

Setup master and worker node VM's

Setup VPS and install docker (all nodes)

Refer: <https://projects-srv2.kondgekar.com/projects/cbofferwall/wiki/0000-preliminary-setup>

Fix swap (all nodes)

Installing kubernetes needs swap should be disabled. Check if swap is enabled and then disable if swap is enabled.

Disable swap

```
sudo swapoff -a
```

Remove / comment out respective swap entry from fstab file

```
sudo nano /etc/fstab
```

```
GNU nano 4.8 /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/sda2 during installation
UUID=f1b6db43-01e1-4851-a53e-b0c4eee7d6a7 / ext4 errors=remount-ro 0 0
# /boot was on /dev/sda1 during installation
UUID=12f6445c-9ef7-440f-bccc-600023baf92b /boot ext4 defaults noatime 0 0
#/swapfile none swap sw 0 0
```

Install kubeadm, Kubelet And Kubectl (all nodes)

Refer: <https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/>

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

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

```
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
```

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

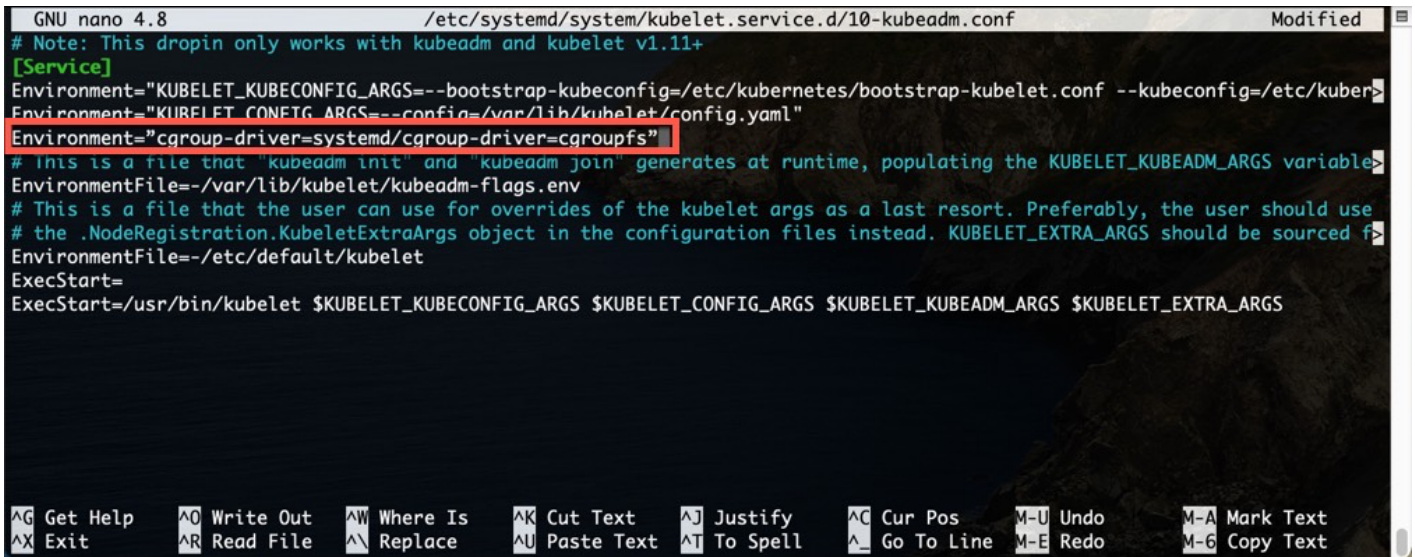
Update Kubernetes Configuration

(all nodes)

```
sudo nano /etc/systemd/system/kubelet.service.d/10-kubeadm.conf
```

This will open a text editor, enter the following line after the last “Environment Variable”:

```
Environment="cgroup-driver=systemd/cgroup-driver=cgroupfs"
```



```
GNU nano 4.8 /etc/systemd/system/kubelet.service.d/10-kubeadm.conf Modified
# Note: This dropin only works with kubeadm and kubelet v1.11+
[Service]
Environment="KUBELET_KUBECONFIG_ARGS=--bootstrap-kubeconfig=/etc/kubernetes/bootstrap-kubelet.conf --kubeconfig=/etc/kubernetes/kubelet.conf"
Environment="KUBELET_CONFIG_ARGS=--config=/var/lib/kubelet/config.yaml"
Environment="cgroup-driver=systemd/cgroup-driver=cgroupfs"
# This is a file that "kubeadm init" and "kubeadm join" generates at runtime, populating the KUBELET_KUBEADM_ARGS variable
EnvironmentFile=/var/lib/kubelet/kubeadm-flags.env
# This is a file that the user can use for overrides of the kubelet args as a last resort. Preferably, the user should use
# the .NodeRegistration.KubeletExtraArgs object in the configuration files instead. KUBELET_EXTRA_ARGS should be sourced from this file.
EnvironmentFile=/etc/default/kubelet
ExecStart=
ExecStart=/usr/bin/kubelet $KUBELET_KUBECONFIG_ARGS $KUBELET_CONFIG_ARGS $KUBELET_KUBEADM_ARGS $KUBELET_EXTRA_ARGS

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos M-U Undo M-A Mark Text
^X Exit ^R Read File ^_ Replace ^U Paste Text ^T To Spell ^_ Go To Line M-E Redo M-6 Copy Text
```

Start kubernetes cluster (on master)

We are going to use [Flannel](#) as a networking for pods

```
sudo kubeadm init --apiserver-advertise-address=<ip-address-of-kmaster-vm> --pod-network-cidr=10.244.0.0/16
```

1. You will get the below output. The commands marked as (1), execute them as a non-root user. This will enable you to use kubectl from the CLI
2. The command marked as (2) should also be saved for future. This will be used to join nodes to your cluster

```
[addons] Applied essential addon: kube-proxy

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 75.119.139.251:6443 --token techiterian@vmi593089 \
--discovery-token-ca-cert-hash sha256:8a71416790924717102701502b307093
```

Execute commands as mentioned above.

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

Verify if cluster is running (on master)

```
kubectl get pods -o wide --all-namespaces
```

```
techiterian@vmi593089:~$ kubectl get pods -o wide --all-namespaces
NAMESPACE   NAME                                     READY   STATUS    RESTARTS   AGE   IP              NODE                                     READINESS GATES
kube-system  coredns-558bd4d5db-6mb2j               0/1     Pending   0           8m41s  <none>          <none>                                <none>
kube-system  coredns-558bd4d5db-kbx4n               0/1     Pending   0           8m41s  <none>          <none>                                <none>
kube-system  etcd-vmi593089.contaboserver.net       1/1     Running   0           8m55s  75.119.139.251 vmi593089.contaboserver.net          <none>
kube-system  kube-apiserver-vmi593089.contaboserver.net 1/1     Running   0           8m46s  75.119.139.251 vmi593089.contaboserver.net          <none>
kube-system  kube-controller-manager-vmi593089.contaboserver.net 1/1     Running   0           8m46s  75.119.139.251 vmi593089.contaboserver.net          <none>
kube-system  kube-proxy-f7kz9                       1/1     Running   0           8m40s  75.119.139.251 vmi593089.contaboserver.net          <none>
kube-system  kube-scheduler-vmi593089.contaboserver.net 1/1     Running   0           8m46s  75.119.139.251 vmi593089.contaboserver.net          <none>
```

Notice that all pods are running except coredns. It will be running once we setup pod network in the next step

Install POD network (Flannel) (on master)

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

```
techiterian@vmi593089:~$ kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
Warning: policy/v1beta1 PodSecurityPolicy is deprecated in v1.21+, unavailable in v1.25+
podsecuritypolicy.policy/psp.flannel.unprivileged 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
techiterian@vmi593089:~$
```

```
kubectl get pods -o wide --all-namespaces
```

```
techiterian@vmi593089:~$ kubectl get pods --all-namespaces -o wide
NAMESPACE   NAME                                     READY   STATUS    RESTARTS   AGE   IP              NODE                                     NOMINATED NODE   READINESS GATES
kube-system  coredns-558bd4d5db-6mb2j              1/1     Running   0          20m   10.244.0.2      vmi593089.contaboserver.net           <none>           <none>
kube-system  coredns-558bd4d5db-kbx4n              1/1     Running   0          20m   10.244.0.3      vmi593089.contaboserver.net           <none>           <none>
kube-system  etcd-vmi593089.contaboserver.net       1/1     Running   0          20m   75.119.139.251  vmi593089.contaboserver.net           <none>           <none>
kube-system  kube-apiserver-vmi593089.contaboserver.net 1/1     Running   0          20m   75.119.139.251  vmi593089.contaboserver.net           <none>           <none>
kube-system  kube-controller-manager-vmi593089.contaboserver.net 1/1     Running   0          20m   75.119.139.251  vmi593089.contaboserver.net           <none>           <none>
kube-system  kube-flannel-ds-5wvrr                 1/1     Running   0          108s   75.119.139.251  vmi593089.contaboserver.net           <none>           <none>
kube-system  kube-proxy-f7kz9                      1/1     Running   0          20m   75.119.139.251  vmi593089.contaboserver.net           <none>           <none>
kube-system  kube-scheduler-vmi593089.contaboserver.net 1/1     Running   0          20m   75.119.139.251  vmi593089.contaboserver.net           <none>           <none>
```

Notice that all pods are now running

Install Kubernetes Dashboard (on master)

Refer: <https://kubernetes.io/docs/tasks/access-application-cluster/web-ui-dashboard/>

Check available nodes (on master)

```
kubectl get nodes
```

```
Last login: Wed May 19 13:30:41 2021 from 49.33.209.127
techiterian@vmi593089:~$ kubectl get nodes
NAME                                STATUS    ROLES    AGE   VERSION
vmi593089.contaboserver.net        Ready    control-plane,master   29m   v1.21.1
techiterian@vmi593089:~$
```

Notice that only one node is available which is a master node.

Add worker node (on worker node)

Use below command to join Kubernetes cluster from worker node

```
kubeadm join <master-node-ip-address>:6443 --token <generated-token> \
--discovery-token-ca-cert-hash <generated-hash>
```

```
techiterian@vmi593088:~$ sudo kubeadm join 75.119.139.251:6443 --token <generated-token> \
--discovery-token-ca-cert-hash <generated-hash>
[sudo] password for techiterian:
[preflight] Running pre-flight checks
[WARNING IsDockerSystemdCheck]: detected "cgroupfs" as the Docker cgroup driver. The recommended driver is
"systemd". Please follow the guide at https://kubernetes.io/docs/setup/cri/
[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 apiservert 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.

techiterian@vmi593088:~$
```

Check if node is added in cluster (on master)

Initially worker node will show status as not ready

```
techiterian@vmi593089:~$ kubectl get nodes
NAME                                STATUS    ROLES    AGE   VERSION
vmi593088.contaboserver.net        NotReady <none>   10s   v1.21.1
vmi593089.contaboserver.net        Ready    control-plane,master 40m   v1.21.1
techiterian@vmi593089:~$
```

Wait for some time and it will be shown as Ready

```
techiterian@vmi593089:~$ kubectl get nodes
NAME                                STATUS    ROLES    AGE   VERSION
vmi593088.contaboserver.net        Ready    <none>   3m25s v1.21.1
vmi593089.contaboserver.net        Ready    control-plane,master 43m   v1.21.1
techiterian@vmi593089:~$
```

“ Kubernetes cluster is now running. You can now run containerized applications and make it available over web using specific setup

Revision #16

Created Wed, May 19, 2021 10:36 AM by [Sudarshan Kondgekar](#)

Updated Wed, May 19, 2021 1:57 PM by [Sudarshan Kondgekar](#)