Kubernetes Installation with Flannel on Ubuntu

Rohan Paranjape and chat gpt

September 5, 2024

Prerequisites

- Ubuntu 20.04 (or later) installed on two or more nodes (one master and at least one worker).
- At least 2 CPUs and 4 GB of RAM on each node.
- Root or sudo access to execute commands.
- All nodes should have a unique hostname.

Step 1: Update the System

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

- `sudo`: Grants superuser privileges to execute administrative commands.
- `apt update`: Updates the package list from repositories.
- `apt upgrade -y`: Installs newer versions of packages; `-y` confirms yes to prompts.

Step 2: Disable Swap

Kubernetes requires swap to be disabled for proper functionality.

```
sudo swapoff -a
```

- `swapoff -a`: Turns off swap immediately, which is needed because Kubernetes won't work with swap enabled.

To ensure swap remains off after reboot, modify the `/etc/fstab` file and comment out any lines that reference swap.

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

Step 3: Install Docker

Kubernetes uses Docker as the container runtime.

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

- `apt install -y docker.io`: Installs Docker, the container runtime that will be used by Kubernetes.

Start and enable Docker to run on boot:

```
sudo systemctl enable docker
sudo systemctl start docker
```

Step 4: Install Kubernetes (kubeadm, kubelet, kubectl)

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

- `apt-transport-https`: Enables access to repositories over HTTPS.
- `curl`: Downloads data from the specified URL.
- `apt-key add`: Adds the Kubernetes GPG key to verify package authenticity.

Add the Kubernetes repository to your system:

```
echo "deb https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee /etc/apt/sources
```

Update the package list and install `kubelet`, `kubeadm`, and `kubectl`:

```
sudo apt update
sudo apt install -y kubelet kubeadm kubectl
```

- `kubelet`: The primary node agent that runs and manages container pods.
- `kubeadm`: Simplifies Kubernetes cluster setup.
- `kubectl`: A command-line tool used to interact with the Kubernetes API.

Ensure these services are enabled:

```
sudo systemctl enable kubelet
```

Step 5: Initialize the Master Node

On the master node, initialize the Kubernetes control plane with `kubeadm`:

```
sudo kubeadm init --pod-network-cidr=10.244.0.0/16
```

- `kubeadm init`: Initializes the Kubernetes control plane (the master).
- `--pod-network-cidr=10.244.0.0/16`: Specifies the CIDR for the pod network.

To allow your user to interact with Kubernetes, set up the kubeconfig file:

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

Step 6: Install Flannel (Pod Network)

Flannel is a simple overlay network provider.

```
kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/ku
```

Step 7: Join Worker Nodes to the Cluster

On each worker node, run the command provided by the master node after initialization. It looks like this:

```
sudo kubeadm join <master-ip>:<port> --token <token> --discovery-token-ca-cert-hash sha25
```

To regenerate the token and hash if you missed them, run this on the master:

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

Step 8: Verify the Setup

On the master node, verify that all nodes have joined the cluster:

```
kubectl get nodes
```