Building a Cluster With Kubeadm

Relevant Documentation

- Install Docker Engine on Ubuntu
- Install Kubeadm
- Creating a Single Control-Plane Cluster With Kubeadm

Lesson Reference

On all nodes, install the container runtime, Docker.

```
sudo apt-get update

sudo apt-get install -y \
  apt-transport-https \
  ca-certificates \
  curl \
  gnupg-agent \
  software-properties-common

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

sudo add-apt-repository \
  "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
  $(lsb_release -cs) \
  stable"

sudo apt-get update

sudo apt-get install -y docker-ce=5:19.03.12~3-0~ubuntu-bionic

sudo apt-mark hold docker-ce
```

Verify Docker is working.

```
sudo docker version
```

Disable swap.

```
sudo swapoff -a

sudo sed -i '/ swap / s/^\(.*\)$/#\1/g' /etc/fstab
```

Install kubeadm, kubelet, and kubectl.

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

cat <<EOF | sudo tee /etc/apt/sources.list.d/kubernetes.list
deb https://apt.kubernetes.io/ kubernetes-xenial main
EOF

sudo apt-get update

sudo apt-get install -y kubelet=1.19.1-00 kubeadm=1.19.1-00 kubectl=1.19.1-00

sudo apt-mark hold kubelet kubeadm kubectl
```

Create a Kubeadm configuration file.

```
vi config.yml
```

Add the basic contents of a file with no custom configuration.

```
apiVersion: kubeadm.k8s.io/v1beta2
kind: ClusterConfiguration
```

On the control plane node only, initialize the cluster and set up kubectl access.

```
sudo kubeadm init --config config.yml

mkdir -p $HOME/.kube

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

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

Verify the cluster is working.

```
kubectl get nodes
```

Install the Calico network add-on.

```
kubectl apply -f https://docs.projectcalico.org/v3.14/manifests/calico.yaml
```

Get the join command.

```
kubeadm token create --print-join-command
```

On all worker nodes, run the join command as root.

```
sudo kubeadm join ...
```

On the control plane node, verify all nodes in your cluster are ready.

```
kubectl get nodes
```