**K3s CI/CD Setup**

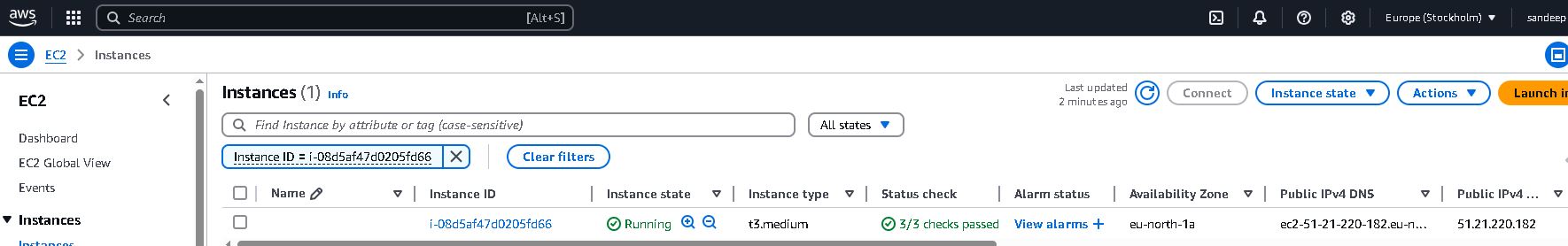
**Overview**

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

```

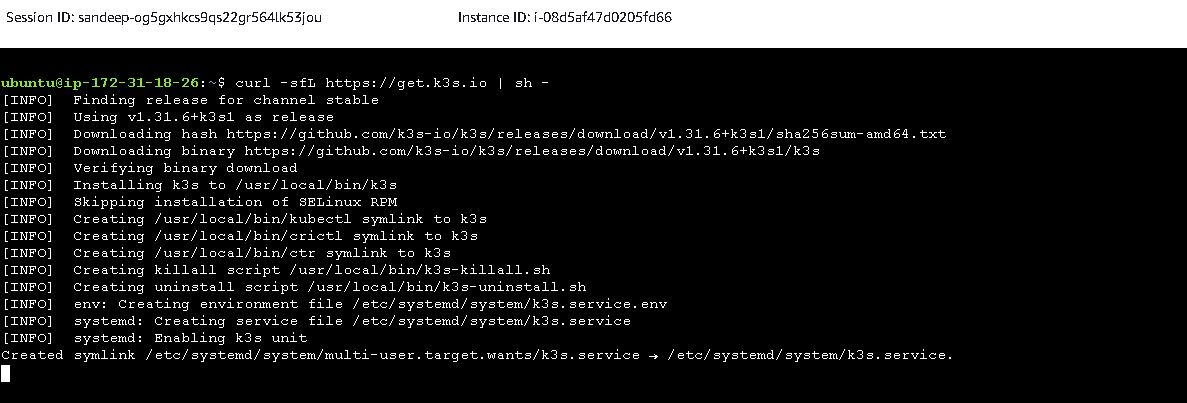
**2. Install K3s**

Download and install K3s using the following command:

```bash

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

```

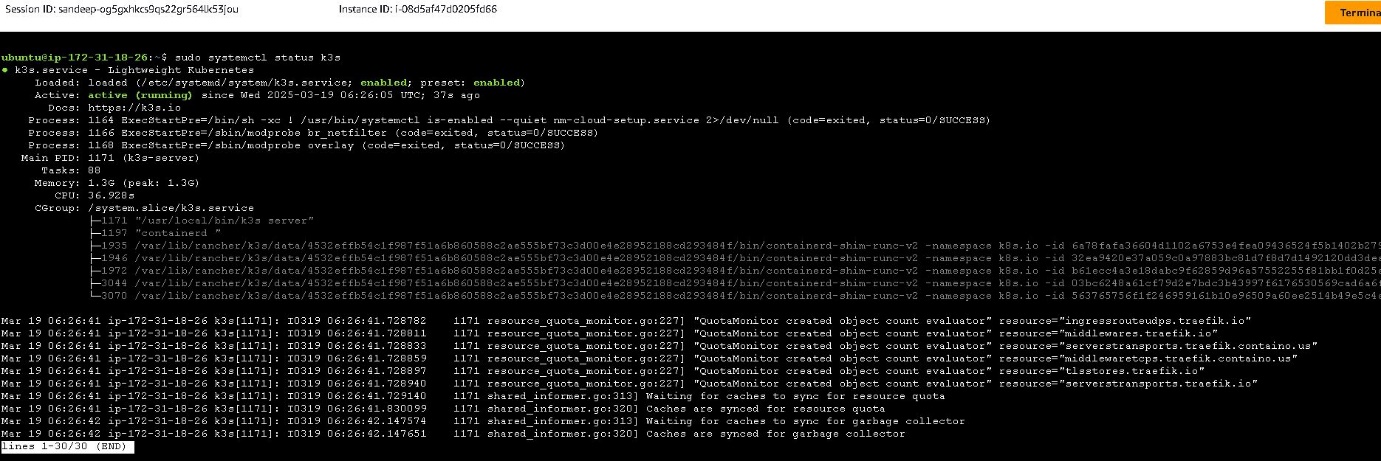


Verify the installation status:

```bash

sudo systemctl status k3s

```



Check if the node is ready:

```bash

sudo k3s kubectl get nodes

```



**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

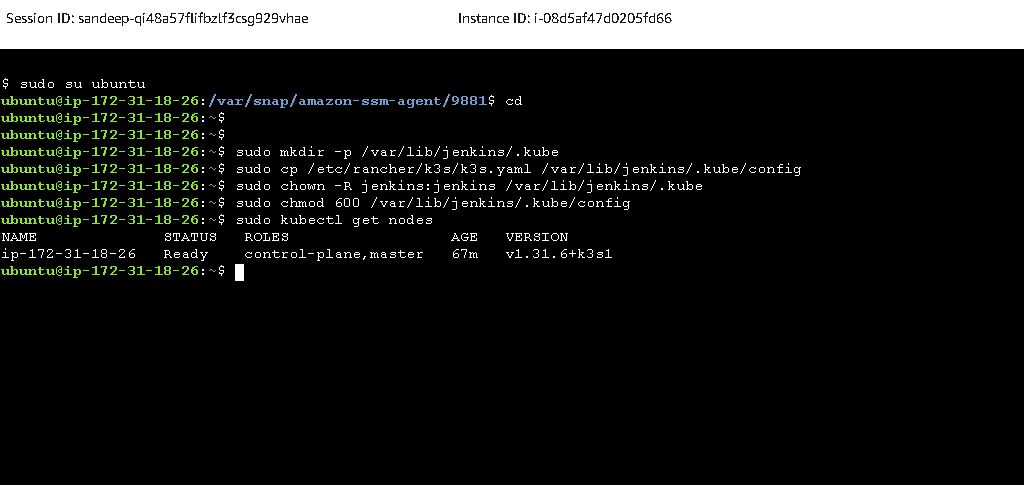
```

Verify the configuration:

```bash

Sudo kubectl get nodes

```



**4. Modify Kubeconfig for CI/CD**

Retrieve the internal IP address of the instance:

```bash

hostname -I | awk '{print $1}'

```

Update the kubeconfig file to replace 127.0.0.1 with the internal IP:

```bash

sudo sed -i "s/127.0.0.1/$(hostname -I | awk '{print $1}')/g" /var/lib/jenkins/.kube/config

```



**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

```

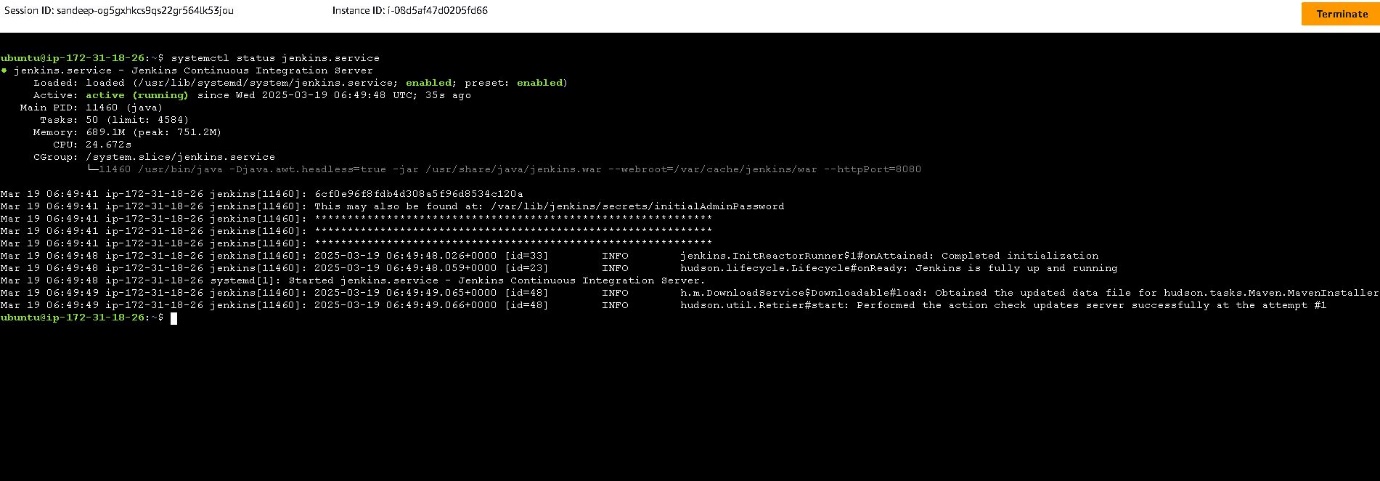
**6. Install Jenkins**

Run the script install.sh

```bash

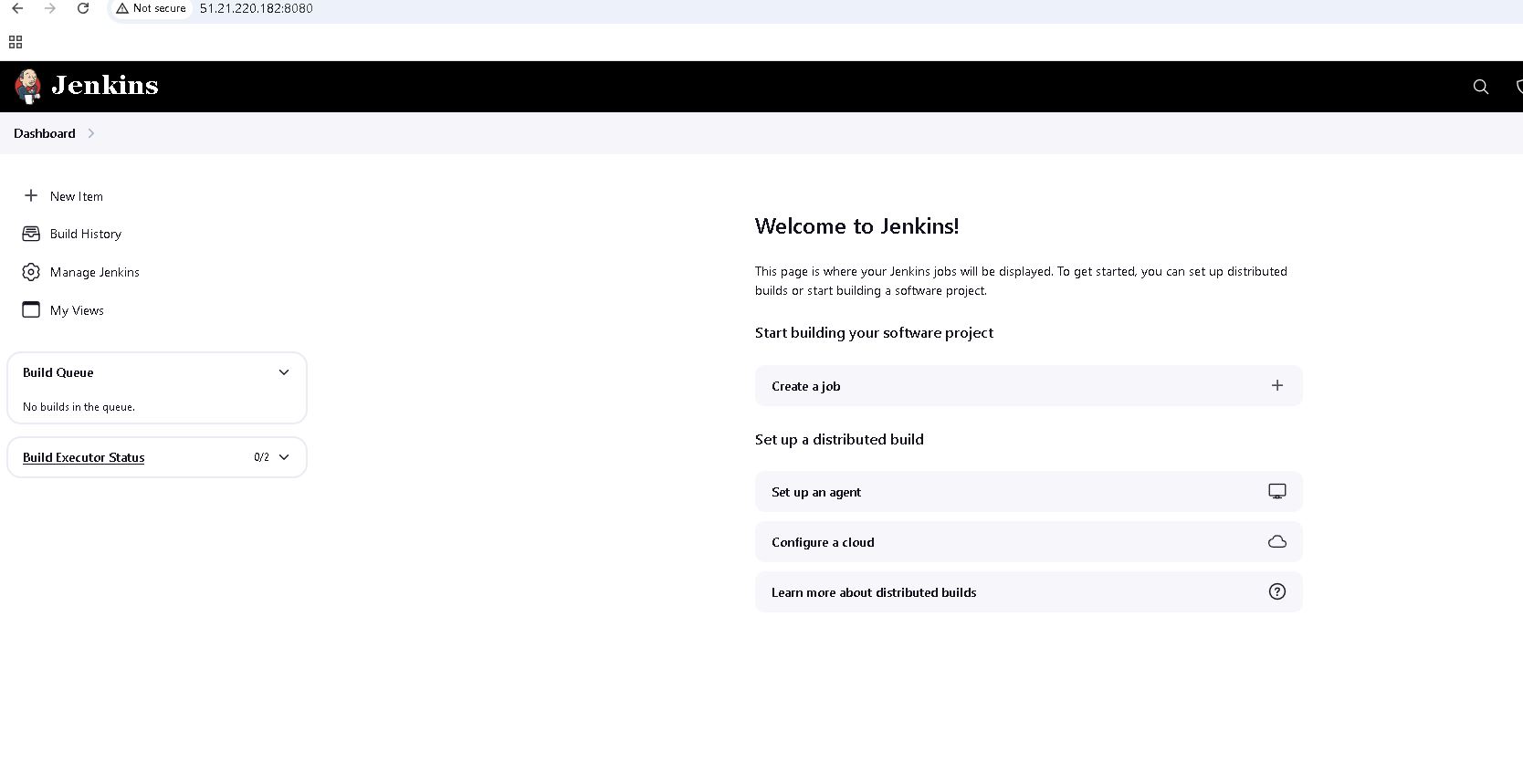
sudo sh -x install.sh

```



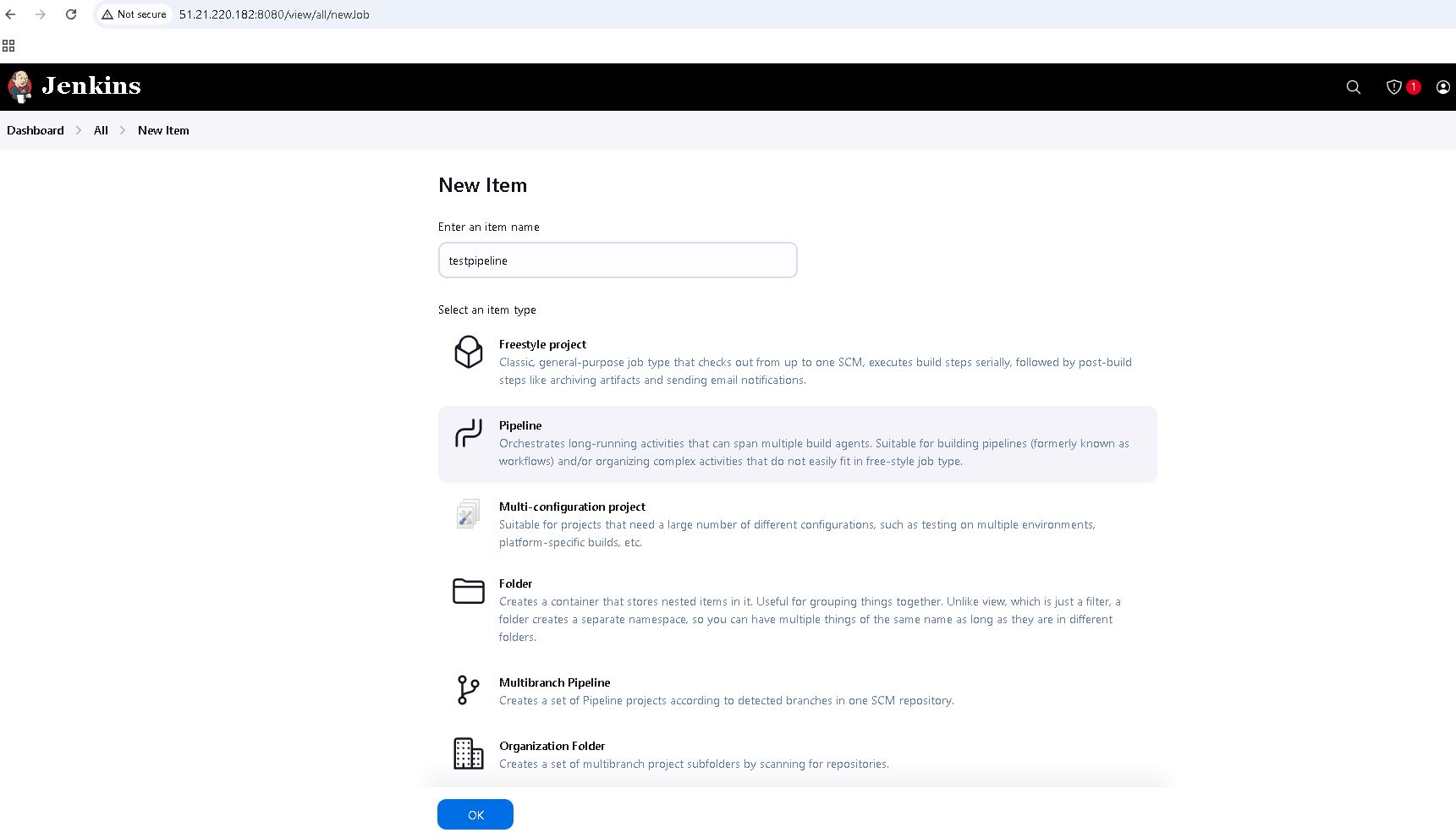
Ensure that port 8080 is open in the security group attached to the instance to allow access to the Jenkins web interface:

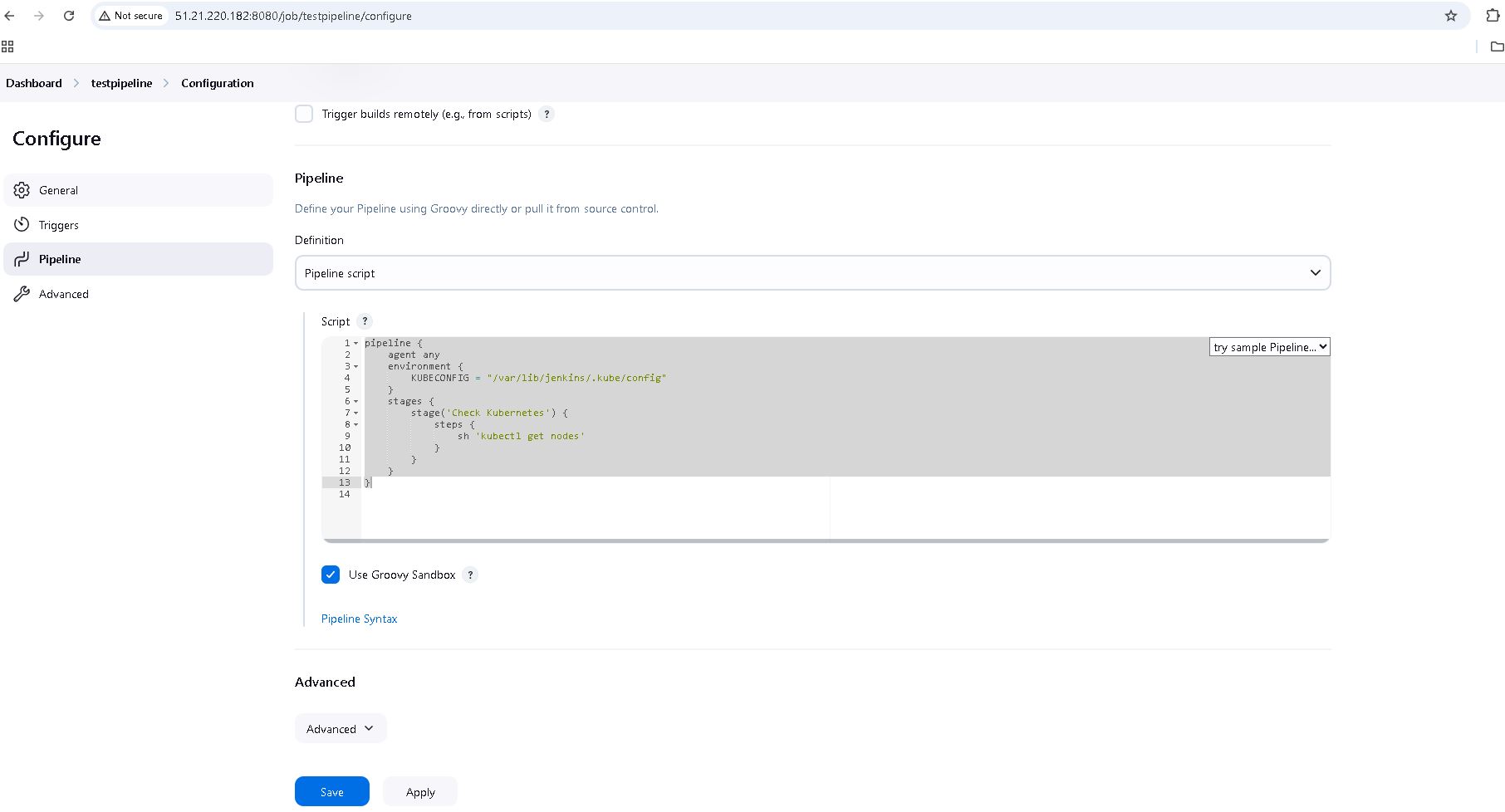
http://<PUBLIC\_IP:8080>



**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'

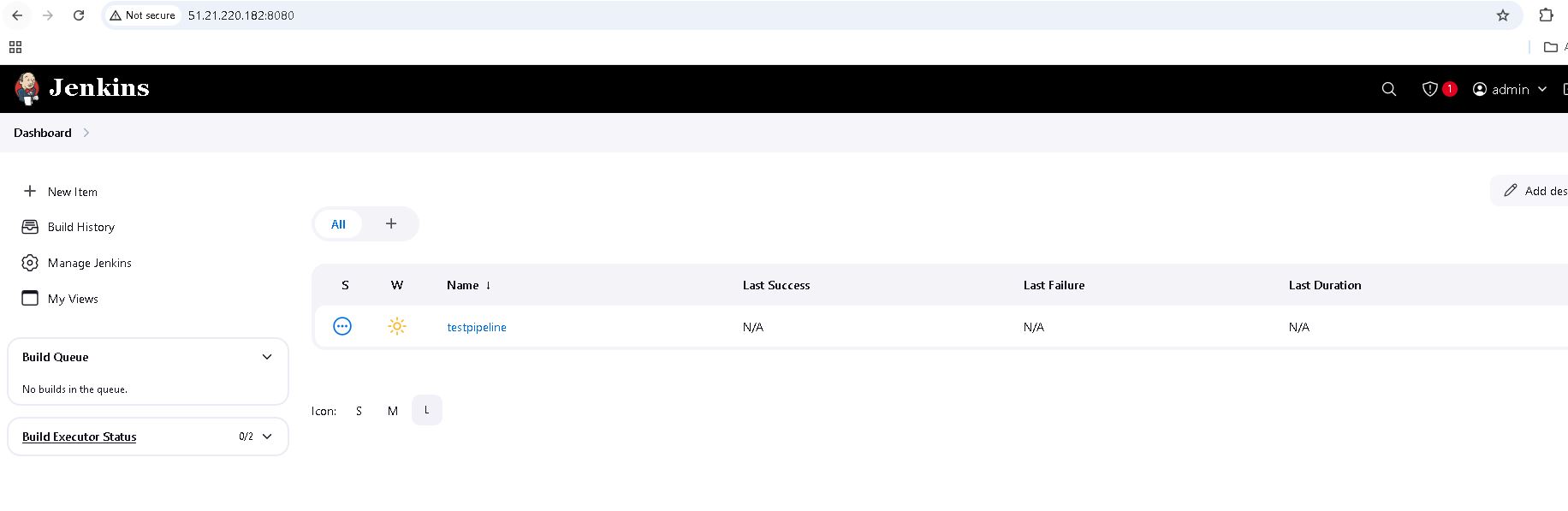
}

}

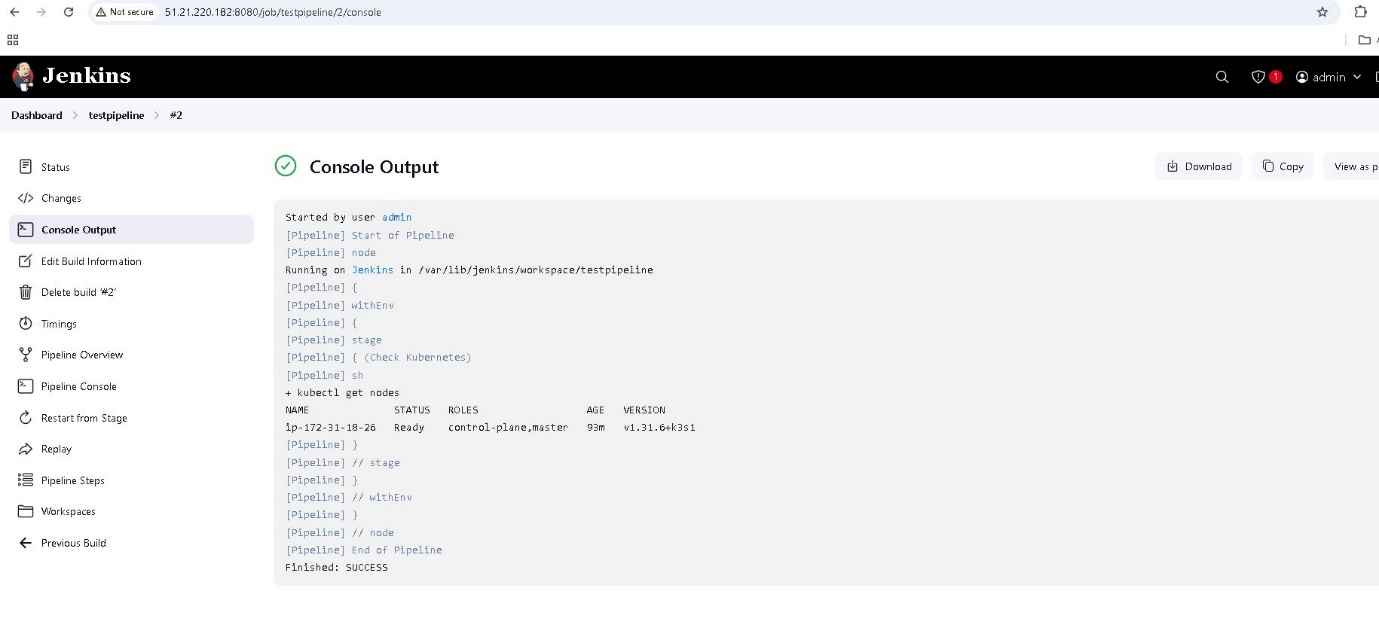
}

}

```



Run the pipeline by selecting **Build Now**.



**8. Configure Kubeconfig in GitHub Actions**

For GitHub Actions, use the following steps to set up and use kubeconfig:

```yaml

- name: Setup Kubeconfig

run: |

echo "$KUBECONFIG\_CONTENT" | base64 --decode > ~/.kube/config

- 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

```