Oliver Radwell

Blog about nearly everything

Provisioning Single-node Kubernetes Cluster using kubeadm on Ubuntu 20.04

2021-05-28 Server Configuration kubeadm, kubernetes

There are many tools out there to provision single-node Kubernetes clusters but kubeadm is the way to go for a production-like set up. Although it is more difficult to create a cluster with kubeadm, with its configuration options you can tweak the cluster to your needs. By following this post you can easily create a Single-node Kubernetes Cluster using kubeadm on Ubuntu 20.04.

```
$ kubectl get all
NAME                             READY   STATUS    RESTARTS   AGE
pod/wordpress-5bbd6f75ff-xv57n   1/1     Running   2          21h
pod/wordpress-mariadb-0          1/1     Running   2          21h

NAME                         TYPE           CLUSTER-IP       EXTERNAL-IP   PORT(S)                      AGE
service/kubernetes           ClusterIP      10.96.0.1        <none>        443/TCP                      45h
service/wordpress            LoadBalancer   10.110.213.79    <pending>     80:30936/TCP,443:32419/TCP   21h
service/wordpress-mariadb    ClusterIP      10.103.175.154   <none>        3306/TCP                     21h

NAME                        READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/wordpress   1/1     1            1           21h

NAME                                   DESIRED   CURRENT   READY   AGE
replicaset.apps/wordpress-5bbd6f75ff   1         1         1       21h

NAME                                 READY   AGE
statefulset.apps/wordpress-mariadb   1/1     21h
```

Install general dependencies

Some packages need to be installed on your system for the commands we're going to use later.

```
sudo apt-get update
sudo apt-get install -y apt-transport-https ca-certificates \
  curl gnupg lsb-release
```

Install docker from official repository

Installing docker from official docker repository as this is the recommended way.

```
# Remove all other versions of docker from your system
sudo apt-get remove -y docker docker-engine \
  docker.io containerd runc

# Add docker GPG key
curl -fsSL https://download.docker.com/linux/ubuntu/gpg \
```

```
  | sudo gpg --dearmor \
  -o /usr/share/keyrings/docker-archive-keyring.gpg

# Add docker apt repository
echo \
  "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-
archive-keyring.gpg] https://download.docker.com/linux/ubuntu
$(lsb_release -cs) stable" \
  | sudo tee /etc/apt/sources.list.d/docker.list

# Fetch the package lists from docker repository
sudo apt-get update

# Install docker and containerd
sudo apt-get install -y docker-ce docker-ce-cli containerd.io
```

Configure docker for kubeadm

We have to do some configuration changes to docker to make it work with Kubernetes or kubeadm pre-flight checks will fail.

```
# Configure docker to use overlay2 storage and systemd
sudo mkdir -p /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

# Restart docker to load new configuration
sudo systemctl restart docker

# Add docker to start up programs
sudo systemctl enable docker

# Allow current user access to docker command line
sudo usermod -aG docker $USER
```

Install kubeadm, kubelet & kubectl

You need to ensure the versions of kubeadm, kubelet and kubectl are compatible.

```
# Add Kubernetes GPG key
sudo curl -fsSLo /usr/share/keyrings/kubernetes-archive-
keyring.gpg \
  https://packages.cloud.google.com/apt/doc/apt-key.gpg

# Add Kubernetes apt repository
echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-
keyring.gpg] https://apt.kubernetes.io/ kubernetes-xenial
main" \
  | sudo tee /etc/apt/sources.list.d/kubernetes.list

# Fetch package list
sudo apt-get update

sudo apt-get install -y kubelet kubeadm kubectl

# Prevent them from being updated automatically
sudo apt-mark hold kubelet kubeadm kubectl
```

Ensure swap is disabled

Swap feature has to be disabled because it is not supported by Kubernetes. See the GitHub issue regarding swap on Kubernetes for details.

```
# See if swap is enabled
swapon --show

# Turn off swap
sudo swapoff -a

# Disable swap completely
sudo sed -i -e '/swap/d' /etc/fstab
```

Create the cluster using kubeadm

It's only a single command to initialise the cluster but it won't be very functional in single-node environments until we make some changes. Note that we're providing "–pod-network-cidr" parameter as required by our CNI plugin (Flannel).

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

Configure kubectl

To be able to access the cluster we have to configure kubectl.

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

Untaint node

We have to untaint the node to allow pods to be deployed to our single-node cluster otherwise your pods will be stuck in pending state.

```
kubectl taint nodes --all node-role.kubernetes.io/master-
```

Install a CNI plugin

For networking to function you have to install a Container Network Interface (CNI) plugin. We're installing flannel.

```
kubectl apply -f
https://raw.githubusercontent.com/coreos/flannel/master/Docume
ntation/kube-flannel.yml
```

Install helm

To install our packages we're installing helm v3.

```
curl
https://raw.githubusercontent.com/helm/helm/master/scripts/get
-helm-3 | bash
```

Install a CSI driver

For storage to work we need to install a Container Storage Interface (CSI) driver. We'll install OpenEBS.

```
# Add openebs repo to helm
helm repo add openebs https://openebs.github.io/charts

kubectl create namespace openebs

helm --namespace=openebs install openebs openebs/openebs
```

Install a test application

To test the cluster you can deploy WordPress. Note that we need to specify the storage class provided by our CSI.

```
# Add bitnami repo to helm
helm repo add bitnami https://charts.bitnami.com/bitnami

helm install wordpress bitnami/wordpress \
  --set=global.storageClass=openebs-hostpath
```

The end

You have successfully created a single-node Kubernetes Cluster using kubeadm on Ubuntu 20.04 and the cluster has everything you need to install your application.

If you'd like to watch this in a video, see below:

References

- Installing kubeadm | Kubernetes
- Creating a cluster with kubeadm | Kubernetes
- Container runtimes | Kubernetes
- Install Docker Engine on Ubuntu | Docker Documentation
- Disable swap on Ubuntu – Grasping Tech
- Kubernetes: Up and Running – 2nd Edition – Amazon

Related posts

- Single-node Kubernetes on Raspberry Pi
- What's the best Kubernetes distribution for local environments?
- Silly but This Is a WordPress Blog Running on Kubernetes

1 thought on "Provisioning Single-node Kubernetes Cluster using kubeadm on Ubuntu 20.04"

Pingback: Latest technical articles & videos. - CertDepot