**Lesson 2 Demo 1**

**Creating and Configuring a Kubernetes Cluster**

**Objective:** To create and configure a Kubernetes cluster

**Tools required:** kubeadm, kubectl

**Prerequisites: kubeadm** and **kubectl** should be installed

**Note 1:** This Assisted Practice is based on Kubernetes version 1.23

**Note 2:** Make sure you have executed **sudo kubeadm reset** on all machines as this will clean up the entire cluster

Steps to be followed:

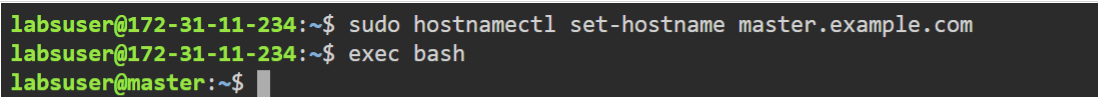
1. Changing the hostnames of all machines
2. Setting up the master node and configuring the cluster
3. Joining the worker nodes to the cluster
4. Verifying the nodes in the cluster

**Step 1: Changing the hostnames of all machines**

1. Type the following command on the master node:

**sudo hostnamectl set-hostname master.example.com**

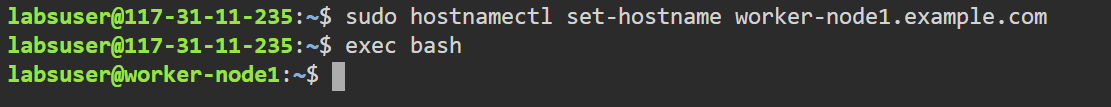
**exec bash**



1. Type the following command on worker 1:

**sudo hostnamectl set-hostname worker-node-1.example.com**

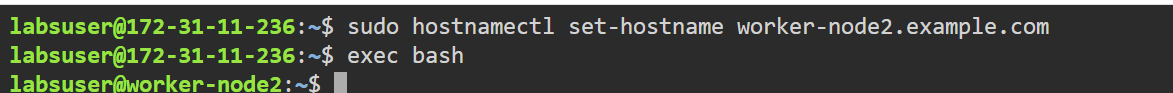
**exec bash**



1. Type the following command on worker 2:

**sudo hostnamectl set-hostname worker-node-2.example.com**

**exec bash**



**Step 2: Setting up the master node and configuring the cluster**

1. Run the following command to initiate **kubeadm**:

**sudo kubeadm init --pod-network-cidr=192.168.0.0/16**

Text

Description automatically generated

Text

Description automatically generated

Note: Run the following command if an error occurs on the master and all nodes while executing kubeadm:

**sudo mkdir /etc/docker**

**cat <<EOF | sudo tee /etc/docker/daemon.json**

**{**

**"exec-opts": ["native.cgroupdriver=systemd"],**

**"log-driver": "json-file",**

**"log-opts": {**

**"max-size": "100m"**

**},**

**"storage-driver": "overlay2"**

**}**

**EOF**

**—------------------------------------------------------------**

**sudo systemctl enable docker**

**sudo systemctl daemon-reload**

**sudo systemctl restart docker**

**sudo swapoff -a**

1. Run the following commands on the master node to allow non-root users to access use kubeadm:

**mkdir -p $HOME/.kube**

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

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

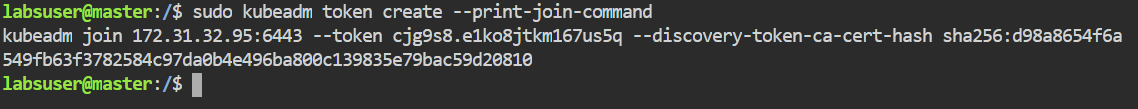
Text

Description automatically generated

**Note:** If the **cp** command asks to overwrite, write **yes** and press enter.

1. Run the following command for joining the worker nodes to the cluster and save it:

**sudo kubeadm token create --print-join-command**



1. Type the following command to add the Weave Net CNI plugin for highly available clusterUse:

**kubectl apply -f https://github.com/flannel-io/flannel/releases/latest/download/kube-flannel.yml**

Text

Description automatically generated

**Step 3: Joining the worker nodes to the cluster**

1. Go to Worker1 node and run the **kubeadm join** command copied in Step 2.3 to join this node as a worker node to the cluster:

**sudo kubeadm join 172.31.32.95:6443 --token cjg9s8.e1ko8jtkm167us5q --discovery-token-ca-cert-hash sha256:d98a8654f6a549fb63f3782584c97da0b4e496ba800c139835e79bac59d20810**

Text

Description automatically generated

1. Go to the **Worker2** node and repeat **Step 3.1** to join this node as a worker node to the cluster:

Text

Description automatically generated

**Step 4: Verifying the nodes in the cluster**

1. Navigate to the master node tab and verify the nodes that are added to the cluster
2. Run the following commands:

**kubectl get nodes**

**kubectl get namespaces**

**kubectl get pods --all-namespaces**

Text

Description automatically generated with medium confidence

Text

Description automatically generated

Text

Description automatically generated

You have successfully added the nodes.

nano fannel.yml

copy paste the below content and save file

kubectl delete -f fannel.yml

kubectl apply -f fannel.yml

kubectl get pods –A