LAB – Lightweight IaaS platforms

Hands on experience with Kubernetes installation

References:

- Kubernetes documentation https://kubernetes.io/docs/home/

Charmed Bundle

- Charmed bundle is a type of juju installation in which everything is automated
- A Yaml file describes which components and where they should be installed
- For each component the configuration is also specified

Deploy Kubernetes

- Download a standard bundle and customize

```
wget https://github.com/charmed-
kubernetes/bundle/blob/main/releases/1.27/bundle.yaml
```

- Deploy the bundle specifying where to install the controller node (node 10) and one worker (node 11)

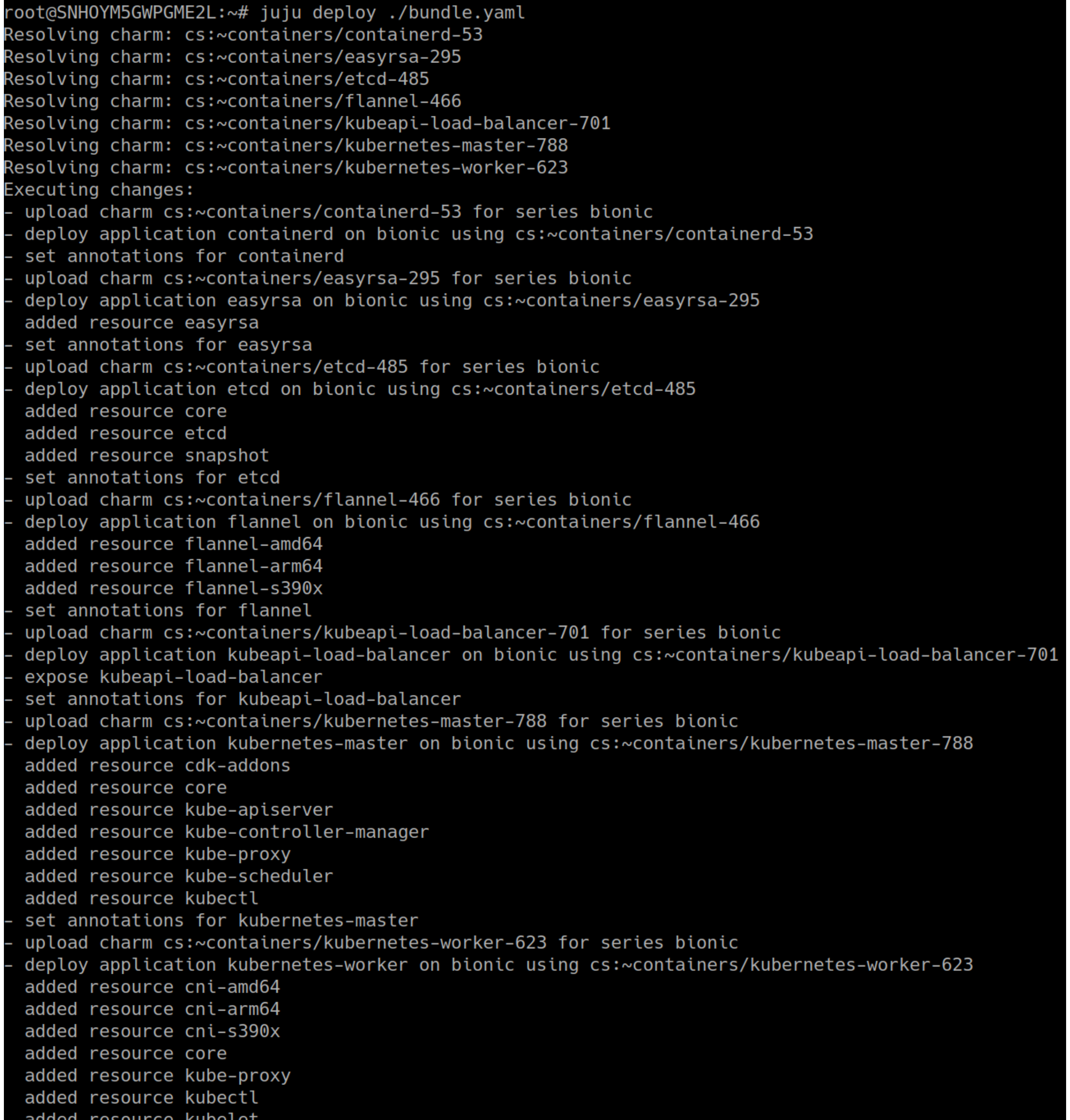
```
juju deploy ./bundle.yaml --map-machines=existing,0=10,1=11
```

- If you have more than one worker extend the role to other machines (node 12 in this case)

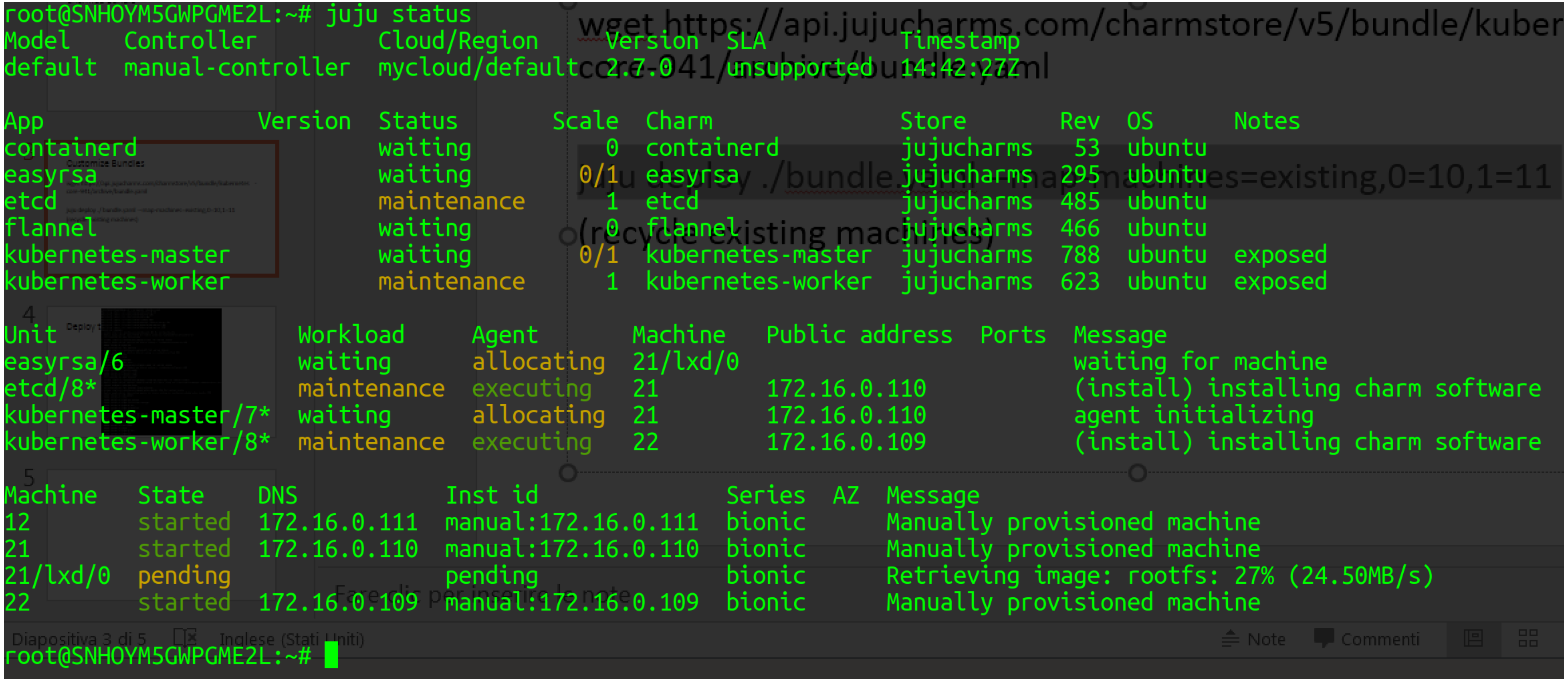
```
juju add-unit --to 12 kubernetes-worker
```

Tutorial

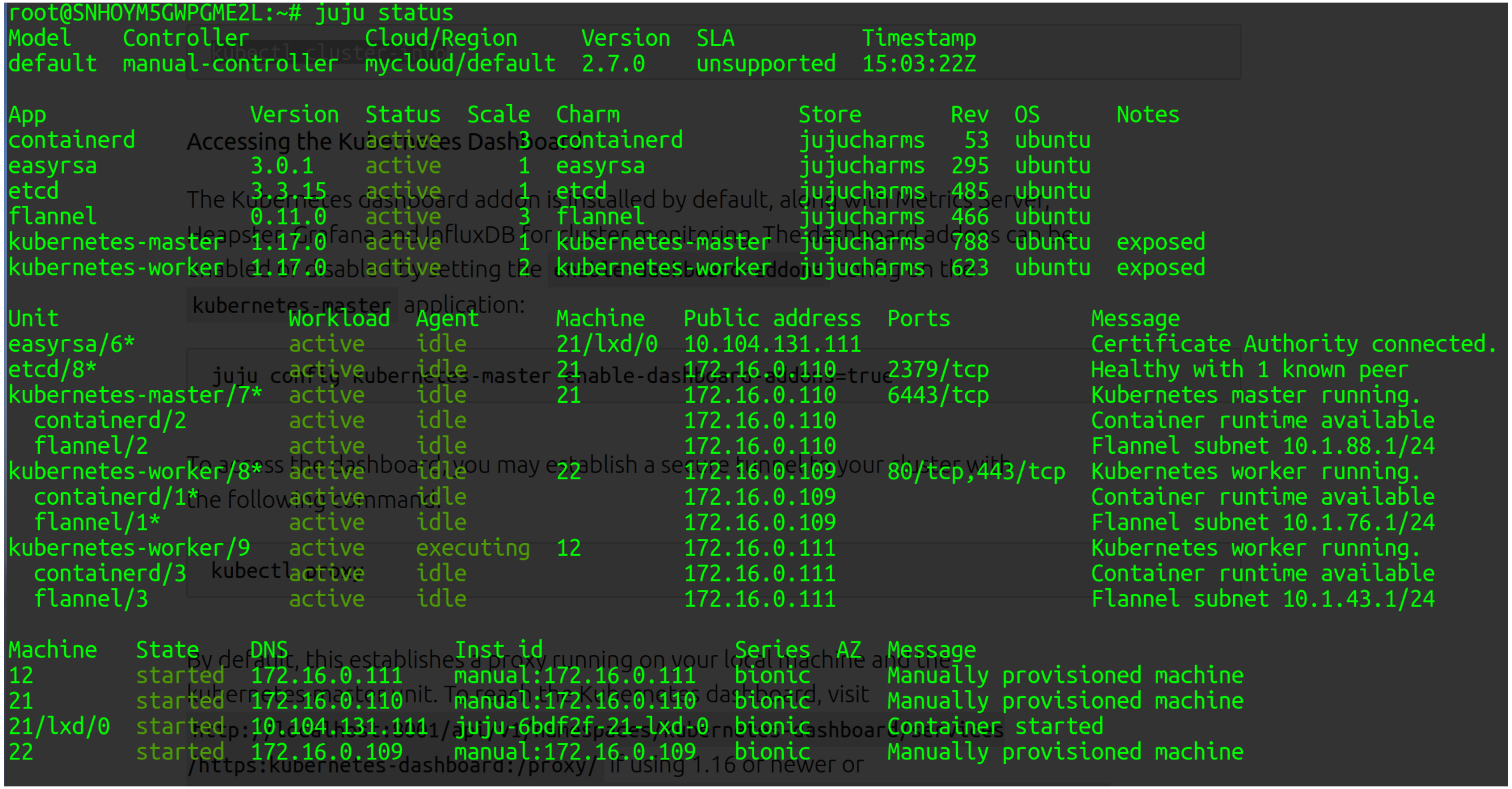
https://jaas.ai/kubernetes-core



```
root@SNHOYM5GWPGME2L:~# juju deploy ./bundle.yaml
Resolving charm: cs:~containers/containerd-53
Resolving charm: cs:~containers/easyrsa-295
Resolving charm: cs:~containers/etcd-485
Resolving charm: cs:~containers/flannel-466
Resolving charm: cs:~containers/kubeapi-load-balancer-701
Resolving charm: cs:~containers/kubernetes-master-788
Resolving charm: cs:~containers/kubernetes-worker-623
Executing changes:
- upload charm cs:~containers/containerd-53 for series bionic
- deploy application containerd on bionic using cs:~containers/containerd-53
- set annotations for containerd
- upload charm cs:~containers/easyrsa-295 for series bionic
- deploy application easyrsa on bionic using cs:~containers/easyrsa-295
  added resource easyrsa
- set annotations for easyrsa
- upload charm cs:~containers/etcd-485 for series bionic
- deploy application etcd on bionic using cs:~containers/etcd-485
  added resource core
  added resource etcd
  added resource snapshot
- set annotations for etcd
- upload charm cs:~containers/flannel-466 for series bionic
- deploy application flannel on bionic using cs:~containers/flannel-466
  added resource flannel-amd64
  added resource flannel-arm64
  added resource flannel-s390x
- set annotations for flannel
- upload charm cs:~containers/kubeapi-load-balancer-701 for series bionic
- deploy application kubeapi-load-balancer on bionic using cs:~containers/kubeapi-load-balancer-701
- expose kubeapi-load-balancer
- set annotations for kubeapi-load-balancer
- upload charm cs:~containers/kubernetes-master-788 for series bionic
- deploy application kubernetes-master on bionic using cs:~containers/kubernetes-master-788
  added resource cdk-addons
  added resource core
  added resource kube-apiserver
  added resource kube-controller-manager
  added resource kube-proxy
  added resource kube-scheduler
  added resource kubectl
- set annotations for kubernetes-master
- upload charm cs:~containers/kubernetes-worker-623 for series bionic
- deploy application kubernetes-worker on bionic using cs:~containers/kubernetes-worker-623
  added resource cni-amd64
  added resource cni-arm64
  added resource cni-s390x
  added resource core
  added resource kube-proxy
  added resource kubectl
  added resource kubelet
```



```
root@SNHOYM5GWPGME2L:~# juju status
Model    Controller          Cloud/Region       Version  SLA          Timestamp
default  manual-controller   mycloud/default    2.7.0    unsupported  14:42:27Z

App                Version  Status       Scale  Charm              Store       Rev  OS      Notes
containerd                  waiting          0  containerd         jujucharms   53  ubuntu
easyrsa                     waiting        0/1  easyrsa            jujucharms  295  ubuntu
etcd                        maintenance      1  etcd               jujucharms  485  ubuntu
flannel                     waiting          0  flannel            jujucharms  466  ubuntu
kubernetes-master           waiting        0/1  kubernetes-master  jujucharms  788  ubuntu  exposed
kubernetes-worker           maintenance      1  kubernetes-worker  jujucharms  623  ubuntu  exposed

Unit                 Workload     Agent       Machine   Public address  Ports  Message
easyrsa/6            waiting      allocating  21/lxd/0                         waiting for machine
etcd/8*              maintenance  executing   21        172.16.0.110           (install) installing charm software
kubernetes-master/7* waiting      allocating  21        172.16.0.110           agent initializing
kubernetes-worker/8* maintenance  executing   22        172.16.0.109           (install) installing charm software

Machine   State    DNS           Inst id               Series  AZ  Message
12        started  172.16.0.111  manual:172.16.0.111   bionic      Manually provisioned machine
21        started  172.16.0.110  manual:172.16.0.110   bionic      Manually provisioned machine
21/lxd/0  pending                pending               bionic      Retrieving image: rootfs: 27% (24.50MB/s)
22        started  172.16.0.109  manual:172.16.0.109   bionic      Manually provisioned machine

root@SNHOYM5GWPGME2L:~#
```



```
root@SNHOYM5GWPGME2L:~# juju status
Model    Controller         Cloud/Region     Version  SLA          Timestamp
default  manual-controller  mycloud/default  2.7.0    unsupported  15:03:22Z

App                Version  Status  Scale  Charm              Store       Rev  OS      Notes
containerd                  active      3  containerd         jujucharms   53  ubuntu
easyrsa            3.0.1    active      1  easyrsa            jujucharms  295  ubuntu
etcd               3.3.15   active      1  etcd               jujucharms  485  ubuntu
flannel            0.11.0   active      3  flannel            jujucharms  466  ubuntu
kubernetes-master  1.17.0   active      1  kubernetes-master  jujucharms  788  ubuntu  exposed
kubernetes-worker  1.17.0   active      2  kubernetes-worker  jujucharms  623  ubuntu  exposed

Unit                   Workload  Agent      Machine   Public address  Ports           Message
easyrsa/6*             active    idle       21/lxd/0  10.104.131.111                  Certificate Authority connected.
etcd/8*                active    idle       21        172.16.0.110    2379/tcp        Healthy with 1 known peer
kubernetes-master/7*   active    idle       21        172.16.0.110    6443/tcp        Kubernetes master running.
  containerd/2         active    idle                 172.16.0.110                    Container runtime available
  flannel/2            active    idle                 172.16.0.110                    Flannel subnet 10.1.88.1/24
kubernetes-worker/8*   active    idle       22        172.16.0.109    80/tcp,443/tcp  Kubernetes worker running.
  containerd/1*        active    idle                 172.16.0.109                    Container runtime available
  flannel/1*           active    idle                 172.16.0.109                    Flannel subnet 10.1.76.1/24
kubernetes-worker/9    active    executing  12        172.16.0.111                    Kubernetes worker running.
  containerd/3         active    idle                 172.16.0.111                    Container runtime available
  flannel/3            active    idle                 172.16.0.111                    Flannel subnet 10.1.43.1/24

Machine   State    DNS             Inst id               Series  AZ  Message
12        started  172.16.0.111    manual:172.16.0.111   bionic      Manually provisioned machine
21        started  172.16.0.110    manual:172.16.0.110   bionic      Manually provisioned machine
21/lxd/0  started  10.104.131.111  juju-6bdf2f-21-lxd-0  bionic      Container started
22        started  172.16.0.109    manual:172.16.0.109   bionic      Manually provisioned machine
```

```
kubectl cluster-info
```

Accessing the Kubernetes Dashboard

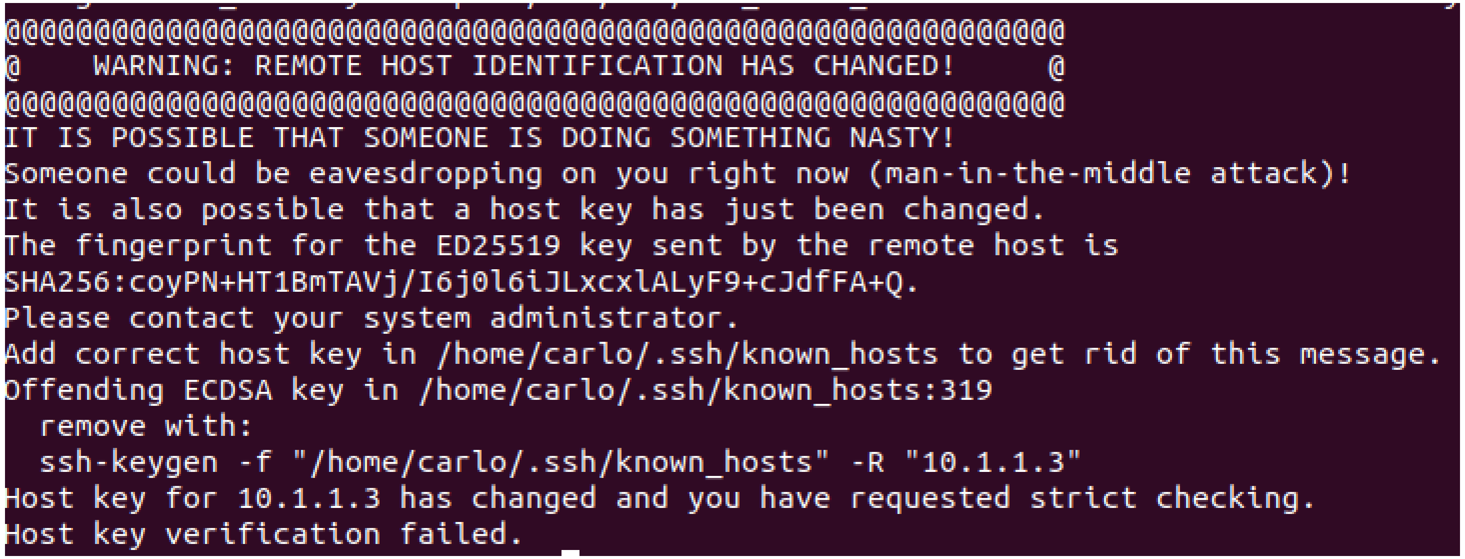
The Kubernetes dashboard addon is installed by default, along with Metrics Server, Heapster, Grafana and InfluxDB for cluster monitoring. The dashboard addons can be enabled or disabled by setting the enable-dashboard-addons config on the kubernetes-master application:

```
juju config kubernetes-master enable-dashboard-addons=true
```

To access the dashboard, you may establish a secure tunnel to your cluster with the following command:

```
kubectl proxy
```

By default, this establishes a proxy running on your local machine and the kubernetes-master unit. To reach the Kubernetes dashboard, visit http://localhost:8001/api/v1/namespaces/kube-system/services/https:kubernetes-dashboard:/proxy/ if using 1.16 or newer or



```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the ED25519 key sent by the remote host is
SHA256:coyPN+HT1BmTAVj/I6j0l6iJLxcxlALyF9+cJdfFA+Q.
Please contact your system administrator.
Add correct host key in /home/carlo/.ssh/known_hosts to get rid of this message.
Offending ECDSA key in /home/carlo/.ssh/known_hosts:319
  remove with:
  ssh-keygen -f "/home/carlo/.ssh/known_hosts" -R "10.1.1.3"
Host key for 10.1.1.3 has changed and you have requested strict checking.
Host key verification failed.
```

Minikube

- An alternative installation method is available for <u>test environment</u>
- This method is named *minikube* in which a Kubernetes installation is run in a single environment
- Minikube explots docker as container engine

Install Docker

- Install some pre-requisites

```
sudo apt install apt-transport-https ca-certificates curl
software-properties-common
```

- Add the key of the official docker repository in the system

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg |
sudo apt-key add -
```

- Add the Docker repository

```
sudo add-apt-repository "deb [arch=amd64]
https://download.docker.com/linux/ubuntu jammy stable“
```

- Install Docker

```
sudo apt install docker-ce
```

Minikube

- Download minikube

```
curl -Lo minikube https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64 \ && chmod +x minikube

mkdir -p /usr/local/bin/

install minikube /usr/local/bin/
```

Install some dependencies

```
apt-get install conntrack

wget https://github.com/Mirantis/cri-dockerd/releases/download/v0.3.1/cri-dockerd_0.3.1.3-0.ubuntu-jammy_amd64.deb
dpkg -i cri-dockerd_0.3.1.3-0.ubuntu-jammy_amd64.deb

wget https://github.com/kubernetes-sigs/cri-tools/releases/download/v1.27.0/crictl-v1.27.0-linux-amd64.tar.gz
tar xvfz crictl-v1.27.0-linux-amd64.tar.gz
install crictl /usr/local/bin/

wget https://github.com/containernetworking/plugins/releases/download/v1.2.0/cni-plugins-linux-amd64-v1.2.0.tgz
mkdir -p /opt/cni/bin
tar xvfz cni-plugins-linux-amd64-v1.2.0.tgz --directory /opt/cni/bin
```

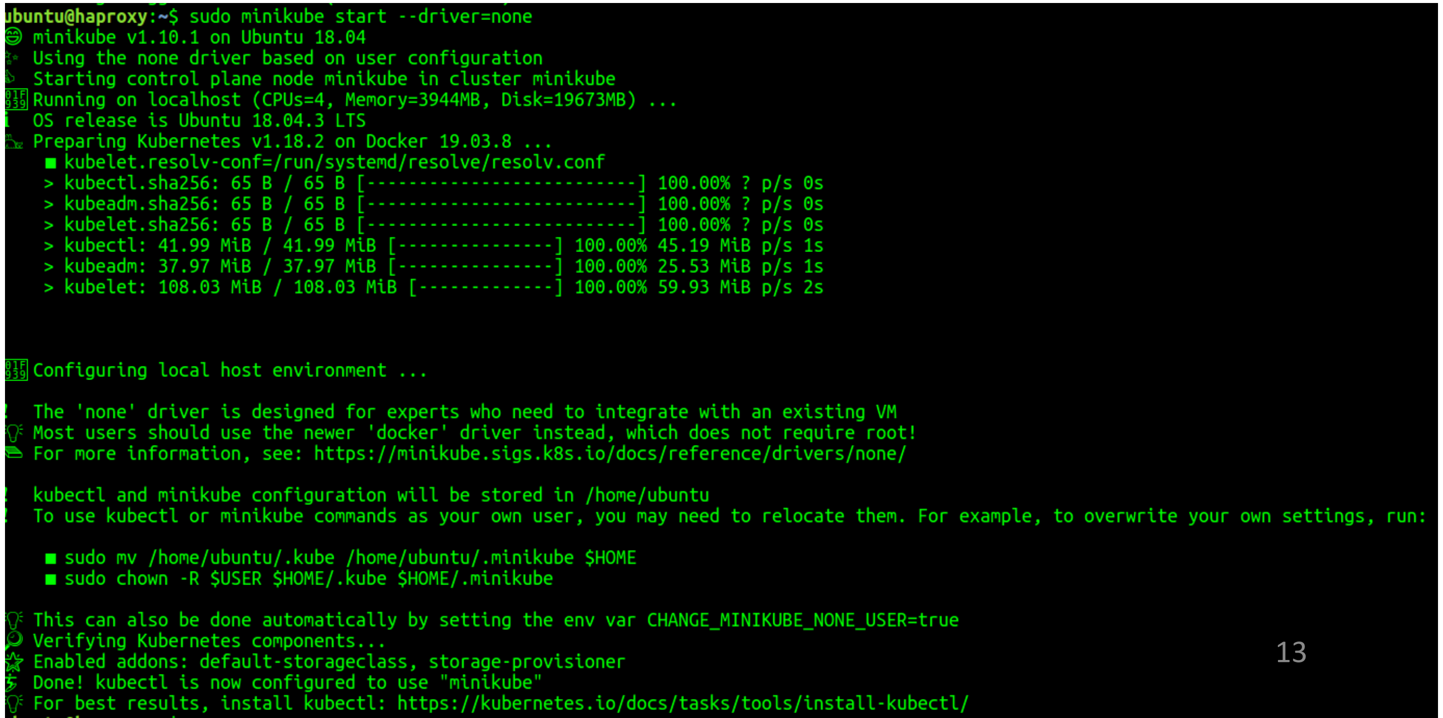
Installation

- Install the environment

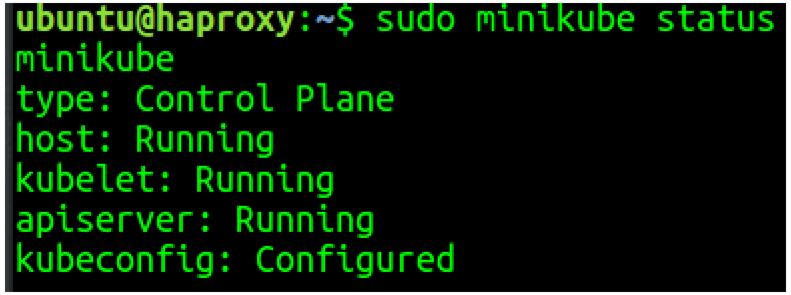
```
minikube start --driver=none
```

- Check the status

```
minikube status
```



```
ubuntu@haproxy:~$ sudo minikube start --driver=none
   minikube v1.10.1 on Ubuntu 18.04
   Using the none driver based on user configuration
   Starting control plane node minikube in cluster minikube
   Running on localhost (CPUs=4, Memory=3944MB, Disk=19673MB) ...
   OS release is Ubuntu 18.04.3 LTS
   Preparing Kubernetes v1.18.2 on Docker 19.03.8 ...
     ■ kubelet.resolv-conf=/run/systemd/resolve/resolv.conf
     > kubectl.sha256: 65 B / 65 B [--------------------------] 100.00% ? p/s 0s
     > kubeadm.sha256: 65 B / 65 B [--------------------------] 100.00% ? p/s 0s
     > kubelet.sha256: 65 B / 65 B [--------------------------] 100.00% ? p/s 0s
     > kubectl: 41.99 MiB / 41.99 MiB [---------------] 100.00% 45.19 MiB p/s 1s
     > kubeadm: 37.97 MiB / 37.97 MiB [---------------] 100.00% 25.53 MiB p/s 1s
     > kubelet: 108.03 MiB / 108.03 MiB [-------------] 100.00% 59.93 MiB p/s 2s



   Configuring local host environment ...

!  The 'none' driver is designed for experts who need to integrate with an existing VM
   Most users should use the newer 'docker' driver instead, which does not require root!
   For more information, see: https://minikube.sigs.k8s.io/docs/reference/drivers/none/

!  kubectl and minikube configuration will be stored in /home/ubuntu
!  To use kubectl or minikube commands as your own user, you may need to relocate them. For example, to overwrite your own settings, run:

     ■ sudo mv /home/ubuntu/.kube /home/ubuntu/.minikube $HOME
     ■ sudo chown -R $USER $HOME/.kube $HOME/.minikube

   This can also be done automatically by setting the env var CHANGE_MINIKUBE_NONE_USER=true
   Verifying Kubernetes components...
   Enabled addons: default-storageclass, storage-provisioner
   Done! kubectl is now configured to use "minikube"
   For best results, install kubectl: https://kubernetes.io/docs/tasks/tools/install-kubectl/
```



```
ubuntu@haproxy:~$ sudo minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
```

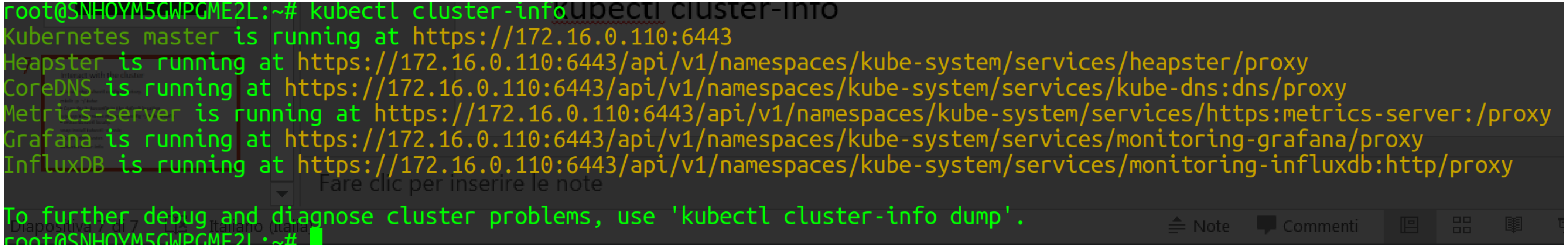
Interact with the cluster

- Install kubectl

```
apt install snapd
snap install kubectl --classic
```

- Query the cluster.

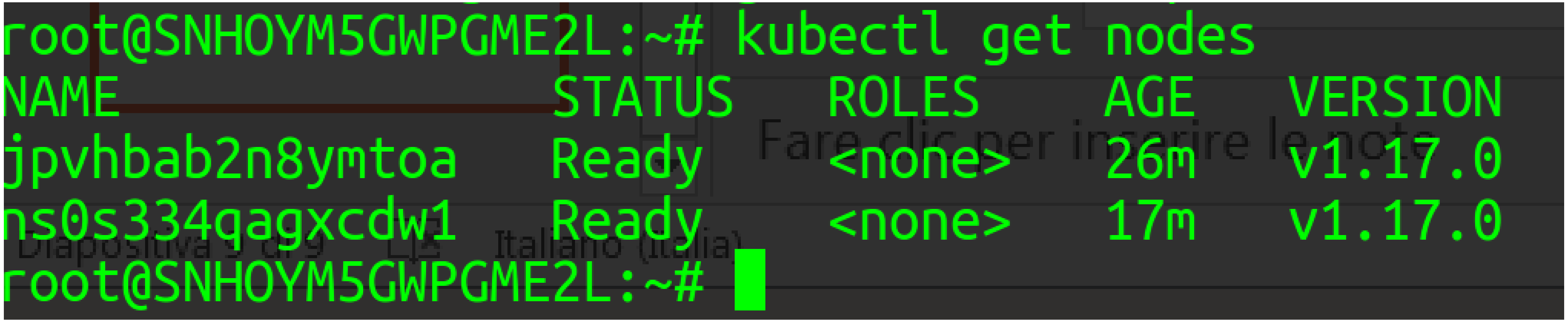
```
kubectl cluster-info
```



```
root@SNHOYM5GWPGME2L:~# kubectl cluster-info
Kubernetes master is running at https://172.16.0.110:6443
Heapster is running at https://172.16.0.110:6443/api/v1/namespaces/kube-system/services/heapster/proxy
CoreDNS is running at https://172.16.0.110:6443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy
Metrics-server is running at https://172.16.0.110:6443/api/v1/namespaces/kube-system/services/https:metrics-server:/proxy
Grafana is running at https://172.16.0.110:6443/api/v1/namespaces/kube-system/services/monitoring-grafana/proxy
InfluxDB is running at https://172.16.0.110:6443/api/v1/namespaces/kube-system/services/monitoring-influxdb:http/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
root@SNHOYM5GWPGME2L:~#
```

Get nodes

`kubectl get nodes`

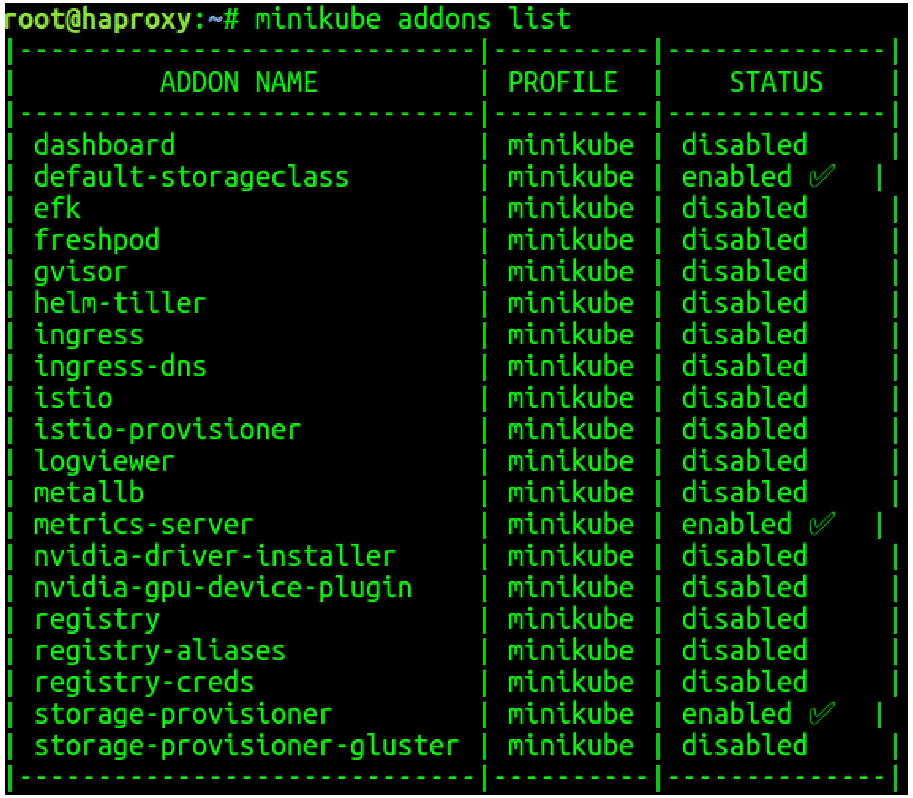


```
root@SNHOYM5GWPGME2L:~# kubectl get nodes
NAME               STATUS   ROLES    AGE   VERSION
jpvhbab2n8ymtoa    Ready    <none>   26m   v1.17.0
ns0s334qagxcdw1    Ready    <none>   17m   v1.17.0
root@SNHOYM5GWPGME2L:~#
```

Activate some plugins (only for minikube)

minikube addons enable metrics-server

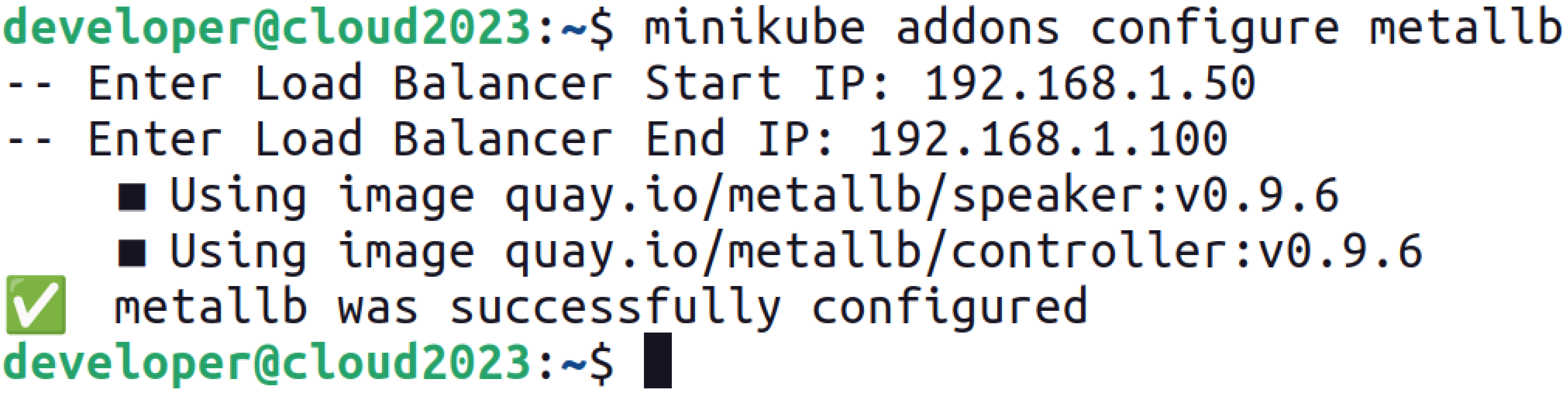
minikube addons enable metallb



```
root@haproxy:~# minikube addons list
|-----------------------------|----------|--------------|
|         ADDON NAME          | PROFILE  |    STATUS    |
|-----------------------------|----------|--------------|
| dashboard                   | minikube | disabled     |
| default-storageclass        | minikube | enabled ✅   |
| efk                         | minikube | disabled     |
| freshpod                    | minikube | disabled     |
| gvisor                      | minikube | disabled     |
| helm-tiller                 | minikube | disabled     |
| ingress                     | minikube | disabled     |
| ingress-dns                 | minikube | disabled     |
| istio                       | minikube | disabled     |
| istio-provisioner           | minikube | disabled     |
| logviewer                   | minikube | disabled     |
| metallb                     | minikube | disabled     |
| metrics-server              | minikube | enabled ✅   |
| nvidia-driver-installer     | minikube | disabled     |
| nvidia-gpu-device-plugin    | minikube | disabled     |
| registry                    | minikube | disabled     |
| registry-aliases            | minikube | disabled     |
| registry-creds              | minikube | disabled     |
| storage-provisioner         | minikube | enabled ✅   |
| storage-provisioner-gluster | minikube | disabled     |
|-----------------------------|----------|--------------|
```

Confiure the metallb

```
minikube addons configure metallb
```



```
developer@cloud2023:~$ minikube addons configure metallb
-- Enter Load Balancer Start IP: 192.168.1.50
-- Enter Load Balancer End IP: 192.168.1.100
    ■ Using image quay.io/metallb/speaker:v0.9.6
    ■ Using image quay.io/metallb/controller:v0.9.6
✅  metallb was successfully configured
developer@cloud2023:~$ █
```