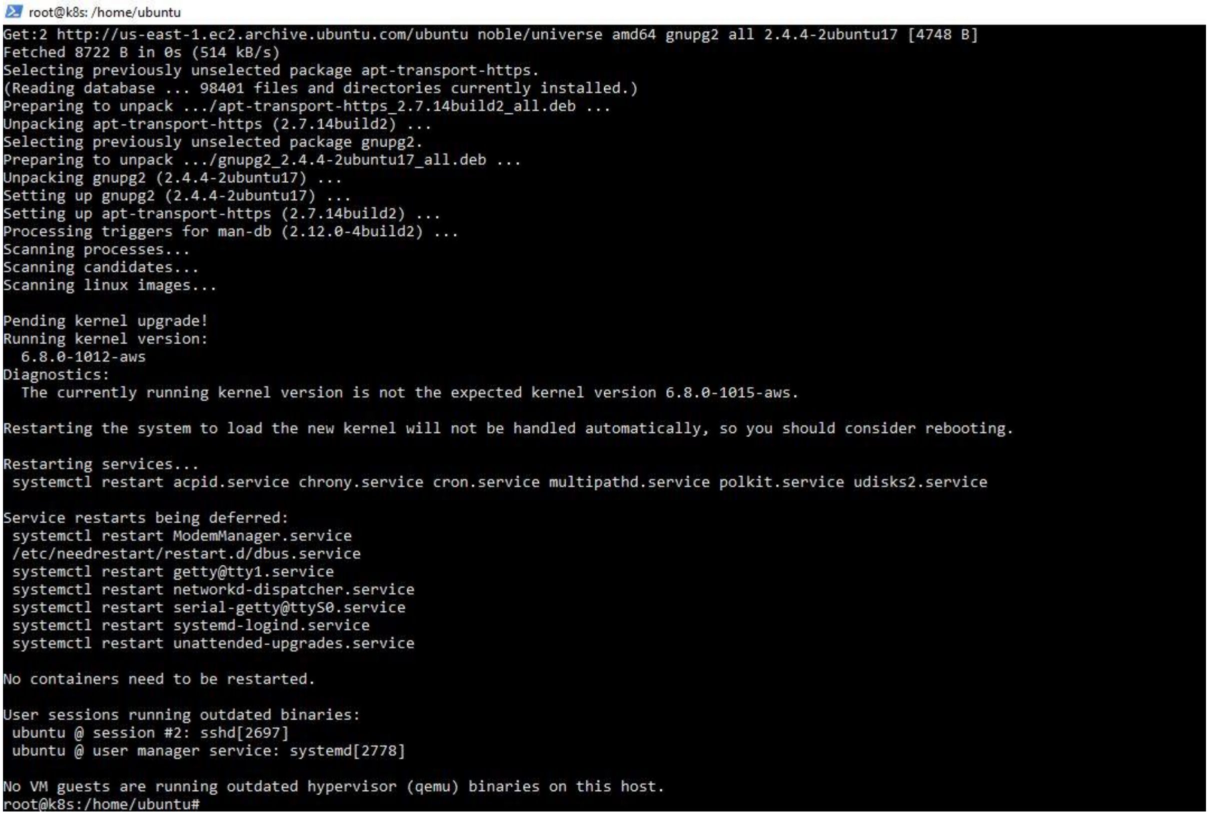
Setup Kubernetes –

Launch EC2 instance

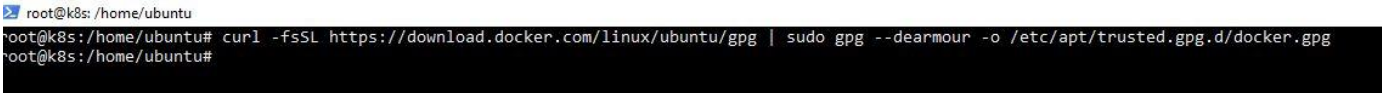
swapoff -a – This command is used to disable all swap space on the system. Swap space is an area on the disk that is used when the system's RAM is fully utilized. By turning off swap, you're instructing the system to stop using swap and rely solely on physical RAM.

apt install -y curl gnupg2 software-properties-common apt-transport-https ca-certificates



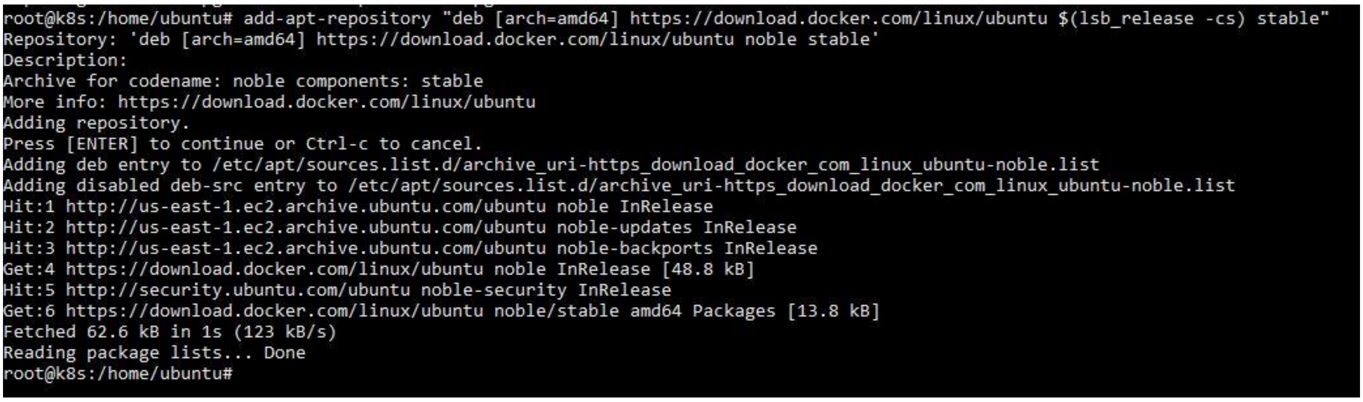
```
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 gnupg2 all 2.4.4-2ubuntu17 [4748 B]
Fetched 8722 B in 0s (514 kB/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 98401 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_2.7.14build2_all.deb ...
Unpacking apt-transport-https (2.7.14build2) ...
Selecting previously unselected package gnupg2.
Preparing to unpack .../gnupg2_2.4.4-2ubuntu17_all.deb ...
Unpacking gnupg2 (2.4.4-2ubuntu17) ...
Setting up gnupg2 (2.4.4-2ubuntu17) ...
Setting up apt-transport-https (2.7.14build2) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning candidates...
Scanning linux images...

Pending kernel upgrade!
Running kernel version:
  6.8.0-1012-aws
Diagnostics:
  The currently running kernel version is not the expected kernel version 6.8.0-1015-aws.

Restarting the system to load the new kernel will not be handled automatically, so you should consider rebooting.

Restarting services...
 systemctl restart acpid.service chrony.service cron.service multipathd.service polkit.service udisks2.service

Service restarts being deferred:
 systemctl restart ModemManager.service
 /etc/needrestart/restart.d/dbus.service
 systemctl restart getty@tty1.service
 systemctl restart networkd-dispatcher.service
 systemctl restart serial-getty@ttyS0.service
 systemctl restart systemd-logind.service
 systemctl restart unattended-upgrades.service

No containers need to be restarted.

User sessions running outdated binaries:
 ubuntu @ session #2: sshd[2697]
 ubuntu @ user manager service: systemd[2778]

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@k8s:/home/ubuntu#
```

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmour -o /etc/apt/trusted.gpg.d/docker.gpg - This command sequence is used to add Docker's official GPG key to your system's list of trusted keys.



```
oot@k8s:/home/ubuntu# curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmour -o /etc/apt/trusted.gpg.d/docker.gpg
oot@k8s:/home/ubuntu#
```

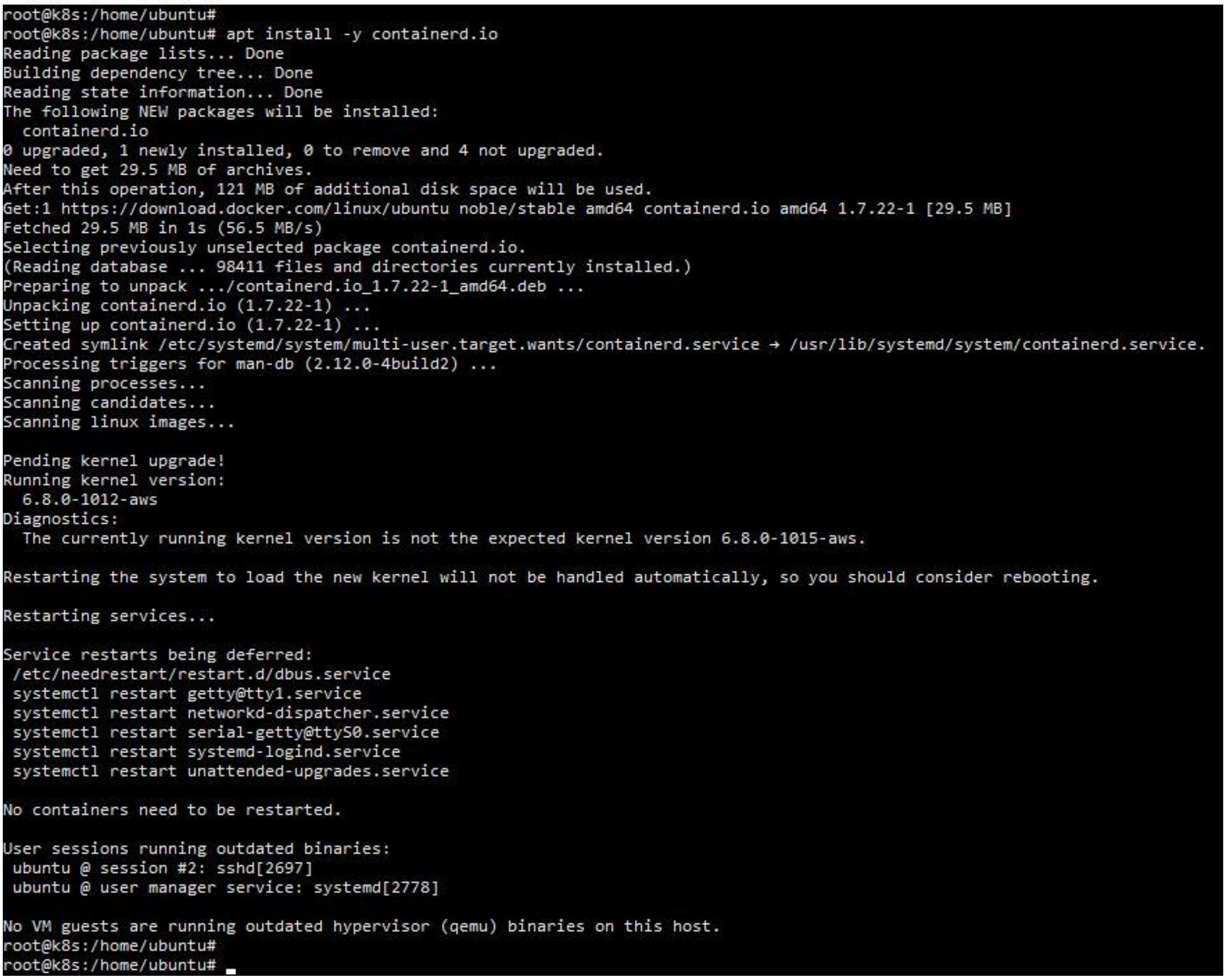
add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" - This command adds Docker's APT repository to your system.



```
root@k8s:/home/ubuntu# add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
Repository: 'deb [arch=amd64] https://download.docker.com/linux/ubuntu noble stable'
Description:
Archive for codename: noble components: stable
More info: https://download.docker.com/linux/ubuntu
Adding repository.
Press [ENTER] to continue or Ctrl-c to cancel.
Adding deb entry to /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-noble.list
Adding disabled deb-src entry to /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-noble.list
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Get:4 https://download.docker.com/linux/ubuntu noble InRelease [48.8 kB]
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:6 https://download.docker.com/linux/ubuntu noble/stable amd64 Packages [13.8 kB]
Fetched 62.6 kB in 1s (123 kB/s)
Reading package lists... Done
root@k8s:/home/ubuntu#
```

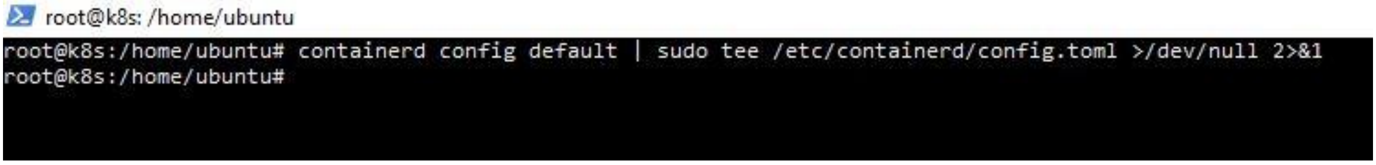
apt update - This command updates the local package database.

apt install -y containerd.io - install the containerd package



```
root@k8s:/home/ubuntu#
root@k8s:/home/ubuntu# apt install -y containerd.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  containerd.io
0 upgraded, 1 newly installed, 0 to remove and 4 not upgraded.
Need to get 29.5 MB of archives.
After this operation, 121 MB of additional disk space will be used.
Get:1 https://download.docker.com/linux/ubuntu noble/stable amd64 containerd.io amd64 1.7.22-1 [29.5 MB]
Fetched 29.5 MB in 1s (56.5 MB/s)
Selecting previously unselected package containerd.io.
(Reading database ... 98411 files and directories currently installed.)
Preparing to unpack .../containerd.io_1.7.22-1_amd64.deb ...
Unpacking containerd.io (1.7.22-1) ...
Setting up containerd.io (1.7.22-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /usr/lib/systemd/system/containerd.service.
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning candidates...
Scanning linux images...

Pending kernel upgrade!
Running kernel version:
  6.8.0-1012-aws
Diagnostics:
  The currently running kernel version is not the expected kernel version 6.8.0-1015-aws.

Restarting the system to load the new kernel will not be handled automatically, so you should consider rebooting.

Restarting services...

Service restarts being deferred:
 /etc/needrestart/restart.d/dbus.service
 systemctl restart getty@tty1.service
 systemctl restart networkd-dispatcher.service
 systemctl restart serial-getty@ttyS0.service
 systemctl restart systemd-logind.service
 systemctl restart unattended-upgrades.service

No containers need to be restarted.

User sessions running outdated binaries:
 ubuntu @ session #2: sshd[2697]
 ubuntu @ user manager service: systemd[2778]

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@k8s:/home/ubuntu#
root@k8s:/home/ubuntu# _
```

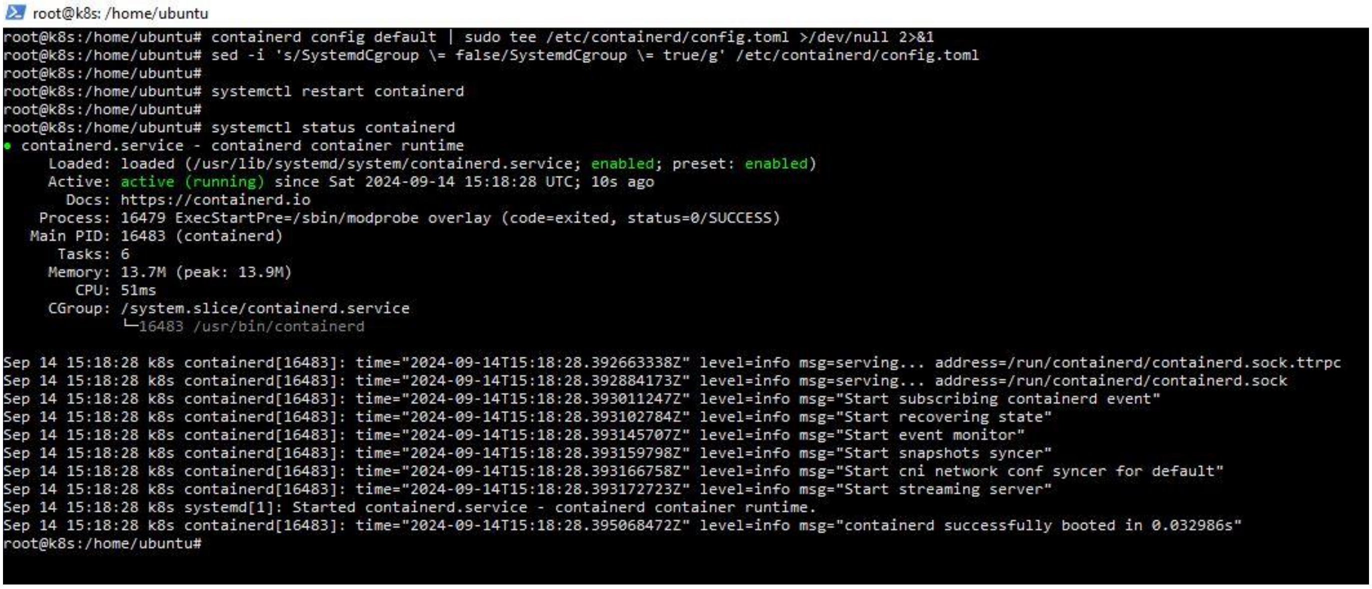
containerd config default | sudo tee /etc/containerd/config.toml >/dev/null 2>&1 - The command is used to generate and save a default configuration file for containerd.



```
root@k8s:/home/ubuntu# containerd config default | sudo tee /etc/containerd/config.toml >/dev/null 2>&1
root@k8s:/home/ubuntu#
```

sed -i 's/SystemdCgroup \= false/SystemdCgroup \= true/g' /etc/containerd/config.toml - This command is used to modify the containerd configuration file.

systemctl restart containerd - Restarts the containerd service

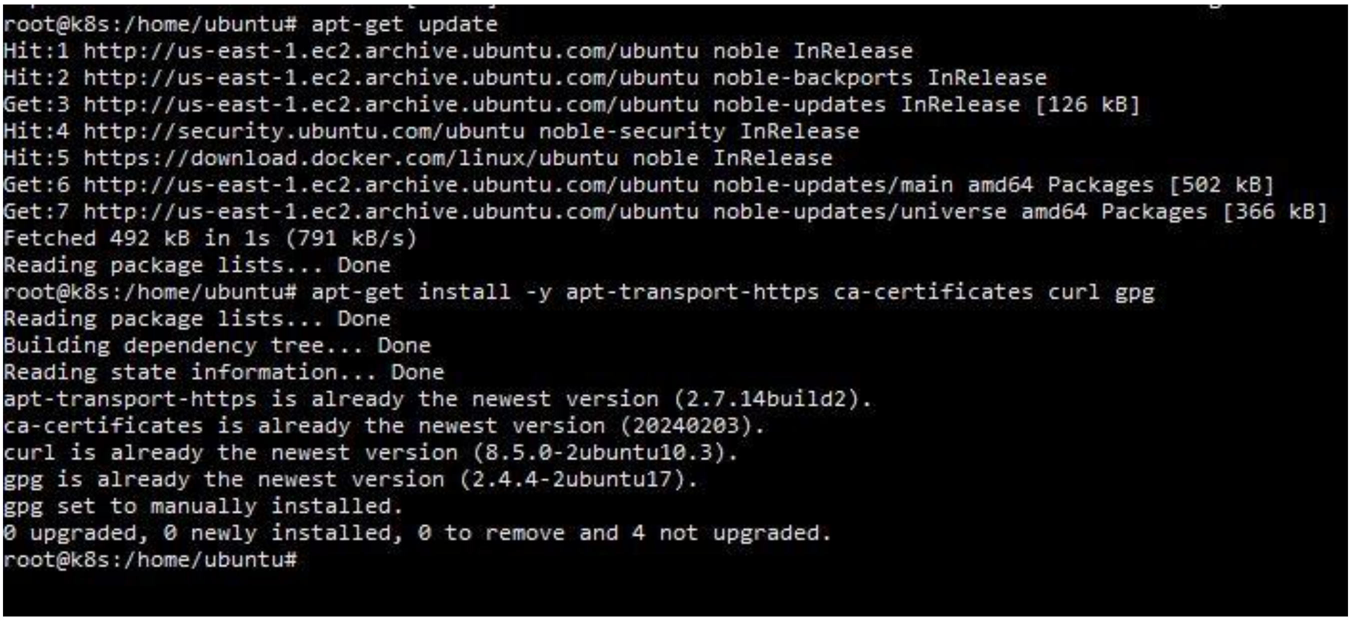


```
root@k8s:/home/ubuntu# containerd config default | sudo tee /etc/containerd/config.toml >/dev/null 2>&1
root@k8s:/home/ubuntu# sed -i 's/SystemdCgroup \= false/SystemdCgroup \= true/g' /etc/containerd/config.toml
root@k8s:/home/ubuntu#
root@k8s:/home/ubuntu# systemctl restart containerd
root@k8s:/home/ubuntu#
root@k8s:/home/ubuntu# systemctl status containerd
● containerd.service - containerd container runtime
     Loaded: loaded (/usr/lib/systemd/system/containerd.service; enabled; preset: enabled)
     Active: active (running) since Sat 2024-09-14 15:18:28 UTC; 10s ago
       Docs: https://containerd.io
    Process: 16479 ExecStartPre=/sbin/modprobe overlay (code=exited, status=0/SUCCESS)
   Main PID: 16483 (containerd)
      Tasks: 6
     Memory: 13.7M (peak: 13.9M)
        CPU: 51ms
     CGroup: /system.slice/containerd.service
             └─16483 /usr/bin/containerd

Sep 14 15:18:28 k8s containerd[16483]: time="2024-09-14T15:18:28.392663338Z" level=info msg=serving... address=/run/containerd/containerd.sock.ttrpc
Sep 14 15:18:28 k8s containerd[16483]: time="2024-09-14T15:18:28.392884173Z" level=info msg=serving... address=/run/containerd/containerd.sock
Sep 14 15:18:28 k8s containerd[16483]: time="2024-09-14T15:18:28.393011247Z" level=info msg="Start subscribing containerd event"
Sep 14 15:18:28 k8s containerd[16483]: time="2024-09-14T15:18:28.393102784Z" level=info msg="Start recovering state"
Sep 14 15:18:28 k8s containerd[16483]: time="2024-09-14T15:18:28.393145707Z" level=info msg="Start event monitor"
Sep 14 15:18:28 k8s containerd[16483]: time="2024-09-14T15:18:28.393159798Z" level=info msg="Start snapshots syncer"
Sep 14 15:18:28 k8s containerd[16483]: time="2024-09-14T15:18:28.393166758Z" level=info msg="Start cni network conf syncer for default"
Sep 14 15:18:28 k8s containerd[16483]: time="2024-09-14T15:18:28.393172723Z" level=info msg="Start streaming server"
Sep 14 15:18:28 k8s systemd[1]: Started containerd.service - containerd container runtime.
Sep 14 15:18:28 k8s containerd[16483]: time="2024-09-14T15:18:28.395068472Z" level=info msg="containerd successfully booted in 0.032986s"
root@k8s:/home/ubuntu#
```

Goto below website:
https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/

Install Kubernetes v1.31 –
apt-get update

# apt-transport-https may be a dummy package; if so, you can skip that package
apt-get install -y apt-transport-https ca-certificates curl gpg



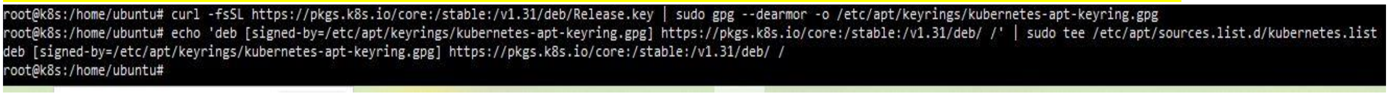
```
root@k8s:/home/ubuntu# apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:5 https://download.docker.com/linux/ubuntu noble InRelease
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [502 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [366 kB]
Fetched 492 kB in 1s (791 kB/s)
Reading package lists... Done
root@k8s:/home/ubuntu# apt-get install -y apt-transport-https ca-certificates curl gpg
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apt-transport-https is already the newest version (2.7.14build2).
ca-certificates is already the newest version (20240203).
curl is already the newest version (8.5.0-2ubuntu10.3).
gpg is already the newest version (2.4.4-2ubuntu17).
gpg set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 4 not upgraded.
root@k8s:/home/ubuntu#
```

# If the directory `/etc/apt/keyrings` does not exist, it should be created before the curl command, reads “sudo mkdir -p -m 755 /etc/apt/keyrings”

curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg

# This overwrites any existing configuration in /etc/apt/sources.list.d/kubernetes.list

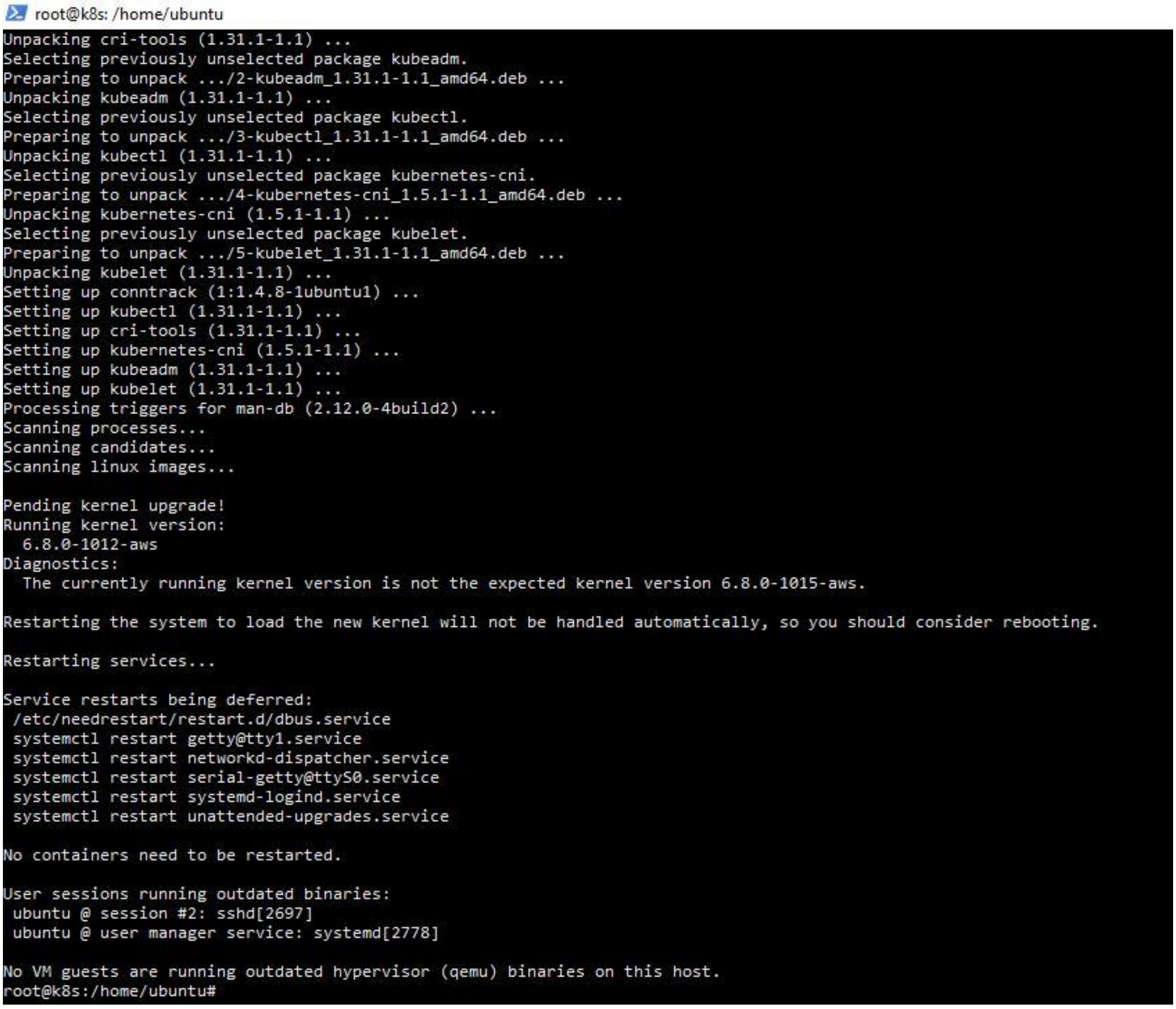
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list



```
root@k8s:/home/ubuntu# curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
root@k8s:/home/ubuntu# echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /
root@k8s:/home/ubuntu#
```

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

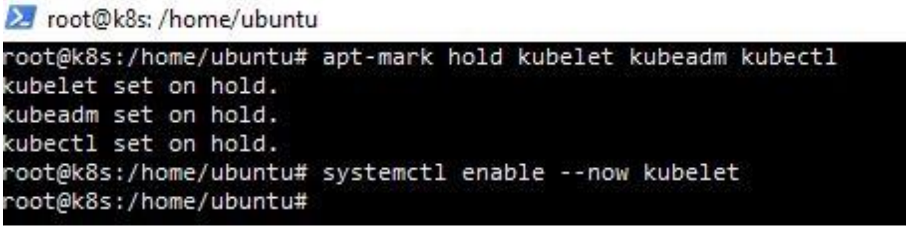
apt-get update
apt-get install -y kubelet kubeadm kubectl
apt-mark hold kubelet kubeadm kubectl



```
Unpacking cri-tools (1.31.1-1.1) ...
Selecting previously unselected package kubeadm.
Preparing to unpack .../2-kubeadm_1.31.1-1.1_amd64.deb ...
Unpacking kubeadm (1.31.1-1.1) ...
Selecting previously unselected package kubectl.
Preparing to unpack .../3-kubectl_1.31.1-1.1_amd64.deb ...
Unpacking kubectl (1.31.1-1.1) ...
Selecting previously unselected package kubernetes-cni.
Preparing to unpack .../4-kubernetes-cni_1.5.1-1.1_amd64.deb ...
Unpacking kubernetes-cni (1.5.1-1.1) ...
Selecting previously unselected package kubelet.
Preparing to unpack .../5-kubelet_1.31.1-1.1_amd64.deb ...
Unpacking kubelet (1.31.1-1.1) ...
Setting up conntrack (1:1.4.8-1ubuntu1) ...
Setting up kubectl (1.31.1-1.1) ...
Setting up cri-tools (1.31.1-1.1) ...
Setting up kubernetes-cni (1.5.1-1.1) ...
Setting up kubeadm (1.31.1-1.1) ...
Setting up kubelet (1.31.1-1.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning candidates...
Scanning linux images...

Pending kernel upgrade!
Running kernel version:
  6.8.0-1012-aws
Diagnostics:
  The currently running kernel version is not the expected kernel version 6.8.0-1015-aws.

Restarting the system to load the new kernel will not be handled automatically, so you should consider rebooting.

Restarting services...

Service restarts being deferred:
 /etc/needrestart/restart.d/dbus.service
 systemctl restart getty@tty1.service
 systemctl restart networkd-dispatcher.service
 systemctl restart serial-getty@ttyS0.service
 systemctl restart systemd-logind.service
 systemctl restart unattended-upgrades.service

No containers need to be restarted.

User sessions running outdated binaries:
 ubuntu @ session #2: sshd[2697]
 ubuntu @ user manager service: systemd[2778]

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@k8s:/home/ubuntu#
```

Enable the kubelet service before running kubeadm

systemctl enable --now kubelet

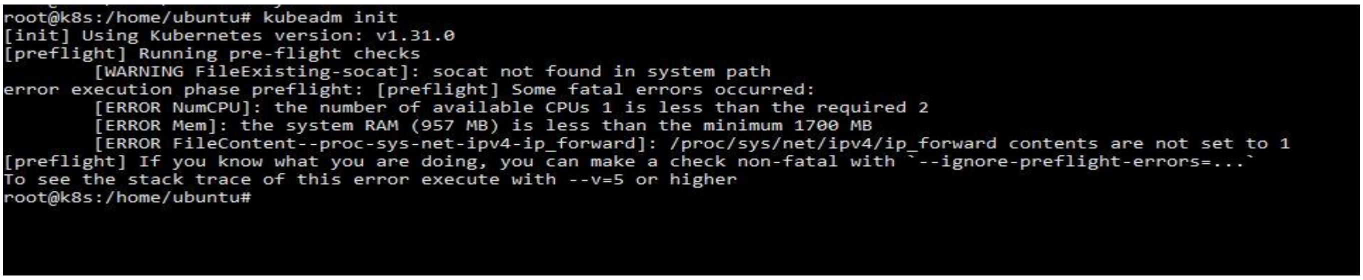


```
root@k8s:/home/ubuntu# apt-mark hold kubelet kubeadm kubectl
kubelet set on hold.
kubeadm set on hold.
kubectl set on hold.
root@k8s:/home/ubuntu# systemctl enable --now kubelet
root@k8s:/home/ubuntu#
```

Kubectl init

[ERROR Mem]: the system RAM (957 MB) is less than the minimum 1700 MB

[ERROR FileContent--proc-sys-net-ipv4-ip_forward]: /proc/sys/net/ipv4/ip_forward contents are not set to 1



```
root@k8s:/home/ubuntu# kubeadm init
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
        [WARNING FileExisting-socat]: socat not found in system path
error execution phase preflight: [preflight] Some fatal errors occurred:
        [ERROR NumCPU]: the number of available CPUs 1 is less than the required 2
        [ERROR Mem]: the system RAM (957 MB) is less than the minimum 1700 MB
        [ERROR FileContent--proc-sys-net-ipv4-ip_forward]: /proc/sys/net/ipv4/ip_forward contents are not set to 1
[preflight] If you know what you are doing, you can make a check non-fatal with `--ignore-preflight-errors=...`
To see the stack trace of this error execute with --v=5 or higher
root@k8s:/home/ubuntu#
```

So, we need to free ram and reinitialise iptables settings

Troubleshooting Steps -

1. Configure the Kernel Module ‘br_netfilter’ in the containerd configuration file.

```
tee /etc/modules-load.d/containerd.conf <<EOF
br_netfilter
EOF
```

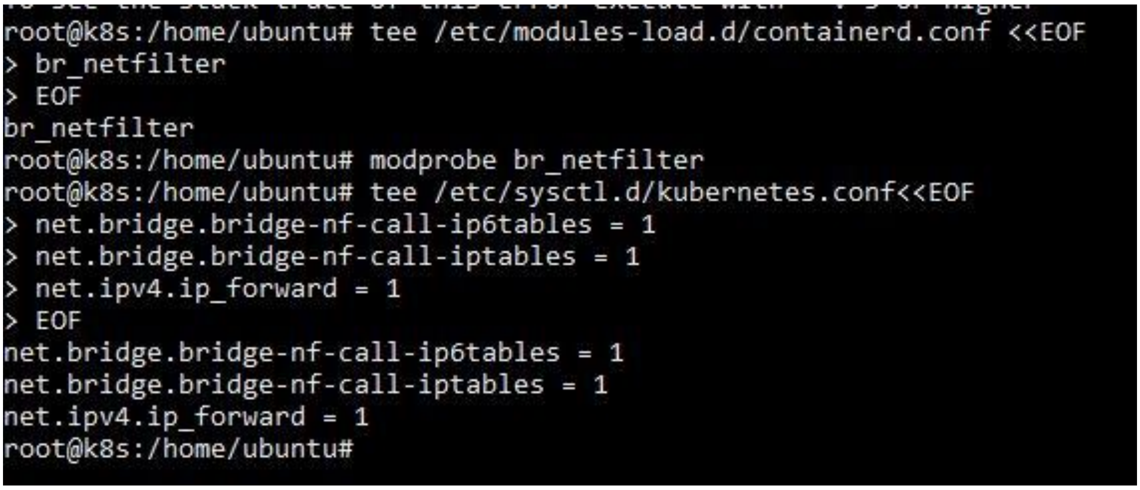
2. Load the br_netfilter modules into the running Linux kernel.

```
modprobe br_netfilter
```

3. Update Iptables Settings.

Note: To ensure packets are properly processed by IP tables during filtering and port forwarding, set the **net.bridge.bridge-nf-call-iptables to ‘1’** in your sysctl configuration file. Otherwise, you may encounter the following error: **[ERROR FileContent–proc-sys-net-ipv4-ip_forward]: /proc/sys/net/ipv4/ip_forward contents are not set to 1.** To avoid this, execute the following command.

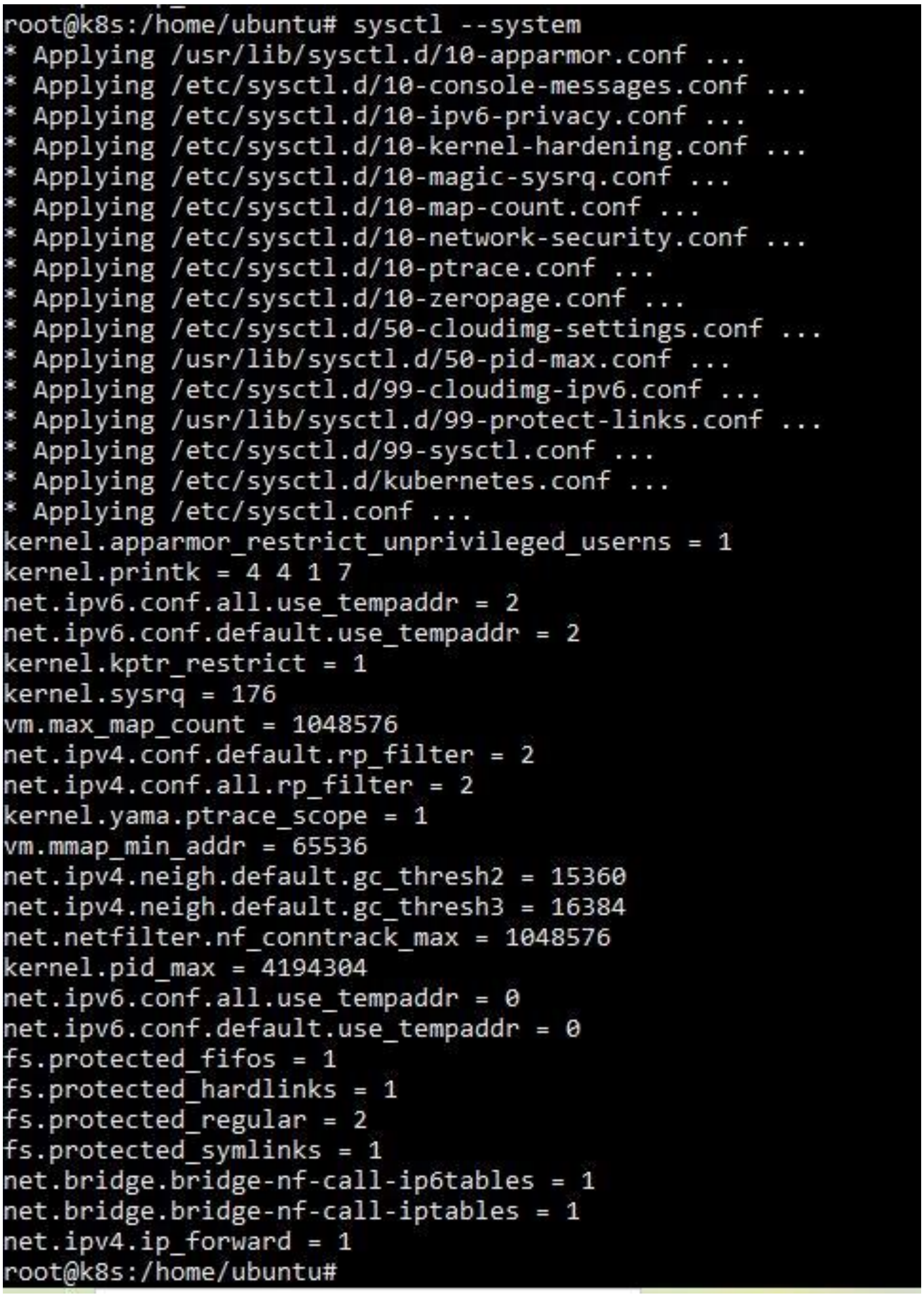
```
tee /etc/sysctl.d/kubernetes.conf<<EOF
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
net.ipv4.ip_forward = 1
EOF
```



```
root@k8s:/home/ubuntu# tee /etc/modules-load.d/containerd.conf <<EOF
> br_netfilter
> EOF
br_netfilter
root@k8s:/home/ubuntu# modprobe br_netfilter
root@k8s:/home/ubuntu# tee /etc/sysctl.d/kubernetes.conf<<EOF
> net.bridge.bridge-nf-call-ip6tables = 1
> net.bridge.bridge-nf-call-iptables = 1
> net.ipv4.ip_forward = 1
> EOF
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
net.ipv4.ip_forward = 1
root@k8s:/home/ubuntu#
```

4. Applying Kernel Settings Without Reboot

```
sysctl –system
```



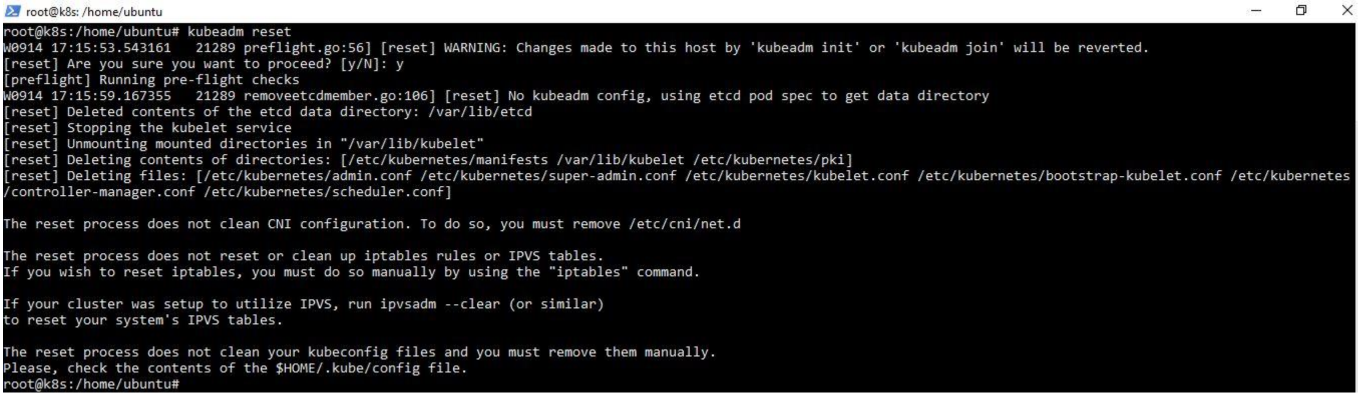
```
root@k8s:/home/ubuntu# sysctl --system
* Applying /usr/lib/sysctl.d/10-apparmor.conf ...
* Applying /etc/sysctl.d/10-console-messages.conf ...
* Applying /etc/sysctl.d/10-ipv6-privacy.conf ...
* Applying /etc/sysctl.d/10-kernel-hardening.conf ...
* Applying /etc/sysctl.d/10-magic-sysrq.conf ...
* Applying /etc/sysctl.d/10-map-count.conf ...
* Applying /etc/sysctl.d/10-network-security.conf ...
* Applying /etc/sysctl.d/10-ptrace.conf ...
* Applying /etc/sysctl.d/10-zeropage.conf ...
* Applying /etc/sysctl.d/50-cloudimg-settings.conf ...
* Applying /usr/lib/sysctl.d/50-pid-max.conf ...
* Applying /etc/sysctl.d/99-cloudimg-ipv6.conf ...
* Applying /usr/lib/sysctl.d/99-protect-links.conf ...
* Applying /etc/sysctl.d/99-sysctl.conf ...
* Applying /etc/sysctl.d/kubernetes.conf ...
* Applying /etc/sysctl.conf ...
kernel.apparmor_restrict_unprivileged_userns = 1
kernel.printk = 4 4 1 7
net.ipv6.conf.all.use_tempaddr = 2
net.ipv6.conf.default.use_tempaddr = 2
kernel.kptr_restrict = 1
kernel.sysrq = 176
vm.max_map_count = 1048576
net.ipv4.conf.default.rp_filter = 2
net.ipv4.conf.all.rp_filter = 2
kernel.yama.ptrace_scope = 1
vm.mmap_min_addr = 65536
net.ipv4.neigh.default.gc_thresh2 = 15360
net.ipv4.neigh.default.gc_thresh3 = 16384
net.netfilter.nf_conntrack_max = 1048576
kernel.pid_max = 4194304
net.ipv6.conf.all.use_tempaddr = 0
net.ipv6.conf.default.use_tempaddr = 0
fs.protected_fifos = 1
fs.protected_hardlinks = 1
fs.protected_regular = 2
fs.protected_symlinks = 1
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
net.ipv4.ip_forward = 1
root@k8s:/home/ubuntu#
```

5. Reset Kubernetes Configuration.

Before reconfiguring, it's crucial to reset the existing Kubernetes configuration to ensure a clean slate.

kubeadm reset

This command will revert any changes made to the cluster configuration, preparing it for a fresh initialization.



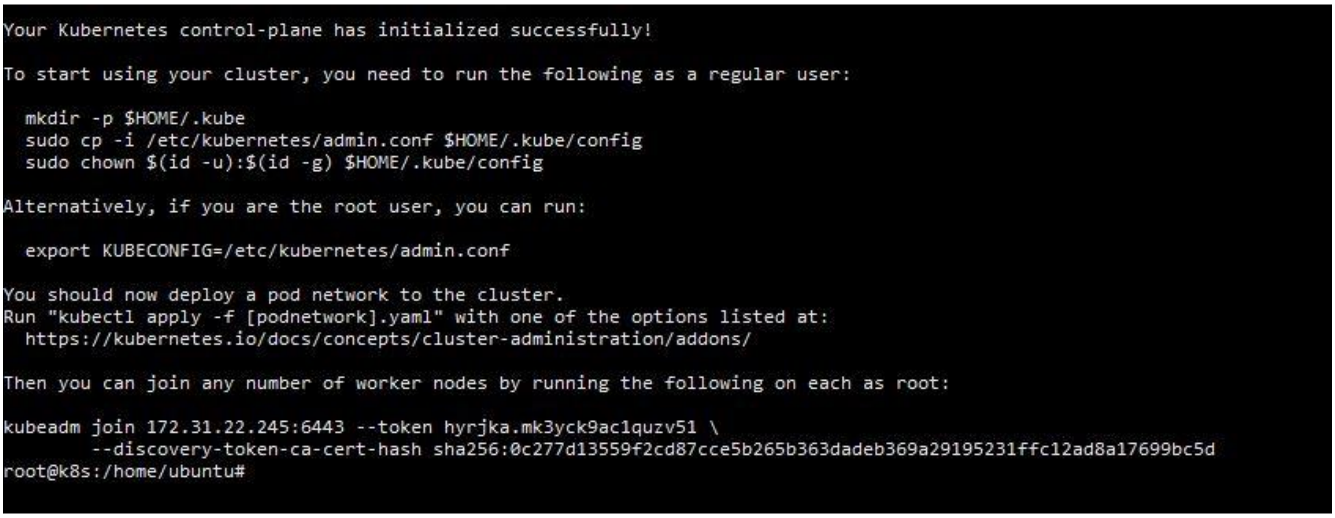
```
root@k8s:/home/ubuntu# kubeadm reset
W0914 17:15:53.543161   21289 preflight.go:56] [reset] WARNING: Changes made to this host by 'kubeadm init' or 'kubeadm join' will be reverted.
[reset] Are you sure you want to proceed? [y/N]: y
[preflight] Running pre-flight checks
W0914 17:15:59.167355   21289 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/super-admin.conf /etc/kubernetes/kubelet.conf /etc/kubernetes/bootstrap-kubelet.conf /etc/kubernetes
/controller-manager.conf /etc/kubernetes/scheduler.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.
root@k8s:/home/ubuntu#
```

6. Reinitialize Kubernetes Cluster.

Once you've verified and potentially adjusted the configuration, proceed with reinitializing the Kubernetes cluster.

kubeadm init

This command will initialize the cluster using the updated configuration.



```
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 172.31.22.245:6443 --token hyrjka.mk3yck9ac1quzv51 \
        --discovery-token-ca-cert-hash sha256:0c277d13559f2cd87cce5b265b363dadeb369a29195231ffc12ad8a17699bc5d
root@k8s:/home/ubuntu#
```

7. 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
cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
chown $(id -u):$(id -g) $HOME/.kube/config

Troubleshooting Finished!

Kubernetes Commands:

1. View Cluster Info

kubectl cluster-info

2. List All Nodes

kubectl get nodes

3. Describe a Node

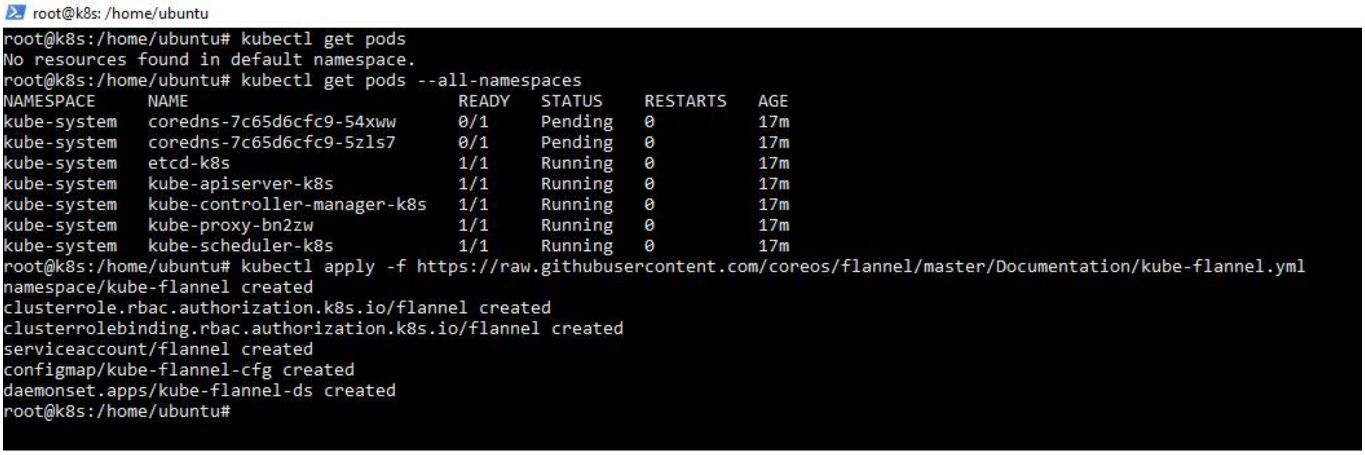
kubectl describe node <node-name>

e.g. kubectl describe node k8s

4. Get Pod Information

kubectl get pods - how to link pods with namespace (later)

No resources found in default namespace. So, we need to add resources.

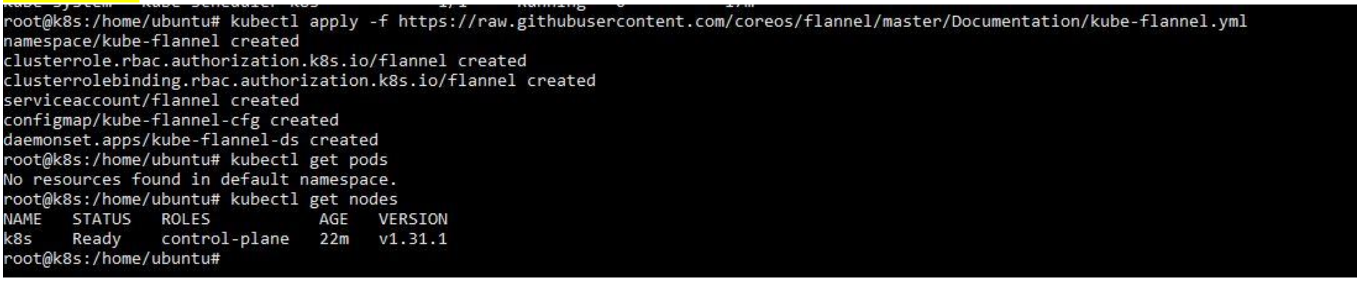


root@k8s: /home/ubuntu

```
root@k8s:/home/ubuntu# kubectl get pods
No resources found in default namespace.
root@k8s:/home/ubuntu# kubectl get pods --all-namespaces
NAMESPACE     NAME                              READY   STATUS    RESTARTS   AGE
kube-system   coredns-7c65d6cfc9-54xww          0/1     Pending   0          17m
kube-system   coredns-7c65d6cfc9-5zls7          0/1     Pending   0          17m
kube-system   etcd-k8s                          1/1     Running   0          17m
kube-system   kube-apiserver-k8s                1/1     Running   0          17m
kube-system   kube-controller-manager-k8s       1/1     Running   0          17m
kube-system   kube-proxy-bn2zw                  1/1     Running   0          17m
kube-system   kube-scheduler-k8s                1/1     Running   0          17m
root@k8s:/home/ubuntu# kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
namespace/kube-flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
serviceaccount/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
root@k8s:/home/ubuntu#
```

We add network using flannel inside kubernetes namespace –

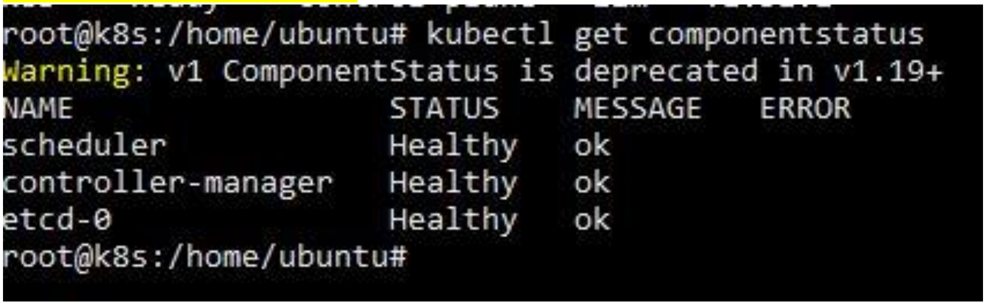
kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml



```
root@k8s:/home/ubuntu# kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
namespace/kube-flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
serviceaccount/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
root@k8s:/home/ubuntu# kubectl get pods
No resources found in default namespace.
root@k8s:/home/ubuntu# kubectl get nodes
NAME   STATUS   ROLES           AGE   VERSION
k8s    Ready    control-plane   22m   v1.31.1
root@k8s:/home/ubuntu#
```

kubectl get pods --all-namespaces

kubectl get componentstatus



```
root@k8s:/home/ubuntu# kubectl get componentstatus
Warning: v1 ComponentStatus is deprecated in v1.19+
NAME                 STATUS    MESSAGE   ERROR
scheduler            Healthy   ok
controller-manager   Healthy   ok
etcd-0               Healthy   ok
root@k8s:/home/ubuntu#
```

5. Get Pods in a Specific Namespace

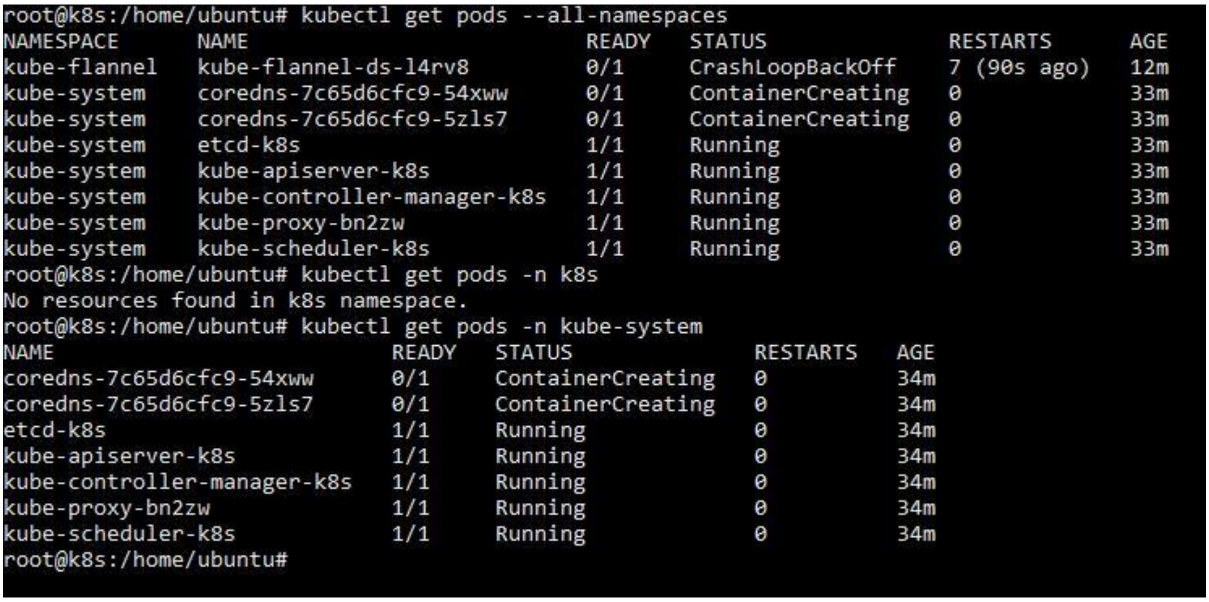
kubectl get pods -n <namespace>

E.g. kubectl get pods -n k8s or kubectl get pods -n kube-system

6. Describe a Pod

kubectl describe pod <pod-name>

kubectl describe pod kube-scheduler-k8s

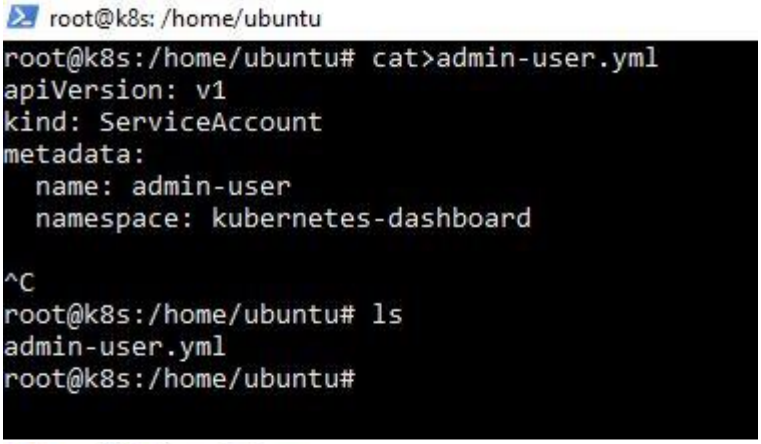


```
root@k8s:/home/ubuntu# kubectl get pods --all-namespaces
NAMESPACE      NAME                             READY   STATUS              RESTARTS      AGE
kube-flannel   kube-flannel-ds-l4rv8            0/1     CrashLoopBackOff    7 (90s ago)   12m
kube-system    coredns-7c65d6cfc9-54xww         0/1     ContainerCreating   0             33m
kube-system    coredns-7c65d6cfc9-5zls7         0/1     ContainerCreating   0             33m
kube-system    etcd-k8s                         1/1     Running             0             33m
kube-system    kube-apiserver-k8s               1/1     Running             0             33m
kube-system    kube-controller-manager-k8s      1/1     Running             0             33m
kube-system    kube-proxy-bn2zw                 1/1     Running             0             33m
kube-system    kube-scheduler-k8s               1/1     Running             0             33m
root@k8s:/home/ubuntu# kubectl get pods -n k8s
No resources found in k8s namespace.
root@k8s:/home/ubuntu# kubectl get pods -n kube-system
NAME                          READY   STATUS              RESTARTS   AGE
coredns-7c65d6cfc9-54xww      0/1     ContainerCreating   0          34m
coredns-7c65d6cfc9-5zls7      0/1     ContainerCreating   0          34m
etcd-k8s                      1/1     Running             0          34m
kube-apiserver-k8s            1/1     Running             0          34m
kube-controller-manager-k8s   1/1     Running             0          34m
kube-proxy-bn2zw              1/1     Running             0          34m
kube-scheduler-k8s            1/1     Running             0          34m
root@k8s:/home/ubuntu#
```

Error from server (NotFound): pods "kube-scheduler-k8s" not found

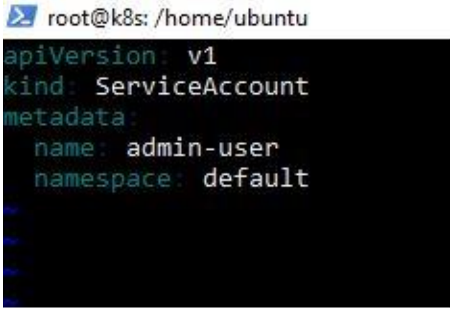
Create a Service Account

```
cat>admin-user.yml
apiVersion: v1
kind: ServiceAccount
metadata:
  name: admin-user
  namespace: default
```



root@k8s: /home/ubuntu

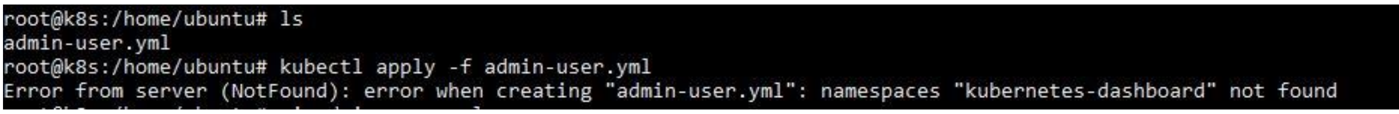
```
root@k8s:/home/ubuntu# cat>admin-user.yml
apiVersion: v1
kind: ServiceAccount
metadata:
  name: admin-user
  namespace: kubernetes-dashboard

^C
root@k8s:/home/ubuntu# ls
admin-user.yml
root@k8s:/home/ubuntu#
```



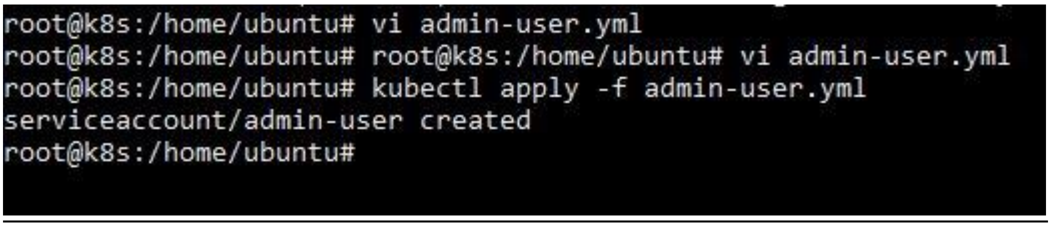
root@k8s: /home/ubuntu

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: admin-user
  namespace: default
~
~
~
~
```

kubectl apply -f admin-user.yml



```
root@k8s:/home/ubuntu# ls
admin-user.yml
root@k8s:/home/ubuntu# kubectl apply -f admin-user.yml
Error from server (NotFound): error when creating "admin-user.yml": namespaces "kubernetes-dashboard" not found
```



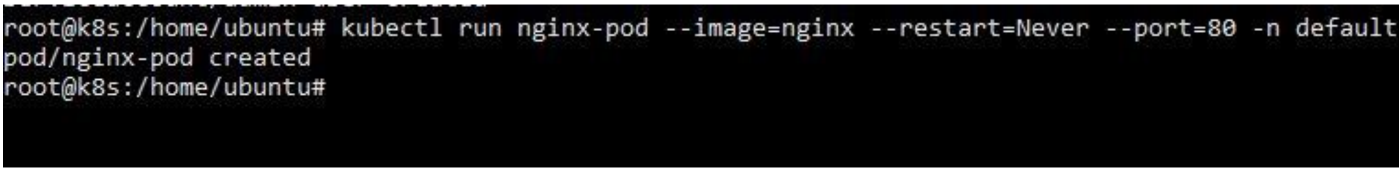
```
root@k8s:/home/ubuntu# vi admin-user.yml
root@k8s:/home/ubuntu# root@k8s:/home/ubuntu# vi admin-user.yml
root@k8s:/home/ubuntu# kubectl apply -f admin-user.yml
serviceaccount/admin-user created
root@k8s:/home/ubuntu#
```

Deploying Nginx on Kubernetes:

Goto : kubernetes nginx pod example

https://medium.com/@muppedaanvesh/deploying-nginx-on-kubernetes-a-quick-guide-04d533414967

kubectl run nginx-pod --image=nginx --restart=Never --port=80 -n default



```
root@k8s:/home/ubuntu# kubectl run nginx-pod --image=nginx --restart=Never --port=80 -n default
pod/nginx-pod created
root@k8s:/home/ubuntu#
```

Apply the YAML file using below command:

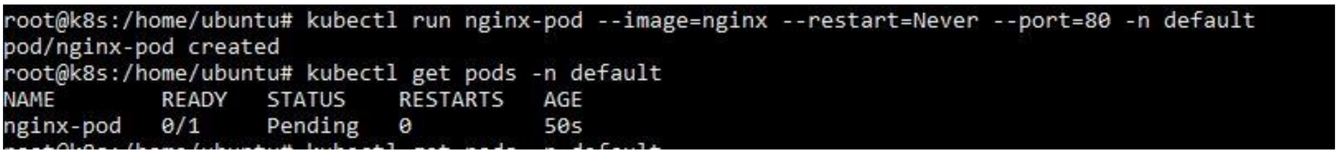
kubectl apply -f nginx-pod.yaml -n default

This command creates a nginx pod in default namespace.

Verify the Pod is Running using below command:

kubectl get pods

kubectl get pods -n default



```
root@k8s:/home/ubuntu# kubectl run nginx-pod --image=nginx --restart=Never --port=80 -n default
pod/nginx-pod created
root@k8s:/home/ubuntu# kubectl get pods -n default
NAME        READY   STATUS    RESTARTS   AGE
nginx-pod   0/1     Pending   0          50s
```

We need to remove taint from master node control-plane in order to run the pod.

Steps –

Remove Taints

kubectl describe node k8s | grep Taints

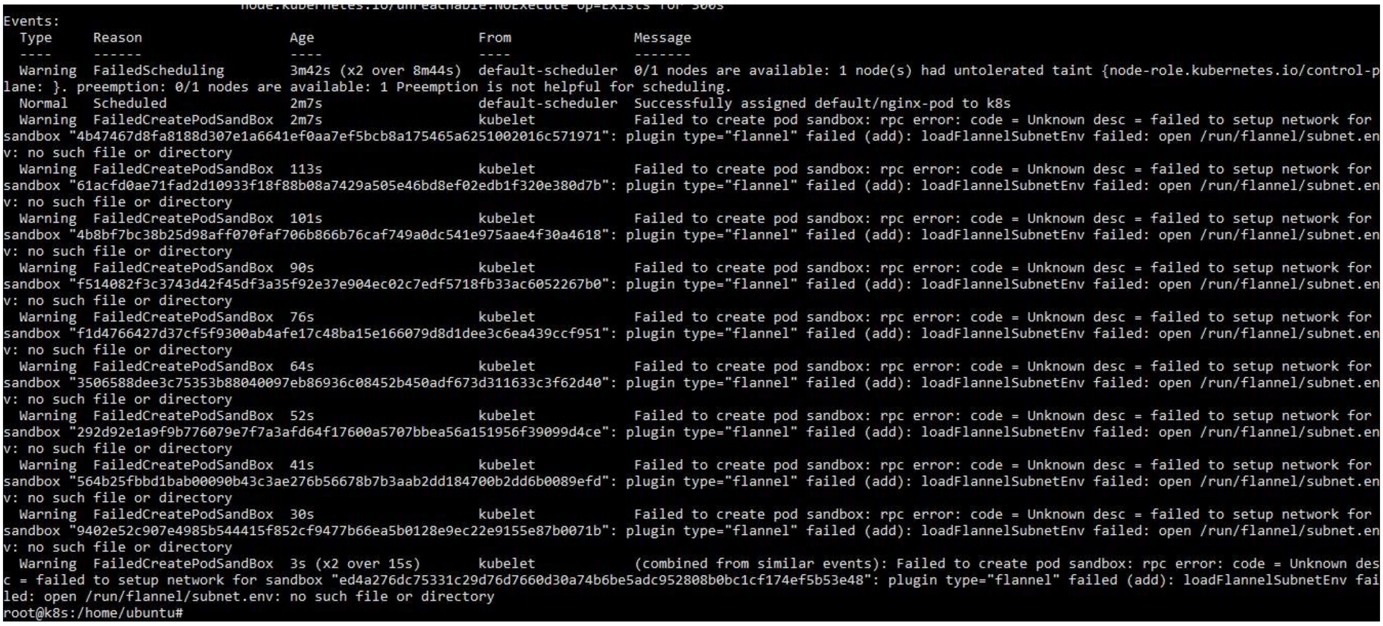
kubectl taint node k8s node-role.kubernetes.io/control-plane:NoSchedule-

kubectl get pods

7. View Pod Logs

kubectl logs <pod-name>

kubectl describe pod nginx-pod



```
                             node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type     Reason                  Age                    From               Message
  ----     ------                  ----                   ----               -------
  Warning  FailedScheduling        3m42s (x2 over 8m44s)  default-scheduler  0/1 nodes are available: 1 node(s) had untolerated taint {node-role.kubernetes.io/control-p
lane: }. preemption: 0/1 nodes are available: 1 Preemption is not helpful for scheduling.
  Normal   Scheduled               2m7s                   default-scheduler  Successfully assigned default/nginx-pod to k8s
  Warning  FailedCreatePodSandBox  2m7s                   kubelet            Failed to create pod sandbox: rpc error: code = Unknown desc = failed to setup network for
sandbox "4b47467d8fa8188d307e1a6641ef0aa7ef5bcb8a175465a6251002016c571971": plugin type="flannel" failed (add): loadFlannelSubnetEnv failed: open /run/flannel/subnet.en
v: no such file or directory
  Warning  FailedCreatePodSandBox  113s                   kubelet            Failed to create pod sandbox: rpc error: code = Unknown desc = failed to setup network for
sandbox "61acfd0ae71fad2d10933f18f88b08a7429a505e46bd8ef02edb1f320e380d7b": plugin type="flannel" failed (add): loadFlannelSubnetEnv failed: open /run/flannel/subnet.en
v: no such file or directory
  Warning  FailedCreatePodSandBox  101s                   kubelet            Failed to create pod sandbox: rpc error: code = Unknown desc = failed to setup network for
sandbox "4b8bf7bc38b25d98aff070faf706b866b76caf749a0dc541e975aae4f30a4618": plugin type="flannel" failed (add): loadFlannelSubnetEnv failed: open /run/flannel/subnet.en
v: no such file or directory
  Warning  FailedCreatePodSandBox  90s                    kubelet            Failed to create pod sandbox: rpc error: code = Unknown desc = failed to setup network for
sandbox "f514082f3c3743d42f45df3a35f92e37e904ec02c7edf5718fb33ac6052267b0": plugin type="flannel" failed (add): loadFlannelSubnetEnv failed: open /run/flannel/subnet.en
v: no such file or directory
  Warning  FailedCreatePodSandBox  76s                    kubelet            Failed to create pod sandbox: rpc error: code = Unknown desc = failed to setup network for
sandbox "f1d4766427d37cf5f9300ab4afe17c48ba15e166079d8d1dee3c6ea439ccf951": plugin type="flannel" failed (add): loadFlannelSubnetEnv failed: open /run/flannel/subnet.en
v: no such file or directory
  Warning  FailedCreatePodSandBox  64s                    kubelet            Failed to create pod sandbox: rpc error: code = Unknown desc = failed to setup network for
sandbox "3506588dee3c75353b88040097eb86936c08452b450adf673d311633c3f62d40": plugin type="flannel" failed (add): loadFlannelSubnetEnv failed: open /run/flannel/subnet.en
v: no such file or directory
  Warning  FailedCreatePodSandBox  52s                    kubelet            Failed to create pod sandbox: rpc error: code = Unknown desc = failed to setup network for
sandbox "292d92e1a9f9b776079e7f7a3afd64f17600a5707bbea56a151956f39099d4ce": plugin type="flannel" failed (add): loadFlannelSubnetEnv failed: open /run/flannel/subnet.en
v: no such file or directory
  Warning  FailedCreatePodSandBox  41s                    kubelet            Failed to create pod sandbox: rpc error: code = Unknown desc = failed to setup network for
sandbox "564b25fbbd1bab00090b43c3ae276b56678b7b3aab2dd184700b2dd6b0089efd": plugin type="flannel" failed (add): loadFlannelSubnetEnv failed: open /run/flannel/subnet.en
v: no such file or directory
  Warning  FailedCreatePodSandBox  30s                    kubelet            Failed to create pod sandbox: rpc error: code = Unknown desc = failed to setup network for
sandbox "9402e52c907e4985b544415f852cf9477b66ea5b0128e9ec22e9155e87b0071b": plugin type="flannel" failed (add): loadFlannelSubnetEnv failed: open /run/flannel/subnet.en
v: no such file or directory
  Warning  FailedCreatePodSandBox  3s (x2 over 15s)       kubelet            (combined from similar events): Failed to create pod sandbox: rpc error: code = Unknown des
c = failed to setup network for sandbox "ed4a276dc75331c29d76d7660d30a74b6be5adc952808b0bc1cf174ef5b53e48": plugin type="flannel" failed (add): loadFlannelSubnetEnv fai
led: open /run/flannel/subnet.env: no such file or directory
root@k8s:/home/ubuntu#
```

8. Execute Command in a Pod

kubectl exec -it <pod-name> -- /bin/

kubectl exec -it nginx-pod -- /bin/

9. Delete Resources from a YAML File

kubectl delete -f <file.yaml>

kubectl delete -f admin-user.yml

11. Scale a Deployment

kubectl scale deployment <deployment-name> --replicas=<number-of-replicas>

kubectl scale deployment tempdeploy --replicas=3

error: no objects passed to scale

12. Get Deployment Information

kubectl get deployments

13. Describe a Deployment

kubectl describe deployment <deployment-name>

14. Get Services

kubectl get services

15. Describe a Service

kubectl describe service <service-name>

kubectl describe service kubernetes

16. Get Namespaces

```
kubectl get namespaces
```

17. Create a Namespace

```
kubectl create namespace <namespace-name>
kubectl create namespace abcd
```

18. Delete a Namespace

```
kubectl delete namespace <namespace-name>
kubectl delete namespace abcd
```

19. Port Forwarding

```
kubectl port-forward <pod-name> <local-port>:<pod-port>
```

20. Check Cluster Health

```
kubectl get componentstatuses
```

Need to Check:

Creating a ClusterRoleBinding

```
cat>ClusterRoleBinding.yml
```

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: admin-user
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: admin-user
  namespace: kubernetes-dashboard
```

Check Status

```
kubectl get all -n kubernetes-dashboard
kubectl get pods --all-namespaces
```

Generate Token

```
kubectl -n kubernetes-dashboard create token admin-user
```