

NEVIS

- [Installation of NEVIS](#)

Installation of NEVIS

Introduction

I decided to install NEVIS inside a kubernetes cluster.

[Installation in Kubernetes Cluster](#)

Installation of kubernetes

[Fedora installation of kubernetes](#)

```
sudo dnf install kubernetes kubernetes-kubeadm kubernetes-client
```

Open firewall ports 6443, 10250

```
sudo systemctl enable kubelet.service
sudo systemctl enable containerd
sudo systemctl start containerd
sudo swapoff -a
sudo dnf install iproute-tc

sudo cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf
overlay
br_netfilter
EOF

sudo modprobe overlay
sudo modprobe br_netfilter

# sysctl params required by setup, params persist across reboots
sudo cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-iptables = 1
```

```
net.bridge.bridge-nf-call-ip6tables = 1
net.ipv4.ip_forward = 1
EOF

# setting DNS correctly
sudo mkdir -p /etc/systemd/resolved.conf.d/
sudo cat <<EOF | sudo tee /etc/systemd/resolved.conf.d/stub-listener.conf
[Resolve]
DNSStubListener=no
EOF

sudo systemctl --system

sudo systemctl enable --now kubelet

sudo kubeadm init

# set KUBELET_KUBEADM_ARGS
sudo tee -a /etc/kubernetes/kubelet.conf <<EOF
KUBELET_LOG_LEVEL=5
KUBELET_KUBEADM_ARGS="--v=4 --logtostderr=true"
EOF
```

Kubelet configuration

[using-kubernetes-kubelet](#)

Accessing the cluster as normal user

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

```
# Allow the control plane machine to also run pods for applications. Otherwise more than one machine is
needed in the cluster.
```

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

Install flannel into the cluster to provide cluster networking. There are many other networking solutions besides flannel. Flannel is straightforward and suitable for this guide.

```
kubectl apply -f https://github.com/coreos/flannel/raw/master/Documentation/kube-flannel.yml
```

Useful commands

```
sudo systemctl restart kubelet
sudo systemctl status kubelet
sudo journalctl -u kubelet
ss -tlnp | grep 6443
kubectl config use-context
kubectl config view
kubectl cluster-info
kubectl get pods --all-namespaces
kubectl get svc -A
kubectl get events --namespace=kube-system
kubectl get nodes -o wide
```

Additional .conf files:

The kubernetes-kubeadm rpm installs an overriding `kubelet` unit file at:

```
/usr/lib/systemd/system/kubelet.service.d/10-kubeadm.conf
```

We strongly recommend to **not** modify either file as any changes could be lost during an update.

As documented by the Kubernetes team (<https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/kubelet-integration/#the-kubelet-drop-in-file-for-systemd>), create the following directory for user managed, system-level systemd `kubelet` overrides:

```
$ sudo mkdir -p /etc/systemd/system/kubelet.service.d/
```

Then create a unit file (`.conf` extension required) and copy the file to the directory listed above. Settings in this file will override settings from either or both of the default systemd files.

misc

Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:

<https://kubernetes.io/docs/concepts/cluster-administration/addons/>

Then you can join any number of worker nodes by running the following on each as root:

```
kubeadm join 192.168.1.35:6443 --token dapwn1.21bvsun7tw95b6j7 \
```

```
[-discovery-token-ca-cert-hash
```

```
sha256:bc878aa0a8db726627f0be2a9bfbec584bde1156114e1af61aa727e2e39302b5
```