Pre-Requisites

https://lcastrose.signin.aws.amazon.com/console

Create the Environment in AWS with the Ubuntu AMI (Ubuntu Server 18.04 LTS (HVM), SSD Volume Type) using 3 instances of T2.Medium:

- 1 Master & 2 Worker Nodes

Create all instances in a new subnet (according Lab Files), then you need to add the following components:

- Internet Gateway
- Default Route
- Security Group allowing all incoming traffic from your subnet

Step 1 - Preparation

Change the hostname for every instance accordingly

sudo hostnamectl set-hostname kubernetes-master

sudo hostnamectl set-hostname worker-1

sudo hostnamectl set-hostname worker-2

```
root@ip-10-1-0-85:/home/ubuntu# hostnamectl set-hostname worker-2
root@ip-10-1-0-85:/home/ubuntu# vi /etc/hosts
```

Perform this task in all three instances

sudo vi /etc/hosts

<IP_Add_K8s_Master> kubernetes-master

<IP_Add_Woker-1> worker-1

<IP_Add_Woker-2> worker-2

```
127.0.0.1 localhost

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts

10.1.0.12 worker-1
10.1.0.85 worker-2
10.1.0.109 kubernetes-master
```

Step 2 - Docker Install

sudo apt update

```
root@ip-10-1-0-85:/home/ubuntu# sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:4 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/universe amd64 Packages [8570 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/universe Translation-en [4941 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/multiverse amd64 Packages [151 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/multiverse Translation-en [108 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [914 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main Translation-en [314 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/restricted amd64 Packages [43.9 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/restricted Translation-en [11.0 kB]
```

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

```
root@ip-10-1-0-85:/home/ubuntu# sudo apt install apt-transport-https ca-certificates curl software-properties-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
ca-certificates is already the newest version (20180409).
ca-certificates set to manually installed.
curl is already the newest version (7.58.0-2ubuntu3.8).
curl set to manually installed.
software-properties-common is already the newest version (0.96.24.32.12).
software-properties-common set to manually installed.
The following NEW packages will be installed:
  apt-transport-https
0 upgraded, 1 newly installed, 0 to remove and 73 not upgraded.
Need to get 1692 B of archives.
After this operation, 153 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 apt-transport-https all 1.6.12 [1692 B]
Fetched 1692 B in 0s (0 B/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 56554 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_1.6.12_all.deb ...
Unpacking apt-transport-https (1.6.12) ...
Setting up apt-transport-https (1.6.12) ...
```

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

```
root@ip-10-1-0-85:/home/ubuntu# curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add
OK
```

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

```
root@ip-10-1-0-85:/home/ubuntu# sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable"
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease
Get:4 https://download.docker.com/linux/ubuntu bionic InRelease [64.4 kB]
Hit:5 http://security.ubuntu.com/ubuntu bionic-security InRelease
Get:6 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages [11.0 kB]
Fetched 75.5 kB in 0s (175 kB/s)
Reading package lists... Done
```

sudo apt update

```
root@ip-10-1-0-85:/home/ubuntu# sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease
Hit:4 https://download.docker.com/linux/ubuntu bionic InRelease
Hit:5 http://security.ubuntu.com/ubuntu bionic-security InRelease
Reading package lists... Done
```

apt-cache policy docker-ce

```
root@ip-10-1-0-85:/home/ubuntu# apt-cache policy docker-ce
docker-ce:
  Installed: (none)
  Candidate: 5:19.03.8~3-0~ubuntu-bionic
  Version table:
     5:19.03.8~3-0~ubuntu-bionic 500
        500 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
     5:19.03.7~3-0~ubuntu-bionic 500
        500 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
     5:19.03.6~3-0~ubuntu-bionic 500
        500 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
     5:19.03.5~3-0~ubuntu-bionic 500
        500 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
     5:19.03.4~3-0~ubuntu-bionic 500
        500 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
     5:19.03.3~3-0~ubuntu-bionic 500
        500 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
     5:19.03.2~3-0~ubuntu-bionic 500
        500 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
     5:19.03.1~3-0~ubuntu-bionic 500
        500 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
     5:19.03.0~3-0~ubuntu-bionic 500
        500 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
     5:18.09.9~3-0~ubuntu-bionic 500
        500 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
```

sudo apt install docker-ce

```
root@ip-10-1-0-85:/home/ubuntu# sudo apt install docker-ce
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  aufs-tools cgroupfs-mount containerd.io docker-ce-cli libltdl7 pigz
The following NEW packages will be installed:
  aufs-tools cgroupfs-mount containerd.io docker-ce docker-ce-cli libltdl7
  pigz
0 upgraded, 7 newly installed, 0 to remove and 73 not upgraded.
Need to get 85.8 MB of archives.
After this operation, 385 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/universe amd64 pigz amd64 2.4-1 [57.4 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/universe amd64 aufs-tools amd64 1:4.9+20170918-1ubuntu1 [104 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/universe amd64 cgroupfs-mount all 1.4 [6320 B]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/main amd64 libltdl7 amd64 2.4.6-2 [38.8 kB]
Get:5 https://download.docker.com/linux/ubuntu bionic/stable amd64 containerd.io amd64 1.2.13-1 [20.1 MB]
Get:6 https://download.docker.com/linux/ubuntu bionic/stable amd64 docker-ce-cli amd64 5:19.03.8~3-0~ubuntu-bionic [42.6 MB]
Get:7 https://download.docker.com/linux/ubuntu bionic/stable amd64 docker-ce amd64 5:19.03.8~3-0~ubuntu-bionic [22.9 MB]
Fetched 85.8 MB in 2s (40.8 MB/s)
Selecting previously unselected package pigz.
(Reading database ... 56558 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.4-1_amd64.deb ...
Unpacking pigz (2.4-1) ...
Selecting previously unselected package aufs-tools.
```

sudo service docker status

```
root@ip-10-1-0-85:/home/ubuntu# sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2020-04-10 21:18:34 UTC; 30s ago
     Docs: https://docs.docker.com
 Main PID: 4082 (dockerd)
    Tasks: 10
   CGroup: /system.slice/docker.service
           └─4082 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Apr 10 21:18:33 worker-2 dockerd[4082]: time="2020-04-10T21:18:33.644794507Z" level=warning msg="Your kernel does not su
Apr 10 21:18:33 worker-2 dockerd[4082]: time="2020-04-10T21:18:33.644833919Z" level=warning msg="Your kernel does not su
Apr 10 21:18:33 worker-2 dockerd[4082]: time="2020-04-10T21:18:33.644843454Z" level=warning msg="Your kernel does not su
Apr 10 21:18:33 worker-2 dockerd[4082]: time="2020-04-10T21:18:33.644991417Z" level=info msg="Loading containers: start.
Apr 10 21:18:33 worker-2 dockerd[4082]: time="2020-04-10T21:18:33.940903879Z" level=info msg="Default bridge (docker0) i
Apr 10 21:18:34 worker-2 dockerd[4082]: time="2020-04-10T21:18:34.132634750Z" level=info msg="Loading containers: done."
Apr 10 21:18:34 worker-2 dockerd[4082]: time="2020-04-10T21:18:34.186882521Z" level=info msg="Docker daemon" commit=afac
Apr 10 21:18:34 worker-2 dockerd[4082]: time="2020-04-10T21:18:34.187001232Z" level=info msg="Daemon has completed initi
Apr 10 21:18:34 worker-2 dockerd[4082]: time="2020-04-10T21:18:34.251250173Z" level=info msg="API listen on /var/run/doc
Apr 10 21:18:34 worker-2 systemd[1]: Started Docker Application Container Engine.
```

Step 3 - Kubernetes Install

curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add

```
root@ip-10-1-0-85:/home/ubuntu# curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add
OK
```

sudo apt-add-repository "deb http://apt.kubernetes.io/ kubernetes-xenial main"

```
root@ip-10-1-0-85:/home/ubuntu# sudo apt-add-repository "deb http://apt.kubernetes.io/ kubernetes-xenial main"
Hit:1 https://download.docker.com/linux/ubuntu bionic InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease
Hit:6 http://security.ubuntu.com/ubuntu bionic-security InRelease
Get:5 https://packages.cloud.google.com/apt kubernetes-xenial InRelease [8993 B]
Get:7 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 Packages [34.8 kB]
Fetched 43.8 kB in 1s (75.1 kB/s)
Reading package lists... Done
```

sudo swapoff -a

```
root@ip-10-1-0-85:/home/ubuntu# sudo swapoff -a
```

Step 4 - Kubeadm Install

sudo apt-get install kubeadm -y

```
root@ip-10-1-0-85:/home/ubuntu# sudo apt-get install kubeadm -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  conntrack cri-tools kubectl kubelet kubernetes-cni socat
The following NEW packages will be installed:
  conntrack cri-tools kubeadm kubectl kubelet kubernetes-cni socat
0 upgraded, 7 newly installed, 0 to remove and 73 not upgraded.
Need to get 52.0 MB of archives.
After this operation, 275 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/main amd64 conntrack amd64 1:1.4.4+snapshot20161117-6ubuntu2 [30.6 kB]
```

kubeadm version

```
root@kubernetes-master:/home/ubuntu# kubeadm version
kubeadm version: &version.Info{Major:"1", Minor:"18", GitVersion:"v1.18.1", GitCommit:"7879fc12a63337efff607952a323df90cdc7a335",
```

Step 5 - Initialize Cluster in Master ONLY

kubeadm init —pod-network-cidr=10.244.0.0/16

```
root@kubernetes-master:/home/ubuntu# sudo kubeadm init --pod-network-cidr=10.244.0.0/16
W0410 20:44:33.659228    6578 configset.go:202] WARNING: kubeadm cannot validate component configs for API groups [kubelet.config.k8s.io kubeproxy.config.k8s.io]
[init] Using Kubernetes version: v1.18.1
[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] 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'
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Starting the kubelet
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [kubernetes-master kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster.local] and IPs [10.96.0.1 10.1.0.109]
[certs] Generating "apiserver-kubelet-client" certificate and key
[certs] Generating "front-proxy-ca" certificate and key
[certs] Generating "front-proxy-client" certificate and key
[certs] Generating "etcd/ca" certificate and key
[certs] Generating "etcd/server" certificate and key
[certs] etcd/server serving cert is signed for DNS names [kubernetes-master localhost] and IPs [10.1.0.109 127.0.0.1 ::1]
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [kubernetes-master localhost] and IPs [10.1.0.109 127.0.0.1 ::1]
[certs] Generating "etcd/healthcheck-client" certificate and key
[certs] Generating "apiserver-etcd-client" certificate and key
```

Save the output command to join worker nodes, as the follow example:

```
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

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 10.1.0.109:6443 --token 0pkqzf.qdozh5j8yae2locm \
    --discovery-token-ca-cert-hash sha256:0d22dfcf6325143a034ce80ccb3b57b9c92a3c49e27f56b3702be3e77befeb7c
```

kubeadm join 10.1.0.109:6443 --token bk4qq9.x3irstlsbkn2o5i4 --discovery-token-ca-cert-hash sha256:0d22dfcf6325143a034ce80ccb3b57b9c92a3c49e27f56b3702be3e77befeb7c

mkdir -p $HOME/.kube

sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config

sudo chown $(id -u):$(id -g) $HOME/.kube/config

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

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

```
root@kubernetes-master:/home/ubuntu# sudo kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
podsecuritypolicy.policy/psp.flannel.unprivileged created
clusterrole.rbac.authorization.k8s.io/flannel configured
clusterrolebinding.rbac.authorization.k8s.io/flannel unchanged
serviceaccount/flannel unchanged
configmap/kube-flannel-cfg configured
daemonset.apps/kube-flannel-ds-amd64 created
daemonset.apps/kube-flannel-ds-arm64 created
daemonset.apps/kube-flannel-ds-arm created
daemonset.apps/kube-flannel-ds-ppc64le created
daemonset.apps/kube-flannel-ds-s390x created
```

kubectl get pods --all-namespaces

```
root@kubernetes-master:/home/ubuntu# kubectl get pods --all-namespaces
NAMESPACE     NAME                                       READY   STATUS    RESTARTS   AGE
kube-system   coredns-66bff467f8-9lwlb                   1/1     Running   0          2m55s
kube-system   coredns-66bff467f8-mfmfn                   0/1     Running   0          2m55s
kube-system   etcd-kubernetes-master                     1/1     Running   0          3m11s
kube-system   kube-apiserver-kubernetes-master           1/1     Running   0          3m11s
kube-system   kube-controller-manager-kubernetes-master  1/1     Running   0          3m11s
kube-system   kube-flannel-ds-amd64-rwcxl                1/1     Running   0          20s
kube-system   kube-proxy-dsxkl                           1/1     Running   0          2m55s
kube-system   kube-scheduler-kubernetes-master           1/1     Running   0          3m11s
```

kubectl get nodes

```
root@kubernetes-master:/home/ubuntu# kubectl get nodes
NAME                STATUS   ROLES    AGE     VERSION
kubernetes-master   Ready    master   3m13s   v1.18.1
```

Step 6 - Add the workers nodes to the cluster

kubeadm join 10.1.0.109:6443 --token bk4qq9.x3irstlsbkn2o5i4 --discovery-token-ca-cert-hash sha256:0d22dfcf6325143a034ce80ccb3b57b9c92a3c49e27f56b3702be3e77befeb7c

```
root@ip-10-1-0-85:/home/ubuntu# kubeadm join 10.1.0.109:6443 --token bk4qq9.x3irstlsbkn2o5i4     --discovery-token-ca-cert-hash sh
a256:0d22dfcf6325143a034ce80ccb3b57b9c92a3c49e27f56b3702be3e77befeb7c
W0410 21:19:56.091403   17168 join.go:346] [preflight] WARNING: JoinControlPane.controlPlane settings will be ignored when control
-plane flag is not set.
[preflight] Running pre-flight checks
        [WARNING IsDockerSystemdCheck]: detected "cgroupfs" as the Docker cgroup driver. The recommended driver is "systemd". Plea
se 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 -oyaml'
[kubelet-start] Downloading configuration for the kubelet from the "kubelet-config-1.18" ConfigMap in the kube-system namespace
[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.
```

Check nodes have been added (Run this command on the master node)

```
ubuntu@kubernetes-master:~$ sudo su
root@kubernetes-master:/home/ubuntu# kubectl get nodes
NAME                STATUS   ROLES    AGE   VERSION
ip-10-1-0-12        Ready    <none>   39m   v1.18.1
kubernetes-master   Ready    master   66m   v1.18.1
worker-2            Ready    <none>   31m   v1.18.1
root@kubernetes-master:/home/ubuntu#
```

Step 7 - Test your cluster

kubectl create deployment nginx-deployment —image=nginx

kubectl get deployments

kubectl get pods

```
root@kubernetes-master:/home/ubuntu# kubectl create deployment nginx-deployment --image=nginx
deployment.apps/nginx-deployment created
root@kubernetes-master:/home/ubuntu# kubectl get deployments
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx              1/1     1            1           2m21s
nginx-deployment   1/1     1            1           9s
root@kubernetes-master:/home/ubuntu# kubectl get pods
NAME                               READY   STATUS    RESTARTS   AGE
nginx-deployment-5969c7f455-89zrk  1/1     Running   0          15s
nginx-f89759699-pj65j              1/1     Running   0          2m27s
root@kubernetes-master:/home/ubuntu#
```

Step 8 - Clear the environment

Kubectl delete deployments nginx-deployment