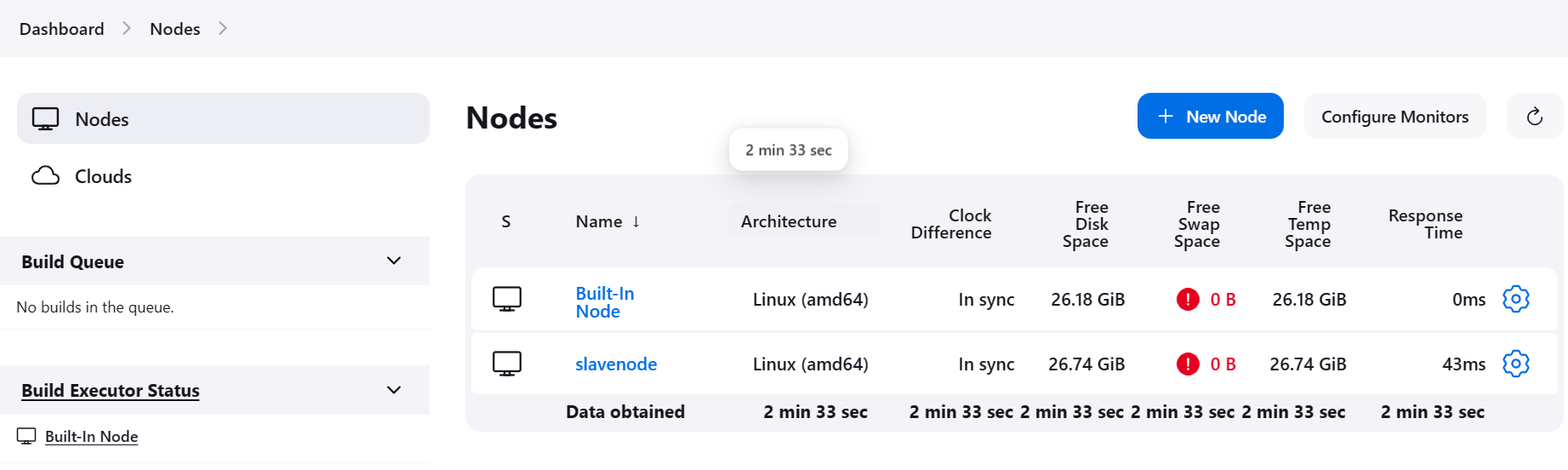
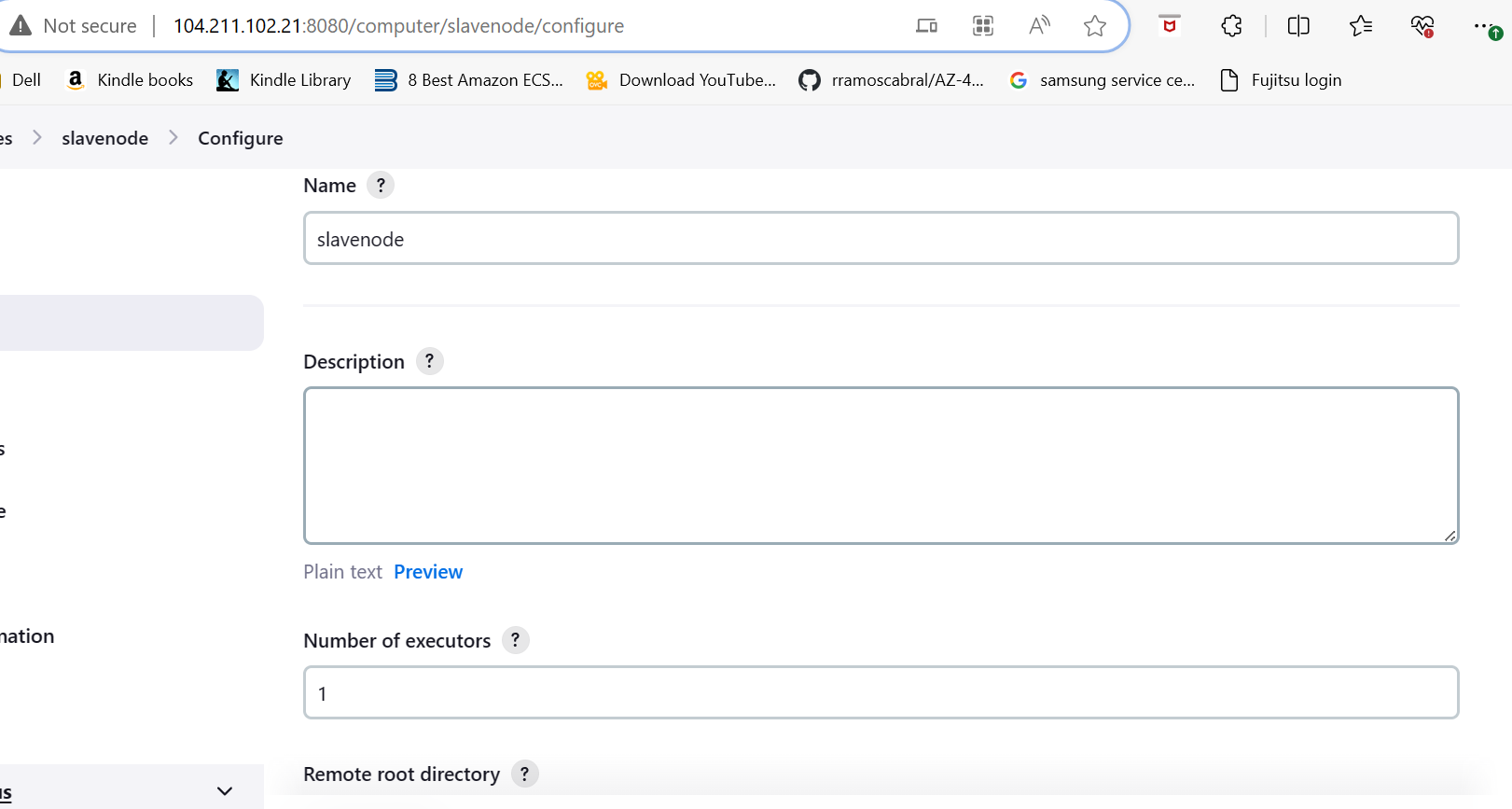


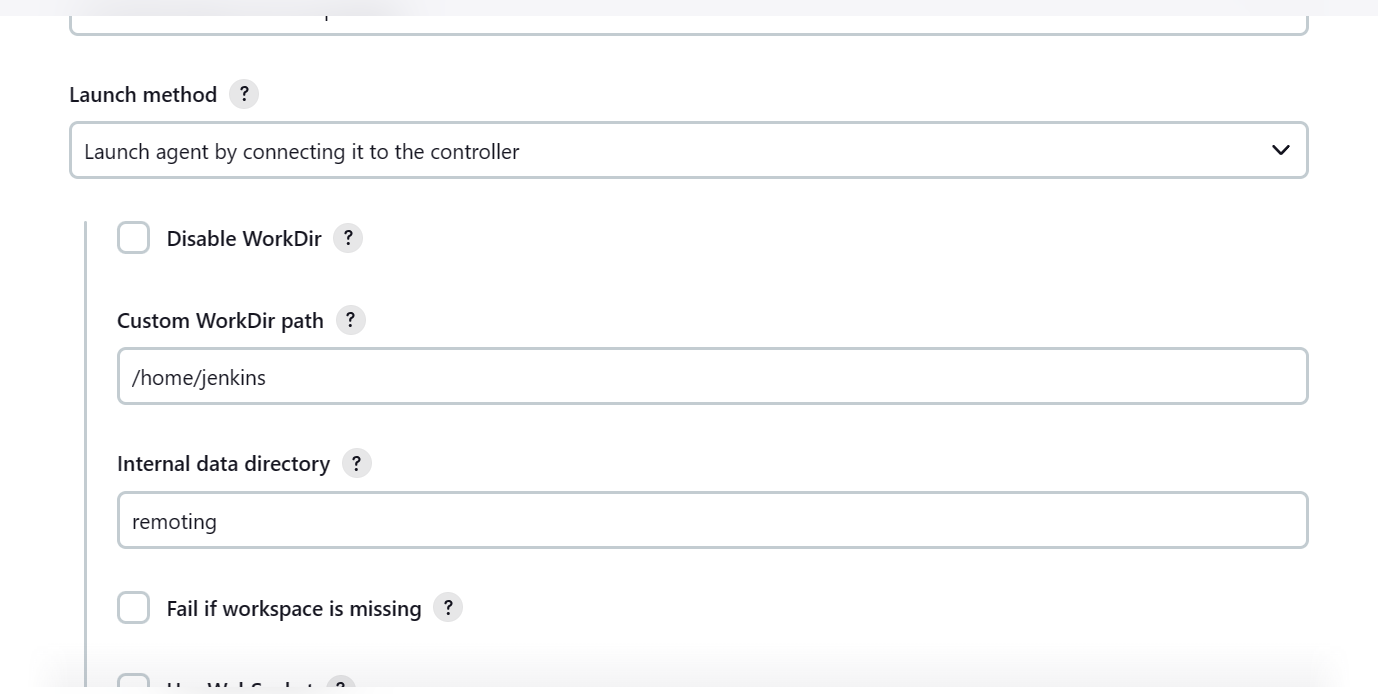
**How to configure the Agent or Slave Node on Jenkins**

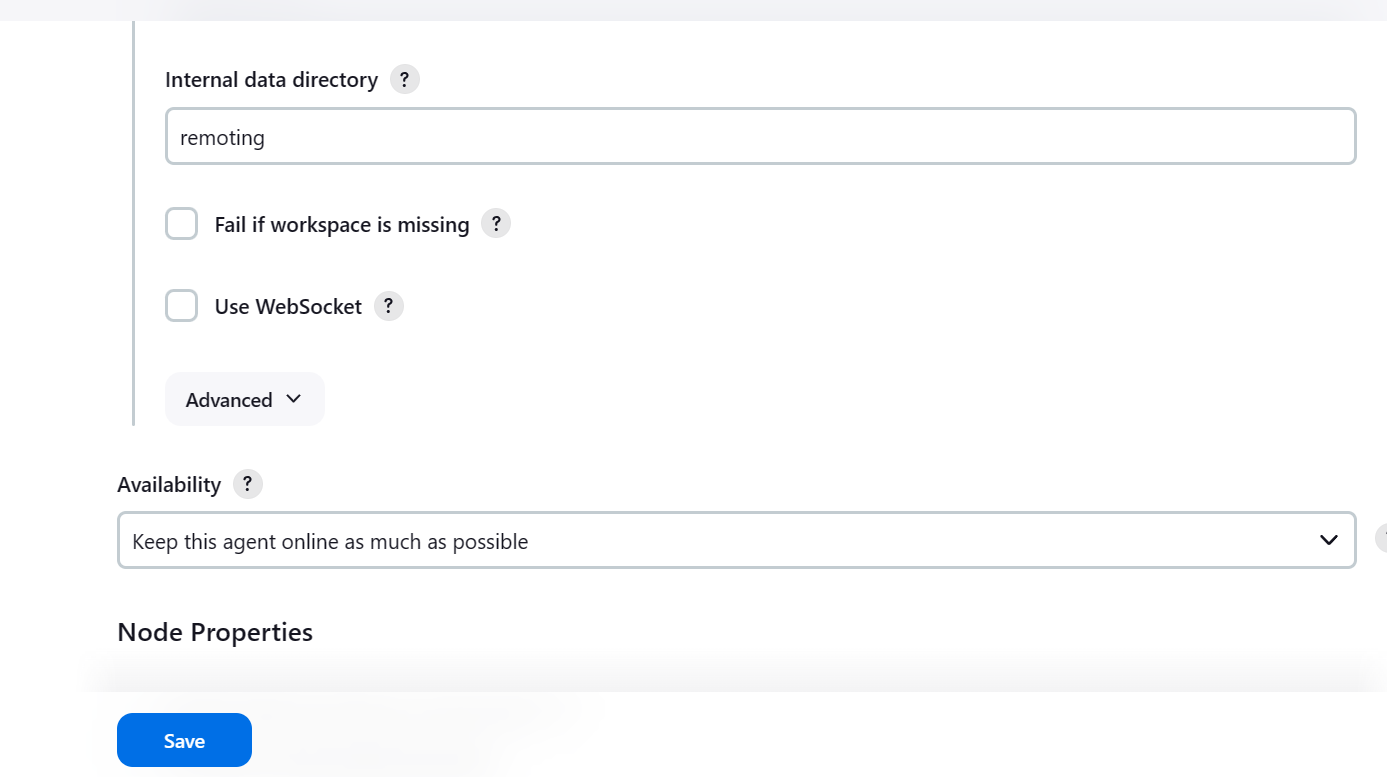
Step 1: Go to Manage Jenkins -> Node ->





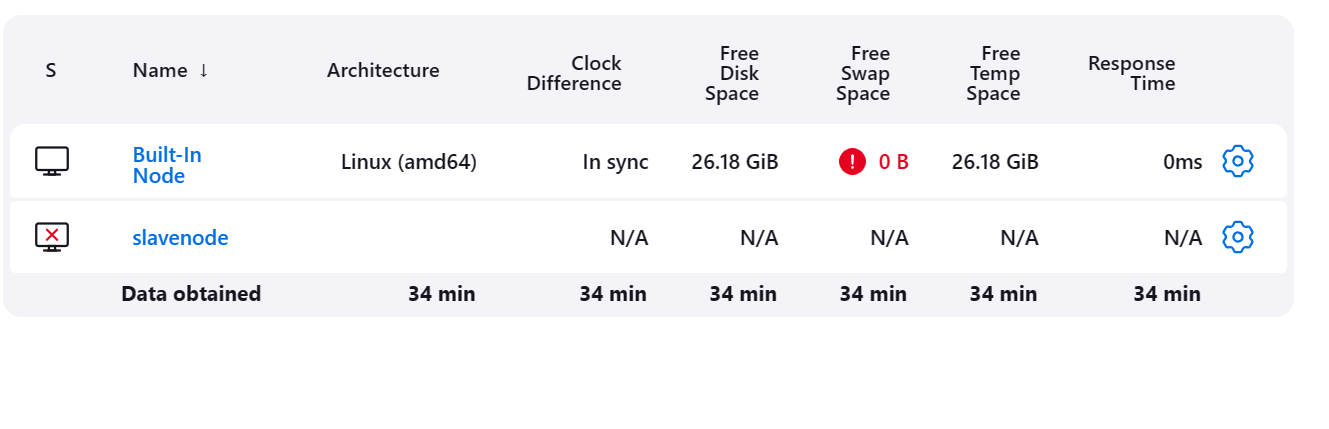


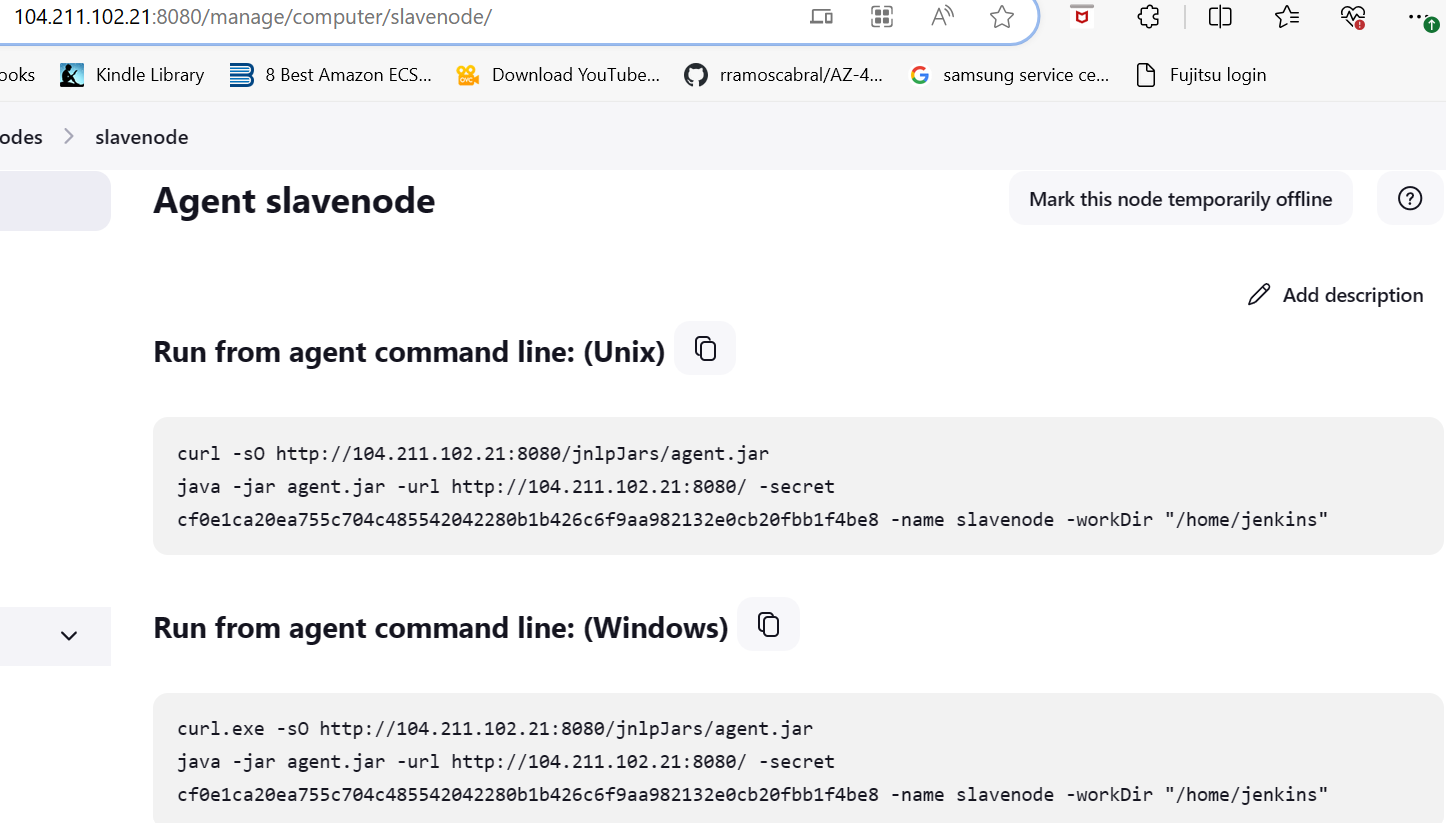




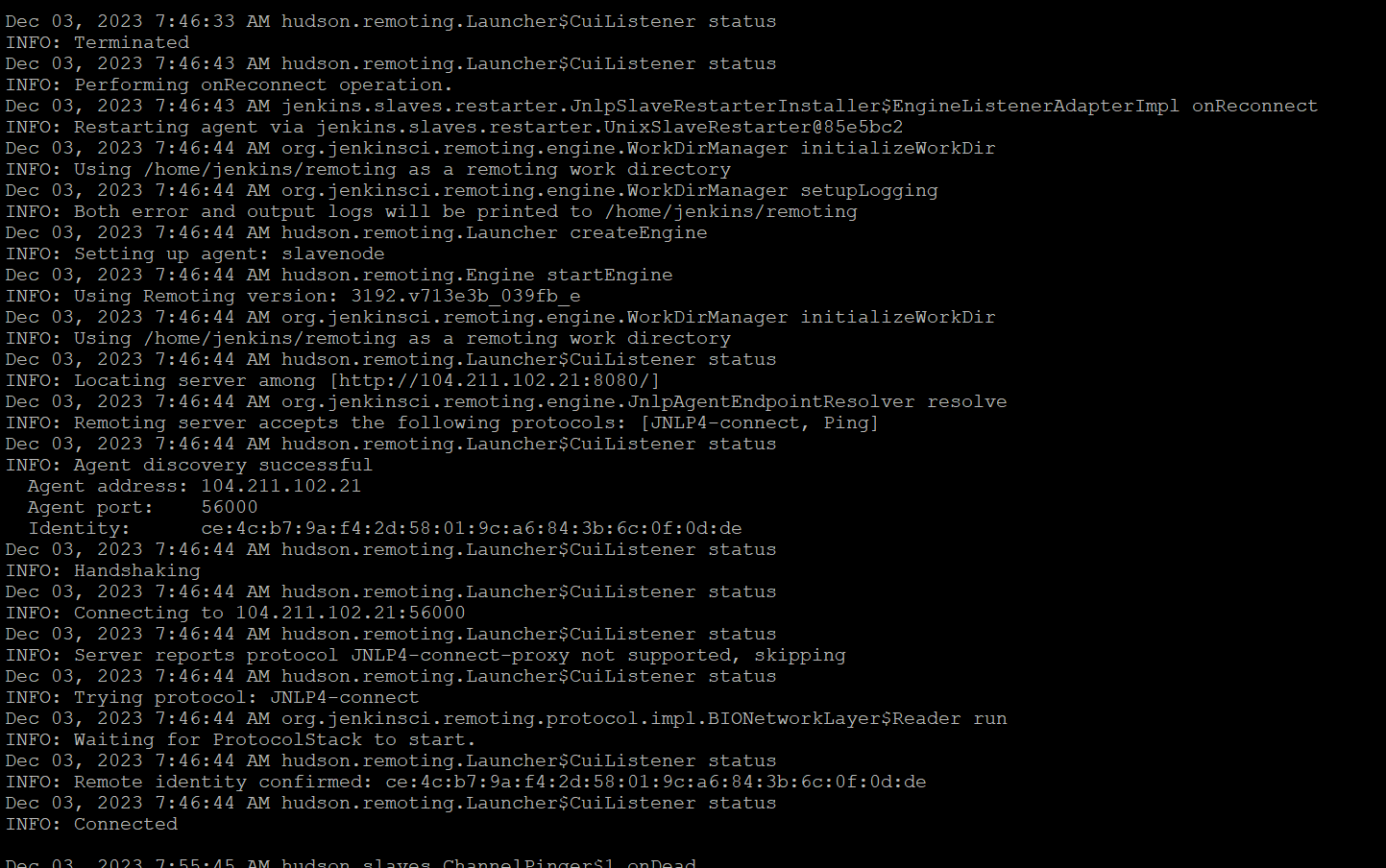
Step 3: Go to Manage -> Security -> Give TCP Port

Step 4: Go to Manage Jenkins -> Node -> Go to Agent Node

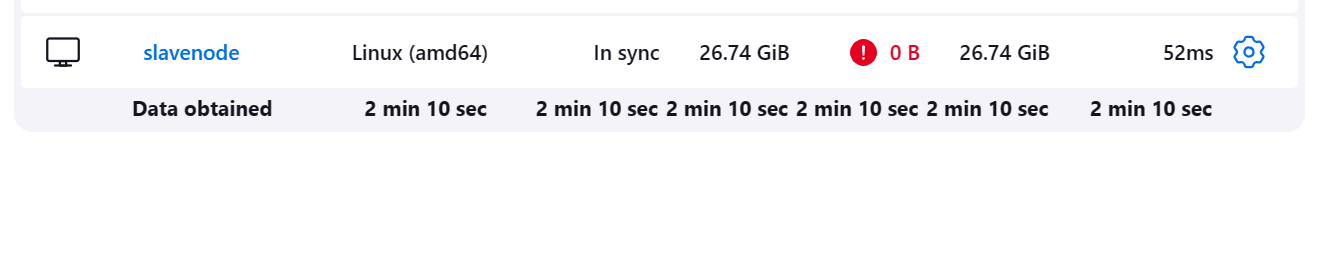




Please run this command on the slave machine (Not in jenkins machine)



You can see the status is Node up cluster :



Step 5: Part of prerequisite , you need to install the dotnetcore and also docker engine on the slave machine , Please Click to this [Docker Installation on UbuntuMachine](https://awstrainingwithjagan089-my.sharepoint.com/personal/jaganrajagopal_awstrainingwithjagan089_onmicrosoft_com/_layouts/15/Doc.aspx?sourcedoc=%7B6F3C5D35-DD2C-4C60-BADE-E5929DB9F0A2%7D&file=Document1.docx&action=default&mobileredirect=true)

**Docker installation setup**

# Add Docker's official GPG key:

sudo apt-get update

sudo apt-get install ca-certificates curl gnupg

sudo install -m 0755 -d /etc/apt/keyrings

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg

sudo chmod a+r /etc/apt/keyrings/docker.gpg

# Add the repository to Apt sources:

echo \

"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \

$(. /etc/os-release && echo "$VERSION\_CODENAME") stable" | \

sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt-get update

sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin

**If you are using Ubuntu 20, you need to download the package from this repo:**

wget <https://packages.microsoft.com/config/ubuntu/20.04/packages-microsoft-prod.deb> -O packages-microsoft-prod.deb sudo dpkg -i packages-microsoft-prod.deb rm packages-microsoft-prod.deb

sudo apt-get update && sudo apt-get install -y dotnet-sdk-6.0

Step 6: Jenkins Pipeline Script :

pipeline {

agent {

label 'slavenodeagent'

}

environment {

DOCKER\_IMAGE\_NAME = "11122233345/dotnetwebapptraining"

DOCKER\_IMAGE\_TAG = "latest"

[dockerHubUser="jaganrajagopalme@gmail.com](mailto:dockerHubUser="jaganrajagopalme@gmail.com)"

dockerHubPassword="xxxx"

}

stages {

stage('Checkout') {

steps {

git changelog: false, credentialsId: 'dotnetjenkins', poll: false, url: '<https://github.com/jaganrajagopal/Jenkinswithdockercomposeup.git>'

}

}

stage('Build') {

steps {

sh 'ls'

sh 'dotnet build Jenkinswithdockercicd.csproj'

}

}

stage('Publish') {

steps {

sh 'dotnet publish Jenkinswithdockercicd.csproj -c Release'

}

}

stage('Build Docker Image') {

steps {

script {

// Build Docker image

sh "docker build -t ${DOCKER\_IMAGE\_NAME}:${DOCKER\_IMAGE\_TAG} ."

}

}

}

stage('Docker Push') {

steps {

script {

sh "docker login -u ${env.dockerHubUser} -p ${env.dockerHubPassword}"

sh 'docker push ${DOCKER\_IMAGE\_NAME}:${DOCKER\_IMAGE\_TAG}'

}

}

}

//just deploying the application on the website

stage('Docker Website'){

steps {

script {

sh "docker run --name dotnetapps -p 80:80 -d ${DOCKER\_IMAGE\_NAME}:${DOCKER\_IMAGE\_TAG}"

}

}

}

}

}