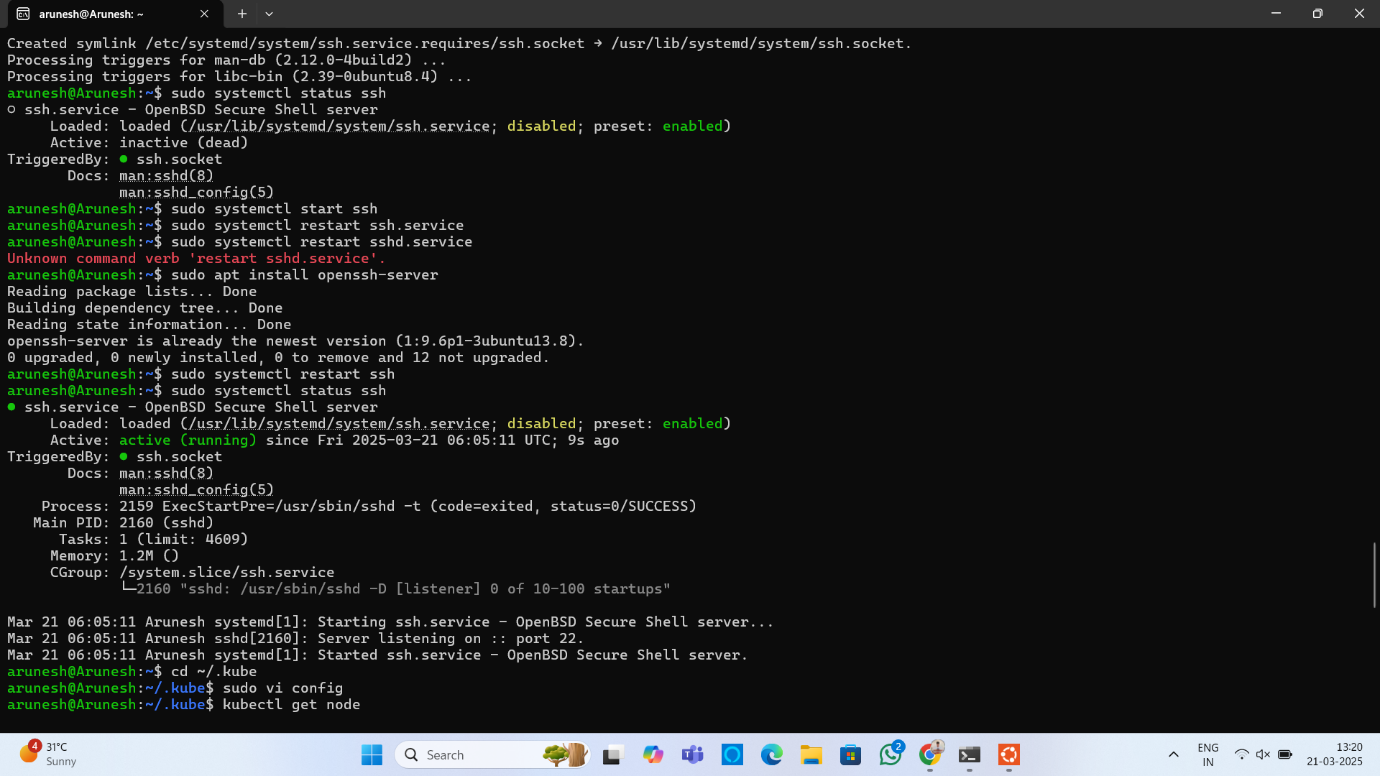
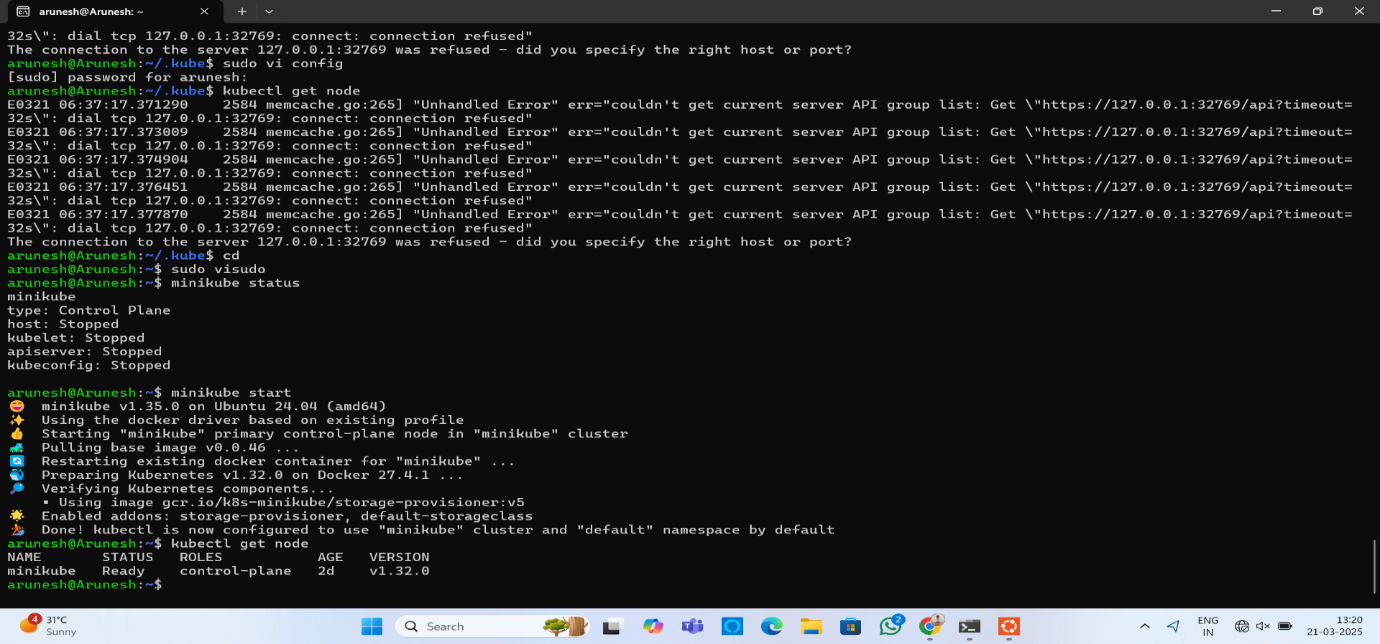
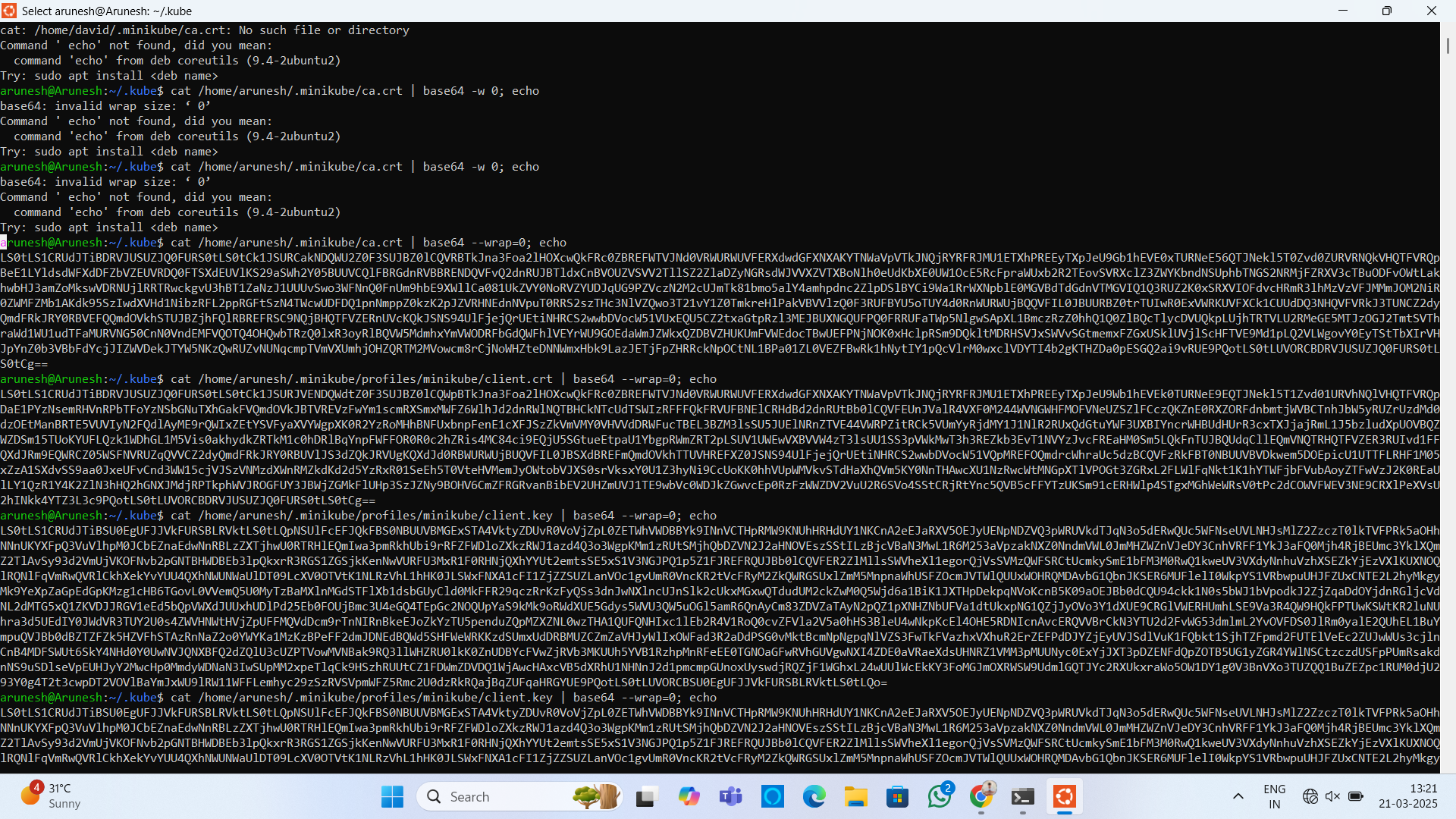
**SSH Service Management**

* sudo systemctl restart ssh.service
* sudo systemctl restart sshd.service
* sudo apt update
* sudo apt install openssh-server
* sudo systemctl restart ssh
* sudo systemct status ssh
* sudo chmod 666 /var/run/docker.sock





**Javaapplication**

Pipeline Script

pipeline {

agent any

stages {

stage('scm') {

steps {

git branch: 'main', url: 'https://github.com/Arunesh155/DevOps.git'

}

}

stage('build') {

steps {

sh "mvn clean"

sh "mvn install"

}

}

stage('build to images') {

steps {

script{

sh 'docker build -t arunesh2005/simpleapplication .'

}}}

stage('push to hub') {

steps {

script{

withDockerRegistry(credentialsId: 'arun', url: 'https://index.docker.io/v1/') {

sh 'docker push arunesh2005/simpleapplication'

}

}

}

}

stage('Deploy App') {

steps {

withKubeConfig(caCertificate: '', clusterName: 'minikube', contextName: 'minikube', credentialsId: 'myminikube', namespace: '', restrictKubeConfigAccess: false, serverUrl: 'https://192.168.39.226:8443'){

sh 'kubectl apply -f deployment.yml --validate=false'

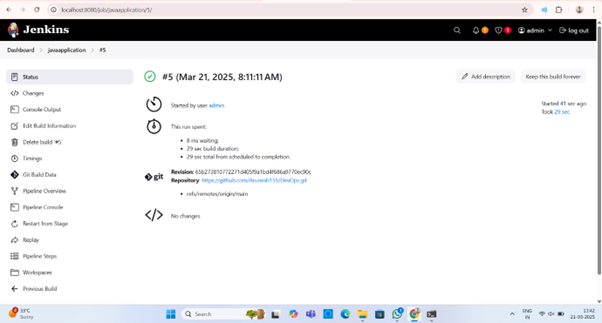
}

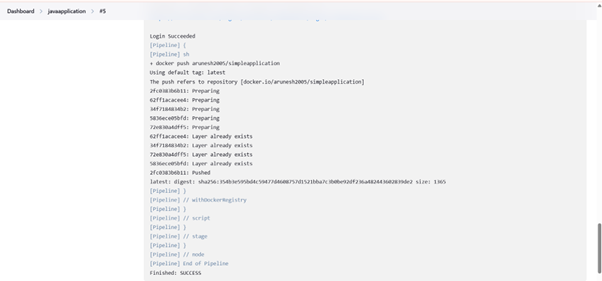
}

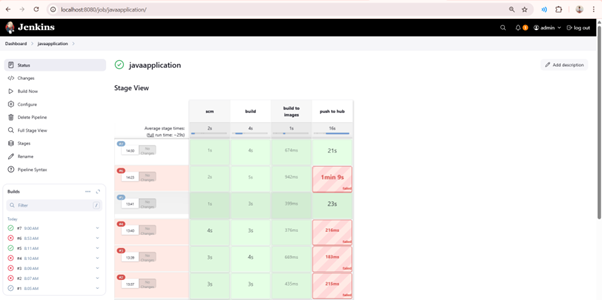
}

}

}







**File: Deploy.yaml**

apiVersion: apps/v1

kind: Deployment

metadata:

name: my-deploy

labels:

name: my-deploy

spec:

replicas: 5

selector:

matchLabels:

apptype: web-backend

strategy:

type: RollingUpdate

template:

metadata:

labels:

apptype: web-backend

spec:

containers:

- name: my-app

image: arunesh2005/simpleapplication

ports:

- containerPort: 9000

apiVersion: v1

kind: Service

metadata:

name: my-service8

labels:

app: my-service

spec:

type: NodePort

ports:

- port: 9000

targetPort: 8080

nodePort: 30002

selector:

apptype: web-backend