Installing prerequisites packages

- Run the **docker version** command to validate Docker.

  docker version

```
root@ip-172-31-86-69:~# docker version
Client:
 Version:           18.09.7
 API version:       1.39
 Go version:        go1.10.1
 Git commit:        2d0083d
 Built:             Wed Jul  3 12:13:59 2019
 OS/Arch:           linux/amd64
 Experimental:      false

Server:
 Engine:
  Version:          18.09.7
  API version:      1.39 (minimum version 1.12)
  Go version:       go1.10.1
  Git commit:       2d0083d
  Built:            Mon Jul  1 19:31:12 2019
  OS/Arch:          linux/amd64
  Experimental:     false
root@ip-172-31-86-69:~#
```

Configuring Kubernetes

- Configure Kubernetes using the procedure below.

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

echo "deb http://apt.kubernetes.io/ kubernetes-xenial main" >/etc/apt/sources.list.d/kubernetes.list

apt-get update

apt-get install -y kubelet kubeadm kubectl

```
root@ip-172-31-86-69:~# curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | apt-key add -
OK
root@ip-172-31-86-69:~# echo "deb http://apt.kubernetes.io/ kubernetes-xenial main" >/etc/apt/sources.list.d/kubernetes.list
root@ip-172-31-86-69:~# apt-get 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 http://security.ubuntu.com/ubuntu bionic-security InRelease
Get:5 https://packages.cloud.google.com/apt kubernetes-xenial InRelease [8993 B]
Get:6 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 Packages [27.5 kB]
Fetched 36.5 kB in 1s (64.9 kB/s)
Reading package lists... Done
root@ip-172-31-86-69:~# apt-get install -y kubelet kubeadm kubectl
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  conntrack cri-tools 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 2 not upgraded.
Need to get 52.9 MB of archives.
After this operation, 280 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]
```

- Initialize Kubernetes to deploy containers using Kubernetes CLI.

  kubeadm init

```
root@ip-172-31-86-69:~# kubeadm init
[init] Using Kubernetes version: v1.15.0
[preflight] Running pre-flight checks
        [WARNING Service-Docker]: docker service is not enabled, please run 'systemctl enable docker.service'
        [WARNING IsDockerSystemdCheck]: detected "cgroupfs" as the Docker cgroup driver. The recommended drive
tes.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"
```

- Once Kubernetes is initialized, configure Kubernetes to start using the Kubernetes cluster.

mkdir -p $HOME/.kube

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

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

```
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 172.31.86.69:6443 --token 7jp400.ldgq81o8qzqwdrwa \
    --discovery-token-ca-cert-hash sha256:50515e1fd7c9454ab794ba72f8d4f5ad30433b3be83126e868817e0114198e9d
root@ip-172-31-86-69:~#
```

- After the cluster gets started, deploy a weave network to the cluster.

export kubever=$(kubectl version | base64 | tr -d '\n')

kubectl apply -f "https://cloud.weave.works/k8s/net?k8s-version=$kubever"

```
root@ip-172-31-86-69:~# export kubever=$(kubectl version | base64 | tr -d '\n')
root@ip-172-31-86-69:~# kubectl apply -f "https://cloud.weave.works/k8s/net?k8s-version=$kubever"
serviceaccount/weave-net created
clusterrole.rbac.authorization.k8s.io/weave-net created
clusterrolebinding.rbac.authorization.k8s.io/weave-net created
role.rbac.authorization.k8s.io/weave-net created
rolebinding.rbac.authorization.k8s.io/weave-net created
daemonset.extensions/weave-net created
root@ip-172-31-86-69:~# kubectl get node
NAME              STATUS     ROLES    AGE   VERSION
ip-172-31-86-69   NotReady   master   12m   v1.15.0
root@ip-172-31-86-69:~# kubectl get node
NAME              STATUS   ROLES    AGE   VERSION
ip-172-31-86-69   Ready    master   12m   v1.15.0
root@ip-172-31-86-69:~#
```

- With weave network deployment, validate that the node is up and running. That will help to deploy a Docker container to the Kubernetes cluster.

kubectl get node

kubectl get pods --all-namespaces

```
root@ip-172-31-86-69:~# kubectl get node
NAME              STATUS   ROLES    AGE   VERSION
ip-172-31-86-69   Ready    master   15m   v1.15.0
root@ip-172-31-86-69:~# kubectl get pods --all-namespaces
NAMESPACE     NAME                                      READY   STATUS    RESTARTS   AGE
kube-system   coredns-5c98db65d4-6x7g2                  1/1     Running   0          15m
kube-system   coredns-5c98db65d4-zzl4t                  1/1     Running   0          15m
kube-system   etcd-ip-172-31-86-69                      1/1     Running   0          13m
kube-system   kube-apiserver-ip-172-31-86-69            1/1     Running   0          14m
kube-system   kube-controller-manager-ip-172-31-86-69   1/1     Running   0          14m
kube-system   kube-proxy-4n9br                          1/1     Running   0          15m
kube-system   kube-scheduler-ip-172-31-86-69            1/1     Running   0          14m
kube-system   weave-net-ht9nf                           2/2     Running   0          3m2s
root@ip-172-31-86-69:~#
```