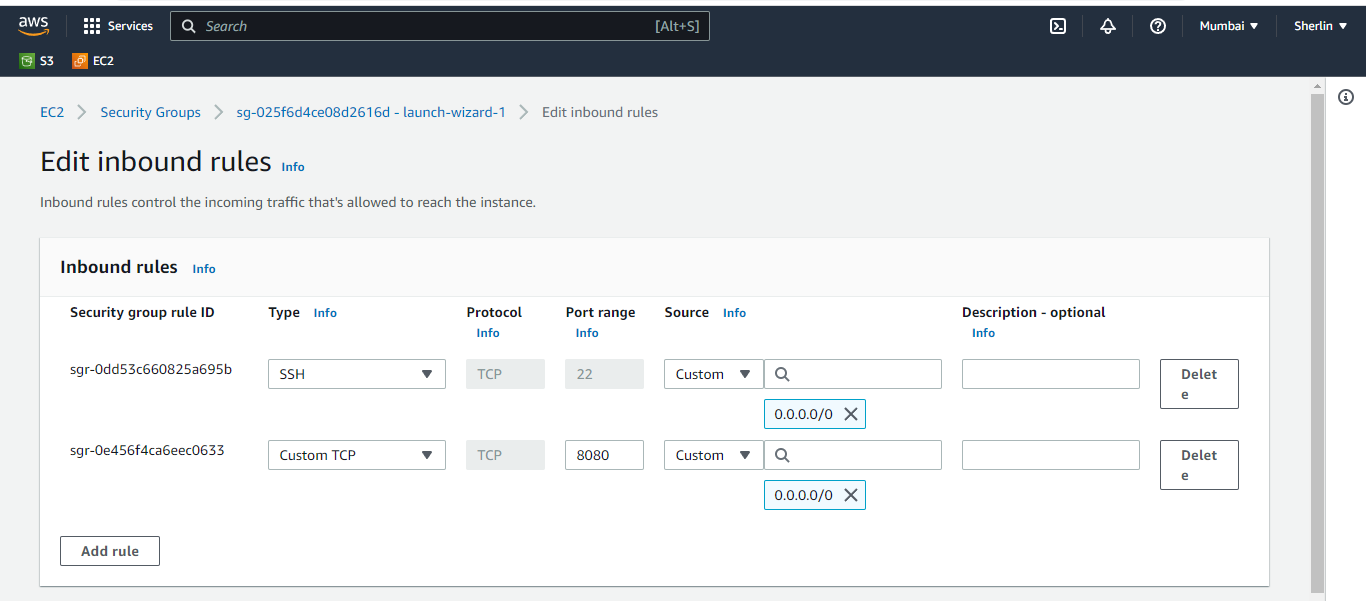
Pipeline Job to build and deploy docker image

1. CI/CD Server

Graphical user interface, text, application

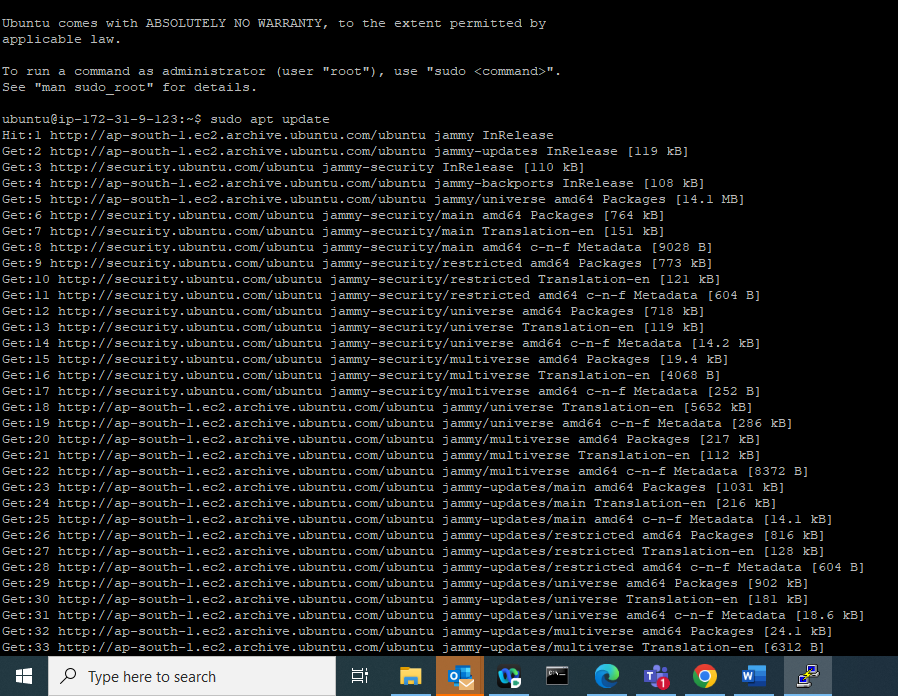
Description automatically generated



In CI/CD server

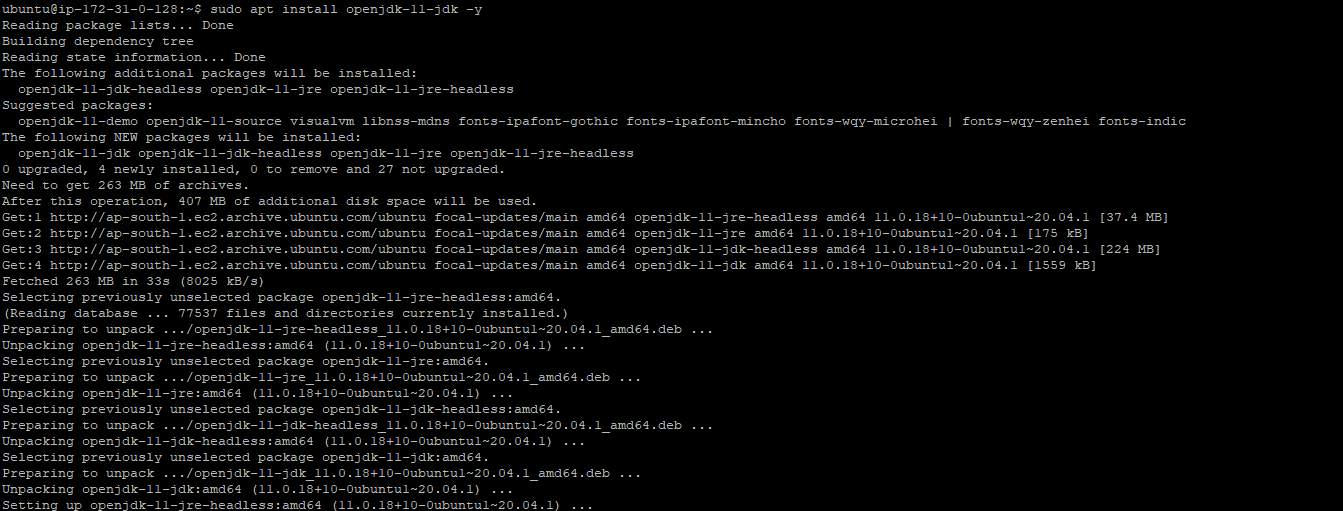
A] Update package Manager

sudo apt update



b] Install Java

sudo apt install openjdk-11-jdk -y



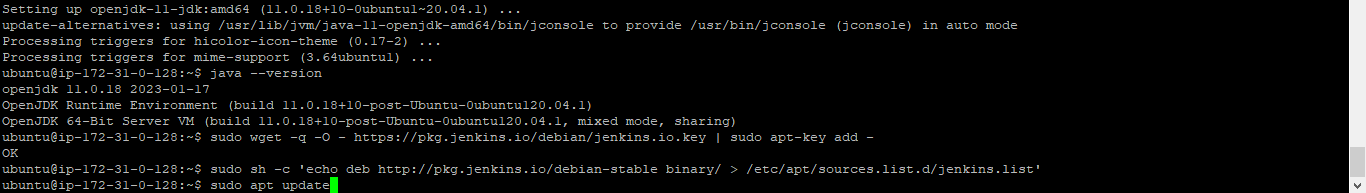
Text

Description automatically generated

c] Install Jenkins

sudo wget -q -O – <https://pkg.jenkins.io/debian/jenkins.io.key> | sudo apt-key add –

sudo sh -c ‘echo deb [https://pkg.jenkins.io/debian-stable binary/](https://pkg.jenkins.io/debian-stable%20binary/) > /etc/apt/sources.list.d/Jenkins.list’



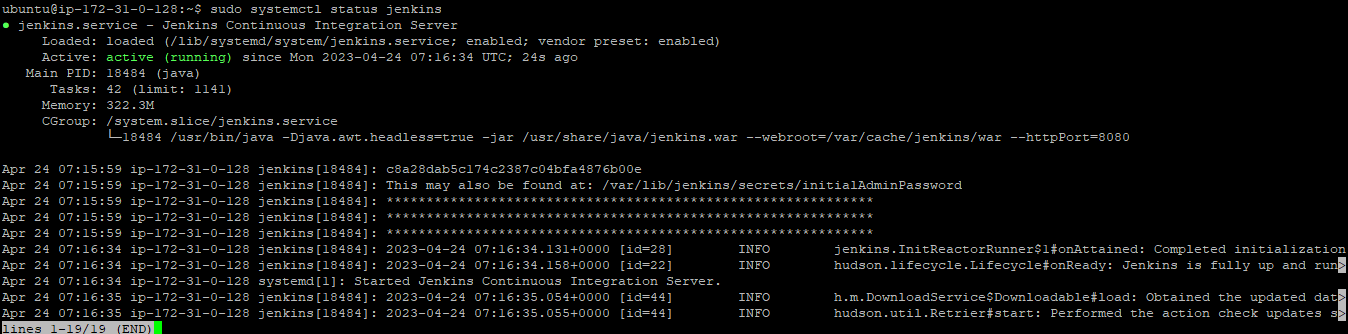
sudo apt update

sudo apt install Jenkins

Text

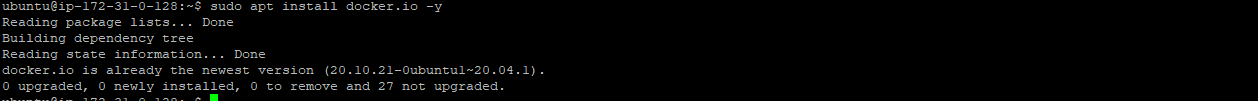
Description automatically generated

sudo systemctl status Jenkins



d] Install Docker

sudo apt install docker.io -y



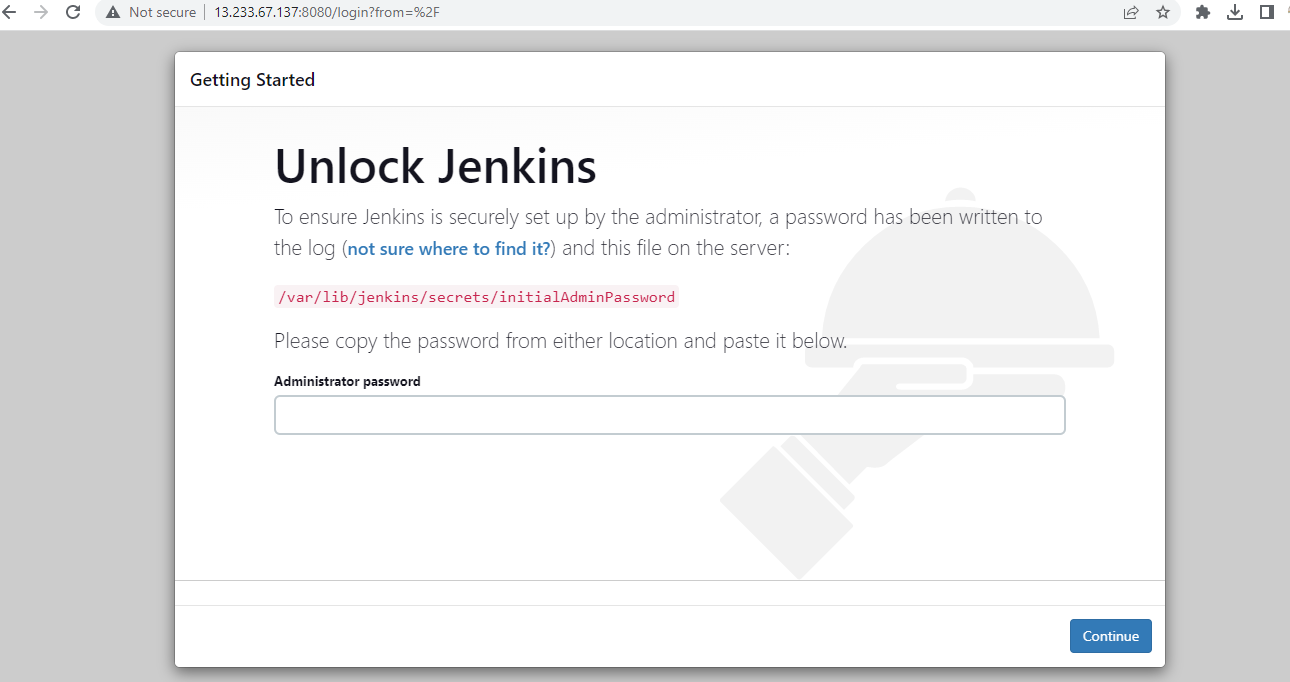
e] Add Jenkins user to docker

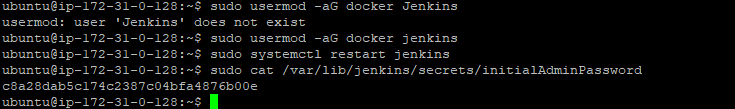
sudo usermod -aG docker jenkins

f] Restart Jenkins

sudo systemctl restart jenkins







Graphical user interface, text, application

Description automatically generated

Install suggested Plugins

Graphical user interface, application, table, Word

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, application, Teams

Description automatically generated

Github



Graphical user interface, application, Teams

Description automatically generated

Text

Description automatically generated with medium confidence

Graphical user interface, text, application

Description automatically generated

node {

//def mvnHome

stage('Git Clone') { // for display purposes

// Get some code from a GitHub repository

git url: 'https://github.com/shearusherly/Java-Jenkins-Docker.git', branch: 'master'

}

stage('Maven CLean Package') {

// Run the maven build

def mavenHome= tool name: "Maven", type: "maven"

sh "${mavenHome}/bin/mvn clean package"

}

}

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

node {

//def buildNumber = BUILD\_NUMBER

stage('Git Clone') { // for display purposes

// Get some code from a GitHub repository

git url: 'https://github.com/shearusherly/Java-Jenkins-Docker.git', branch: 'master'

}

stage('Maven CLean Package') {

// Run the maven build

def mavenHome= tool name: "Maven", type: "maven"

sh "${mavenHome}/bin/mvn clean package"

}

stage('Build Docker Image'){

sh 'docker build -t sherlinsharmi97/java-web-app-docker-demo .'

}

}

Graphical user interface, text, application, email

Description automatically generated

Text

Description automatically generated with medium confidence

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

A computer screen capture

Description automatically generated with medium confidence

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

node {

//def buildNumber = BUILD\_NUMBER

stage('Git Clone') { // for display purposes

// Get some code from a GitHub repository

git url: 'https://github.com/shearusherly/Java-Jenkins-Docker.git', branch: 'master'

}

stage('Maven CLean Package') {

// Run the maven build

def mavenHome= tool name: "Maven", type: "maven"

sh "${mavenHome}/bin/mvn clean package"

}

stage('Build Docker Image'){

sh 'docker build -t sherlinsharmi97/java-web-app-docker-demo .'

}

stage('Docker Login and Push Docker Image'){

withCredentials([string(credentialsId: 'DockerHubPwd', variable: 'DockerHubPwd')]) {

sh "docker login -u sherlinsharmi97 -p ${DockerHubPwd}"

}

sh 'docker push sherlinsharmi97/java-web-app-docker-demo'

}

}

Graphical user interface, application, table

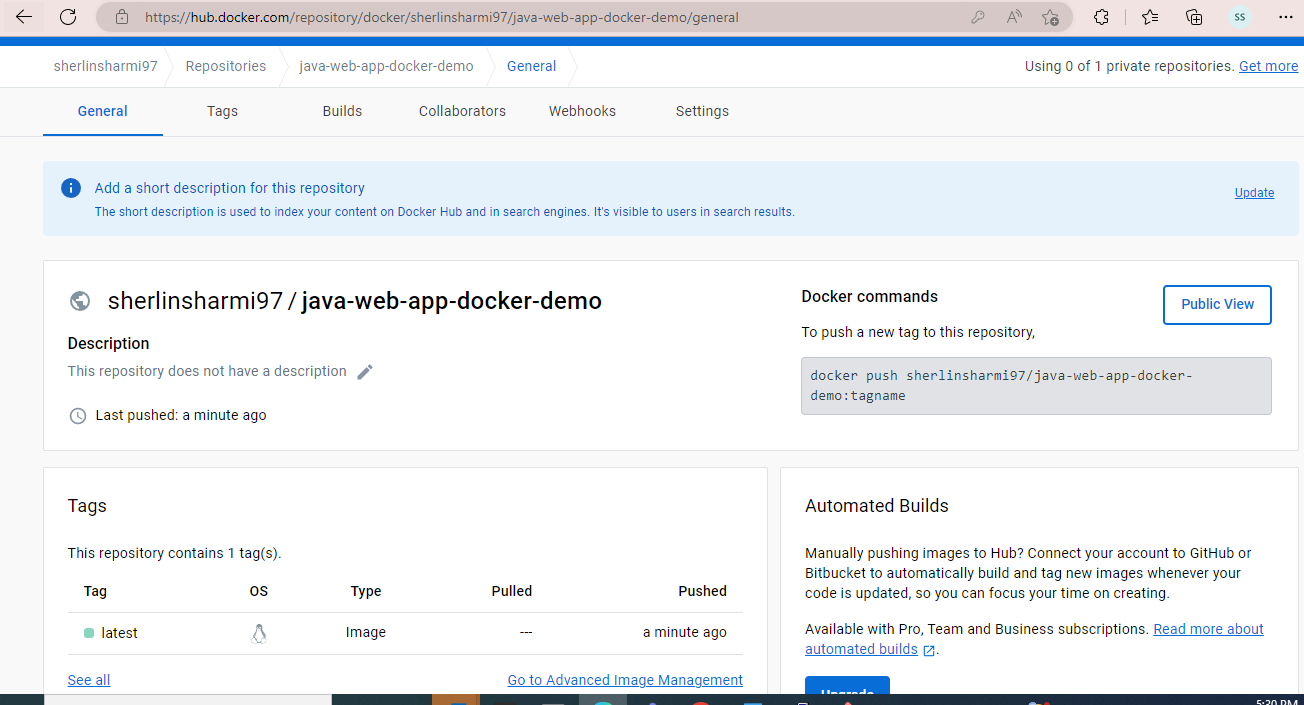
Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, table

Description automatically generated with medium confidence



Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

node {

//def buildNumber = BUILD\_NUMBER

stage('Git Clone') { // for display purposes

// Get some code from a GitHub repository

git url: 'https://github.com/shearusherly/Java-Jenkins-Docker.git', branch: 'master'

}

stage('Maven CLean Package') {

// Run the maven build

def mavenHome= tool name: "Maven", type: "maven"

sh "${mavenHome}/bin/mvn clean package"

}

stage('Build Docker Image'){

sh 'docker build -t sherlinsharmi97/java-web-app-docker-demo .'

}

stage('Docker Login and Push Docker Image'){

withCredentials([string(credentialsId: 'DockerHubPwd', variable: 'DockerHubPwd')]) {

sh "docker login -u sherlinsharmi97 -p ${DockerHubPwd}"

}

sh 'docker push sherlinsharmi97/java-web-app-docker-demo'

}

stage('Deploy Application as Docker Container in docker deployment server'){

sshagent(['Docker\_Dev\_Server\_SSH\_U']) {

sh 'ssh -o StrictHostKeyChecking=no ubuntu@172.31.14.121 docker rm -f javawebappcontainer || true'

sh "ssh -o StrictHostKeyChecking=no ubuntu@172.31.14.121 docker run -d -p 8080:8080 --name javawebappcontainer sherlinsharmi97/java-web-app-docker-demo"

}

}

}

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

Text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application, table

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

1. Deployment server

Graphical user interface, text, application, email

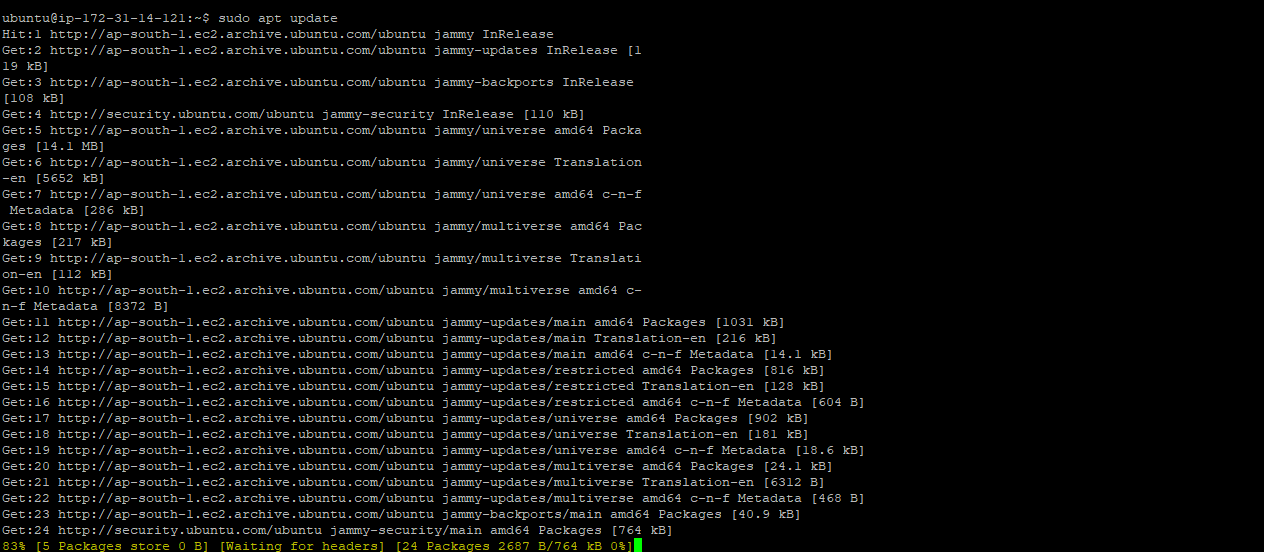
Description automatically generated

Graphical user interface, application

Description automatically generated

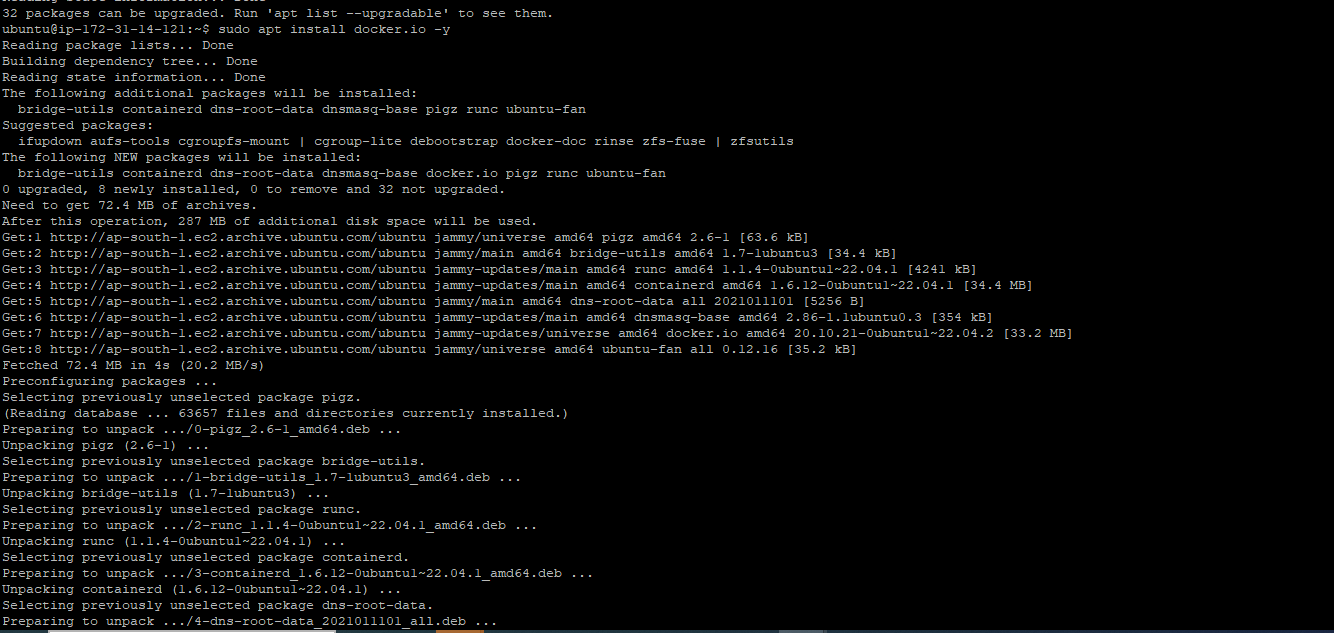
A] Update Package Manager

sudo apt update



B] Install docker

sudo apt install docker.io -y

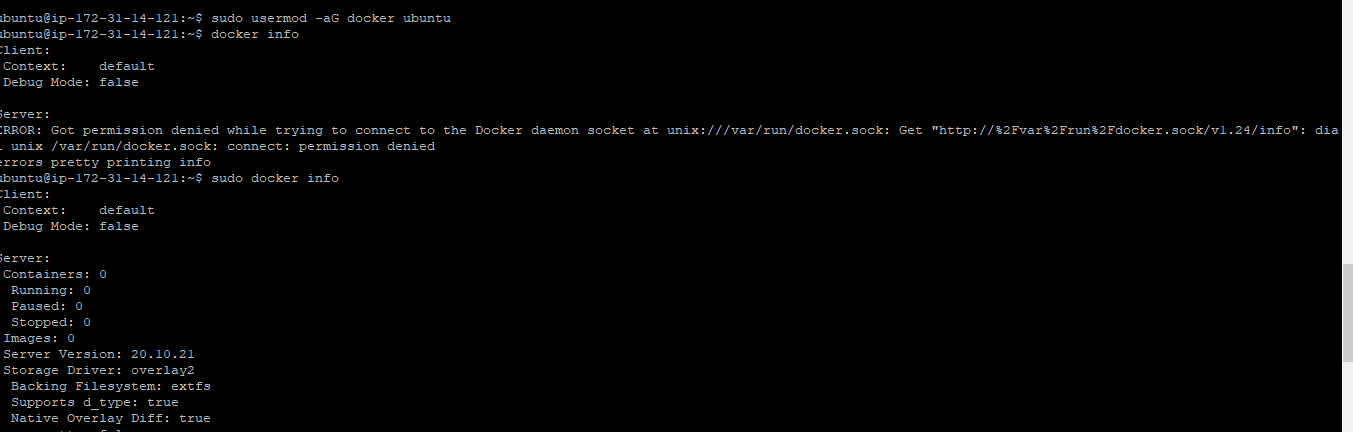


C] Add ubuntu user to docker

sudo usermod -aG docker ubuntu



D] docker info



Graphical user interface, text

Description automatically generated

After Commit

A screenshot of a computer

Description automatically generated

Graphical user interface, application, table

Description automatically generated

Table

Description automatically generated with medium confidence

Push Docker image to ECR

node {

//def buildNumber = BUILD\_NUMBER

stage('Git Clone') { // for display purposes

// Get some code from a GitHub repository

git url: 'https://github.com/shearusherly/Java-Jenkins-Docker.git', branch: 'master'

}

stage('Maven CLean Package') {

// Run the maven build

def mavenHome= tool name: "Maven", type: "maven"

sh "${mavenHome}/bin/mvn clean package"

}

stage('Maven Test')

{

def mvnHome= tool name: "Maven", type: "maven"

sh "${mvnHome}/bin/mvn test"

}

stage('Build Docker Image'){

sh 'docker build -t sherlinsharmi97/java\_web\_app\_docker\_demo .'

}

stage('Docker Login and Push Docker Image'){

withCredentials([string(credentialsId: 'DockerHubPwd', variable: 'DockerHubPwd')]) {

sh "docker login -u sherlinsharmi97 -p ${DockerHubPwd}"

}

sh 'docker push sherlinsharmi97/java\_web\_app\_docker\_demo'

}

stage('Pushing to ECR') {

sh 'aws ecr get-login-password --region ap-south-1 | docker login --username AWS --password-stdin 487838928633.dkr.ecr.ap-south-1.amazonaws.com'

sh 'docker build -t java\_web\_app\_docker\_demo .'

sh 'docker tag java\_web\_app\_docker\_demo:latest 487838928633.dkr.ecr.ap-south-1.amazonaws.com/java\_web\_app\_docker\_demo:latest'

sh 'docker push 487838928633.dkr.ecr.ap-south-1.amazonaws.com/java\_web\_app\_docker\_demo:latest'

}

stage('Deploy Application as Docker Container in docker deployment server'){

sshagent(['Docker\_Dev\_Server\_SSH\_U']) {

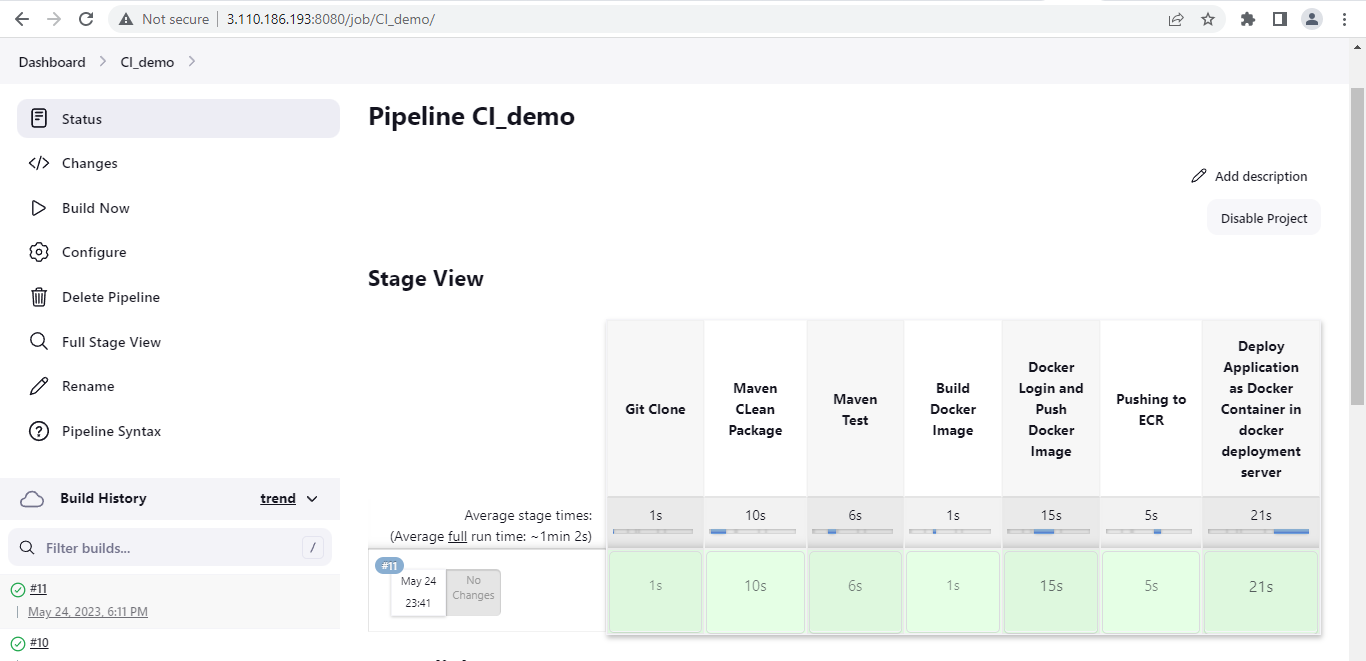
sh 'ssh -o StrictHostKeyChecking=no ubuntu@172.31.39.131 docker rm -f javawebappcontainer || true'

sh "ssh -o StrictHostKeyChecking=no ubuntu@172.31.39.131 docker run -d -p 8080:8080 --name javawebappcontainer sherlinsharmi97/java\_web\_app\_docker\_demo"

}

}

}



A screenshot of a computer

Description automatically generated with medium confidence

7. Our Web app Deployed into the Deployment Server

Graphical user interface, text

Description automatically generated