1. Design a Kubernetes Cluster

2. Install Kubernetes Masters and Nodes

3. Configure secure cluster communications

4. Configure a highly-available Kubernetes cluster

5. Know where to get the Kubernetes release binaries

6. Provision underlying infrastructure to deploy a Kubernetes cluster

7. Choose a network solution

8. Choose your Kubernetes infrastructure configuration

9. Run end-to-end tests on your cluster

10. Analyze end-to-end test results

11. Run Node end-to-end Tests

12. Install and use kubeadm to install, configure, and manage Kubernetes clusters

12.1. Install Master

show

Follow instructions using native packages management. In more details:

Enter root and update, upgrade the system.

```
sudo -i
apt-get update && apt-get upgrade -y
```

Follow instructions to install docker.

```
apt-get install -y docker.io

cat > /etc/docker/daemon.json <<EOF
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opts": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
EOF

mkdir -p /etc/systemd/system/docker.service.d

# Restart docker.
systemctl daemon-reload
systemctl restart docker
```

Add new repo for kubernetes.

```
vim /etc/apt/sources.list.d/kubernetes.list

deb http://apt.kubernetes.io/http://apt.kubernetes.io/ kubernetes-xenial main
```

Add a GPG key for the package, and update with new repo.

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

Install kubeadm, kubelet, and kubectl.

```
apt-get install -y kubeadm=1.14.1-00 kubelet=1.14.1-00 kubectl=1.14.1-00
```

Install Container Networking Interface (CNI). Follows intrustions to use Calico for Network Policy. Download calico.yaml file. Looking for the CALICO_IPv4POOL_CIDR. Note that the default is 192.168.0.0/16. If you are using a different pod CIDR, change it accordingly in the calico.yaml file.

```
wget https://docs.projectcalico.org/v3.8/manifests/calico.yamlhttps://docs.projectcalico.org/v3.8/manifests/calico.yaml
cat calico.yaml | grep -a2 CIDR
```

Need to turn off all swap devices before initalizing the master. Initialize the master using the following command.

```
kubeadm init --kubernetes-version=1.14.1 --pod-network-cidr=192.168.0.0/16 | tee kubeadm-init.out
```

Save the kubeadm join output to use when adding workers to the cluster.

```
kubeadm join 10.0.2.15:6443 --token m3jpro.pvufj1envk6mx3g5 \
    --discovery-token-ca-cert-hash sha256:822885260222721f04296d96d490ddf9de568cd3507b735a1c21a5185755042e
```

Exit root and following instructions for the regular user.

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

Copy calico.yaml file from root to regular user space, and apply the calico file with kubectl.

```
sudo cp /root/calico.yaml .
kubectl apply -f calico.yaml
```

Apply kubectl autocompletion.

```
source <(kubectl completion bash) # setup autocomplete in bash into the current shell, bash-completion package should be installed first.
echo "source <(kubectl completion bash)" >> ~/.bashrc # add autocomplete permanently to your bash shell.
```

notes for installing locally using VirtualBox

Follow some instructions here. The problem is the ip used in kubeadm join instruction. Kubeadm has generated the ip of NAT interface and the correct ip is the ip of master host.

In general, get the IP Addresses of the Nodes. For example:

- Master - 20.0.0.11/24
- Worker1 - 20.0.0.21/24

Run the below command on master:

```
kubeadm init --apiserver-advertise-address=20.0.0.11 --kubernetes-version=1.14.1 --pod-network-cidr=192.168.0.0/16 | tee kubeadm-init.out
```

The output should be something like:

```
kubeadm join 20.0.0.11:6443 --token 5pfs0f.70axkqvb6jzte28i \
    --discovery-token-ca-cert-hash sha256:f0a201b4355a3ed345f055afa1f0a70ade9ee8048bab6641fbbb779c3653bc9b
```

Another problem of using VirtualBox:

This happens if you're running a multi-box setup because the kubelet on the workers end up binding services to the wrong ethernet interface. This error is the master attempting a connection to the wrong address in order to pull logs. The fix is to modify the config for the kubelet, as described in: Fix 1 OR Fix 2.

Add "--node-ip" to '/var/lib/kubelet/kubeadm-flags.env':

```
cat /var/lib/kubelet/kubeadm-flags.env
KUBELET_KUBEADM_ARGS=--cgroup-driver=systemd --network-plugin=cni --pod-infra-container-image=k8s.gcr.io/pause:3.1 --node-ip=20.0.0.21
```

Restart Kubelet:

```
systemctl daemon-reload && systemctl restart kubelet
```

12.2. Install worker

show

Install docker similarly as above in Master.

Similar to master installation with less commands.

```
sudo -i
apt-get update && sudo apt-get install -y apt-transport-https
curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpghttps://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -
echo "deb https://apt.kubernetes.io/https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee -a /etc/apt/sources.list.d/kubernetes.list
apt-get update
apt-get install -y kubeadm=1.14.1-00 kubelet=1.14.1-00 kubectl=1.14.1-00
```

Need to turn off all swap devices before joining the cluster using the output of kubeadm init above.

```
kubeadm join 10.0.2.15:6443 --token m3jpro.pvufj1envk6mx3g5 \
    --discovery-token-ca-cert-hash sha256:822885260222721f04296d96d490ddf9de568cd3507b735a1c21a5185755042e
```