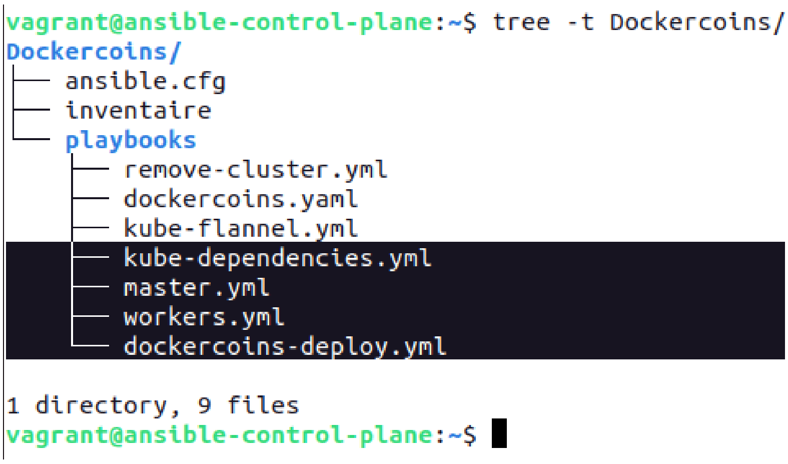
Lab 8 - Ansible

Dans ce Lab, on va utiliser Ansible pour créer un cluster k8s et déployer l'application Dockercoins. Les 4 playbooks qu'on va appliquer :

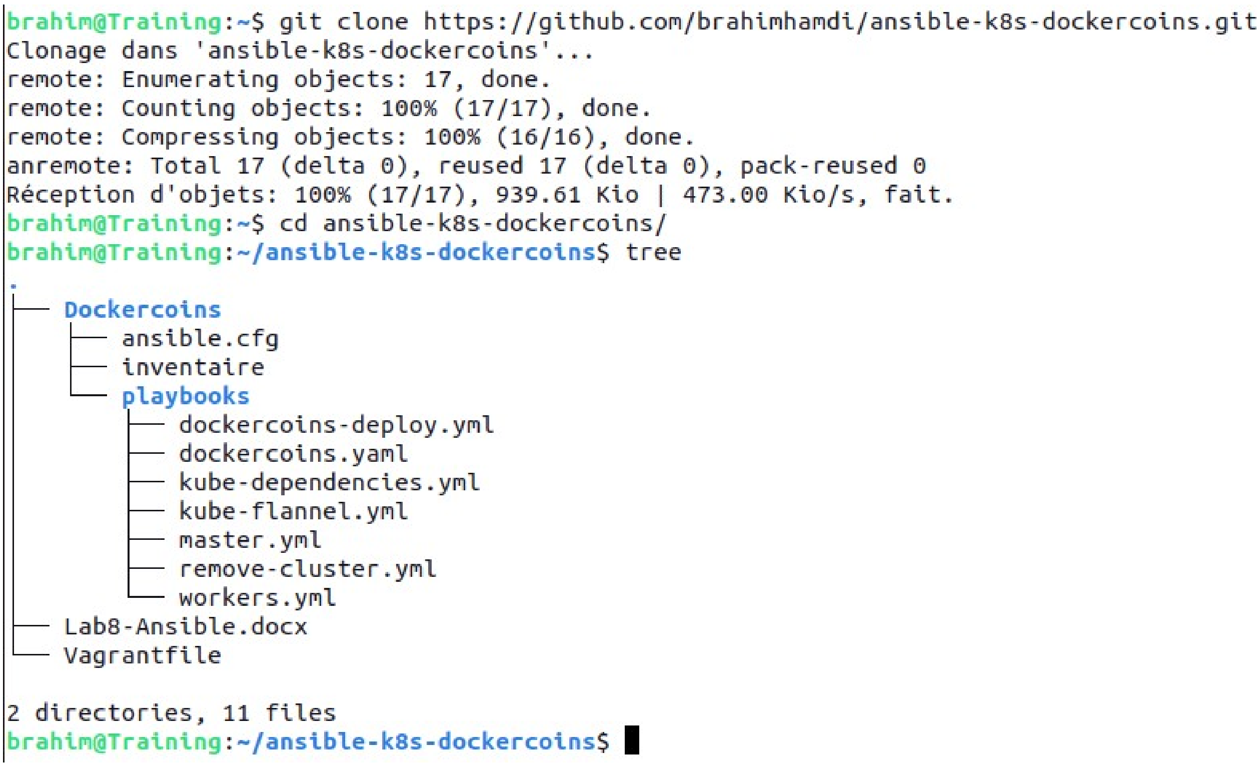
- Installation des pré-requis sur les 3 VMs du cluster (master + 2 nodes) : ***kube-dependencies.yml***
- Initialisation du cluster sur master : ***master.yml***
- Joindre les 2 workers au cluster : ***workers.yml***
- Déployer l'application Dockercoins sur le cluster : ***dockercoins-deploy.yml***



```
vagrant@ansible-control-plane:~$ tree -t Dockercoins/
Dockercoins/
├── ansible.cfg
├── inventaire
└── playbooks
    ├── remove-cluster.yml
    ├── dockercoins.yaml
    ├── kube-flannel.yml
    ├── kube-dependencies.yml
    ├── master.yml
    ├── workers.yml
    └── dockercoins-deploy.yml

1 directory, 9 files
vagrant@ansible-control-plane:~$
```

Préparation de l'environnement

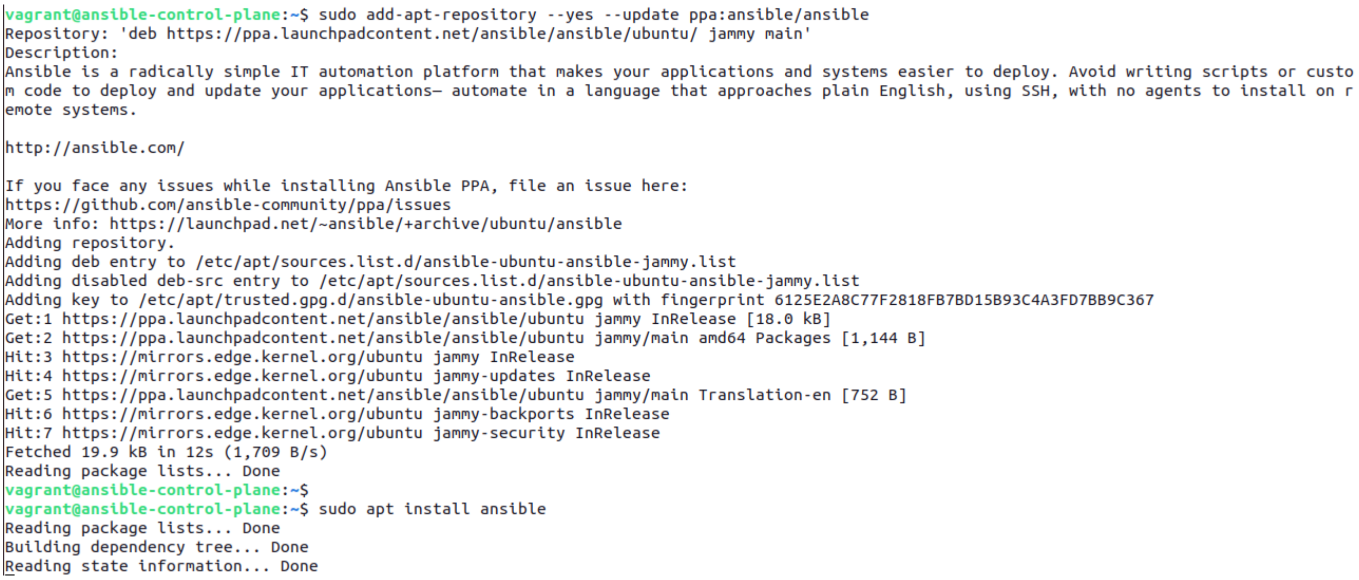
1. Git cloner le dépôt: https://github.com/brahimhamdi/ansible-k8s-dockercoins.git



```
brahim@Training:~$ git clone https://github.com/brahimhamdi/ansible-k8s-dockercoins.git
Clonage dans 'ansible-k8s-dockercoins'...
remote: Enumerating objects: 17, done.
remote: Counting objects: 100% (17/17), done.
remote: Compressing objects: 100% (16/16), done.
anremote: Total 17 (delta 0), reused 17 (delta 0), pack-reused 0
Réception d'objets: 100% (17/17), 939.61 Kio | 473.00 Kio/s, fait.
brahim@Training:~$ cd ansible-k8s-dockercoins/
brahim@Training:~/ansible-k8s-dockercoins$ tree
.
├── Dockercoins
│   ├── ansible.cfg
│   ├── inventaire
│   └── playbooks
│       ├── dockercoins-deploy.yml
│       ├── dockercoins.yaml
│       ├── kube-dependencies.yml
│       ├── kube-flannel.yml
│       ├── master.yml
│       ├── remove-cluster.yml
│       └── workers.yml
├── Lab8-Ansible.docx
└── Vagrantfile

2 directories, 11 files
brahim@Training:~/ansible-k8s-dockercoins$ 
```

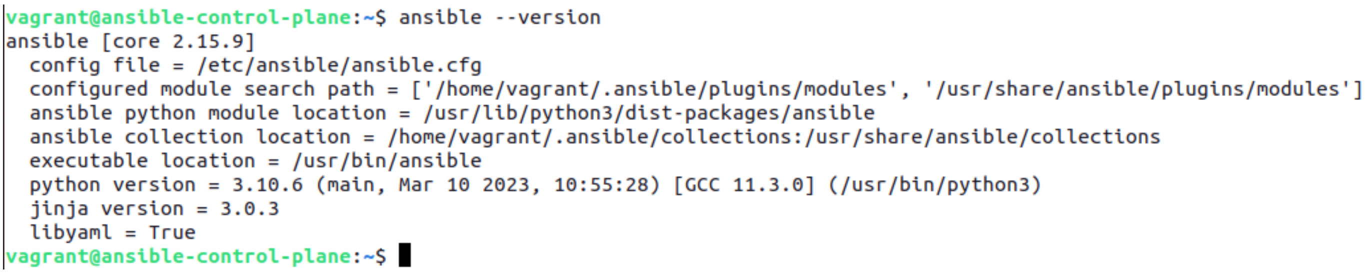
2. Déployer l'environnement Vagrant en utilisant la commande ***vagrant up*** (Pour libérer les ressources arrêter toutes les VMs des Labs précédents avant).

3. Installer ansible sur la VM *ansible-control-plane*. Quel est la version d'ansible ?



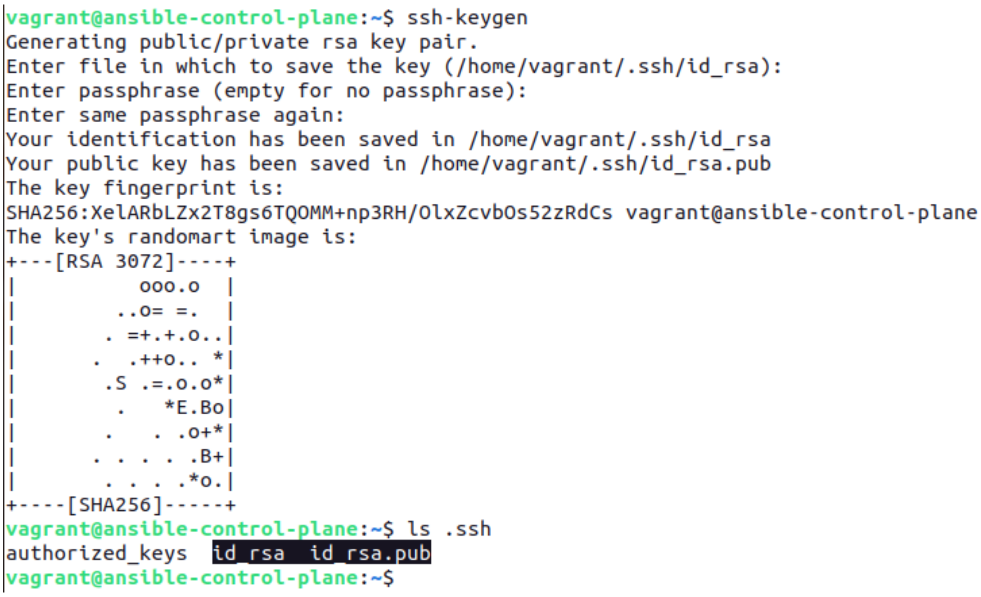
```
vagrant@ansible-control-plane:~$ sudo add-apt-repository --yes --update ppa:ansible/ansible
Repository: 'deb https://ppa.launchpadcontent.net/ansible/ansible/ubuntu/ jammy main'
Description:
Ansible is a radically simple IT automation platform that makes your applications and systems easier to deploy. Avoid writing scripts or custo
m code to deploy and update your applications— automate in a language that approaches plain English, using SSH, with no agents to install on r
emote systems.

http://ansible.com/

If you face any issues while installing Ansible PPA, file an issue here:
https://github.com/ansible-community/ppa/issues
More info: https://launchpad.net/~ansible/+archive/ubuntu/ansible
Adding repository.
Adding deb entry to /etc/apt/sources.list.d/ansible-ubuntu-ansible-jammy.list
Adding disabled deb-src entry to /etc/apt/sources.list.d/ansible-ubuntu-ansible-jammy.list
Adding key to /etc/apt/trusted.gpg.d/ansible-ubuntu-ansible.gpg with fingerprint 6125E2A8C77F2818FB7BD15B93C4A3FD7BB9C367
Get:1 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu jammy InRelease [18.0 kB]
Get:2 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu jammy/main amd64 Packages [1,144 B]
Hit:3 https://mirrors.edge.kernel.org/ubuntu jammy InRelease
Hit:4 https://mirrors.edge.kernel.org/ubuntu jammy-updates InRelease
Get:5 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu jammy/main Translation-en [752 B]
Hit:6 https://mirrors.edge.kernel.org/ubuntu jammy-backports InRelease
Hit:7 https://mirrors.edge.kernel.org/ubuntu jammy-security InRelease
Fetched 19.9 kB in 12s (1,709 B/s)
Reading package lists... Done
vagrant@ansible-control-plane:~$
vagrant@ansible-control-plane:~$ sudo apt install ansible
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

…



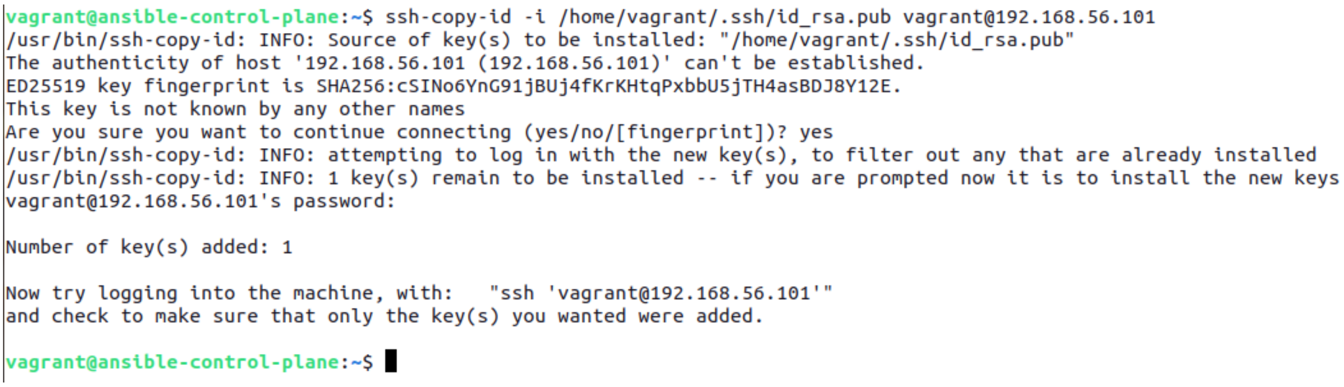
```
vagrant@ansible-control-plane:~$ ansible --version
ansible [core 2.15.9]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/vagrant/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  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
vagrant@ansible-control-plane:~$ 
```

4. Générer une paire de clés RSA sur la VM *ansible-control-plane*

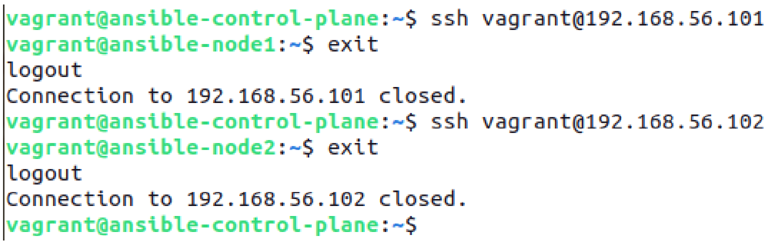


```
vagrant@ansible-control-plane:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/vagrant/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/vagrant/.ssh/id_rsa
Your public key has been saved in /home/vagrant/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:XelARbLZx2T8gs6TQOMM+np3RH/OlxZcvbOs52zRdCs vagrant@ansible-control-plane
The key's randomart image is:
+---[RSA 3072]----+
|          ooo.o  |
|        ..o= =.  |
|       . =+.+.o..|
|      .  .++o.. *|
|       .S .=.o.o*|
|        .   *E.Bo|
|       .   . .o+*|
|      . . . . .B+|
|       . . . .*o.|
+----[SHA256]-----+
vagrant@ansible-control-plane:~$ ls .ssh
authorized_keys  id_rsa  id_rsa.pub
vagrant@ansible-control-plane:~$
```

5. Copier la clé publique sur les 2 nodes (*ansible-node1* et *ansible-node2*) et tester la connexion SSH

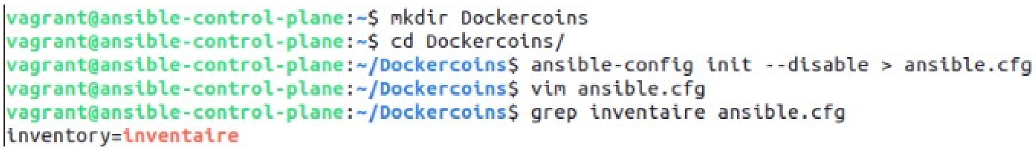


```
vagrant@ansible-control-plane:~$ ssh-copy-id -i /home/vagrant/.ssh/id_rsa.pub vagrant@192.168.56.101
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/vagrant/.ssh/id_rsa.pub"
The authenticity of host '192.168.56.101 (192.168.56.101)' can't be established.
ED25519 key fingerprint is SHA256:cSINo6YnG91jBUj4fKrKHtqPxbbU5jTH4asBDJ8Y12E.
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@192.168.56.101's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'vagrant@192.168.56.101'"
and check to make sure that only the key(s) you wanted were added.

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



```
vagrant@ansible-control-plane:~$ ssh vagrant@192.168.56.101
vagrant@ansible-node1:~$ exit
logout
Connection to 192.168.56.101 closed.
vagrant@ansible-control-plane:~$ ssh vagrant@192.168.56.102
vagrant@ansible-node2:~$ exit
logout
Connection to 192.168.56.102 closed.
vagrant@ansible-control-plane:~$
```

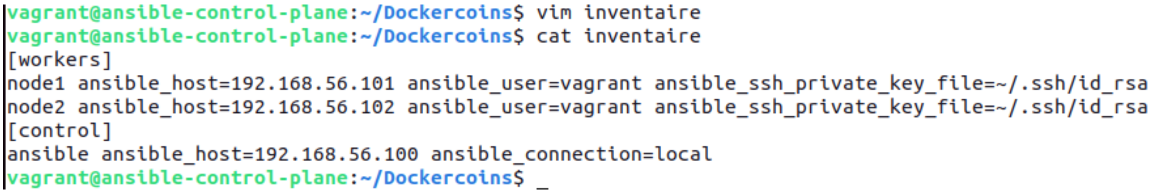
6. Créer le repertoire de travail *Dockercoins* et y ajouter le fichier *ansible.cfg*. Dans ansible.cfg, modifier le nom du fichier inventaire (de ***/etc/ansible/hosts*** à ***inventaire***) et supprimer le commentaire (**;**)



```
vagrant@ansible-control-plane:~$ mkdir Dockercoins
vagrant@ansible-control-plane:~$ cd Dockercoins/
vagrant@ansible-control-plane:~/Dockercoins$ ansible-config init --disable > ansible.cfg
vagrant@ansible-control-plane:~/Dockercoins$ vim ansible.cfg
vagrant@ansible-control-plane:~/Dockercoins$ grep inventaire ansible.cfg
inventory=inventaire
```

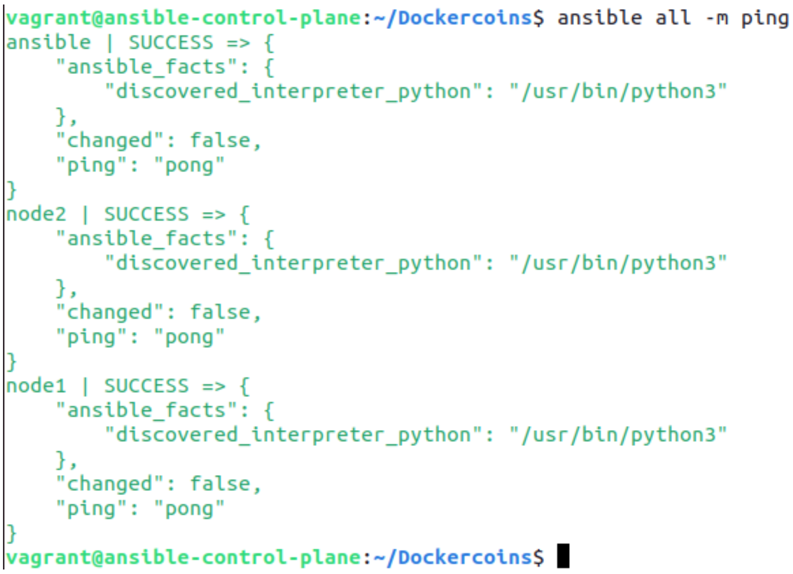
7. Ajouter le fichier *inventaire* au répertoire Dockercoins. Dans ce fichier, on va définir 2 groupes d'hôtes:

- control: hôte **Ansible** (en même temps master k8s pour simplifier les choses)
- workers: hôtes **node1** et **node2**



```
vagrant@ansible-control-plane:~/Dockercoins$ vim inventaire
vagrant@ansible-control-plane:~/Dockercoins$ cat inventaire
[workers]
node1 ansible_host=192.168.56.101 ansible_user=vagrant ansible_ssh_private_key_file=~/.ssh/id_rsa
node2 ansible_host=192.168.56.102 ansible_user=vagrant ansible_ssh_private_key_file=~/.ssh/id_rsa
[control]
ansible ansible_host=192.168.56.100 ansible_connection=local
vagrant@ansible-control-plane:~/Dockercoins$ _
```

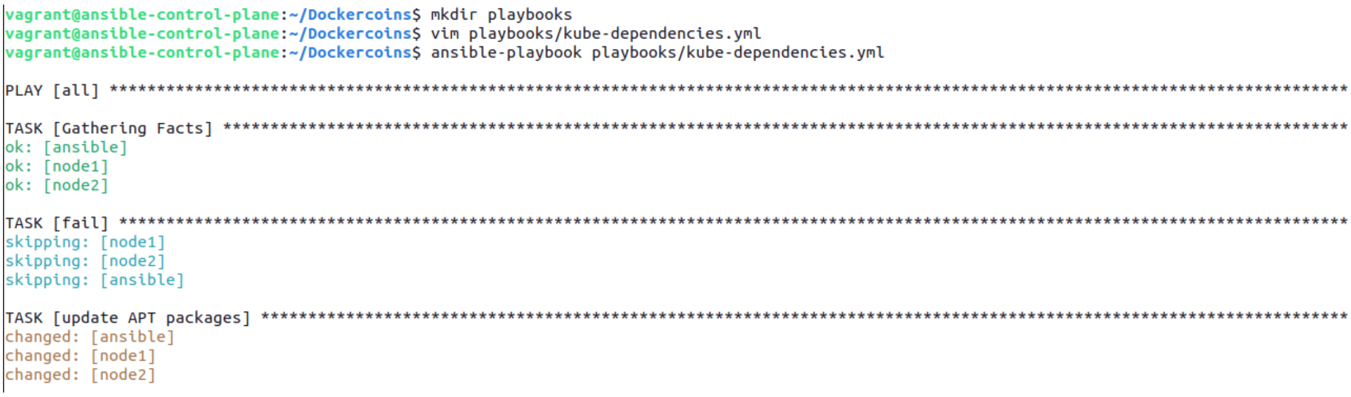
8. Tester la connexion avec la commade ad-hoc en utilisant le module *ping*



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

Initialisation du cluster

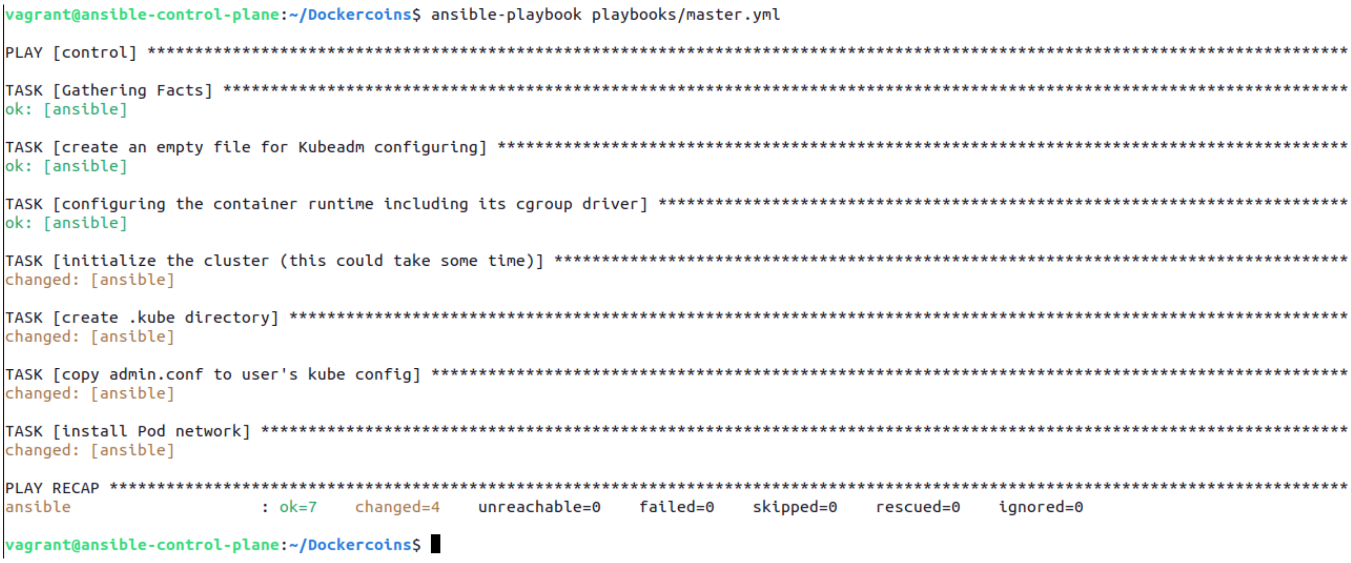
9. Dans le répertoire *playbooks*, créer et appliquer le premier playbook ***kube-dependencies.yml*** qui permet de préparer tous les noeuds du futur cluster kubernetes: charger modules, activer fonctionnalités, installer les paquets (containerd, kubelet, kubectl, kubeadm, …), désactiver le swap, etc ...



```
vagrant@ansible-control-plane:~/Dockercoins$ mkdir playbooks
vagrant@ansible-control-plane:~/Dockercoins$ vim playbooks/kube-dependencies.yml
vagrant@ansible-control-plane:~/Dockercoins$ ansible-playbook playbooks/kube-dependencies.yml

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

TASK [Gathering Facts] ***********************************************************************************************************************
ok: [ansible]
ok: [node1]
ok: [node2]

TASK [fail] **********************************************************************************************************************************
skipping: [node1]
skipping: [node2]
skipping: [ansible]

TASK [update APT packages] *******************************************************************************************************************
changed: [ansible]
changed: [node1]
changed: [node2]
```

…

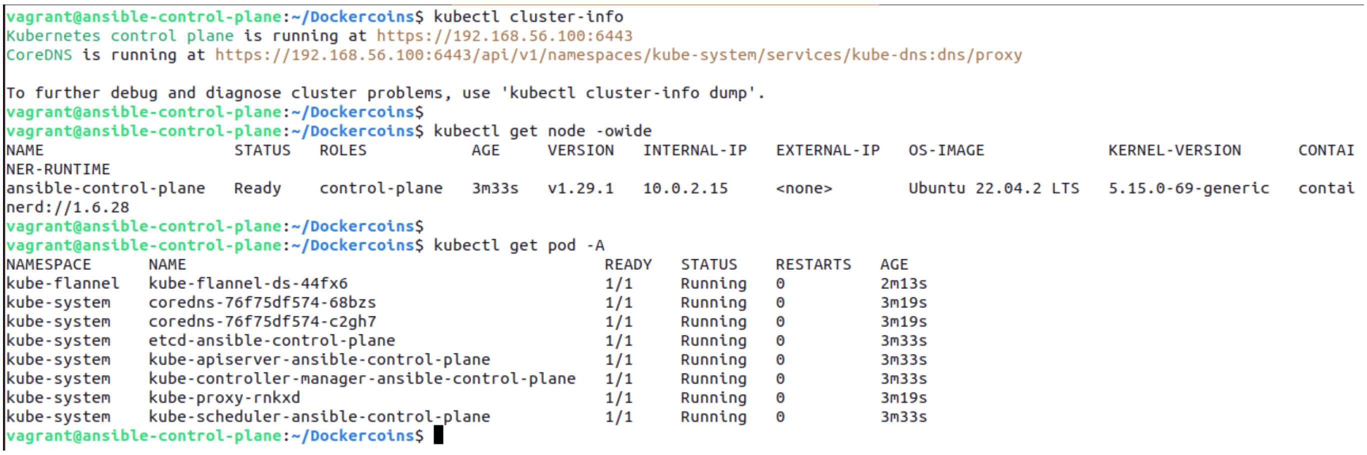
10. Sous le répertoire *playbooks*, créer et appliquer le deuxième playbook ***master.yml*** qui permet d’initialiser le cluster k8s sur le noeud master.



```
vagrant@ansible-control-plane:~/Dockercoins$ ansible-playbook playbooks/master.yml

PLAY [control] ******************************************************************************************************************************

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

TASK [create an empty file for Kubeadm configuring] *****************************************************************************************
ok: [ansible]

TASK [configuring the container runtime including its cgroup driver] ************************************************************************
ok: [ansible]

TASK [initialize the cluster (this could take some time)] ***********************************************************************************
changed: [ansible]

TASK [create .kube directory] ***************************************************************************************************************
changed: [ansible]

TASK [copy admin.conf to user's kube config] ************************************************************************************************
changed: [ansible]

TASK [install Pod network] ******************************************************************************************************************
changed: [ansible]

PLAY RECAP **********************************************************************************************************************************
ansible                    : ok=7    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

vagrant@ansible-control-plane:~/Dockercoins$ █
```

…

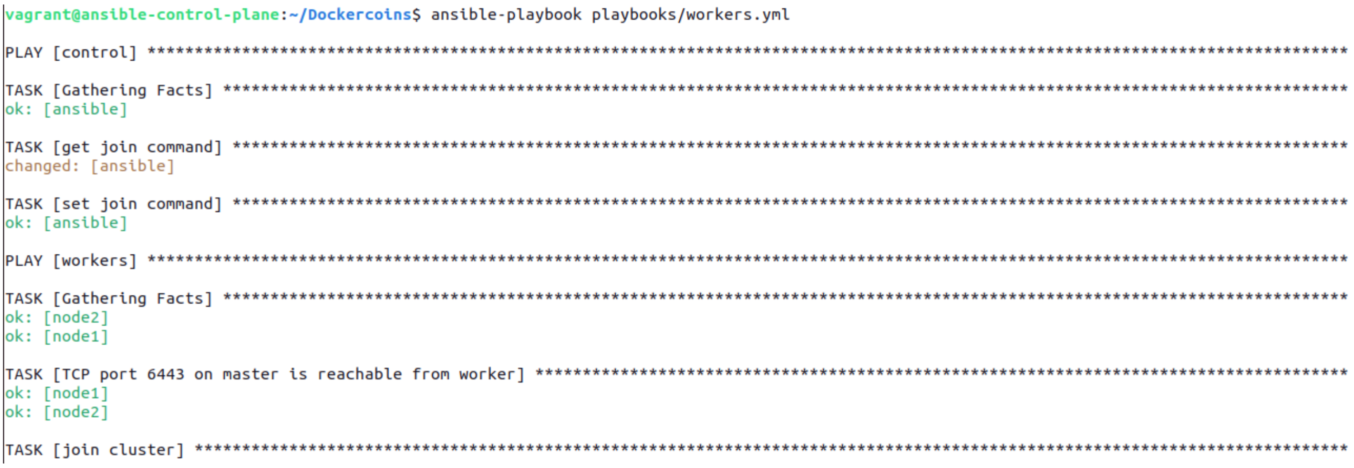
11. Vérifier que le cluster a été bien initialisé et que le master est “*Ready”*.



```
vagrant@ansible-control-plane:~/Dockercoins$ kubectl cluster-info
Kubernetes control plane is running at https://192.168.56.100:6443
CoreDNS is running at https://192.168.56.100:6443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
vagrant@ansible-control-plane:~/Dockercoins$
vagrant@ansible-control-plane:~/Dockercoins$ kubectl get node -owide
NAME                    STATUS   ROLES           AGE     VERSION   INTERNAL-IP   EXTERNAL-IP   OS-IMAGE             KERNEL-VERSION      CONTAI
NER-RUNTIME
ansible-control-plane   Ready    control-plane   3m33s   v1.29.1   10.0.2.15     <none>        Ubuntu 22.04.2 LTS   5.15.0-69-generic   contai
nerd://1.6.28
vagrant@ansible-control-plane:~/Dockercoins$
vagrant@ansible-control-plane:~/Dockercoins$ kubectl get pod -A
NAMESPACE      NAME                                            READY   STATUS    RESTARTS   AGE
kube-flannel   kube-flannel-ds-44fx6                           1/1     Running   0          2m13s
kube-system    coredns-76f75df574-68bzs                        1/1     Running   0          3m19s
kube-system    coredns-76f75df574-c2gh7                        1/1     Running   0          3m19s
kube-system    etcd-ansible-control-plane                      1/1     Running   0          3m33s
kube-system    kube-apiserver-ansible-control-plane            1/1     Running   0          3m33s
kube-system    kube-controller-manager-ansible-control-plane   1/1     Running   0          3m33s
kube-system    kube-proxy-rnkxd                                1/1     Running   0          3m19s
kube-system    kube-scheduler-ansible-control-plane            1/1     Running   0          3m33s
vagrant@ansible-control-plane:~/Dockercoins$ █
```

Joindre les workers

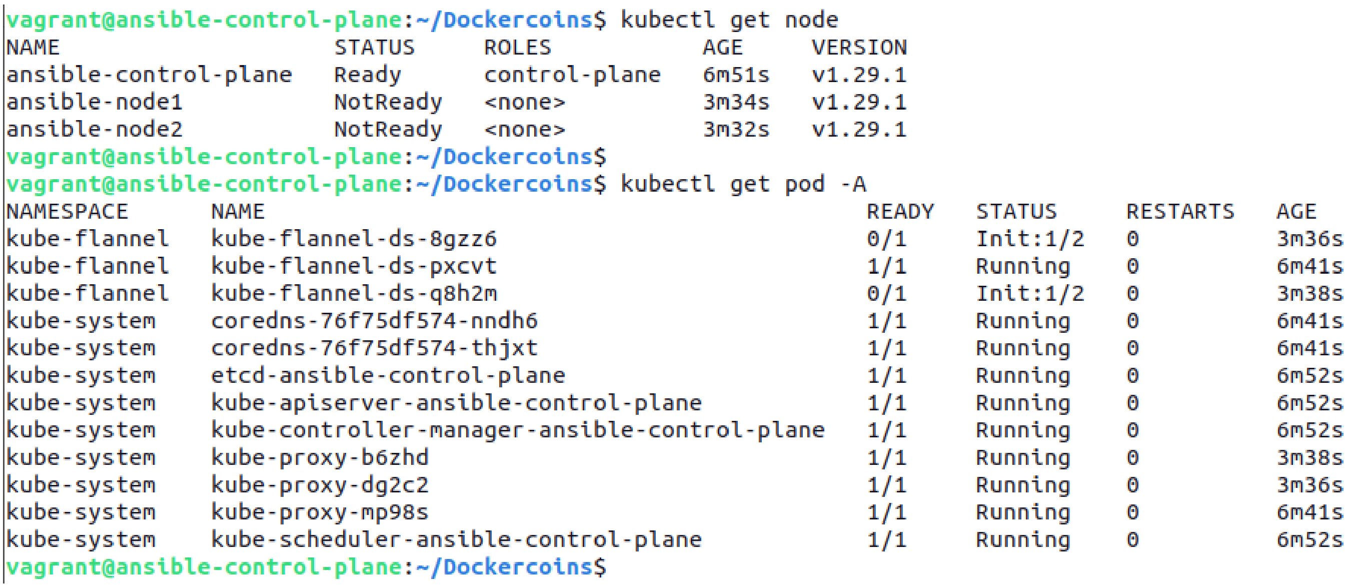
12. Créer et appliquer le playbook ***workers.yml*** pour joindre *node1* et *node2* au cluster.



```
vagrant@ansible-control-plane:~/Dockercoins$ ansible-playbook playbooks/workers.yml

PLAY [control] ******************************************************************************************************

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

TASK [get join command] *********************************************************************************************
changed: [ansible]

TASK [set join command] *********************************************************************************************
ok: [ansible]

PLAY [workers] ******************************************************************************************************

TASK [Gathering Facts] **********************************************************************************************
ok: [node2]
ok: [node1]

TASK [TCP port 6443 on master is reachable from worker] *************************************************************
ok: [node1]
ok: [node2]

TASK [join cluster] *************************************************************************************************
```

…

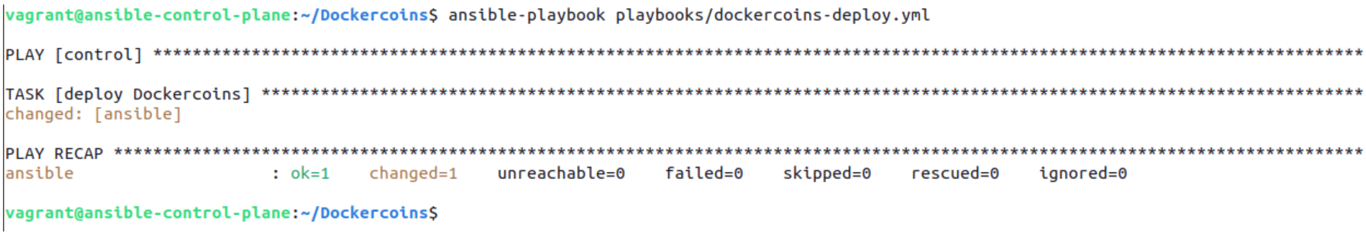
13. Vérifier que les 2 nodes workers sont *Ready*.



```
vagrant@ansible-control-plane:~/Dockercoins$ kubectl get node
NAME                    STATUS     ROLES           AGE     VERSION
ansible-control-plane   Ready      control-plane   6m51s   v1.29.1
ansible-node1           NotReady   <none>          3m34s   v1.29.1
ansible-node2           NotReady   <none>          3m32s   v1.29.1
vagrant@ansible-control-plane:~/Dockercoins$
vagrant@ansible-control-plane:~/Dockercoins$ kubectl get pod -A
NAMESPACE      NAME                                            READY   STATUS     RESTARTS   AGE
kube-flannel   kube-flannel-ds-8gzz6                           0/1     Init:1/2   0          3m36s
kube-flannel   kube-flannel-ds-pxcvt                           1/1     Running    0          6m41s
kube-flannel   kube-flannel-ds-q8h2m                           0/1     Init:1/2   0          3m38s
kube-system    coredns-76f75df574-nndh6                        1/1     Running    0          6m41s
kube-system    coredns-76f75df574-thjxt                        1/1     Running    0          6m41s
kube-system    etcd-ansible-control-plane                      1/1     Running    0          6m52s
kube-system    kube-apiserver-ansible-control-plane            1/1     Running    0          6m52s
kube-system    kube-controller-manager-ansible-control-plane   1/1     Running    0          6m52s
kube-system    kube-proxy-b6zhd                                1/1     Running    0          3m38s
kube-system    kube-proxy-dg2c2                                1/1     Running    0          3m36s
kube-system    kube-proxy-mp98s                                1/1     Running    0          6m41s
kube-system    kube-scheduler-ansible-control-plane            1/1     Running    0          6m52s
vagrant@ansible-control-plane:~/Dockercoins$
```

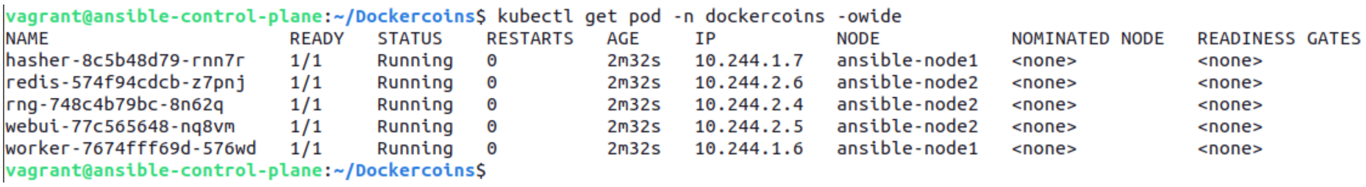
Déploiement de Dockercoins

14. Créer et appliquer le playbook ***dockercoins.yaml*** qui permet de déployer l’application Dockercoins sur le nouveau cluster k8s



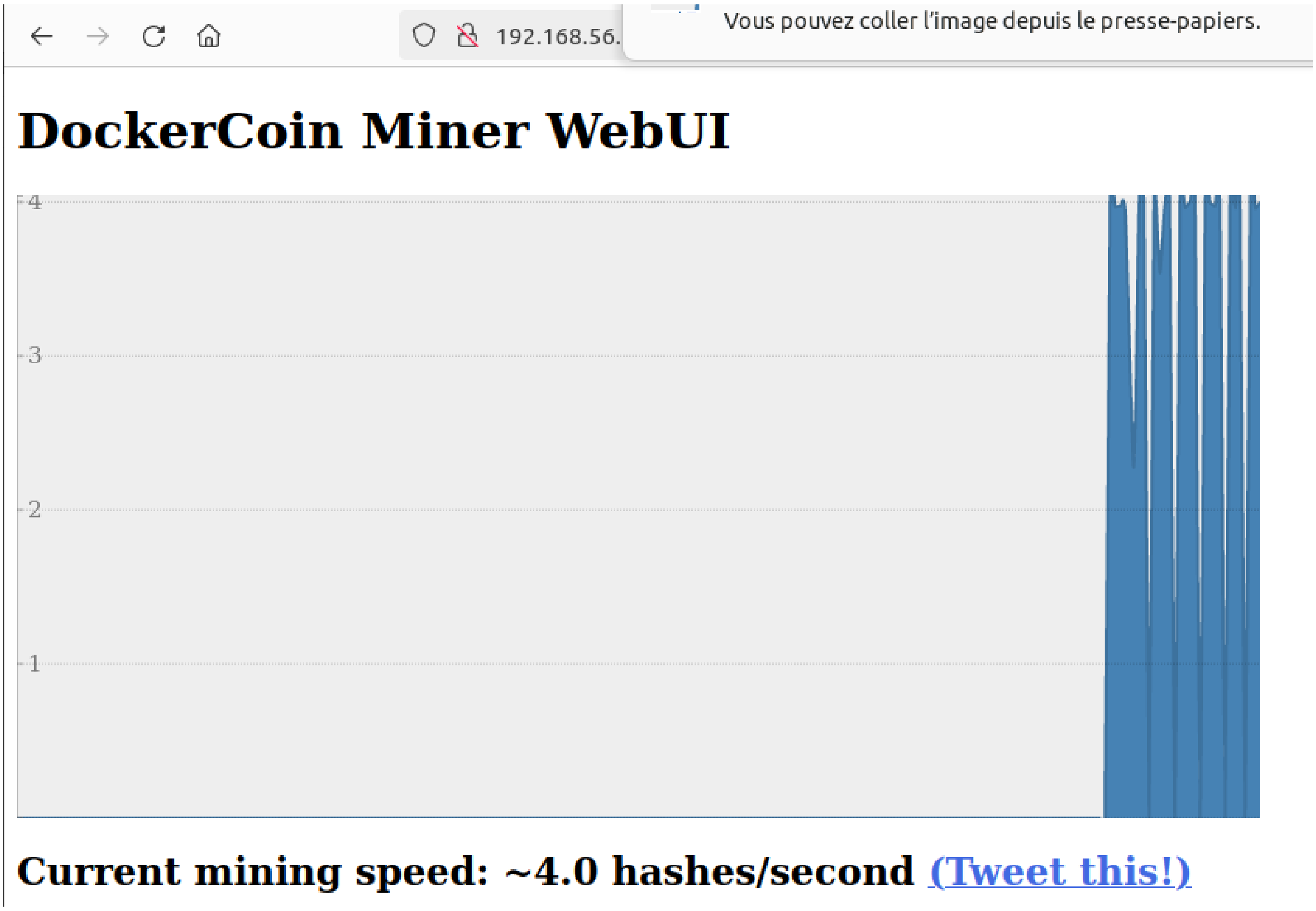
```
vagrant@ansible-control-plane:~/Dockercoins$ ansible-playbook playbooks/dockercoins-deploy.yml

PLAY [control] *******************************************************************************************************

TASK [deploy Dockercoins] ********************************************************************************************
changed: [ansible]

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

vagrant@ansible-control-plane:~/Dockercoins$
```

15. Vérifier que tous les pods applicatifs sont à l’état *Running*.



```
vagrant@ansible-control-plane:~/Dockercoins$ kubectl get pod -n dockercoins -owide
NAME                     READY   STATUS    RESTARTS   AGE    IP           NODE            NOMINATED NODE   READINESS GATES
hasher-8c5b48d79-rnn7r   1/1     Running   0          2m32s  10.244.1.7   ansible-node1   <none>           <none>
redis-574f94cdcb-z7pnj   1/1     Running   0          2m32s  10.244.2.6   ansible-node2   <none>           <none>
rng-748c4b79bc-8n62q     1/1     Running   0          2m32s  10.244.2.4   ansible-node2   <none>           <none>
webui-77c565648-nq8vm    1/1     Running   0          2m32s  10.244.2.5   ansible-node2   <none>           <none>
worker-7674fff69d-576wd  1/1     Running   0          2m32s  10.244.1.6   ansible-node1   <none>           <none>
vagrant@ansible-control-plane:~/Dockercoins$
```

16. En utilisant le navigateur, afficher l’interface de l’application.



192.168.56.

Vous pouvez coller l’image depuis le presse-papiers.

DockerCoin Miner WebUI

Current mining speed: ~4.0 hashes/second (Tweet this!)