

How To Install Kubernetes (K8s) Cluster On RHEL 8

By Pradeep Kumar / Last Updated: August 20, 2024 / 6 Minutes Of Reading



In this guide, we will walk you step-by-step on how you can install a Kubernetes (k8s) cluster on RHEL 8. We will demonstrate this using one Master and one worker node which we will add to our cluster.

Also known as **k8s**, Kubernetes is an opensource, and portable container orchestration platform for automating the deployment and management of containerized applications. Kubernetes was originally created by Google in the Go programming language. Currently, it is maintained by Cloud Native Computing Foundation.

Lab setup

- Master node: master-node-k8 192.168.1.130
- Worker node: worker-node-1-k8 192.168.1.140

NOTE: Steps 1 to 6 should be applied to both the Master and the worker node.

Step 1) Disable Swap Space

For best performance, Kubernetes requires that swap is disabled on the host system. This is because memory swapping can significantly lead to instability and performance degradation.

To disable swap space, run the command:

...

```
$ sudo swapoff -a
```

To make the changes persistent, run

```
$ sudo sed -i 's/^([^#]*)$/#1/g' /etc/fstab
```

Step 2) Disable SELinux



Additionally, we need to disable SELinux and set it to 'permissive' in order to allow smooth communication between the nodes and the pods.

```
$ sudo vi /etc/selinux/config
```

Change the SELINUX value from enforcing to permissive.

```
SELINUX=permissive
```

Alternatively, you use the sed command as follows.

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

Step 3) Configure Networking In Master And Worker Node

Some additional network configuration is required for your master and worker nodes to communicate effectively. On each node, edit the `/etc/hosts` file.

...

```
$ sudo vi /etc/hosts
```

Next, update the entries as shown

```
192.168.1.130 master-node-k8
192.168.1.140 worker-node-1-k8
```

Save and exit the configuration file. Next, install the traffic control utility package:

```
$ sudo dnf install -y iproute-tc
```

Step 4) Allow Firewall Rules For K8s

For seamless communication between the Master and worker node, you need to configure the firewall and allow some pertinent ports and services as outlined below.



On Master node, allow following ports,

```
$ sudo firewall-cmd --permanent --add-port=6443/tcp
$ sudo firewall-cmd --permanent --add-port=2379-2380/tcp
$ sudo firewall-cmd --permanent --add-port=10250/tcp
$ sudo firewall-cmd --permanent --add-port=10251/tcp
$ sudo firewall-cmd --permanent --add-port=10252/tcp
$ sudo firewall-cmd --reload
```

On Worker node, allow following ports,

```
$ sudo firewall-cmd --permanent --add-port=10250/tcp
$ sudo firewall-cmd --permanent --add-port=30000-32767/tcp
$ sudo firewall-cmd --reload
```

Step 5) Install CRI-O Container Runtime

Kubernetes requires a container runtime for pods to run. Kubernetes 1.23 and later versions require that you install a container runtime that conforms with the [Container Runtime](#) Interface.



A Container Runtime is an application that supports running containers. Kubernetes supports the following Container Runtime:

- Containerd
- CRI-O
- Docker Engine
- Mirantis Container Runtime

In this guide, we will install CRI-O which is a high-level container runtime. To do so, we need to enable two crucial kernel modules – [overlay](#) and [br_netfilter](#) modules.

To achieve this, we need to configure the prerequisites as follows:



First, create a modules configuration file for Kubernetes.

```
$ sudo vi /etc/modules-load.d/k8s.conf
```

Add these lines and save the changes

```
overlay  
br_netfilter
```

Then load both modules using the `modprobe` command.

...

```
$ sudo modprobe overlay  
$ sudo modprobe br_netfilter
```

Next, configure the required sysctl parameters as follows

```
$ sudo vi /etc/sysctl.d/k8s.conf
```

Add the following lines:

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

Save the changes and exit. To confirm the changes have been applied, run the command:

```
$ sudo sysctl --system
```

To install CRI-O, set the `$CRIO_VERSION` environment variable to match your CRI-O version. For instance, to install CRI-O version 1.30 set the `$CRIO_VERSION` as shown:

```
⚙ $ export CRIO_VERSION=v1.30
```

```
$ cat <<EOF | sudo tee /etc/yum.repos.d/cri-o.repo
[cri-o]
name=CRI-O
baseurl=https://pkgs.k8s.io/addons:/cri-o:/stable:/$CRIO_VERSION/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/addons:/cri-o:/stable:/$CRIO_VERSION/rpm/repodata/repomd.xml.key
EOF
```

Then use the DNF package manager to install CRI-O:

...

```
$ sudo dnf install -y cri-o
```

```
[linuxtech@master-node-k8 ~]$ sudo dnf install -y cri-o
Updating Subscription Management repositories.
CRI-O                               3.4 kB/s | 5.7 kB    00:01
Dependencies resolved.
=====
Package                        Architecture      Version              Repository           Size
=====
Installing:
cri-o                          x86_64            1.30.4-150500.1.1   cri-o                18 M
Transaction Summary
=====
Install 1 Package
Total download size: 18 M
Installed size: 70 M
Downloading Packages:
cri-o-1.30.4-150500.1.1.x86_64.rpm                                7.4 MB/s | 18 MB    00:02
Total
CRI-O                               7.3 MB/s | 18 MB    00:02
                               2.1 kB/s | 1.7 kB    00:00
```

Next, enable CRI-O on boot time and start it:

```
$ sudo systemctl enable crio
$ sudo systemctl start crio
```

Step 6) Install Kubernetes Packages

With everything required for Kubernetes to work installed, let us go ahead and install Kubernetes packages like kubelet, kubeadm and kubectl.



...

Set the Kubernetes version variable as shown below:

```
$ KUBERNETES_VERSION=v1.30
```

Next, create a Kubernetes repository file.

```
$ cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/$KUBERNETES_VERSION/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/$KUBERNETES_VERSION/rpm/repodata/repomd.xml.key
EOF
```

Save the changes and exit. Finally, install k8s package as follows. At time of writing this guide, Kubernetes version 1.30 is available.



```
$ sudo dnf install kubelet kubeadm kubectl -y
```

```
[linuxtechi@master-node-k8 ~]$
[linuxtechi@master-node-k8 ~]$ sudo dnf install kubelet kubeadm kubectl -y
Updating Subscription Management repositories.
Kubernetes
Dependencies resolved.
8.5 kB/s | 15 kB    00:01
=====
Package                                Architecture    Version          Repository        Size
-----
Installing:
kubeadm                                x86_64          1.30.4-150500.1.1  kubernetes        10 M
kubectl                                x86_64          1.30.4-150500.1.1  kubernetes        10 M
kubelet                                x86_64          1.30.4-150500.1.1  kubernetes        17 M
Installing dependencies:
conntrack-tools                        x86_64          1.4.4-11.el8      rhel-8-for-x86_64-baseos-rpms 204 k
cri-tools                              x86_64          1.30.1-150500.1.1  kubernetes        8.6 M
kubernetes-cni                         x86_64          1.4.0-150500.1.1  kubernetes        6.7 M
libnetfilter_cthelper                  x86_64          1.0.0-15.el8      rhel-8-for-x86_64-baseos-rpms 24 k
libnetfilter_cttimeout                 x86_64          1.0.0-11.el8      rhel-8-for-x86_64-baseos-rpms 24 k
libnetfilter_queue                     x86_64          1.0.4-3.el8       rhel-8-for-x86_64-baseos-rpms 31 k
socat                                  x86_64          1.7.4.1-1.el8     rhel-8-for-x86_64-appstream-rpms 323 k
Transaction Summary
-----
Install 10 Packages
Total download size: 54 M
Installed size: 293 M
```

⚙ Once installed, be sure to enable and start Kubelet service.

At this juncture, we are all set to install Kubernetes cluster.

...

Step 7) Create A Kubernetes Cluster

We are going to initialize a Kubernetes cluster using the kubeadm command as follows. This initializes a control plane in the master node.

```
$ sudo kubeadm init --pod-network-cidr=192.168.10.0/16
```

Once the control plane is created, you will be required to carry out some additional commands to start using the cluster.

```
[addons] Applied essential addon: CoreDNS
[addons] Applied essential addon: kube-proxy

Your Kubernetes control-plane has initialized successfully!

To start using your cluster, you need to run the following as a regular user:

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

Alternatively, if you are the root user, you can run:

export KUBECONFIG=/etc/kubernetes/admin.conf

You should now deploy a pod network to the cluster.
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 192.168.1.130:6443 --token 64upvp.eluqhf4jmdobsvns \
--discovery-token-ca-cert-hash sha256:d3ec95d2b1de276b977db339c0a36e80dd745534dcc14f10b9e94ac1bc8e871b
[linuxtechi@master-node-k8 ~]$
```

...

Therefore, run the commands, sequentially.

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



At the very end of the output, you will be given the command to run on worker nodes to join the cluster. We will come to

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

Step 8) Install Calico Pod Network Add-On

The next step is to install [Calico CNI](#) (Container Network Interface). It is an opensource project used to provide container networking and security. After installing Calico CNI, nodes state will change to Ready state, DNS service inside the cluster would be functional and containers can start communicating with each other.

Calico provides scalability, high performance, and interoperability with existing Kubernetes workloads. It can be deployed on-premises and on popular cloud technologies such as Google Cloud, AWS and Azure.

To install Calico CNI, run the following command from the master node

```
$ kubectl apply -f https://raw.githubusercontent.com/projectcalico/calico/v3.28.0/manifests/calico
```

To confirm if the pods have started, run the command:

```
$ kubectl get pods -n kube-system
```

You should see that each pod is 'READY' and has the 'RUNNING' status as shown in the third column.

```
[linuxtech@master-node-k8 ~]$ kubectl get pods -n kube-system
NAME                                READY   STATUS    RESTARTS   AGE
calico-kube-controllers-564985c589-st2r2  1/1     Running   0           58s
calico-node-v8hbm                     1/1     Running   0           58s
coredns-7db6d8ff4d-kb8h7              1/1     Running   0          7m43s
coredns-7db6d8ff4d-s9b59              1/1     Running   0          7m43s
etcd-master-node-k8                   1/1     Running   0          7m54s
kube-apiserver-master-node-k8          1/1     Running   0          7m59s
kube-controller-manager-master-node-k8  1/1     Running   0          7m54s
kube-proxy-w4ztj                       1/1     Running   0          7m44s
kube-scheduler-master-node-k8          1/1     Running   0          7m54s
[linuxtech@master-node-k8 ~]$
```

...

To verify the master node's availability in the cluster, run the command:

```
$ kubectl get nodes
NAME             STATUS    ROLES    AGE   VERSION
master-node-k8   Ready     control-plane   9m56s   v1.30.4
$
```

In addition, you can retrieve more information using the `-o wide` options.

```
$ kubectl get nodes -o wide
```

```
[linuxtech@master-node-k8 ~]$ kubectl get nodes -o wide
NAME             STATUS    ROLES    AGE   VERSION   INTERNAL-IP   EXTERNAL-IP   OS-IMAGE                                     KERNEL-VERSION   CONTAINER-ENGINE
master-node-k8   Ready     control-plane   10m   v1.30.4   192.168.1.130   <none>         Red Hat Enterprise Linux 8.10 (otterlei)   4.18.0-553.el8.x86_64   cri-o
```



```
$ kubectl get pods --all-namespaces
```

```
[linuxtechi@master-node-k8 ~]$ kubectl get pods --all-namespaces
NAMESPACE   NAME                                     READY   STATUS    RESTARTS   AGE
kube-system  calico-kube-controllers-564985c589-st2r2 1/1     Running   0           6m18s
kube-system  calico-node-v8hbm                        1/1     Running   0           6m18s
kube-system  coredns-7db6d8ff4d-kb8h7                1/1     Running   0           13m
kube-system  coredns-7db6d8ff4d-s9b59                1/1     Running   0           13m
kube-system  etcd-master-node-k8                     1/1     Running   0           13m
kube-system  kube-apiserver-master-node-k8            1/1     Running   0           13m
kube-system  kube-controller-manager-master-node-k8   1/1     Running   0           13m
kube-system  kube-proxy-w4ztj                        1/1     Running   0           13m
kube-system  kube-scheduler-master-node-k8            1/1     Running   0           13m
[linuxtechi@master-node-k8 ~]$
[linuxtechi@master-node-k8 ~]$
```

Step 9) Adding Worker Node To The Cluster

To add the worker node to the Kubernetes cluster, follow step 1 up until Step 6. Once you are done, run the command generated by the master node for joining a worker node to the cluster. In our case, this will be:

```
$ sudo kubeadm join 192.168.1.130:6443 --token 64upvp.eluqhf4jmdobsvns \
--discovery-token-ca-cert-hash sha256:d3ec95d2b1de276b977db339c0a36e80dd745534dcc14f10b9e94ac1bc8e{
```



If all goes well, you should get the notification that the node has joined the cluster. Repeat the same procedure for other nodes in case you have multiple worker nodes

Now, head back to the master node and, once again, verify the nodes in your cluster. This time around, the worker node will appear in the list on nodes in the cluster,

...

```
$ kubectl get nodes
```



Conclusion

That was a walk through of how you can install a Kubernetes Cluster on RHEL 8. Your feedback on this guide is welcome.

About The Author



Pradeep Kumar

I am a Cloud Consultant with over 15 years of experience in Linux, Kubernetes, cloud technologies (AWS, Azure, OpenStack), automation (Ansible, Terraform), and DevOps. I hold certifications like RHCA, CKA, CKAD, CKS, AWS, and Azure.

← Previous Post

Next Post →

25 Thoughts On “How To Install Kubernetes (K8s) Cluster On RHEL 8”

Older Comments

GANESH

MARCH 19, 2023 AT 8:32 PM

Hi Getting below exception in cordns pod and calico-kube-controllers pod can any one suggest what is the is the solution:

coredns: exec /coredns: operation not permitted

calico-kube-controllers: exec /usr/bin/kube-controllers: permission denied

Thanks in advance !

Reply

SAM NTURANABO

APRIL 4, 2023 AT 7:51 PM

This tutorial is awesome. Thanks alot

Reply



Hi Pradeep, you rock, I installed the k8s on rocky 9.1, it worked. Only thing to take care firewall-cmd does not come enabled. So u install first with sudo yum install firewalld
sudo systemctl enable firewalld --now. After which the step 4 can be executed. Since firewalld is not installed, same as can be skipped as well, as firewall is not installed and enabled. Else it was cakewalk.... Great... thanks a ton...

Reply

DAVID

MAY 12, 2023 AT 6:30 PM

Great tutorial, after going through a lot of different documents this one help me to set up my kubernetes cluster. Thank you

Reply

PARSHVA SHAH

JUNE 7, 2023 AT 4:44 AM

Hi

we were able to setup the cluster with the above mentioned steps.

K8s :- V1.26.1

OS : - RHEL 8.7

VM is hosted using Azure cloud provider.

Core dns logs :-

CoreDNS-1.9.3

linux/amd64, go1.18.2, 45b0a11

[ERROR] plugin/errors: 2 2568389657905608835.8295431261288812352. HINFO: read udp 192.168.54.66:34391->168.63.129.16:53: read: no route to host

[ERROR] plugin/errors: 2 2568389657905608835.8295431261288812352. HINFO: read udp 192.168.54.66:34275->168.63.129.16:53: read: no route to host

[ERROR] plugin/errors: 2 2568389657905608835.8295431261288812352. HINFO: read udp 192.168.54.66:59435->168.63.129.16:53: read: no route to host

Unable to reach internet from inside the pod. nslookup fails while connecting to internet.

Do you have idea on this ?

Reply

DINESH

SEPTEMBER 27, 2023 AT 4:26 AM

Hi Parshva Shah,

We are facing the same issue in azure. Stuck in this for while . Did you resolve it?. We are also using azure

Reply

DEBJIT KAR

JUNE 16, 2023 AT 9:30 PM



The installation went through well. But when I am doing `kubectl` sometimes it is getting connected and fetching details and sometimes or most of the time it is complaining `connection refused`

DORON VEEDER

JULY 18, 2023 AT 1:59 PM

Excellent Documentation! Thanks!!

A few notes:

* I've struggled a bit to handle things behind a proxy and find a solution here – “<https://jbn1233.medium.com/docker-cri-o-behind-http-proxy-4a5645a9ff7b>”

[Reply](#)**YOGESWARAN THANGARAJ**

AUGUST 14, 2024 AT 6:57 AM

Hello everyone, Now I am trying the same repository, but it has the following error...

Updating Subscription Management repositories.

Kubernetes 0.0 B/s | 0 B 02:04

Errors during downloading metadata for repository 'kubernetes':

```
- Curl error (28): Timeout was reached for https://prod-cdn.packages.k8s.io/repositories/iscv:/kubernetes:/core:/stable:/v1.31/rpm/repodata/repomd.xml [Connection timed out after 30757 milliseconds]
- Curl error (28): Timeout was reached for https://prod-cdn.packages.k8s.io/repositories/iscv:/kubernetes:/core:/stable:/v1.31/rpm/repodata/repomd.xml [Connection timed out after 30746 milliseconds]
- Curl error (28): Timeout was reached for https://prod-cdn.packages.k8s.io/repositories/iscv:/kubernetes:/core:/stable:/v1.31/rpm/repodata/repomd.xml [Connection timed out after 30646 milliseconds]
- Curl error (28): Timeout was reached for https://prod-cdn.packages.k8s.io/repositories/iscv:/kubernetes:/core:/stable:/v1.31/rpm/repodata/repomd.xml [Connection timed out after 30740 milliseconds]
```

Error: Failed to download metadata for repo 'kubernetes': Cannot download repomd.xml: Cannot download repodata/repomd.xml: All mirrors were tried

[Reply](#)**PRADEEP KUMAR**

AUGUST 27, 2024 AT 9:04 AM

Hi Yogeswaran,

I have updated this post along with the repository URL, Please follow the steps. It should work now

[Reply](#)[Older Comments](#)

Leave A Comment

Your email address will not be published. Required fields are marked *



Type here..

Name*

Email*

Website

[Post Comment »](#)

Search...




Join Our Newsletter!

Name*

Email Address*

[Subscribe](#)

Recent Posts

- How to Setup SFTP Server on Debian 12
- How to Install PHP 8.3 on Ubuntu 24.04 (Simple Guide)
- How to Add New Worker Node to Existing Kubernetes Cluster
- How to Install Python 3 on RHEL 9 (Red Hat Enterprise Linux)
- How to Install Docker Desktop on Ubuntu 24.04
-  How to Upgrade Linux Mint 21.3 to Linux Mint 22
- How to Deploy Kubernetes Cluster on RHEL 9 with Kubeadm

How Install Kubernetes on Ubuntu 24.04 (Step-by-Step Guide)

How to Boot Ubuntu 24.04 into Rescue Mode

How to Install VirtualBox Guest Additions on Ubuntu 24.04























[About Us](#)[Privacy Policy](#)[Contact Us](#)[Write For LinuxTechi](#)

LinuxTechi: Linux Howtos, Tutorials & Guides Copyright © 2024. All Rights Reserved.

Sponsored Content

Raccomandato da



Potente arma contro la micosi.

bettermethod24.com





Una nave creduta scomparsa nel Triangolo delle Bermude, trovata 20 anni dopo

DailyBreak

