

# K3s CI/CD Setup

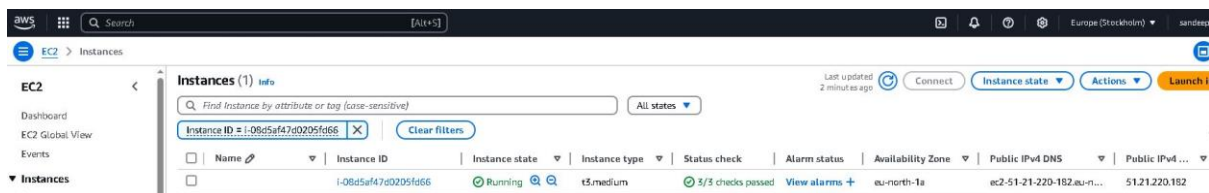
## Overview

This repository provides a comprehensive guide for installing K3s on a virtual machine and configuring the kubeconfig file for CI/CD pipelines. K3s is a lightweight Kubernetes distribution, making it ideal for self-hosted CI/CD environments.

## ## Installation Steps

### 1. Set Up the Virtual Machine

Ensure that you have a virtual machine (Ubuntu 20.04/22.04 recommended) with internet access. Update the system packages:



```
```bash
```

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

```
```
```

### 2. Install K3s

Download and install K3s using the following command:

```
```bash
```

```
curl -sfL https://get.k3s.io | sh -
```

```
```
```

```
ubuntu@ip-172-31-18-26:~$ curl -sL https://get.k3s.io | sh -
[INFO] Finding release for channel stable
[INFO] Using v1.31.6+k3s1 as release
[INFO] Downloading hash https://github.com/k3s-io/k3s/releases/download/v1.31.6+k3s1/sha256sum-amd64.txt
[INFO] Downloading binary https://github.com/k3s-io/k3s/releases/download/v1.31.6+k3s1/k3s
[INFO] Verifying binary download
[INFO] Installing k3s to /usr/local/bin/k3s
[INFO] Skipping installation of SELinux RPM
[INFO] Creating /usr/local/bin/kubectl symlink to k3s
[INFO] Creating /usr/local/bin/crictl symlink to k3s
[INFO] Creating /usr/local/bin/ctr symlink to k3s
[INFO] Creating killall script /usr/local/bin/k3s-killall.sh
[INFO] Creating uninstall script /usr/local/bin/k3s-uninstall.sh
[INFO] env: Creating environment file /etc/systemd/system/k3s.service.env
[INFO] systemd: Creating service file /etc/systemd/system/k3s.service
[INFO] systemd: Enabling k3s unit
Created symlink /etc/systemd/system/multi-user.target.wants/k3s.service → /etc/systemd/system/k3s.service.
```

Verify the installation status:

```
``bash
```

```
sudo systemctl status k3s
```

```
````
```

```
ubuntu@ip-172-31-18-26:~$ sudo systemctl status k3s
* k3s.service - Lightweight Kubernetes
   Loaded: loaded (/etc/systemd/system/k3s.service; enabled; preset: enabled)
   Active: active (running) since Wed 2025-03-19 06:26:05 UTC; 37s ago
     Docs: https://k3s.io
   Process: 1164 ExecStartPre=/bin/sh -xc /usr/bin/systemctl is-enabled --quiet nm-cloud-setup.service 2>/dev/null (code=exited, status=0/SUCCESS)
   Process: 1166 ExecStartPre=/sbin/mkfsprobe br_netfilter (code=exited, status=0/SUCCESS)
   Process: 1168 ExecStartPre=/sbin/mkfsprobe overlay (code=exited, status=0/SUCCESS)
   Main PID: 1171 (k3s-server)
      Tasks: 88
     Memory: 1.3G (peak: 1.3G)
        CPU: 36.928s
     CGroup: /system.slice/k3s.service
             └─1171 /usr/local/bin/k3s server
               └─1197 "containerd"
                 └─1939 /var/lib/rancher/k3s/data/4532effb34c1f987f51a6b60589dc2ae555b473c3d400e4e28952188cd293484f/bin/containerd-shim-runc-v2 --namespace k8s.io -id 6470fafa3660841102a6753efea09436524e5b1402b27
                 └─1946 /var/lib/rancher/k3s/data/4532effb34c1f987f51a6b60589dc2ae555b473c3d400e4e28952188cd293484f/bin/containerd-shim-runc-v2 --namespace k8s.io -id 3f4423c2f40590a97883b3914748471492120482a-
                 └─1975 /var/lib/rancher/k3s/data/4532effb34c1f987f51a6b60589dc2ae555b473c3d400e4e28952188cd293484f/bin/containerd-shim-runc-v2 --namespace k8s.io -id 851cc4c1a18dab3cf452857696a57751255f81bb1f0d15-
                 └─3044 /var/lib/rancher/k3s/data/4532effb34c1f987f51a6b60589dc2ae555b473c3d400e4e28952188cd293484f/bin/containerd-shim-runc-v2 --namespace k8s.io -id 03bc6248a61c779d2e7bd3b43997f6176530568cd6a6-
                 └─3070 /var/lib/rancher/k3s/data/4532effb34c1f987f51a6b60589dc2ae555b473c3d400e4e28952188cd293484f/bin/containerd-shim-runc-v2 --namespace k8s.io -id 563765756f1216959161b1069650fa60ee2514b49e5d4-

Mar 19 06:26:41 ip-172-31-18-26 k3s[1171]: 10319 06:26:41.728782 1171 resource_quota_monitor.go:227] "QuotaMonitor created object count evaluator" resource="ingressrouteops.traefik.io"
Mar 19 06:26:41 ip-172-31-18-26 k3s[1171]: 10319 06:26:41.728811 1171 resource_quota_monitor.go:227] "QuotaMonitor created object count evaluator" resource="middlewares.traefik.io"
Mar 19 06:26:41 ip-172-31-18-26 k3s[1171]: 10319 06:26:41.728833 1171 resource_quota_monitor.go:227] "QuotaMonitor created object count evaluator" resource="servertransports.traefik.containo.us"
Mar 19 06:26:41 ip-172-31-18-26 k3s[1171]: 10319 06:26:41.728859 1171 resource_quota_monitor.go:227] "QuotaMonitor created object count evaluator" resource="middlewares.traefik.containo.us"
Mar 19 06:26:41 ip-172-31-18-26 k3s[1171]: 10319 06:26:41.728897 1171 resource_quota_monitor.go:227] "QuotaMonitor created object count evaluator" resource="liststores.traefik.io"
Mar 19 06:26:41 ip-172-31-18-26 k3s[1171]: 10319 06:26:41.729140 1171 shared_informer.go:313] Waiting for caches to sync for resource quota
Mar 19 06:26:41 ip-172-31-18-26 k3s[1171]: 10319 06:26:41.830099 1171 shared_informer.go:320] Caches are synced for resource quota
Mar 19 06:26:42 ip-172-31-18-26 k3s[1171]: 10319 06:26:42.147574 1171 shared_informer.go:313] Waiting for caches to sync for garbage collector
Mar 19 06:26:42 ip-172-31-18-26 k3s[1171]: 10319 06:26:42.147651 1171 shared_informer.go:320] Caches are synced for garbage collector
lines 1-30/30 (END)
```

Check if the node is ready:

```
``bash
```

```
sudo k3s kubectl get nodes
```

```
````
```

```
ubuntu@ip-172-31-18-26:~$ sudo k3s kubectl get nodes
NAME                STATUS    ROLES    AGE   VERSION
ip-172-31-18-26     Ready    control-plane,master   2m9s   v1.31.6+k3s1
ubuntu@ip-172-31-18-26:~$
```

### 3. Retrieve and Configure Kubeconfig ( This can be done after Step 6 )

Create the necessary directory for the Jenkins user:

```
```bash
```

```
sudo mkdir -p /var/lib/jenkins/.kube
```

Copy the K3s configuration file:

Adjust permissions for Jenkins:

```
sudo cp /etc/rancher/k3s/k3s.yaml /var/lib/jenkins/.kube/config
```

```
sudo chown -R jenkins:jenkins /var/lib/jenkins/.kube
```

```
sudo chmod 600 /var/lib/jenkins/.kube/config
```

```
```
```

Verify the configuration:

```
```bash
```

```
Sudo kubectl get nodes
```

```
```
```



## 5. Use K3s in CI/CD Pipelines

For Jenkins, GitHub Actions, or GitLab CI, set up `kubeconfig` as a secret and use it in pipelines.

## Example: Jenkins

```
``bash

export KUBECONFIG=/path/to/kubeconfig

kubectl get pods

``
```

## 6. Install Jenkins

Run the script install.sh

```
``bash

sudo sh -x install.sh

``
```

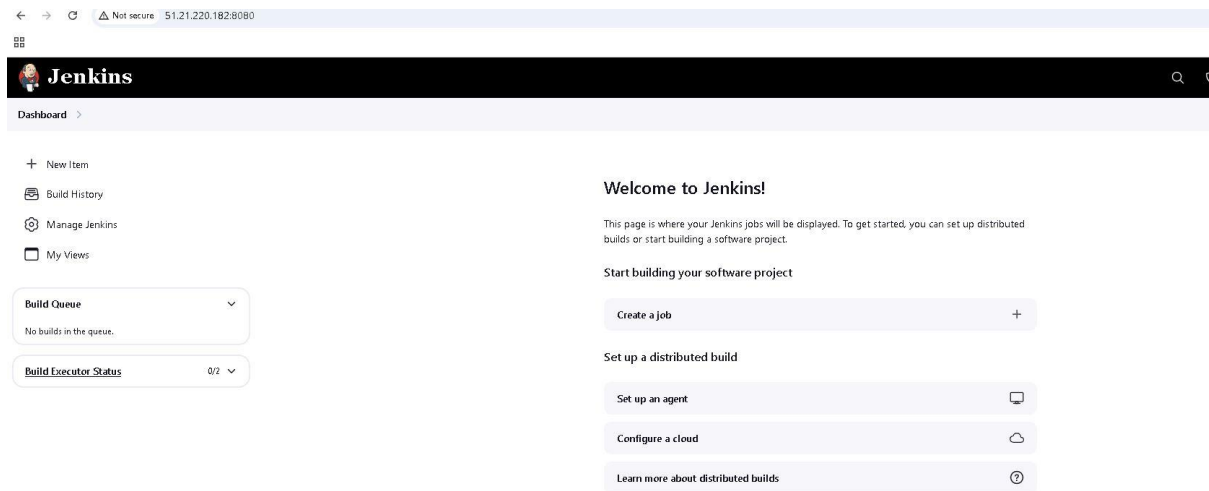
```
Session ID: sandeep-og5gthks9q22gr564u55jju Instance ID: i-08d5af47d0205fc66 Terminate

ubuntu@ip-172-31-18-26:~$ systemctl status jenkins.service
* jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; preset: enabled)
   Active: active (running) since Wed 2025-03-19 06:49:48 UTC; 35s ago
     Main PID: 11460 (java)
       Tasks: 50 (limit: 4584)
      Memory: 489.1M (peak: 751.2M)
         CPU: 24.67s
    CGroup: /system.slice/jenkins.service
            └─11460 /usr/bin/java -Djava.net.preferIPv4=true -jar /usr/share/java/jenkins.war --webroot"/var/cache/jenkins/war" --httpPort=8080

Mar 19 06:49:41 ip-172-31-18-26 jenkins[11460]: 6cf0e96f8fbb4d308a5f96d80534c120a
Mar 19 06:49:41 ip-172-31-18-26 jenkins[11460]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Mar 19 06:49:41 ip-172-31-18-26 jenkins[11460]: *****
Mar 19 06:49:41 ip-172-31-18-26 jenkins[11460]: *****
Mar 19 06:49:41 ip-172-31-18-26 jenkins[11460]: *****
Mar 19 06:49:48 ip-172-31-18-26 jenkins[11460]: 2025-03-19 06:49:48.026+0000 [id=23] INFO Jenkins.InitSeatorsRunner$1$onAttained: Completed initialization
Mar 19 06:49:48 ip-172-31-18-26 jenkins[11460]: 2025-03-19 06:49:48.059+0000 [id=23] INFO Hudson.Lifecycle.Lifecycle$onReady: Jenkins is fully up and running
Mar 19 06:49:48 ip-172-31-18-26 systemd[1]: Started jenkins.service - Jenkins Continuous Integration Server.
Mar 19 06:49:49 ip-172-31-18-26 jenkins[11460]: 2025-03-19 06:49:49.065+0000 [id=48] INFO hudson.DownloadService$downloadable$1$ed: Obtained the updated data file for hudson.tasks.Maven.MavenInstaller
Mar 19 06:49:49 ip-172-31-18-26 jenkins[11460]: 2025-03-19 06:49:49.066+0000 [id=48] INFO Hudson.Util.Retrier$Start: Performed the action check updates server successfully at the attempt #1
ubuntu@ip-172-31-18-26:~$
```

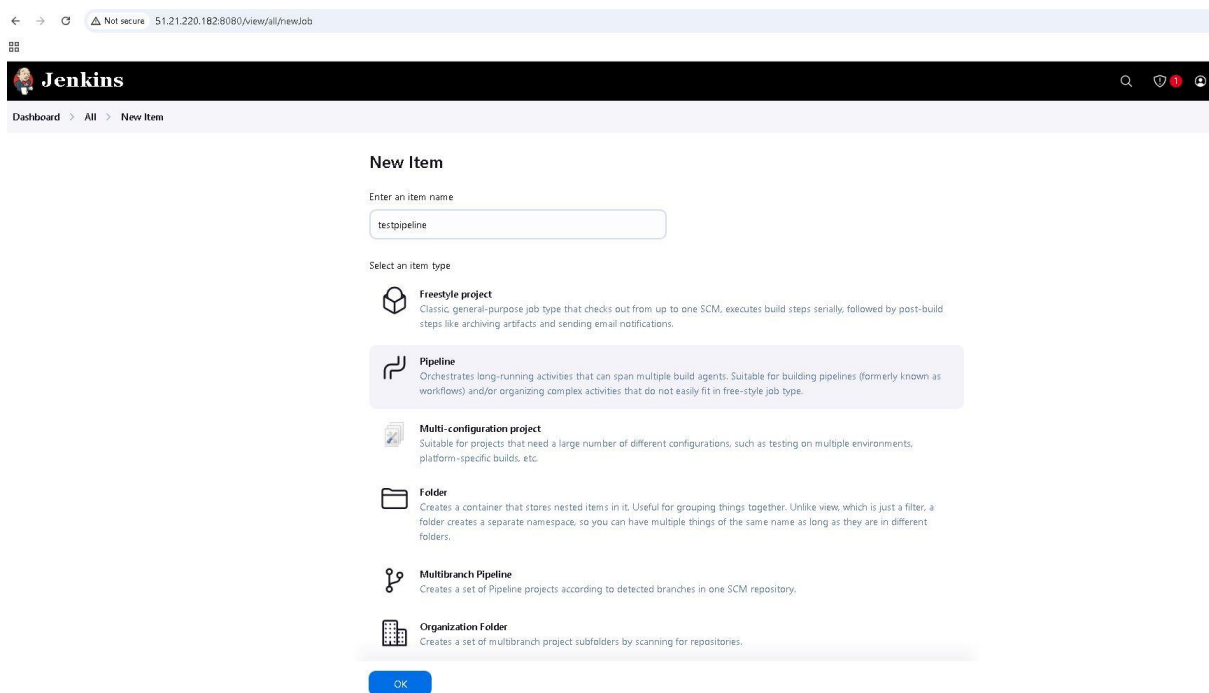
Ensure that port 8080 is open in the security group attached to the instance to allow access to the Jenkins web interface:

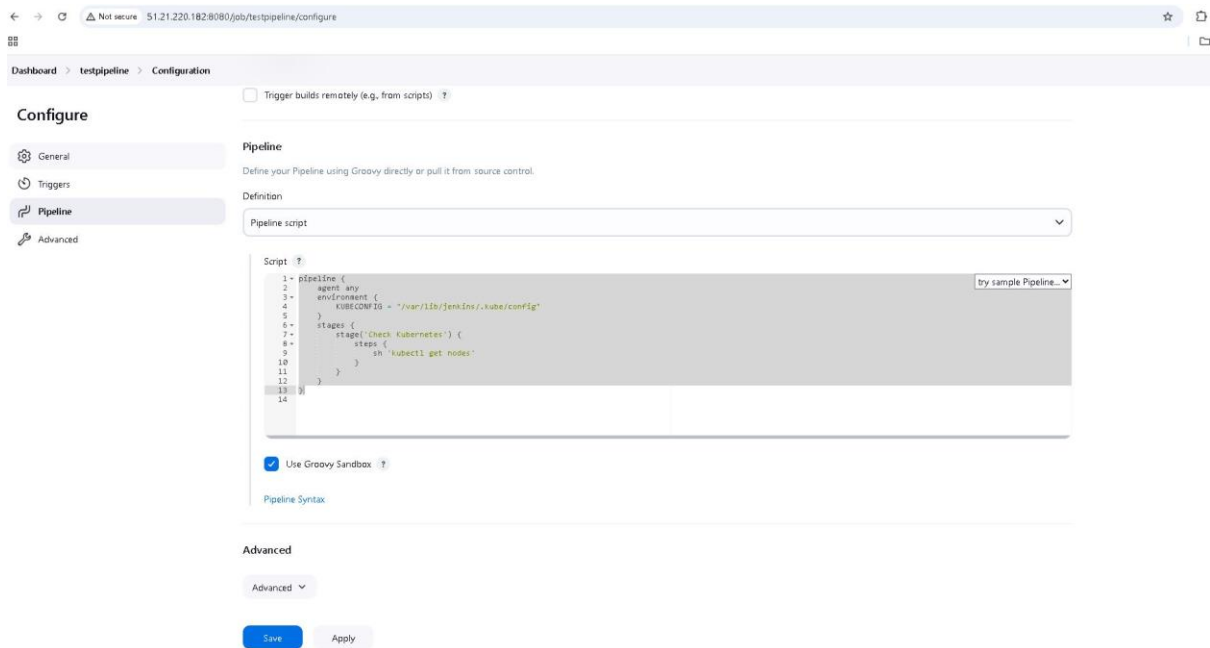
[http://<PUBLIC\\_IP>:8080](http://<PUBLIC_IP>:8080)



## 7. Verify KUBECONFIG in a Jenkins Pipeline

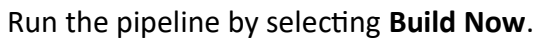
Create a new Jenkins pipeline and configure it as follows:





## Pipeline Script Example:

```
``groovy
pipeline {
    agent any
    environment {
        KUBECONFIG = "/var/lib/jenkins/.kube/config"
    }
    stages {
        stage('Check Kubernetes') {
            steps {
                sh 'kubectl get nodes'
            }
        }
    }
}
```



```
```yaml
```

run: |

**SANDEEP**



```
- name: Deploy

run: kubectl apply -f deployment.yaml

'''
```

## Conclusion

This guide provides a structured approach to setting up K3s, configuring kubeconfig for CI/CD pipelines, and integrating Kubernetes with Jenkins or GitHub Actions. By following these steps, you can efficiently deploy and manage applications in a self-hosted Kubernetes environment.

Script – install.sh

```
'''bash

#!/bin/bash

# Install Java

apt-get update


# Install required packages

apt-get update

apt-get install -y gnupg curl

sudo apt install openjdk-17-jdk -y


# Add the correct Jenkins repository key

curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo tee \

/usr/share/keyrings/jenkins-keyring.asc > /dev/null


# Add Jenkins repository
```

```
echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]
https://pkg.jenkins.io/debian-stable binary/" | sudo tee \
/etc/apt/sources.list.d/jenkins.list > /dev/null

# Update package lists again
apt-get update

# Install Java and Jenkins
apt-get install -y openjdk-17-jdk jenkins

# Enable and start Jenkins service
systemctl enable jenkins
systemctl start jenkins

# Install AWS CLI
apt-get install -y unzip
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
./aws/install
```

...