Kubernetes Installation Guide on Fedora (Latest Kubernetes v1.33.x)

1. Prepare Both Master and Worker Nodes

sudo dnf check-update && sudo dnf install -y dnf-plugins-core curl

sudo -i

2. Add Google GPG Keys for Kubernetes Packages

sudo rpm --import <https://packages.cloud.google.com/yum/doc/yum-key.gpg>

# Import Google GPG key (no dearmor)

sudo rpm --import https://packages.cloud.google.com/yum/doc/yum-key.gpg

3. Add Kubernetes Repository (v1.33)

sudo tee cat <<EOF > /etc/yum.repos.d/kubernetes.repo

[kubernetes]

name=Kubernetes

baseurl=https://pkgs.k8s.io/core:/stable:/v1.33/rpm/

enabled=1

gpgcheck=1

gpgkey=https://pkgs.k8s.io/core:/stable:/v1.33/rpm/repodata/repomd.xml.key

EOF

4. Set SELinux to Permissive Mode

# Temporarily set SELinux to permissive

sudo setenforce 0

# Permanently set SELinux to permissive

sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config

5. Disable Swap

6. Enable Required Kernel Modules and Sysctl Settings

sudo swapoff -a

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

**1. Load the br\_netfilter module immediately**

sudo modprobe br\_netfilter

This enables the module **now** (immediate effect).

**2. Persist the setting (load on reboot)**

echo 'br\_netfilter' | sudo tee /etc/modules-load.d/k8s.conf

This tells the system to automatically load br\_netfilter every time the system boots.

**✅ Optional (for completeness): Add overlay too**

echo -e "overlay\nbr\_netfilter" | sudo tee /etc/modules-load.d/k8s.conf

**3. Verify the module is loaded**

lsmod | grep br\_netfilter

Would you like me to give you a script to run this on all your nodes?

Ask ChatGPT

Top of Form

Bottom of Form

cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf

net.bridge.bridge-nf-call-ip6tables = 1

net.bridge.bridge-nf-call-iptables = 1

net.ipv4.ip\_forward = 1

EOF

# Apply the changes system-wide

sudo sysctl --system

7. Remove Docker Repo if Present (Avoid Conflicts)

dnf config-manager --disable docker-ce-stable || true

rm -f /etc/yum.repos.d/docker-ce.repo || true

dnf clean all

8. Install and Configure containerd

# 1. Install containerd

sudo dnf install -y containerd

# 2. Create the config directory

sudo mkdir -p /etc/containerd

# 3. Generate and write default containerd config

sudo containerd config default | sudo tee /etc/containerd/config.toml > /dev/null

# 4. Set containerd to use systemd cgroup driver (important for Kubernetes compatibility)

sudo sed -i 's/SystemdCgroup = false/SystemdCgroup = true/' /etc/containerd/config.toml

# 5. Enable and start containerd

sudo systemctl enable --now containerd

# 6. Verify containerd is running

systemctl status containerd

9. Install Kubernetes Tools

sudo dnf install -y kubelet kubeadm kubectl

10. Enable and Start kubelet

sudo systemctl daemon-reexec

sudo systemctl enable --now kubelet

11. Initialize Kubernetes Master Node (Only on Master)

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

12. Configure kubectl for Your User (on Master)

# 1. Create the .kube directory (you already did this, safe to repeat)

mkdir -p $HOME/.kube

# 2. Copy the kubeconfig file from root-owned directory

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

# 3. Give ownership of the file to your current user

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

13. Deploy Pod Network (calico master only)

# Radhe Radhe

kubectl apply -f https://raw.githubusercontent.com/projectcalico/calico/v3.27.0/manifests/calico.yaml

14. Join Worker Nodes to Cluster

After master initialization, the kubeadm join ... command output and run it on each worker node.

Summary

Step Command Summary

Prep system dnf check-update && dnf install -y dnf-plugins-core curl

Add GPG keys `curl ...

Add repo Create /etc/yum.repos.d/kubernetes.repo with v1.33 URL

SELinux setenforce 0 and modify /etc/selinux/config

Disable swap swapoff -a + comment swap in /etc/fstab

Kernel & sysctl Load br\_netfilter, set sysctl params, sysctl --system

containerd Install + configure + start

Kubernetes tools Install kubelet kubeadm kubectl

kubelet service Enable & start kubelet

Master init kubeadm init --pod-network-cidr=10.244.0.0/16

Setup kubectl admin.conf to user dir

Pod network Apply Flannel

Join workers Use kubeadm join command from master output

kubeadm join 172.31.5.96:6443 --token f5au9h.7tgukvfbasvf6ij2 --discovery-token-ca-cert-hash sha256:c7f8a95dd92cbbda9ab8128208dd72415d67addbf93a876aaafff81739179762

**Here’s the breakdown:**

| **Step No.** | **Step Description** | **Master Node** | **Worker Node** |
| --- | --- | --- | --- |
| 1 to 10 | System prep, repo, SELinux, swap off, kernel modules, containerd, kubelet install and start | Same | Same |
| **11** | **Initialize Kubernetes Cluster** | kubeadm init --pod-network-cidr=10.244.0.0/16 (only on master) | **Skip** (do NOT run kubeadm init on worker) |
| **12** | **Configure kubectl** | admin.conf to $HOME/.kube/config | Skip (workers don't run kubectl management commands) |
| **13** | **Deploy Pod Network** | Run pod network YAML (e.g., Flannel) | Skip |
| **14** | **Join Cluster** | N/A | Run the kubeadm join <master-ip>:6443 --token ... --discovery-token-ca-cert-hash ... command |

kubeadm join 172.31.34.236:6443 --token zcu30z.yiktbyic6puw71bu \

--discovery-token-ca-cert-hash sha256:8de4a13fe8d71dc12003412d1260a10c4aaa40b4f8397f00f608ea72a692b9e0