.local] and IPs [10.96.0.1 192.168.205.100]",
        "[certs] Generating \"apiserver-kubelet-client\" certificate and key",
        "[certs] Generating \"front-proxy-ca\" certificate and key",
        "[certs] Generating \"front-proxy-client\" certificate and key",
        "[certs] Generating \"etcd/ca\" certificate and key",
        "[certs] Generating \"etcd/server\" certificate and key",
        "[certs] etcd/server serving cert is signed for DNS names [k8s-master localhost] and IPs [192.168.205.100 127.0.0.1 ::1]",
        "[certs] Generating \"etcd/peer\" certificate and key",
        "[certs] etcd/peer serving cert is signed for DNS names [k8s-master localhost] and IPs [192.168.205.100 127.0.0.1 ::1]",
        "[certs] Generating \"etcd/healthcheck-client\" certificate and key",
        "[certs] Generating \"apiserver-etcd-client\" certificate and key",
        "[certs] Generating \"sa\" key and public key",
        "[kubeconfig] Using kubeconfig folder \"/etc/kubernetes\"",
        "[kubeconfig] Writing \"admin.conf\" kubeconfig file",
        "[kubeconfig] Writing \"kubelet.conf\" kubeconfig file",
        "[kubeconfig] Writing \"controller-manager.conf\" kubeconfig file",
        "[kubeconfig] Writing \"scheduler.conf\" kubeconfig file",
        "[kubelet-start] Writing kubelet environment file with flags to file \"/var/lib/kubelet/kubeadm-flags.env\"",
        "[kubelet-start] Writing kubelet configuration to file \"/var/lib/kubelet/config.yaml\"",
        "[kubelet-start] Starting the kubelet",
        "[control-plane] Using manifest folder \"/etc/kubernetes/manifests\"",
        "[control-plane] Creating static Pod manifest for \"kube-apiserver\"",
        "[control-plane] Creating static Pod manifest for \"kube-controller-manager\"",
        "[control-plane] Creating static Pod manifest for \"kube-scheduler\"",
        "[etcd] Creating static Pod manifest for local etcd in \"/etc/kubernetes/manifests\"",
        "[wait-control-plane] Waiting for the kubelet to boot up the control plane as static Pods from directory \"/etc/kubernetes/manifests\". This can take up to 4m0s",
        "[apiclient] All control plane components are healthy after 15.004742 seconds",
        "[upload-config] Storing the configuration used in ConfigMap \"kubeadm-config\" in the \"kube-system\" Namespace",
        "[kubelet] Creating a ConfigMap \"kubelet-config\" in namespace kube-system with the configuration for the kubelets in the cluster",
        "[upload-certs] Skipping phase. Please see --upload-certs",
        "[mark-control-plane] Marking the node k8s-master as control-plane by adding the labels: [node-role.kubernetes.io/control-plane node.kubernetes.io/exclude-from-external-load-balancers]",
        "[mark-control-plane] Marking the node k8s-master as control-plane by adding the taints [node-role.kubernetes.io/control-plane:NoSchedule]",
        "[bootstrap-token] Using token: gyr6y9.595r88ubggo554gu",
```

```
kubeadm join 192.168.205.100:6443 --token pa60na.1f9pydcz2pnegr9z --discovery-token-ca-cert-hash sha256:e8f6e9b2d686b7fab068a581b59d29d917bb7ebb138417dcf1fd2c863c21b766 vagrant@k8s-m
aster:~/Lab8_ansible$ logout
brahim@Training:~/k8s-lab$ vagrant ssh k8s-worker1
Last login: Mon Jun 19 10:18:50 2023 from 10.0.2.2
vagrant@k8s-worker1:~$ sudo kubeadm reset
W0619 10:20:02.032947   42443 preflight.go:56] [reset] WARNING: Changes made to this host by 'kubeadm init' or 'kubeadm join' will be reverted.
[reset] Are you sure you want to proceed? [y/N]: y
[preflight] Running pre-flight checks
W0619 10:20:03.697545   42443 removeetcdmember.go:106] [reset] No kubeadm config, using etcd pod spec to get data directory
[reset] Deleted contents of the etcd data directory: /var/lib/etcd
[reset] Stopping the kubelet service
[reset] Unmounting mounted directories in "/var/lib/kubelet"
[reset] Deleting contents of directories: [/etc/kubernetes/manifests /var/lib/kubelet /etc/kubernetes/pki]
[reset] Deleting files: [/etc/kubernetes/admin.conf /etc/kubernetes/kubelet.conf /etc/kubernetes/bootstrap-kubelet.conf /etc/kubernetes/controller-manager.conf /etc/kubernetes/schedu
ler.conf]

The reset process does not clean CNI configuration. To do so, you must remove /etc/cni/net.d

The reset process does not reset or clean up iptables rules or IPVS tables.
If you wish to reset iptables, you must do so manually by using the "iptables" command.

If your cluster was setup to utilize IPVS, run ipvsadm --clear (or similar)
to reset your system's IPVS tables.

The reset process does not clean your kubeconfig files and you must remove them manually.
Please, check the contents of the $HOME/.kube/config file.
vagrant@k8s-worker1:~$ sudo kubeadm join 192.168.205.100:6443 --token pa60na.1f9pydcz2pnegr9z --discovery-token-ca-cert-hash sha256:e8f6e9b2d686b7fab068a581b59d29d917bb7ebb138417dcf1
fd2c863c21b766
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap...

This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.

vagrant@k8s-worker1:~$
```

```
brahim@Training:~/k8s-lab$ vagrant ssh k8s-master
Last login: Mon Jun 19 10:19:36 2023 from 10.0.2.2
vagrant@k8s-master:~$ kubectl get node -o wide
NAME          STATUS   ROLES           AGE     VERSION   INTERNAL-IP       EXTERNAL-IP   OS-IMAGE             KERNEL-VERSION      CONTAINER-RU
NTIME
k8s-master    Ready    control-plane   6m57s   v1.27.3   192.168.205.100   <none>        Ubuntu 20.04.5 LTS   5.4.0-139-generic   containerd:/
/1.6.21
k8s-worker1   Ready    <none>          95s     v1.27.3   192.168.205.101   <none>        Ubuntu 20.04.5 LTS   5.4.0-139-generic   containerd:/
/1.6.21
k8s-worker2   Ready    <none>          19s     v1.27.3   192.168.205.102   <none>        Ubuntu 20.04.5 LTS   5.4.0-139-generic   containerd:/
/1.6.21
vagrant@k8s-master:~$
```

15. Create and apply the playbook to deploy dockercoins. Check if dockercoins has been deployed.

```yaml
- hosts: ansible
  gather_facts: false
  tasks:
  - name: apply manifest yaml
    shell: |
      kubectl apply -f dockercoins.yaml
      sleep 30
      kubectl get all -n dockercoins
    register: dockercoins_deploy
  - name: Verify dockercoins deployment
    debug:
      msg: "{{ dockercoins_deploy.stdout_lines | list }}"
```

```
vagrant@k8s-master:~/Lab8_ansible$ ansible-playbook playbook_dockercoins.yml

PLAY [ansible] *****************************************************************************************

TASK [apply manifest yaml] *****************************************************************************
changed: [ansible]

TASK [Verify dockercoins deployment] *******************************************************************
ok: [ansible] => {
    "msg": [
        "namespace/dockercoins created",
        "deployment.apps/worker created",
        "deployment.apps/rng created",
        "deployment.apps/hasher created",
        "deployment.apps/redis created",
        "deployment.apps/webui created",
        "service/rng created",
        "service/hasher created",
        "service/redis created",
        "service/webui created",
        "NAME                          READY   STATUS    RESTARTS   AGE",
        "pod/hasher-7f9d944db9-nvz6d   1/1     Running   0          31s",
        "pod/redis-78579d7b98-9js9m    1/1     Running   0          31s",
        "pod/rng-544477487c-6h86f      1/1     Running   0          32s",
        "pod/webui-c9697458-j8gmm      1/1     Running   0          31s",
        "pod/worker-5f7877988-wrls5    1/1     Running   0          32s",
        "",
        "NAME             TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE",
        "service/hasher   ClusterIP   10.105.149.194   <none>        80/TCP         31s",
        "service/redis    ClusterIP   10.101.124.115   <none>        6379/TCP       30s",
        "service/rng      ClusterIP   10.103.243.155   <none>        80/TCP         31s",
        "service/webui    NodePort    10.100.33.238    <none>        80:30001/TCP   30s",
        "",
        "NAME                     READY   UP-TO-DATE   AVAILABLE   AGE",
        "deployment.apps/hasher   1/1     1            1           32s",
        "deployment.apps/redis    1/1     1            1           31s",
        "deployment.apps/rng      1/1     1            1           32s",
        "deployment.apps/webui    1/1     1            1           31s",
        "deployment.apps/worker   1/1     1            1           32s",
        "",
        "NAME                                 DESIRED   CURRENT   READY   AGE",
        "replicaset.apps/hasher-7f9d944db9    1         1         1       31s",
        "replicaset.apps/redis-78579d7b98     1         1         1       31s",
```

- Using the same playbook, delete dockercoins from k8s cluster.

```
- hosts: ansible
  gather_facts: false
  tasks:
  - name: deploy dockercoins application on k8s
    shell: |
      kubectl apply -f dockercoins.yaml
      sleep 30
      kubectl get all -n dockercoins
    register: dockercoins_deploy
    tags:
    - deploy_dockercoins
  - name: remove dockercoins application from k8s
    shell: |
      kubectl delete -f dockercoins.yaml
      sleep 30
      kubectl get all -n dockercoins
    register: dockercoins_deploy
    tags:
    - remove_dockercoins

  - name: Verify dockercoins deployment
    debug:
      msg: "{{ dockercoins_deploy.stdout_lines | list }}"
    tags:
    - deploy_dockercoins
    - remove_dockercoins
```

```
vagrant@k8s-master:~/Lab8_ansible$ ansible-playbook playbook_dockercoins.yml --tags=remove_dockercoins

PLAY [ansible] *****************************************************************************************************

TASK [remove dockercoins application from k8s] *********************************************************************
changed: [ansible]

TASK [Verify dockercoins deployment] *******************************************************************************
ok: [ansible] => {
    "msg": []
}

PLAY RECAP *********************************************************************************************************
ansible                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

vagrant@k8s-master:~/Lab8_ansible$
```