Kubernetes Cluster Creation With Kubeadm

By Shiva

Prerequisites

- Create 3 servers in the AWS console as mentioned in the class.
- Name them 1 as master and others as worker-1 and worker-2
- Connect to all the three servers.

Step - 1

Switch to root user with below command.

```
sudo su -
```

Clone the git repository on to the three servers. Below is the git command.

```
git clone https://github.com/shivagande26/kubeadm-scripts.git
```

Step - 2

Go to the /kubeadm-scripts/scripts directory on all the servers in the cloned repository. Execute common.sh script with below command in all the 3 servers.

```
./common.sh
```

Step - 3

In the master machine, Execute other script name master.sh with below command.

Note: master.sh should only be executed on master.sh

```
./master.sh
```

Step - 4

After the successful execution check the status of the installation on master with the below command.

```
kubectl get --raw='/ready?verbose'
```

Step - 5

We should join the worker nodes to the cluster, For that follow below. Execute the below command and generate the token on the master node.

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

Above command will generate the command along with a token.

Step - 6

Execute the above generate command in all the worker nodes to join the worker nodes with master and form a cluster.

Step - 7

On the master node label the worker nodes with below command.

```
kubectl label node <node_name> node-role.kubernetes.io/worker=worker
```

Then you can execute the kubectl command and interact with cluster. For example execute "kubectl get pods"

Step - 8

Once the above step is completed go to kubeadm-scripts/manifests folder in the cloned repo and apply “metrics-server.yaml” with the below command.

```
kubectl apply -f metrics-server.yaml
```

If you want to access the cluster from your local laptop follow below.

Prerequisites

- Install kubectl on the local machine. I have mentioned about installation in the previous document.
- Once installed there will be a ~/.kube/config file. If it is not present, create that file in that path.

Step - 1

On the master node go to /etc/kubernetes/ path with the below command.

```
cd /etc/kubernetes/
```

Display the content of the admin.conf file with the below command.

```
cat admin.conf
```

Step - 2

Copy all the content. Paste the copied content in ~/.kube/config file. You should be able to connect to the cluster with kubectl commands.

For example execute “kubectl get nodes”

If you want to access the cluster from the client follow below steps.

Step - 1

Launch EC2 machine and connect/login to it. Go to root user with below command.

```
sudo su -
```

Step - 2

Install kubectl and helm in the client with below commands.

Kubectl Installation

```
curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"

sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl

kubectl version --client
```

Helm Installation

```
curl https://baltocdn.com/helm/signing.asc | gpg --dearmor | sudo tee /usr/share/keyrings/helm.gpg > /dev/null

sudo apt-get install apt-transport-https --yes

echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/helm.gpg] https://baltocdn.com/helm/stable/debian/ all main" | sudo tee /etc/apt/sources.list.d/helm-stable-debian.list

sudo apt-get update
```

```
sudo apt-get install helm
```

Step - 3

Once installed, Create a directory called .kube. Inside it create a file with name config. Add the content from the master node from /etc/kubernetes/admin.conf file to this config file with below commands.

Note: Make sure these commands will be used as root user.

```
mkdir .kube

cd .kube

vi config
```

Click on ‘i’ for insert mode as we are using vi editor.
Paste the content copied from /etc/kubernetes/admin.conf file here.
Click on ‘esc’.
Enter :wq and click on enter.

Now verify that you are able to access the cluster with the below command.

```
kubectl get nodes
```

Congratulations! You are connected to the cluster now from the client.
