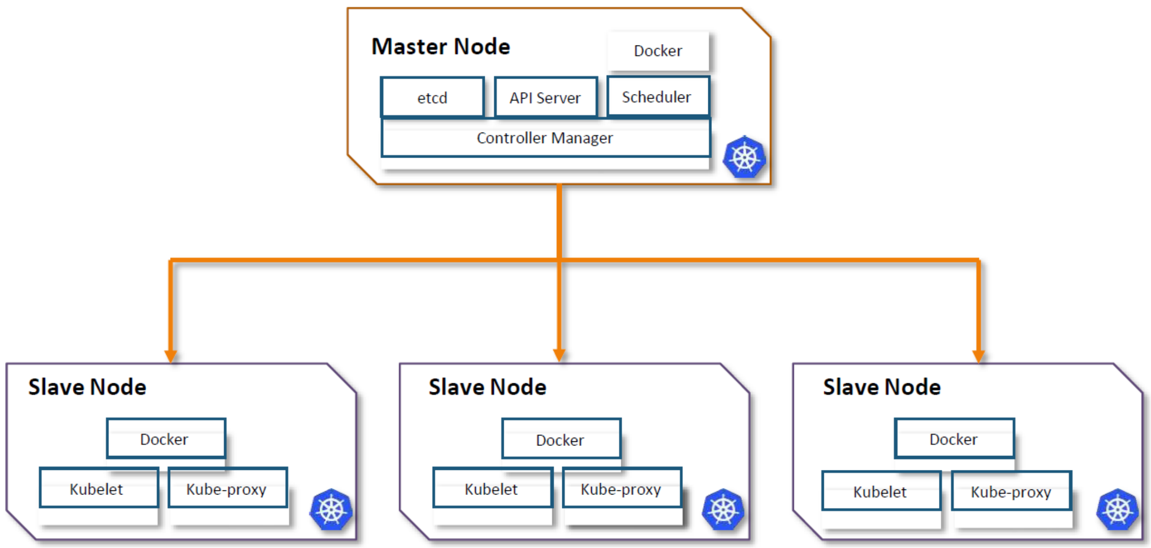
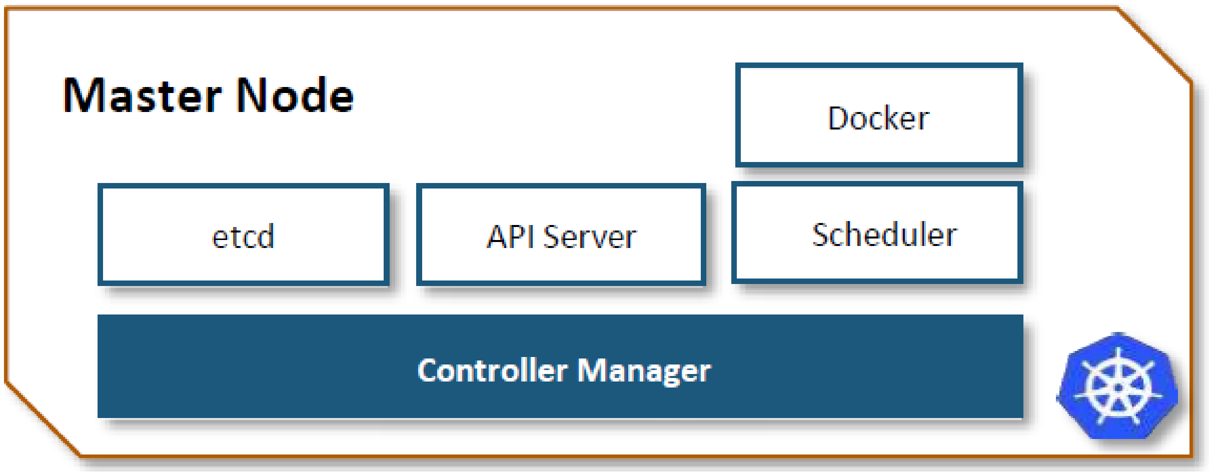
Kubernetes

Kubernetes is an open-source container orchestration software
It was originally developed by Google
It was first released on July 21st 2015
It is the ninth most active repository on GitHub in terms of number of commits

Kubernetes Architecture



Kubernetes Architecture – Master Components



ETCD:

It is a highly available distributed key value store, which is used to store cluster wide secrets. It is only accessible by Kubernetes API server, as it has sensitive information.

APISERVER:

It exposes the Kubernetes API. The Kubernetes API is the front-end for Kubernetes Control Plane, and is used to deploy and execute all operations in Kubernetes

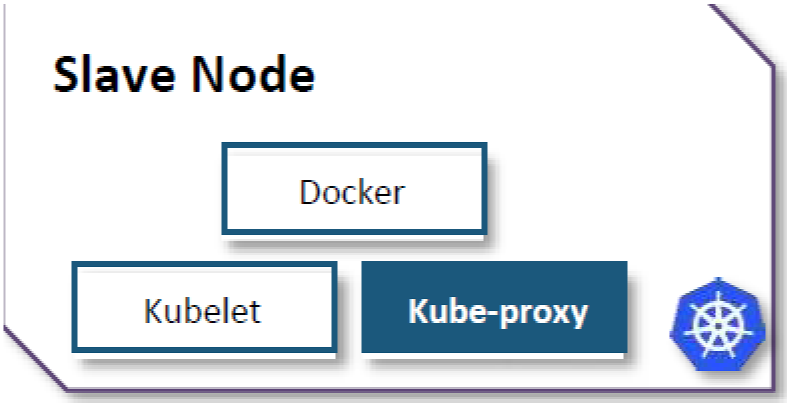
SCHEDULER

The scheduler takes care of scheduling of all the processes, Dynamic Resource Management and manages present and future events on the cluster

CONTROLLER MANAGER

The controller manager, runs all the controllers on the Kubernetes Cluster. Although each controller, is a separate process, but to reduce complexity, all the controllers are compiled into a single process. They are as follows: Node Controller, Replication Controller, Endpoints Controller, Service Accounts and Token Controllers

Kubernetes Architecture –Slave Components

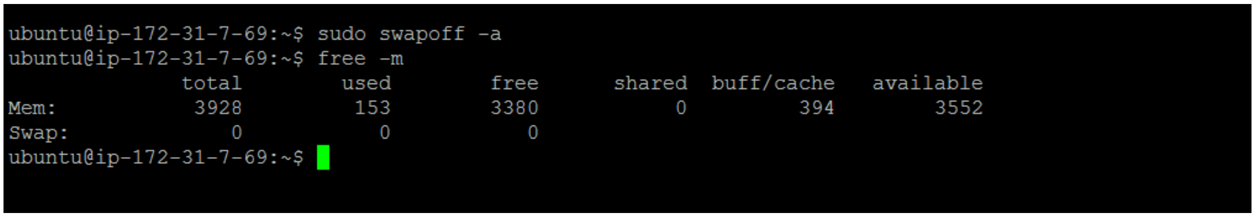


Kubelet
Kubelet takes the specification from the API server, and ensures the application is running according to the specifications which were mentioned. Each node has it's kubelet service

KUBEPROXY
This proxy service runs on each node and helps in making services available to the external host. It helps in connection forwarding to the correct

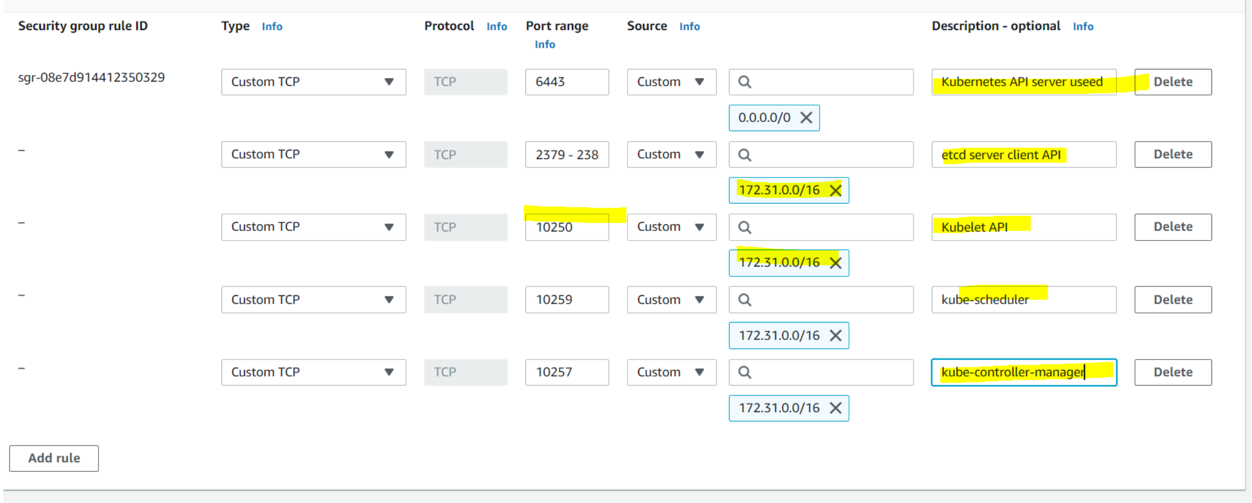
resources, it is also capable of doing primitive load balancing

Hands-on: Installing Kubernetes using kubeadm



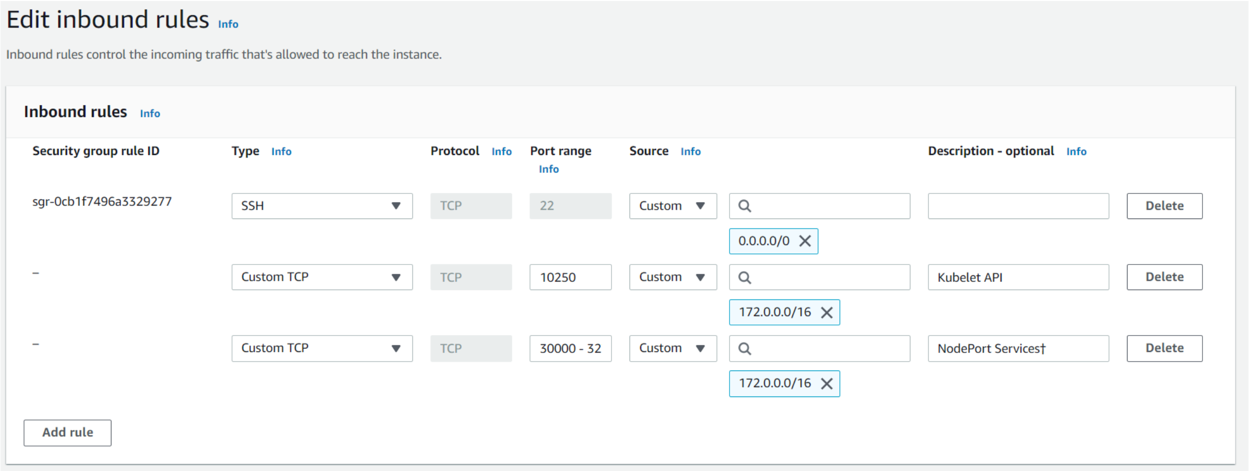
```
ubuntu@ip-172-31-7-69:~$ sudo swapoff -a
ubuntu@ip-172-31-7-69:~$ free -m
              total        used        free      shared  buff/cache   available
Mem:           3928         153        3380           0         394        3552
Swap:             0           0           0
ubuntu@ip-172-31-7-69:~$
```

Ports opened on security group for kube master



Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info		Description - optional Info	
sgr-08e7d914412350329	Custom TCP	TCP	6443	Custom	0.0.0.0/0	Kubernetes API server useed	Delete
–	Custom TCP	TCP	2379 - 238	Custom	172.31.0.0/16	etcd server client API	Delete
–	Custom TCP	TCP	10250	Custom	172.31.0.0/16	Kubelet API	Delete
–	Custom TCP	TCP	10259	Custom	172.31.0.0/16	kube-scheduler	Delete
–	Custom TCP	TCP	10257	Custom	172.31.0.0/16	kube-controller-manager	Delete

Ports opened for worker node security node

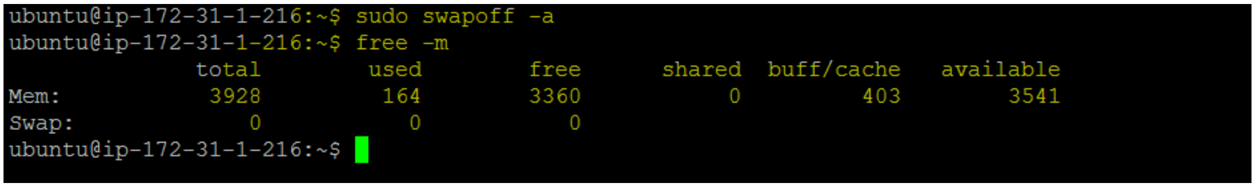


Edit inbound rules Info

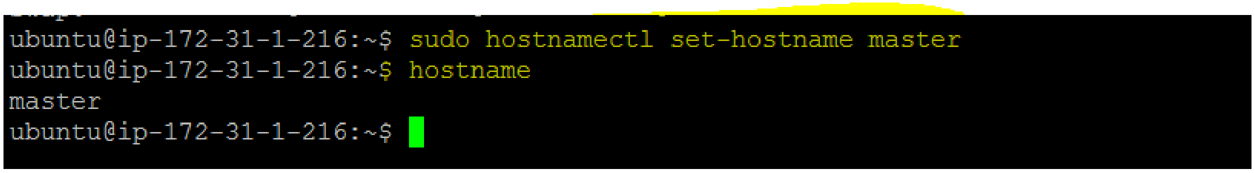
Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info		Description - optional Info	
sgr-0cb1f7496a3329277	SSH	TCP	22	Custom	0.0.0.0/0		Delete
–	Custom TCP	TCP	10250	Custom	172.0.0.0/16	Kubelet API	Delete
–	Custom TCP	TCP	30000 - 32	Custom	172.0.0.0/16	NodePort Services†	Delete



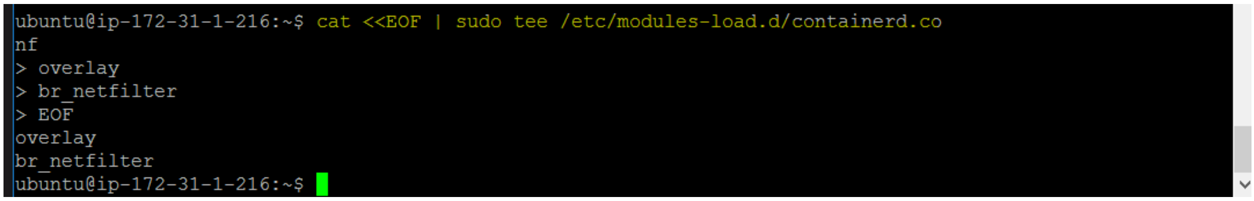
```
ubuntu@ip-172-31-1-216:~$ sudo swapoff -a
ubuntu@ip-172-31-1-216:~$ free -m
              total        used        free      shared  buff/cache   available
Mem:           3928         164        3360           0         403        3541
Swap:             0           0           0
ubuntu@ip-172-31-1-216:~$
```



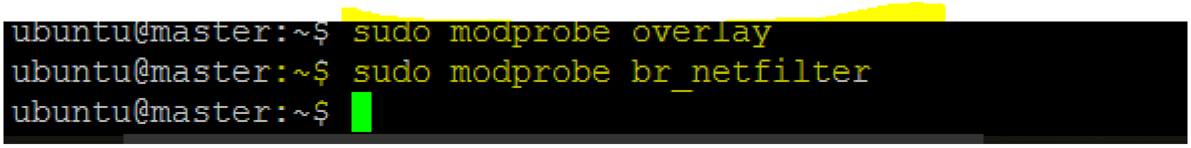
```
ubuntu@ip-172-31-1-216:~$ sudo hostnamectl set-hostname master
ubuntu@ip-172-31-1-216:~$ hostname
master
ubuntu@ip-172-31-1-216:~$
```

```
ubuntu@ip-172-31-1-216:~$ sudo swapoff -a
ubuntu@ip-172-31-1-216:~$ free -m
          total     used     free    shared  buff/cache   available
Mem:        3928      164     3360        0       403       3541
Swap:          0        0        0
ubuntu@ip-172-31-1-216:~$ sudo hostnamectl set-hostname master
ubuntu@ip-172-31-1-216:~$ hostname
master
```

https://kubernetes.io/docs/setup/production-environment/container-runtimes/#containered

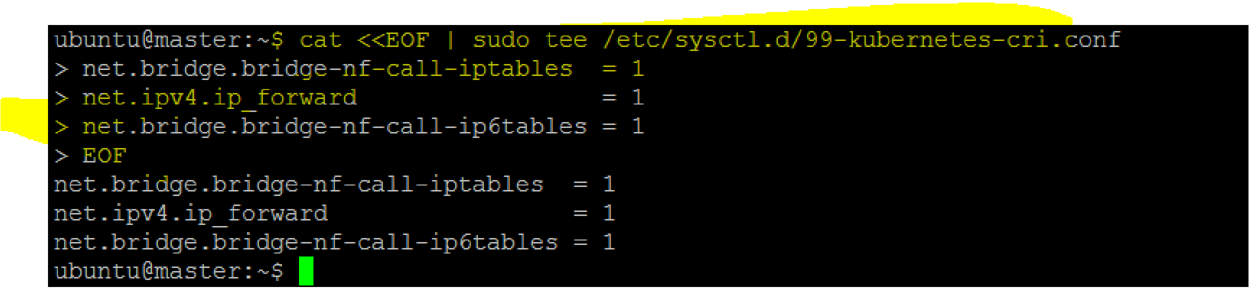


```
ubuntu@ip-172-31-1-216:~$ cat <<EOF | sudo tee /etc/modules-load.d/containerd.co
nf
> overlay
> br_netfilter
> EOF
overlay
br_netfilter
ubuntu@ip-172-31-1-216:~$
```

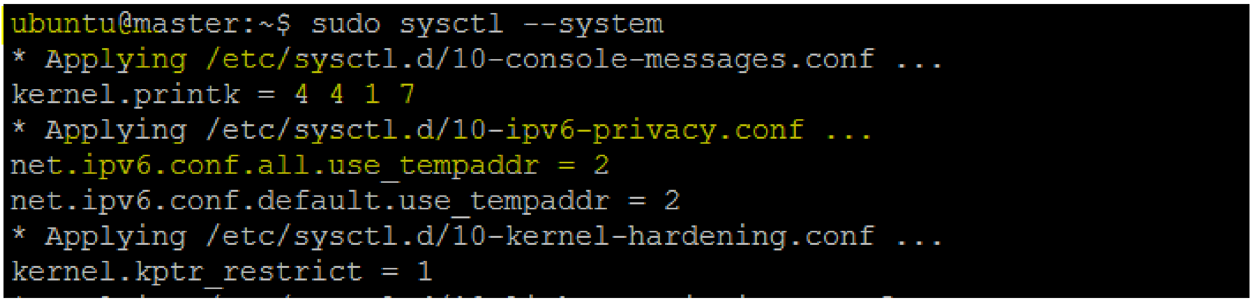


```
ubuntu@master:~$ sudo modprobe overlay
ubuntu@master:~$ sudo modprobe br_netfilter
ubuntu@master:~$
```

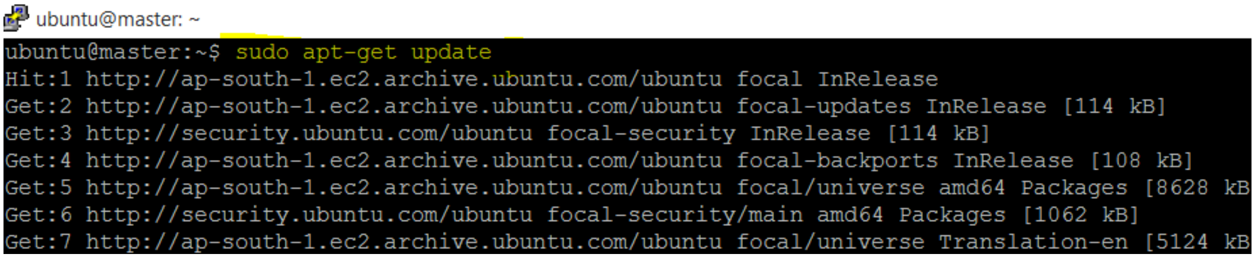
https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/#check-required-ports



```
ubuntu@master:~$ cat <<EOF | sudo tee /etc/sysctl.d/99-kubernetes-cri.conf
> net.bridge.bridge-nf-call-iptables  = 1
> net.ipv4.ip_forward                 = 1
> net.bridge.bridge-nf-call-ip6tables = 1
> EOF
net.bridge.bridge-nf-call-iptables  = 1
net.ipv4.ip_forward                 = 1
net.bridge.bridge-nf-call-ip6tables = 1
ubuntu@master:~$
```

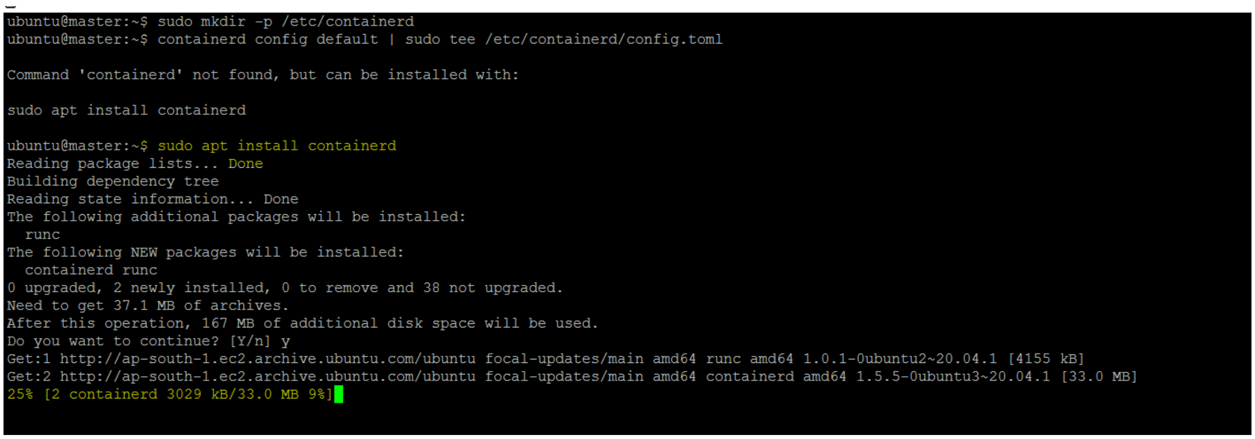


```
ubuntu@master:~$ sudo sysctl --system
* Applying /etc/sysctl.d/10-console-messages.conf ...
kernel.printk = 4 4 1 7
* Applying /etc/sysctl.d/10-ipv6-privacy.conf ...
net.ipv6.conf.all.use_tempaddr = 2
net.ipv6.conf.default.use_tempaddr = 2
* Applying /etc/sysctl.d/10-kernel-hardening.conf ...
kernel.kptr_restrict = 1
```

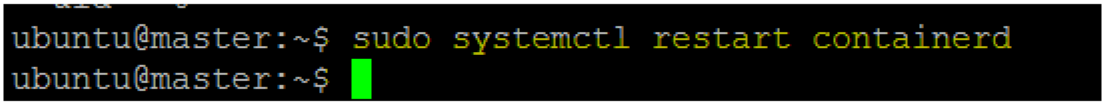


```
ubuntu@master:~$ sudo apt-get update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 Packages [8628 kB
Get:6 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [1062 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/universe Translation-en [5124 kB
```

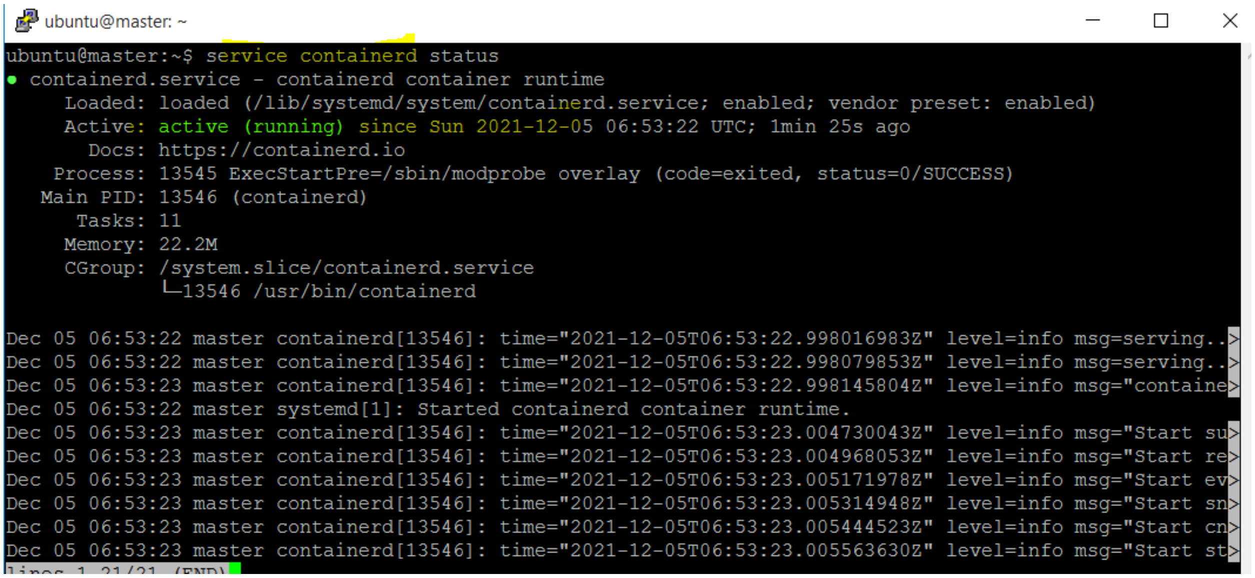
Install containerd:



```
ubuntu@master:~$ sudo mkdir -p /etc/containerd
ubuntu@master:~$ containerd config default | sudo tee /etc/containerd/config.toml

Command 'containerd' not found, but can be installed with:

sudo apt install containerd

ubuntu@master:~$ sudo apt install containerd
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  runc
The following NEW packages will be installed:
  containerd runc
0 upgraded, 2 newly installed, 0 to remove and 38 not upgraded.
Need to get 37.1 MB of archives.
After this operation, 167 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 runc amd64 1.0.1-0ubuntu2~20.04.1 [4155 kB]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 containerd amd64 1.5.5-0ubuntu3~20.04.1 [33.0 MB]
25% [2 containerd 3029 kB/33.0 MB 9%]
```



```
ubuntu@master:~$ sudo systemctl restart containerd
ubuntu@master:~$
```



```
ubuntu@master:~$ service containerd status
● containerd.service - containerd container runtime
     Loaded: loaded (/lib/systemd/system/containerd.service; enabled; vendor preset: enabled)
     Active: active (running) since Sun 2021-12-05 06:53:22 UTC; 1min 25s ago
       Docs: https://containerd.io
    Process: 13545 ExecStartPre=/sbin/modprobe overlay (code=exited, status=0/SUCCESS)
   Main PID: 13546 (containerd)
      Tasks: 11
     Memory: 22.2M
     CGroup: /system.slice/containerd.service
             └─13546 /usr/bin/containerd

Dec 05 06:53:22 master containerd[13546]: time="2021-12-05T06:53:22.998016983Z" level=info msg=serving..>
Dec 05 06:53:22 master containerd[13546]: time="2021-12-05T06:53:22.998079853Z" level=info msg=serving..>
Dec 05 06:53:23 master containerd[13546]: time="2021-12-05T06:53:22.998145804Z" level=info msg="containe>
Dec 05 06:53:22 master systemd[1]: Started containerd container runtime.
Dec 05 06:53:23 master containerd[13546]: time="2021-12-05T06:53:23.004730043Z" level=info msg="Start su>
Dec 05 06:53:23 master containerd[13546]: time="2021-12-05T06:53:23.004968053Z" level=info msg="Start re>
Dec 05 06:53:23 master containerd[13546]: time="2021-12-05T06:53:23.005171978Z" level=info msg="Start ev>
Dec 05 06:53:23 master containerd[13546]: time="2021-12-05T06:53:23.005314948Z" level=info msg="Start sn>
Dec 05 06:53:23 master containerd[13546]: time="2021-12-05T06:53:23.005444523Z" level=info msg="Start cn>
Dec 05 06:53:23 master containerd[13546]: time="2021-12-05T06:53:23.005563630Z" level=info msg="Start st>
lines 1-21/21 (END)
```

Update the apt package index and install packages needed to use the Kubernetes apt repository:

```
sudo apt-get update
sudo apt-get install -y apt-transport-https ca-certificates curl
```

Download the Google Cloud public signing key:

```
sudo curl -fsSLo /usr/share/keyrings/kubernetes-archive-keyring.gpg
https://packages.cloud.google.com/apt/doc/apt-key.gpg
```

Add the Kubernetes apt repository:

```
echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg]
https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee
/etc/apt/sources.list.d/kubernetes.list
```

Update apt package index, install kubelet, kubeadm and kubectl, and pin their version:

```
sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
sudo apt-mark hold kubelet kubeadm kubectl
```

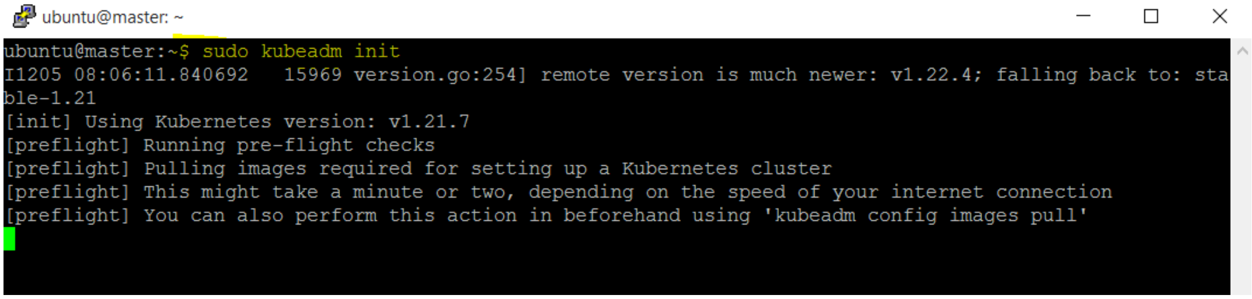
To check the kubernetes all versions

```
apt-cache madison kubeadm
```

```
sudo apt-get install -y kubelet=1.21.0-00 kubeadm=1.21.0-00 kubectl=1.21.0-00

sudo apt-mark hold kubelet kubeadm kubectl
```

```
ubuntu@master:~$ service kubelet status

● kubelet.service - kubelet: The Kubernetes Node Agent

     Loaded: loaded (/lib/systemd/system/kubelet.service; enabled; vendor preset: enabled)

    Drop-In: /etc/systemd/system/kubelet.service.d

             └─10-kubeadm.conf

     Active: activating (auto-restart) (Result: exit-code) since Sun 2021-12-05 08:02:36 UTC; 1s
ago

       Docs: https://kubernetes.io/docs/home/

    Process: 15797 ExecStart=/usr/bin/kubelet $KUBELET_KUBECONFIG_ARGS
$KUBELET_CONFIG_ARGS $KUBELET_KUB>

   Main PID: 15797 (code=exited, status=1/FAILURE)
```

```
ubuntu@master:~$ kubeadm version

kubeadm version: &version.Info{Major:"1", Minor:"21", GitVersion:"v1.21.0",
GitCommit:"cb303e613a121a29364f75cc67d3d580833a7479", GitTreeState:"clean",
BuildDate:"2021-04-08T16:30:03Z", GoVersion:"go1.16.1", Compiler:"gc",
Platform:"linux/amd64"}

===========================================================

ubuntu@master:~$ kubectl version

Client Version: version.Info{Major:"1", Minor:"21", GitVersion:"v1.21.0",
GitCommit:"cb303e613a121a29364f75cc67d3d580833a7479", GitTreeState:"clean",
BuildDate:"2021-04-08T16:31:21Z", GoVersion:"go1.16.1", Compiler:"gc",
Platform:"linux/amd64"}

The connection to the server localhost:8080 was refused - did you specify the right host or
port?
```



```
ubuntu@master:~$ sudo kubeadm init
I1205 08:06:11.840692   15969 version.go:254] remote version is much newer: v1.22.4; falling back to: sta
ble-1.21
[init] Using Kubernetes version: v1.21.7
[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'
```

Your Kubernetes control-plane has initialized successfully!

To start using your cluster, you need to run the following as a regular user:

```
  mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

Alternatively, if you are the root user, you can run:

```
  export KUBECONFIG=/etc/kubernetes/admin.conf
```

You should now deploy a pod network to the cluster.

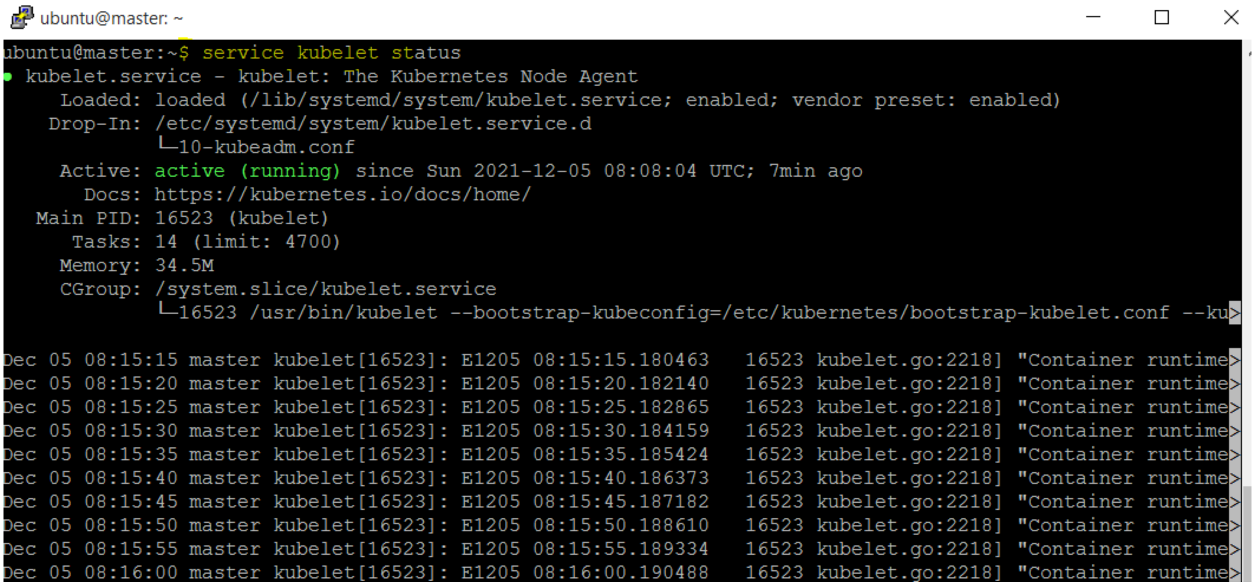
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:

  https://kubernetes.io/docs/concepts/cluster-administration/addons/

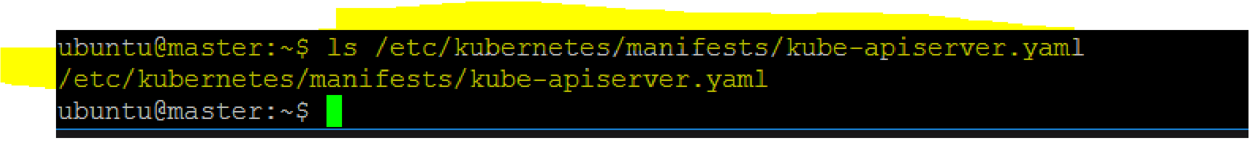
Then you can join any number of worker nodes by running the following on each as root:

```
kubeadm join 172.31.1.216:6443 --token ehc3z5.9nggelpp039euf5k \
        --discovery-token-ca-cert-hash
sha256:534fc66db53f1b4e93ec822337e38e353155550384519aa3758fabdfc4f0f162
```

```
ubuntu@master:~$ ls /etc/kubernetes/pki
apiserver-etcd-client.crt    apiserver.crt  etcd               front-proxy-client.key
apiserver-etcd-client.key    apiserver.key  front-proxy-ca.crt     sa.key
apiserver-kubelet-client.crt  ca.crt        front-proxy-ca.key     sa.pub
apiserver-kubelet-client.key  ca.key        front-proxy-client.crt
```



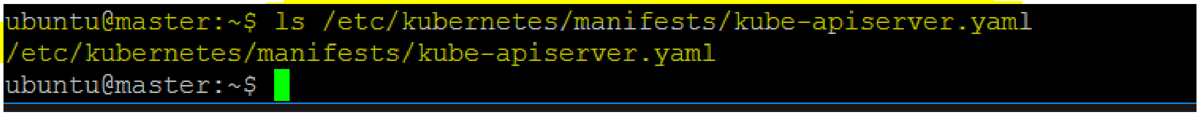
```
ubuntu@master:~$ service kubelet status
● kubelet.service - kubelet: The Kubernetes Node Agent
     Loaded: loaded (/lib/systemd/system/kubelet.service; enabled; vendor preset: enabled)
    Drop-In: /etc/systemd/system/kubelet.service.d
             └─10-kubeadm.conf
     Active: active (running) since Sun 2021-12-05 08:08:04 UTC; 7min ago
       Docs: https://kubernetes.io/docs/home/
   Main PID: 16523 (kubelet)
      Tasks: 14 (limit: 4700)
     Memory: 34.5M
     CGroup: /system.slice/kubelet.service
             └─16523 /usr/bin/kubelet --bootstrap-kubeconfig=/etc/kubernetes/bootstrap-kubelet.conf --ku>

Dec 05 08:15:15 master kubelet[16523]: E1205 08:15:15.180463   16523 kubelet.go:2218] "Container runtime>
Dec 05 08:15:20 master kubelet[16523]: E1205 08:15:20.182140   16523 kubelet.go:2218] "Container runtime>
Dec 05 08:15:25 master kubelet[16523]: E1205 08:15:25.182865   16523 kubelet.go:2218] "Container runtime>
Dec 05 08:15:30 master kubelet[16523]: E1205 08:15:30.184159   16523 kubelet.go:2218] "Container runtime>
Dec 05 08:15:35 master kubelet[16523]: E1205 08:15:35.185424   16523 kubelet.go:2218] "Container runtime>
Dec 05 08:15:40 master kubelet[16523]: E1205 08:15:40.186373   16523 kubelet.go:2218] "Container runtime>
Dec 05 08:15:45 master kubelet[16523]: E1205 08:15:45.187182   16523 kubelet.go:2218] "Container runtime>
Dec 05 08:15:50 master kubelet[16523]: E1205 08:15:50.188610   16523 kubelet.go:2218] "Container runtime>
Dec 05 08:15:55 master kubelet[16523]: E1205 08:15:55.189334   16523 kubelet.go:2218] "Container runtime>
Dec 05 08:16:00 master kubelet[16523]: E1205 08:16:00.190488   16523 kubelet.go:2218] "Container runtime>
```



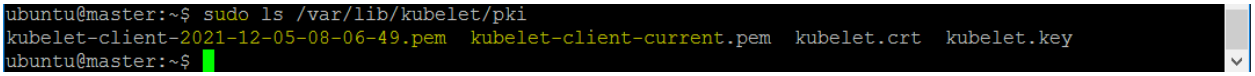
```
ubuntu@master:~$ ls /etc/kubernetes/manifests/kube-apiserver.yaml
/etc/kubernetes/manifests/kube-apiserver.yaml
ubuntu@master:~$
```

```
ubuntu@master:~$ sudo cat /etc/kubernetes/manifests/kube-apiserver.yaml
apiVersion: v1
kind: Pod
metadata:
  annotations:
    kubeadm.kubernetes.io/kube-apiserver.advertise-address.endpoint: 172.31.1.216:6443
  creationTimestamp: null
  labels:
    component: kube-apiserver
    tier: control-plane
  name: kube-apiserver
  namespace: kube-system
spec:
  containers:
  - command:
    - kube-apiserver
    - --advertise-address=172.31.1.216
    - --allow-privileged=true
    - --authorization-mode=Node,RBAC
    - --client-ca-file=/etc/kubernetes/pki/ca.crt
    - --enable-admission-plugins=NodeRestriction
    - --enable-bootstrap-token-auth=true
    - --etcd-cafile=/etc/kubernetes/pki/etcd/ca.crt
    - --etcd-certfile=/etc/kubernetes/pki/apiserver-etcd-client.crt
    - --etcd-keyfile=/etc/kubernetes/pki/apiserver-etcd-client.key
    - --etcd-servers=https://127.0.0.1:2379
    - --insecure-port=0
    - --kubelet-client-certificate=/etc/kubernetes/pki/apiserver-kubelet-client.crt
    - --kubelet-client-key=/etc/kubernetes/pki/apiserver-kubelet-client.key
    - --kubelet-preferred-address-types=InternalIP,ExternalIP,Hostname
    - --proxy-client-cert-file=/etc/kubernetes/pki/front-proxy-client.crt
- --proxy-client-key-file=/etc/kubernetes/pki/front-proxy-client.key
- --requestheader-allowed-names=front-proxy-client
- --requestheader-client-ca-file=/etc/kubernetes/pki/front-proxy-ca.crt
- --requestheader-extra-headers-prefix=X-Remote-Extra-
- --requestheader-group-headers=X-Remote-Group
- --requestheader-username-headers=X-Remote-User
- --secure-port=6443
- --service-account-issuer=https://kubernetes.default.svc.cluster.local
- --service-account-key-file=/etc/kubernetes/pki/sa.pub
- --service-account-signing-key-file=/etc/kubernetes/pki/sa.key
- --service-cluster-ip-range=10.96.0.0/12
- --tls-cert-file=/etc/kubernetes/pki/apiserver.crt
- --tls-private-key-file=/etc/kubernetes/pki/apiserver.key
image: k8s.gcr.io/kube-apiserver:v1.21.7
imagePullPolicy: IfNotPresent
livenessProbe:
  failureThreshold: 8
  httpGet:
    host: 172.31.1.216
    path: /livez
    port: 6443
    scheme: HTTPS
  initialDelaySeconds: 10
  periodSeconds: 10
  timeoutSeconds: 15
name: kube-apiserver
readinessProbe:
  failureThreshold: 3
  httpGet:
    host: 172.31.1.216
    path: /readyz
    port: 6443
    scheme: HTTPS
  periodSeconds: 1
  timeoutSeconds: 15
resources:
  requests:
    cpu: 250m
startupProbe:
  failureThreshold: 24
  httpGet:
    host: 172.31.1.216
    path: /livez
    port: 6443
    scheme: HTTPS
  initialDelaySeconds: 10
  periodSeconds: 10
  timeoutSeconds: 15
volumeMounts:
- mountPath: /etc/ssl/certs
  name: ca-certs
  readOnly: true
- mountPath: /etc/ca-certificates
  name: etc-ca-certificates
```

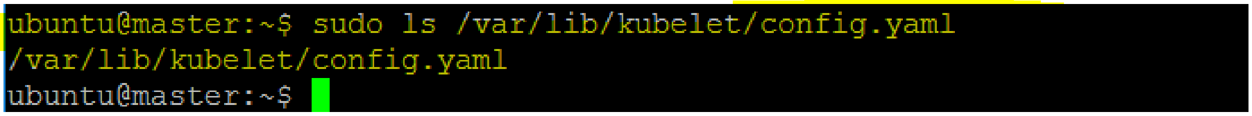
```
    readOnly: true
  - mountPath: /etc/pki
    name: etc-pki
    readOnly: true
  - mountPath: /etc/kubernetes/pki
    name: k8s-certs
    readOnly: true
  - mountPath: /usr/local/share/ca-certificates
    name: usr-local-share-ca-certificates
    readOnly: true
  - mountPath: /usr/share/ca-certificates
    name: usr-share-ca-certificates
    readOnly: true
hostNetwork: true
priorityClassName: system-node-critical
volumes:
- hostPath:
    path: /etc/ssl/certs
    type: DirectoryOrCreate
  name: ca-certs
- hostPath:
    path: /etc/ca-certificates
    type: DirectoryOrCreate
  name: etc-ca-certificates
- hostPath:
    path: /etc/pki
    type: DirectoryOrCreate
    name: etc-pki
  - hostPath:
      path: /etc/kubernetes/pki
      type: DirectoryOrCreate
    name: k8s-certs
  - hostPath:
      path: /usr/local/share/ca-certificates
      type: DirectoryOrCreate
    name: usr-local-share-ca-certificates
  - hostPath:
      path: /usr/share/ca-certificates
      type: DirectoryOrCreate
    name: usr-share-ca-certificates
status: {}
```



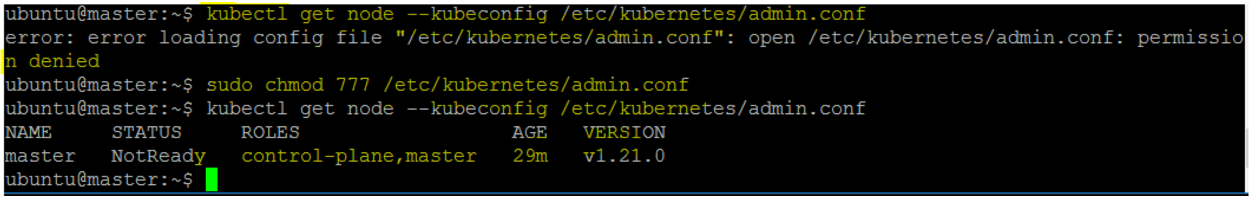
```
ubuntu@master:~$ ls /etc/kubernetes/manifests/kube-apiserver.yaml
/etc/kubernetes/manifests/kube-apiserver.yaml
ubuntu@master:~$
```



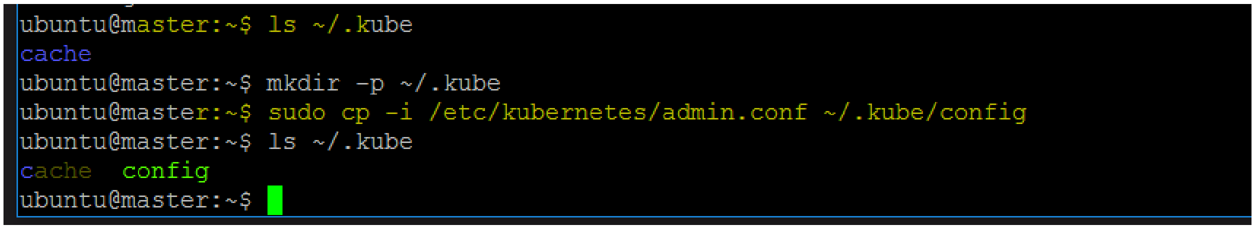
```
ubuntu@master:~$ sudo ls /var/lib/kubelet/pki
kubelet-client-2021-12-05-08-06-49.pem  kubelet-client-current.pem  kubelet.crt  kubelet.key
ubuntu@master:~$
```



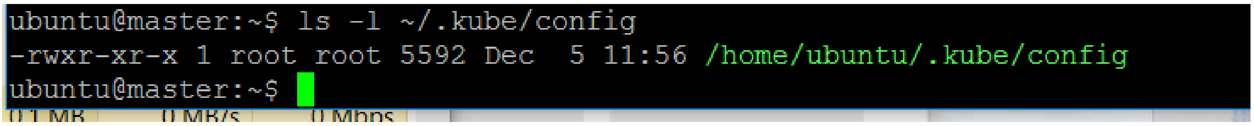
```
ubuntu@master:~$ sudo ls /var/lib/kubelet/config.yaml
/var/lib/kubelet/config.yaml
ubuntu@master:~$
```



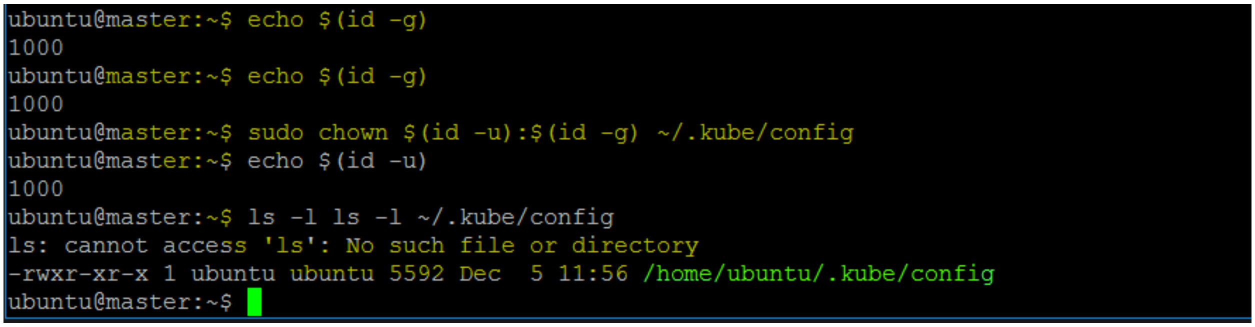
```
ubuntu@master:~$ kubectl get node --kubeconfig /etc/kubernetes/admin.conf
error: error loading config file "/etc/kubernetes/admin.conf": open /etc/kubernetes/admin.conf: permissio
n denied
ubuntu@master:~$ sudo chmod 777 /etc/kubernetes/admin.conf
ubuntu@master:~$ kubectl get node --kubeconfig /etc/kubernetes/admin.conf
NAME     STATUS     ROLES                  AGE   VERSION
master   NotReady   control-plane,master   29m   v1.21.0
ubuntu@master:~$
```



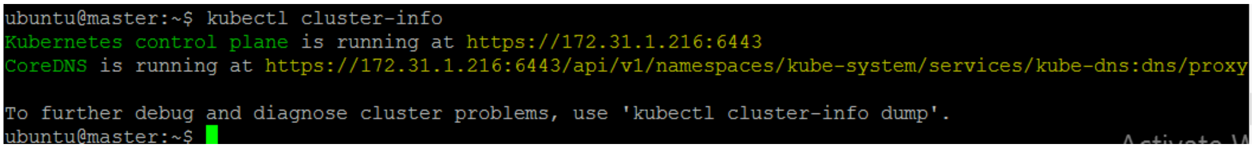
```
ubuntu@master:~$ ls ~/.kube
cache
ubuntu@master:~$ mkdir -p ~/.kube
ubuntu@master:~$ sudo cp -i /etc/kubernetes/admin.conf ~/.kube/config
ubuntu@master:~$ ls ~/.kube
cache  config
ubuntu@master:~$
```



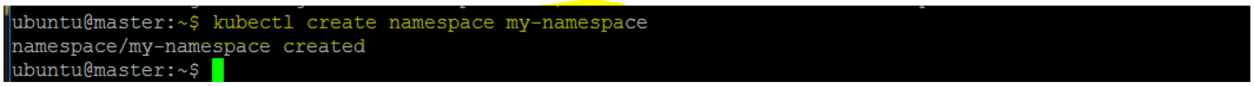
```
ubuntu@master:~$ ls -l ~/.kube/config
-rwxr-xr-x 1 root root 5592 Dec  5 11:56 /home/ubuntu/.kube/config
ubuntu@master:~$
```



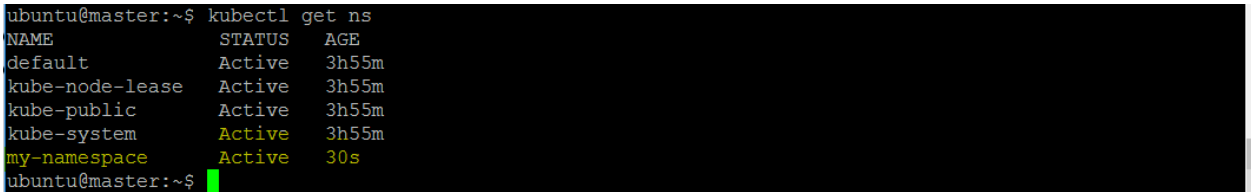
```
ubuntu@master:~$ echo $(id -g)
1000
ubuntu@master:~$ echo $(id -g)
1000
ubuntu@master:~$ sudo chown $(id -u):$(id -g) ~/.kube/config
ubuntu@master:~$ echo $(id -u)
1000
ubuntu@master:~$ ls -l ls -l ~/.kube/config
ls: cannot access 'ls': No such file or directory
-rwxr-xr-x 1 ubuntu ubuntu 5592 Dec  5 11:56 /home/ubuntu/.kube/config
ubuntu@master:~$
```



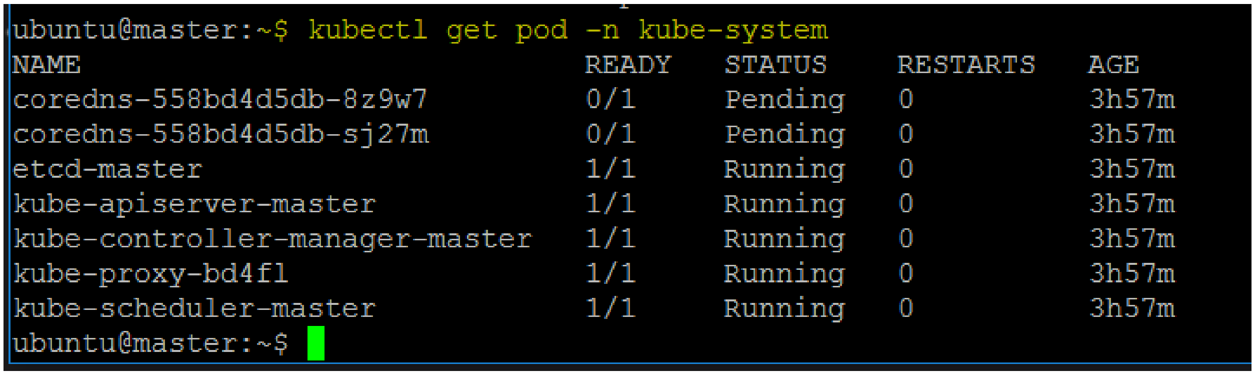
```
ubuntu@master:~$ kubectl cluster-info
Kubernetes control plane is running at https://172.31.1.216:6443
CoreDNS is running at https://172.31.1.216:6443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
ubuntu@master:~$
```



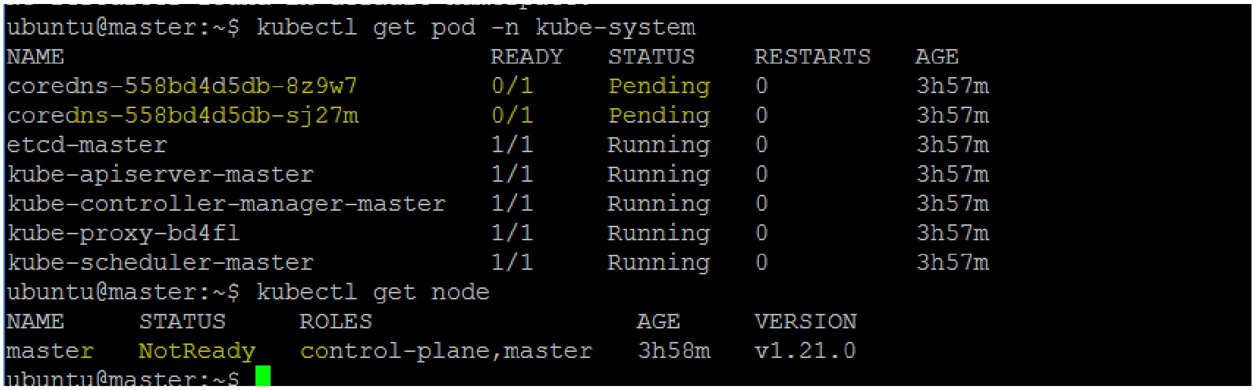
```
ubuntu@master:~$ kubectl create namespace my-namespace
namespace/my-namespace created
ubuntu@master:~$
```



```
ubuntu@master:~$ kubectl get ns
NAME              STATUS   AGE
default           Active   3h55m
kube-node-lease   Active   3h55m
kube-public       Active   3h55m
kube-system       Active   3h55m
my-namespace      Active   30s
ubuntu@master:~$
```



```
ubuntu@master:~$ kubectl get pod -n kube-system
NAME                               READY   STATUS    RESTARTS   AGE
coredns-558bd4d5db-8z9w7           0/1     Pending   0          3h57m
coredns-558bd4d5db-sj27m           0/1     Pending   0          3h57m
etcd-master                        1/1     Running   0          3h57m
kube-apiserver-master              1/1     Running   0          3h57m
kube-controller-manager-master     1/1     Running   0          3h57m
kube-proxy-bd4fl                   1/1     Running   0          3h57m
kube-scheduler-master              1/1     Running   0          3h57m
ubuntu@master:~$
```



```
ubuntu@master:~$ kubectl get pod -n kube-system
NAME                               READY   STATUS    RESTARTS   AGE
coredns-558bd4d5db-8z9w7           0/1     Pending   0          3h57m
coredns-558bd4d5db-sj27m           0/1     Pending   0          3h57m
etcd-master                        1/1     Running   0          3h57m
kube-apiserver-master              1/1     Running   0          3h57m
kube-controller-manager-master     1/1     Running   0          3h57m
kube-proxy-bd4fl                   1/1     Running   0          3h57m
kube-scheduler-master              1/1     Running   0          3h57m
ubuntu@master:~$ kubectl get node
NAME     STATUS     ROLES                  AGE     VERSION
master   NotReady   control-plane,master   3h58m   v1.21.0
ubuntu@master:~$
```

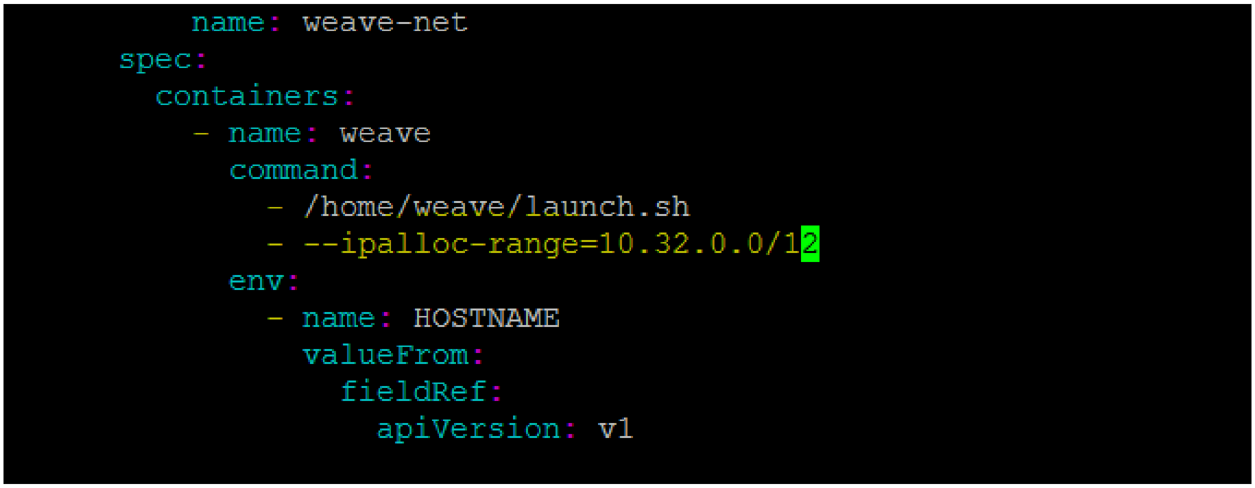
https://www.weave.works/docs/net/latest/kubernetes/kube-addon/

```
kubectl apply -f "https://cloud.weave.works/k8s/net?k8s-version=$(kubectl version | base64 | tr -d '\n')"
```

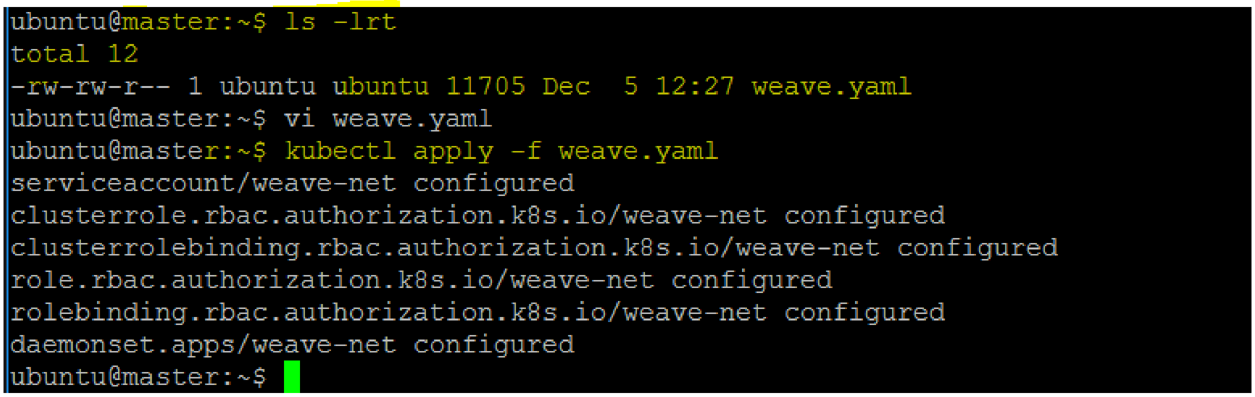
kubectl apply -f https://cloud.weave.works/k8s/net?k8s-version=$(kubectl version | base64 | tr -d '\n')

wget "https://cloud.weave.works/k8s/net?k8s-version=$(kubectl version | base64 | tr -d '\n')" -O weave.yaml

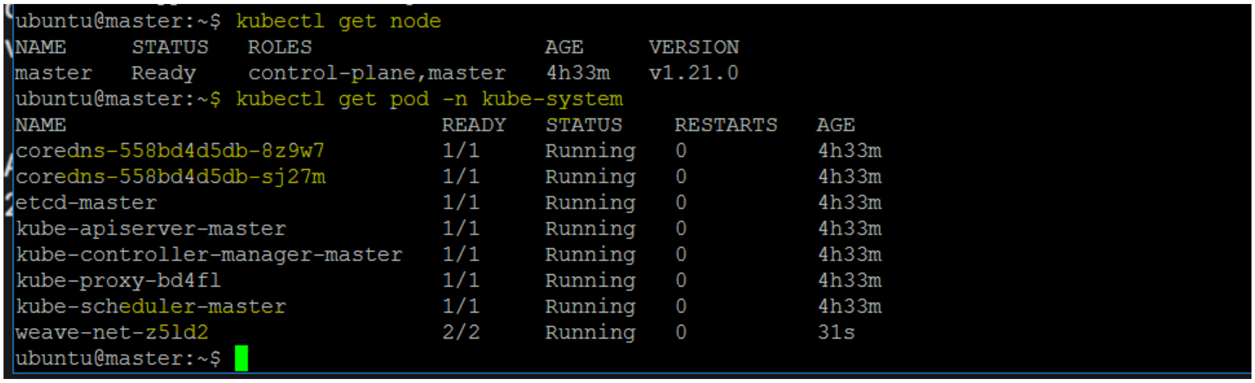
10.32.0.0/12



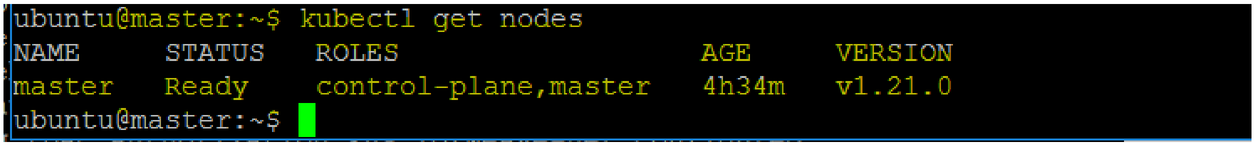
```
          name: weave-net
      spec:
        containers:
          - name: weave
            command:
              - /home/weave/launch.sh
              - --ipalloc-range=10.32.0.0/12
            env:
              - name: HOSTNAME
                valueFrom:
                  fieldRef:
                    apiVersion: v1
```



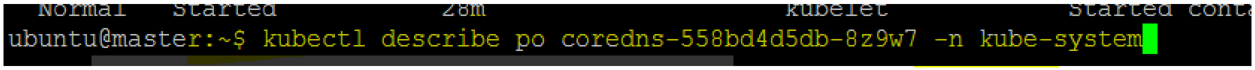
```
ubuntu@master:~$ ls -lrt
total 12
-rw-rw-r-- 1 ubuntu ubuntu 11705 Dec  5 12:27 weave.yaml
ubuntu@master:~$ vi weave.yaml
ubuntu@master:~$ kubectl apply -f weave.yaml
serviceaccount/weave-net configured
clusterrole.rbac.authorization.k8s.io/weave-net configured
clusterrolebinding.rbac.authorization.k8s.io/weave-net configured
role.rbac.authorization.k8s.io/weave-net configured
rolebinding.rbac.authorization.k8s.io/weave-net configured
daemonset.apps/weave-net configured
ubuntu@master:~$
```



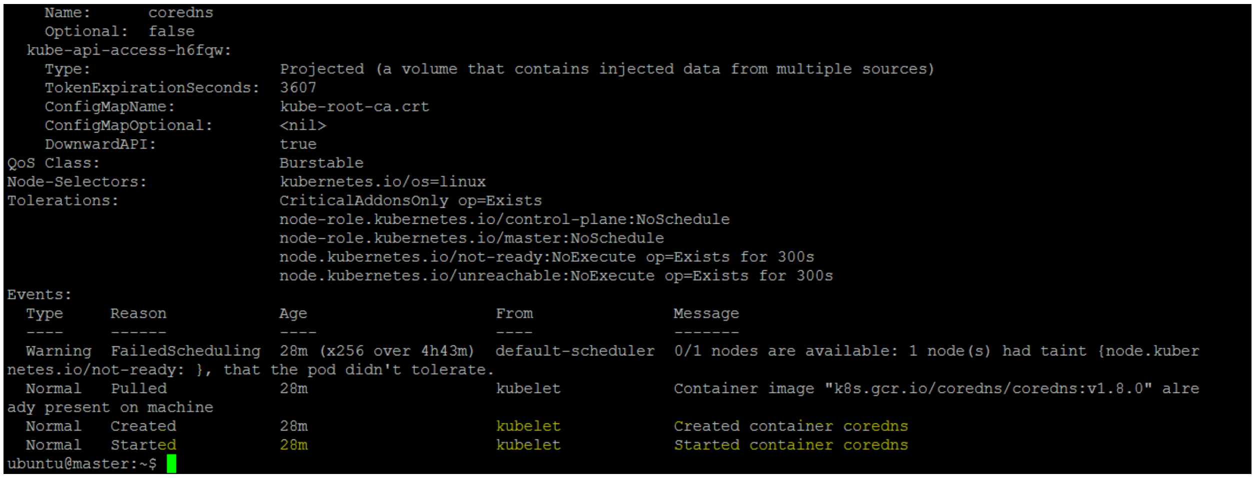
```
ubuntu@master:~$ kubectl get node
NAME      STATUS   ROLES                  AGE     VERSION
master    Ready    control-plane,master   4h33m   v1.21.0
ubuntu@master:~$ kubectl get pod -n kube-system
NAME                             READY   STATUS    RESTARTS   AGE
coredns-558bd4d5db-8z9w7         1/1     Running   0          4h33m
coredns-558bd4d5db-sj27m         1/1     Running   0          4h33m
etcd-master                      1/1     Running   0          4h33m
kube-apiserver-master            1/1     Running   0          4h33m
kube-controller-manager-master   1/1     Running   0          4h33m
kube-proxy-bd4fl                 1/1     Running   0          4h33m
kube-scheduler-master            1/1     Running   0          4h33m
weave-net-z5ld2                  2/2     Running   0          31s
ubuntu@master:~$
```



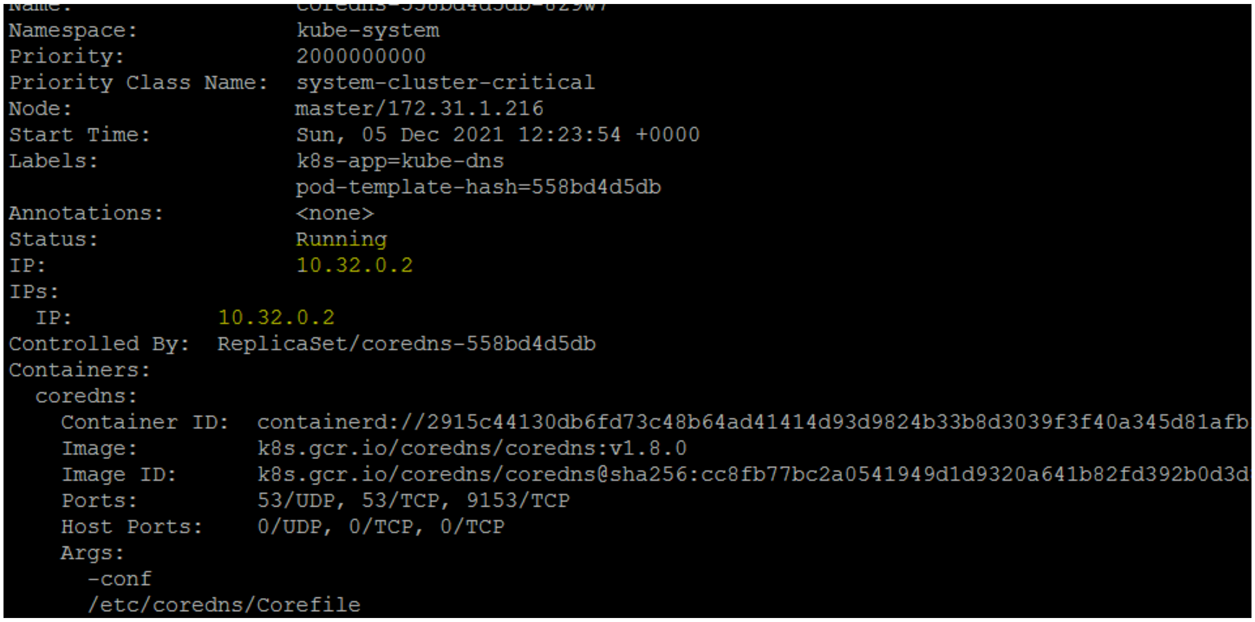
```
ubuntu@master:~$ kubectl get nodes
NAME     STATUS   ROLES                  AGE     VERSION
master   Ready    control-plane,master   4h34m   v1.21.0
ubuntu@master:~$
```



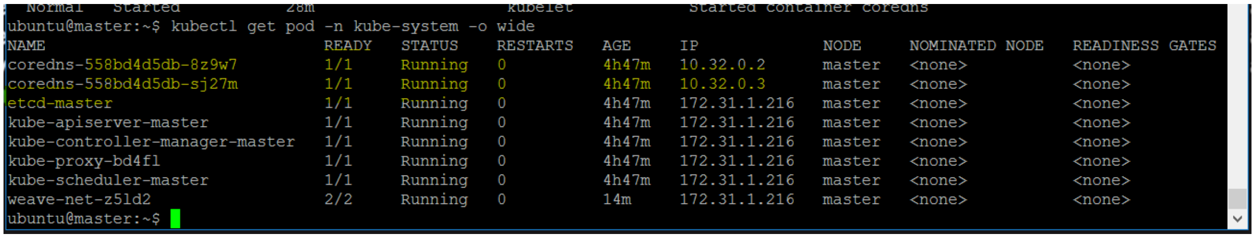
```
  Normal   Started              28m                        kubelet            Started conta
ubuntu@master:~$ kubectl describe po coredns-558bd4d5db-8z9w7 -n kube-system
```



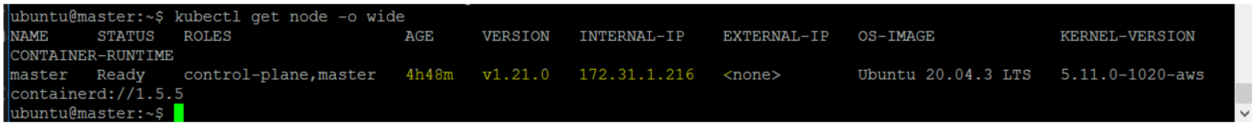
```
    Name:      coredns
    Optional:  false
  kube-api-access-h6fqw:
    Type:                    Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    ConfigMapOptional:       <nil>
    DownwardAPI:             true
QoS Class:                   Burstable
Node-Selectors:              kubernetes.io/os=linux
Tolerations:                 CriticalAddonsOnly op=Exists
                             node-role.kubernetes.io/control-plane:NoSchedule
                             node-role.kubernetes.io/master:NoSchedule
                             node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                             node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type     Reason            Age                    From               Message
  ----     ------            ----                   ----               -------
  Warning  FailedScheduling  28m (x256 over 4h43m)  default-scheduler  0/1 nodes are available: 1 node(s) had taint {node.kuber
netes.io/not-ready: }, that the pod didn't tolerate.
  Normal   Pulled            28m                    kubelet            Container image "k8s.gcr.io/coredns/coredns:v1.8.0" alre
ady present on machine
  Normal   Created           28m                    kubelet            Created container coredns
  Normal   Started           28m                    kubelet            Started container coredns
ubuntu@master:~$
```



```
Name:                 coredns-558bd4d5db-8z9w7
Namespace:            kube-system
Priority:             2000000000
Priority Class Name:  system-cluster-critical
Node:                 master/172.31.1.216
Start Time:           Sun, 05 Dec 2021 12:23:54 +0000
Labels:               k8s-app=kube-dns
                      pod-template-hash=558bd4d5db
Annotations:          <none>
Status:               Running
IP:                   10.32.0.2
IPs:
  IP:           10.32.0.2
Controlled By:  ReplicaSet/coredns-558bd4d5db
Containers:
  coredns:
    Container ID:  containerd://2915c44130db6fd73c48b64ad41414d93d9824b33b8d3039f3f40a345d81afb
    Image:         k8s.gcr.io/coredns/coredns:v1.8.0
    Image ID:      k8s.gcr.io/coredns/coredns@sha256:cc8fb77bc2a0541949d1d9320a641b82fd392b0d3d
    Ports:         53/UDP, 53/TCP, 9153/TCP
    Host Ports:    0/UDP, 0/TCP, 0/TCP
    Args:
      -conf
      /etc/coredns/Corefile
```

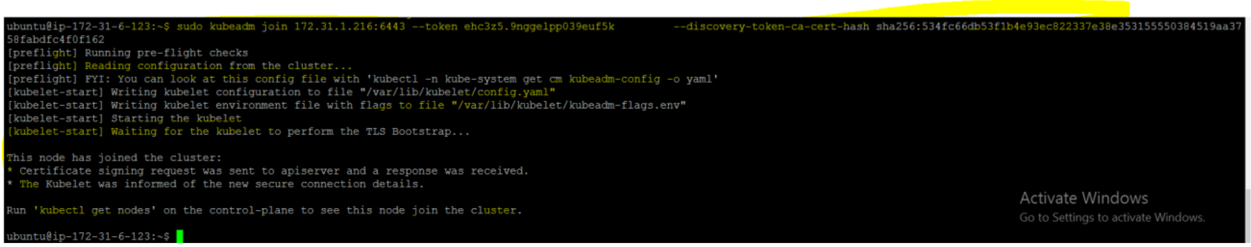


```
  Normal   Started          28m                 kubelet            Started container coredns
ubuntu@master:~$ kubectl get pod -n kube-system -o wide
NAME                             READY   STATUS    RESTARTS   AGE    IP            NODE     NOMINATED NODE   READINESS GATES
coredns-558bd4d5db-8z9w7         1/1     Running   0          4h47m  10.32.0.2     master   <none>           <none>
coredns-558bd4d5db-sj27m         1/1     Running   0          4h47m  10.32.0.3     master   <none>           <none>
etcd-master                      1/1     Running   0          4h47m  172.31.1.216  master   <none>           <none>
kube-apiserver-master            1/1     Running   0          4h47m  172.31.1.216  master   <none>           <none>
kube-controller-manager-master   1/1     Running   0          4h47m  172.31.1.216  master   <none>           <none>
kube-proxy-bd4fl                 1/1     Running   0          4h47m  172.31.1.216  master   <none>           <none>
kube-scheduler-master            1/1     Running   0          4h47m  172.31.1.216  master   <none>           <none>
weave-net-z5ld2                  2/2     Running   0          14m    172.31.1.216  master   <none>           <none>
ubuntu@master:~$
```



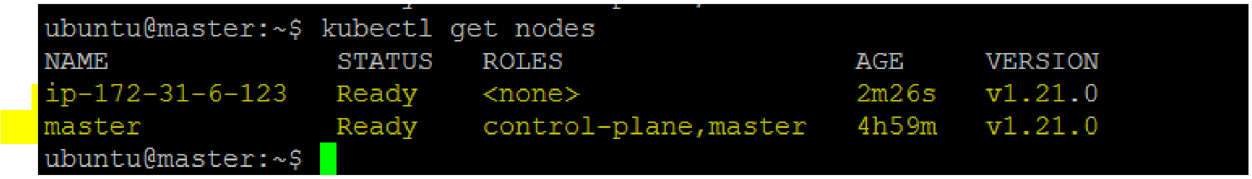
```
ubuntu@master:~$ kubectl get node -o wide
NAME     STATUS   ROLES                  AGE     VERSION   INTERNAL-IP    EXTERNAL-IP   OS-IMAGE             KERNEL-VERSION
CONTAINER-RUNTIME
master   Ready    control-plane,master   4h48m   v1.21.0   172.31.1.216   <none>        Ubuntu 20.04.3 LTS   5.11.0-1020-aws
containerd://1.5.5
ubuntu@master:~$
```

Worker node adding to master node

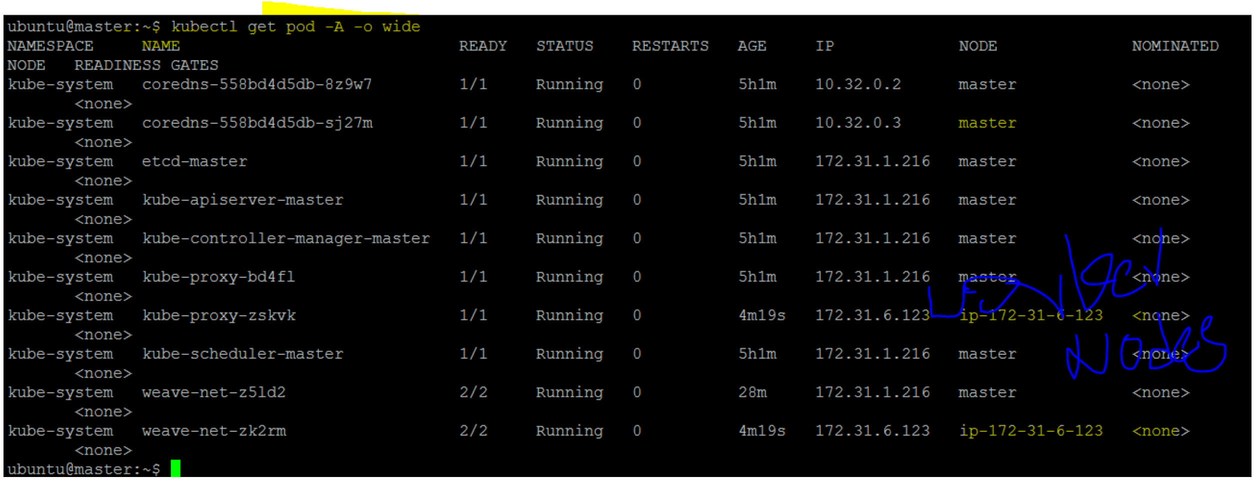


```
ubuntu@ip-172-31-6-123:~$ sudo kubeadm join 172.31.1.216:6443 --token ehc3z5.9nggelpp039euf5k         --discovery-token-ca-cert-hash sha256:534fc66db53f1b4e93ec822337e38e353155550384519aa37
58fabdfc4f0f162
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[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.

ubuntu@ip-172-31-6-123:~$
```

Activate Windows
Go to Settings to activate Windows.



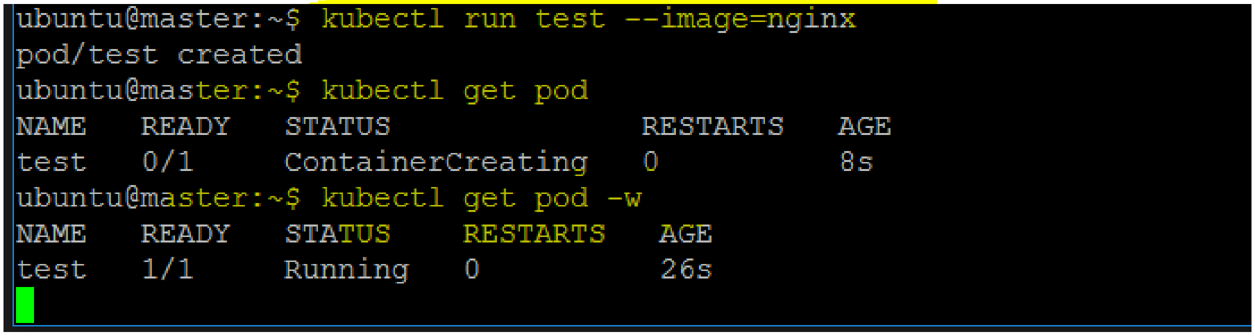
```
ubuntu@master:~$ kubectl get nodes
NAME              STATUS   ROLES                  AGE     VERSION
ip-172-31-6-123   Ready    <none>                 2m26s   v1.21.0
master            Ready    control-plane,master   4h59m   v1.21.0
ubuntu@master:~$
```



```
ubuntu@master:~$ kubectl get pod -A -o wide
NAMESPACE     NAME                             READY   STATUS    RESTARTS   AGE     IP             NODE              NOMINATED
NODE   READINESS GATES
kube-system   coredns-558bd4d5db-8z9w7         1/1     Running   0          5h1m    10.32.0.2      master            <none>
         <none>
kube-system   coredns-558bd4d5db-sj27m         1/1     Running   0          5h1m    10.32.0.3      master            <none>
         <none>
kube-system   etcd-master                      1/1     Running   0          5h1m    172.31.1.216   master            <none>
         <none>
kube-system   kube-apiserver-master            1/1     Running   0          5h1m    172.31.1.216   master            <none>
         <none>
kube-system   kube-controller-manager-master   1/1     Running   0          5h1m    172.31.1.216   master            <none>
         <none>
kube-system   kube-proxy-bd4fl                 1/1     Running   0          5h1m    172.31.1.216   master            <none>
         <none>
kube-system   kube-proxy-zskvk                 1/1     Running   0          4m19s   172.31.6.123   ip-172-31-6-123   <none>
         <none>
kube-system   kube-scheduler-master            1/1     Running   0          5h1m    172.31.1.216   master            <none>
         <none>
kube-system   weave-net-z5ld2                  2/2     Running   0          28m     172.31.1.216   master            <none>
         <none>
kube-system   weave-net-zk2rm                  2/2     Running   0          4m19s   172.31.6.123   ip-172-31-6-123   <none>
         <none>
ubuntu@master:~$
```

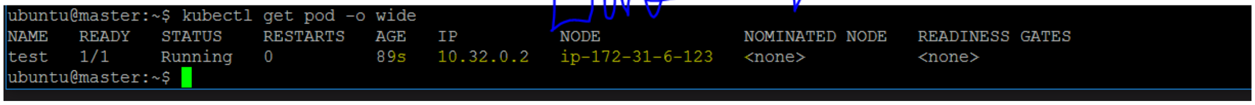
Worker Nodes

How to run the NGINX pod



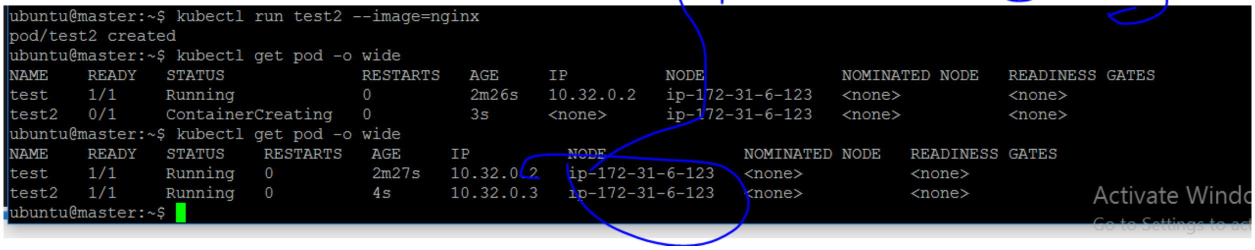
```
ubuntu@master:~$ kubectl run test --image=nginx
pod/test created
ubuntu@master:~$ kubectl get pod
NAME    READY   STATUS              RESTARTS   AGE
test    0/1     ContainerCreating   0          8s
ubuntu@master:~$ kubectl get pod -w
NAME    READY   STATUS    RESTARTS   AGE
test    1/1     Running   0          26s
```

Worker Node



```
ubuntu@master:~$ kubectl get pod -o wide
NAME   READY   STATUS    RESTARTS   AGE   IP          NODE               NOMINATED NODE   READINESS GATES
test   1/1     Running   0          89s   10.32.0.2   ip-172-31-6-123    <none>           <none>
ubuntu@master:~$
```

Worker Nodes



```
ubuntu@master:~$ kubectl run test2 --image=nginx
pod/test2 created
ubuntu@master:~$ kubectl get pod -o wide
NAME    READY   STATUS              RESTARTS   AGE     IP          NODE               NOMINATED NODE   READINESS GATES
test    1/1     Running             0          2m26s   10.32.0.2   ip-172-31-6-123    <none>           <none>
test2   0/1     ContainerCreating   0          3s      <none>      ip-172-31-6-123    <none>           <none>
ubuntu@master:~$ kubectl get pod -o wide
NAME    READY   STATUS    RESTARTS   AGE     IP          NODE               NOMINATED NODE   READINESS GATES
test    1/1     Running   0          2m27s   10.32.0.2   ip-172-31-6-123    <none>           <none>
test2   1/1     Running   0          4s      10.32.0.3   ip-172-31-6-123    <none>           <none>
ubuntu@master:~$
```