How do you install and configure multi-node Kubernetes cluster?

srinathchalla@outlook.com

Installing Kubernetes using kubeadm

Concept

Udemy

Objectives

Concept

a. Overview of kubeadm
b. kubeadm commands

Review Demo

a. Pre-reqs
b. Installing software applications
c. Configuring Master
d. Join Worker Node
e. Test

srinathchalla@outlook.com

kubeadm

Kubeadm helps with installing and configuring Kubernetes cluster

srinathchalla@outlook.com

kubeadm commands

kubeadm init //on master

```
kubeadm init [flags]
```

kubeadm join //on worker

```
kubeadm join –-token [] --discovery-token-ca-cert-hash []
```

kubeadm token

```
kubeadm token [create|delete|list|generate] [flags]
```

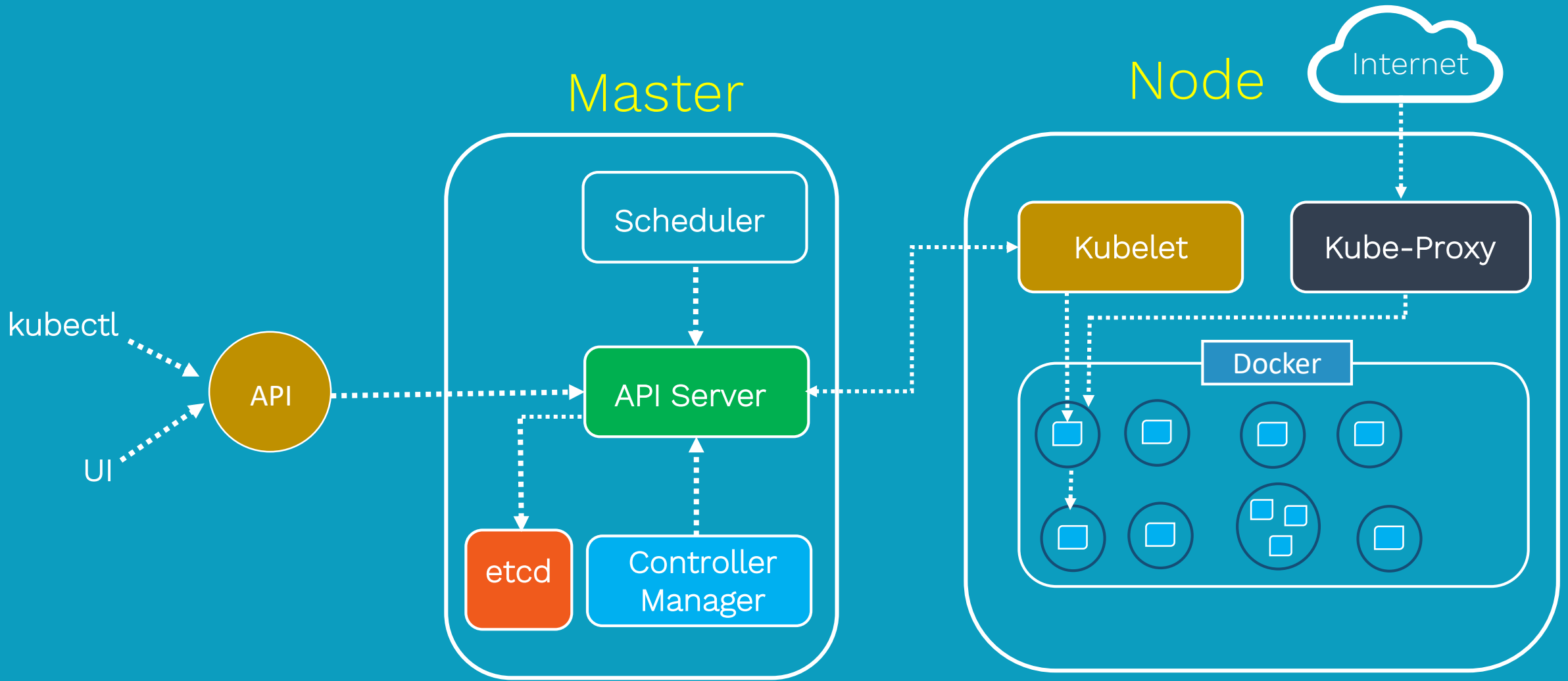
kubeadm version

```
Kubeadm version [flags]
```

kubeadm upgrade

```
kubeadm upgrade plan [version] [flags]
```

srinathchalla@outlook.com



Kubernetes Architecture

srinathchalla@outlook.com

Configuring Kubernetes by kubeadm

Installing | Testing

srinathchalla@outlook.com

Pre-reqs

- 3GB or more RAM
- 3 CPU or more
- Full network connectivity among all machines in the cluster.
- Disable SWAP on all nodes
- Disable SELinux on all nodes

Steps

1. Create VMs which are part of k8s cluster (master & worker nodes)

2. Disable SELinux and SWAP on all nodes

3. Install Kubeadm, kubelet, kubectl and Docker on all nodes
    - Start and enable docker and kubelet on all nodes

4. Initialize the master node

5. Configure Pod network

6. Join worker nodes to the cluster

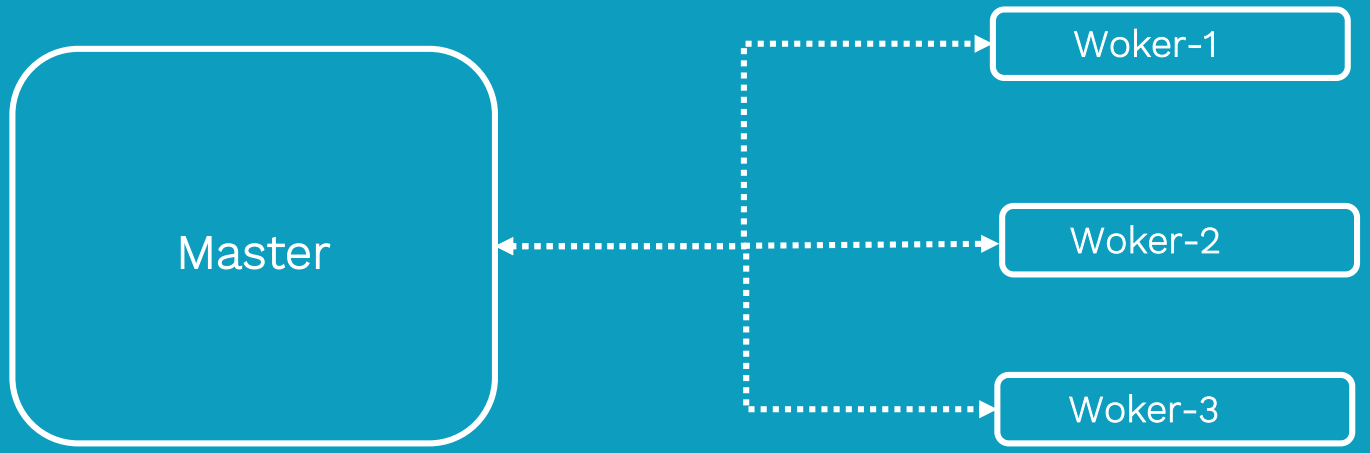
srinathchalla@outlook.com

kubeadm

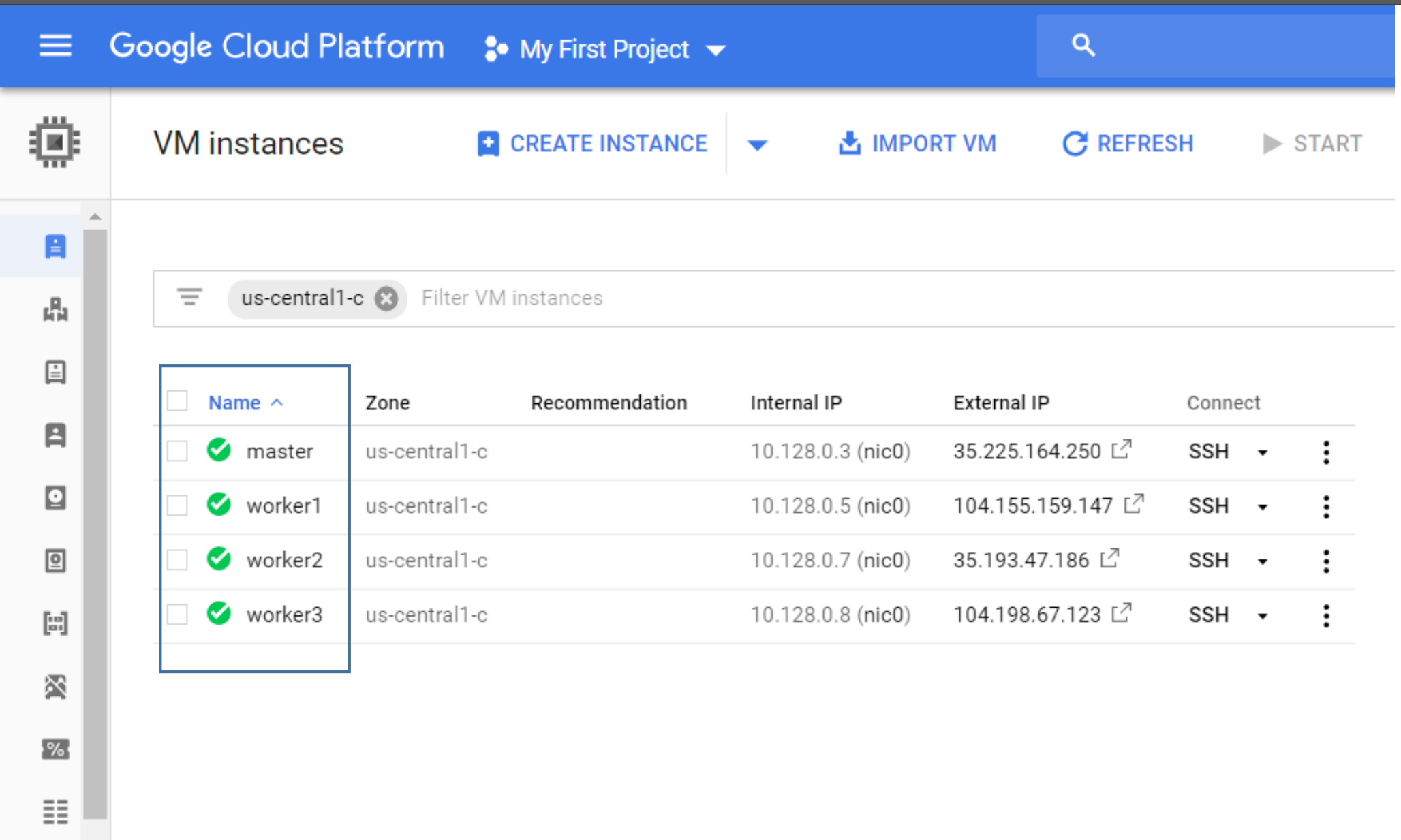
Review Demo

srinathchalla@outlook.com

Review Demo



1. Create VMs



srinathchalla@outlook.com

2. Disable SWAP and SELinux on all nodes

Disable SWAP on all nodes:

```
root@master:$  swapoff -a
```

Disable SELinux on all nodes:

```
root@master:$ setenforce 0
root@master:$ sed -i 's/enforcing/disabled/g' /etc/selinux/config
root@master:$ grep disabled /etc/selinux/config | grep -v '#'
SELINUX=disabled
```

Reboot all nodes

⚠ Perform all above steps on all worker nodes as well

srinathchalla@outlook.com

3a. Install Docker

Install Docker:

```
root@master:#     yum update -y
root@master:#     yum install -y docker
```

Start and enable Docker:

```
root@master:#     systemctl start docker
root@master:#     systemctl enable docker
root@master:#     systemctl status docker
```

```
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2018-09-03 13:25:42 UTC; 1s ago
     Docs: http://docs.docker.com
 Main PID: 13833 (dockerd-current)
   CGroup: /system.slice/docker.service
           ├─13833 /usr/bin/dockerd-current --add-runtime docker-runc=/usr/libexec/docker/docker-runc-current --default-ru...
           └─13837 /usr/bin/docker-containerd-current -l unix:///var/run/docker/libcontainerd/docker-containerd.sock --met...

Sep 03 13:25:40 master dockerd-current[13833]: time="2018-09-03T13:25:40.795598959Z" level=warning msg="Docker could.."
...
Sep 03 13:25:42 master systemd[1]: Started Docker Application Container Engine.
Sep 03 13:25:42 master dockerd-current[13833]: time="2018-09-03T13:25:42.319356249Z" level=info msg="API listen on /v...sock"
Hint: Some lines were ellipsized, use -l to show in full.
```

Perform all above steps on all worker nodes as well

srinathchalla@outlook.com

3b. Install kubeadm, kubelet, Kubectl

Add Kubernetes Repo:

```
root@master:# cat <<EOF > /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86_64
enabled=1
gpgcheck=1
repo_gpgcheck=1
gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg
https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
exclude=kube*
EOF
```

Install kubelet, kubeadm, kubectl and start kubelet:

```
root@master:# yum install -y kubeadm kubelet kubectl --disableexcludes=kubernetes
root@master:# systemctl enable kubelet && systemctl start kubelet
```

Perform all above steps on all worker nodes as well

srinathchalla@outlook.com

```
[root@master ~]# cat <<EOF >  /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
EOF

[root@master ~]# sysctl --system
```

NOTE: Applicable only if you are on RHEL/CentOS 7

srinathchalla@outlook.com

4. Initialize master node

Only on *master* node:

```
root@master:#   kubeadm init --pod-network-cidr=10.240.0.0/16

...
Your Kubernetes master 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/

You can now join any number of machines by running the following on each node
as root:

  kubeadm join --token a2dc82.7e936a7ba007f01e 10.240.0.1:6443 --discovery-token-ca-cert-hash
sha256:30aca9f9c04f829a13c925224b34c47df0a784e9ba94e132a983658a70ee2914
```

srinathchalla@outlook.com

5. Configure Pod network

Only on *master* node:

```
root@master:# kubectl apply -f \
              https://raw.githubusercontent.com/coreos/flannel/v0.9.1/Documentation/kube-flannel.yml
```

```
root@master:# kubectl get pods --all-namespaces
NAMESPACE     NAME                                READY     STATUS    RESTARTS   AGE
default       kube-flannel-ds-3hsdj               2/2       Running   0          1m
kube-system   dummy-2088944543-69gpw              1/1       Running   0          4m
kube-system   etcd-master1                        1/1       Running   0          3m
kube-system   kube-apiserver-master1              1/1       Running   0          4m
kube-system   kube-controller-manager-master1     1/1       Running   0          3m
kube-system   kube-discovery-1769846148-n4fjq     1/1       Running   0          4m
kube-system   kube-dns-2924299975-fx7lv           4/4       Running   0          3m
kube-system   kube-proxy-k87q7                    1/1       Running   0          3m
kube-system   kube-scheduler-master1              1/1       Running   0          3m
```

srinathchalla@outlook.com

6. Join workers nodes to the cluster

Only on *worker* nodes:

```
root@master:# kubeadm join --token a2dc82.7e936a7ba007f01e 10.240.0.1:6443 --discovery-token-ca-cert-hash
              sha256:30aca9f9c04f829a13c925224b34c47df0a784e9ba94e132a983658a70ee2914
```

```
root@master:# kubeadm token create --print-join-command
```

srinathchalla@outlook.com

Testing

```
[root@master ~]# kubectl get no
NAME      STATUS    ROLES     AGE       VERSION
master    Ready     master    29m       v1.11.2
worker1   Ready     <none>    26m       v1.11.2
worker2   Ready     <none>    24m       v1.11.2
worker3   Ready     <none>    24m       v1.11.2
```

```
[root@master ~]# kubectl run kubernetes-bootcamp --image=gcr.io/google-samples/kubernetes-bootcamp:v1 --port=8080
deployment.apps/kubernetes-bootcamp created
```

```
[root@master ~]# kubectl get po
NAME                                  READY     STATUS    RESTARTS   AGE
kubernetes-bootcamp-69bf88c8c-mcn9p   1/1       Running   0          21s
```

srinathchalla@outlook.com

Summary

Kubeadm installs and configures minimum viable Kubernetes cluster

kubeadm commands

Steps:

1. Create VMs which are part of k8s cluster (master & worker nodes)
2. Disable SELinux and SWAP on all nodes
3. Install Docker, kubelet, kubeadm, and kubectl on all nodes
    - Start and enable docker and kubelet on all nodes
4. Initialize the master node
5. Configure Pod network
6. Join worker nodes to the cluster

srinathchalla@outlook.com

Coming up...

Demo

kubeadm