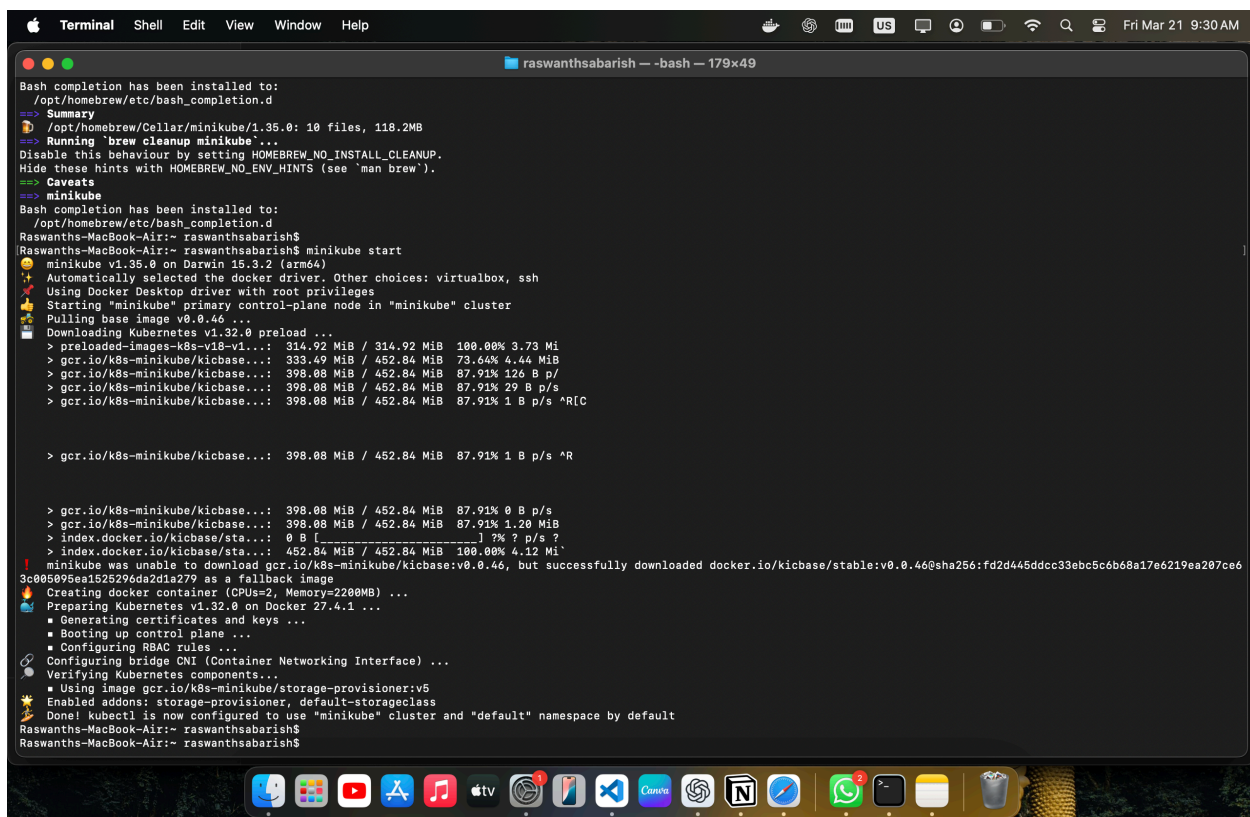


Devops class guvi (DAY-3)

19 March 2025 (DAY-3)

Install Minikube- on macOS using Homebrew:



```
Terminal Shell Edit View Window Help
raswanthasabarish - bash - 179x49

Bash completion has been installed to:
/opt/homebrew/etc/bash_completion.d

=> Summary
  /opt/homebrew/Cellar/minikube/1.35.0: 10 files, 118.2MB
=> Running 'brew cleanup minikube'...
Disable this behaviour by setting HOMEBREW_NO_INSTALL_CLEANUP.
Hide these hints with HOMEBREW_NO_ENV_HINTS (see 'man brew').
=> Caveats
=> minikube
Bash completion has been installed to:
/opt/homebrew/etc/bash_completion.d
Raswanths-MacBook-Air:~ raswanthasabarish$ minikube start
Raswanths-MacBook-Air:~ raswanthasabarish$ minikube start
minikube v1.35.0 on Darwin 15.3.2 (arm64)
🔧 Automatically selected the docker driver. Other choices: virtualbox, ssh
🔧 Using Docker Desktop driver with root privileges
🔧 Starting "minikube" primary control-plane node in "minikube" cluster
🔧 Pulling base image v0.0.46 ...
🔧 Downloading Kubernetes v1.32.0 preload ...
> preloaded-images-k8s-v18-v1...: 314.92 MiB / 314.92 MiB 100.00% 3.73 MiB/s
> gcr.io/k8s-minikube/kicbase...: 333.49 MiB / 452.84 MiB 73.64% 4.44 MiB/s
> gcr.io/k8s-minikube/kicbase...: 398.08 MiB / 452.84 MiB 87.91% 126 B/s
> gcr.io/k8s-minikube/kicbase...: 398.08 MiB / 452.84 MiB 87.91% 29 B/s
> gcr.io/k8s-minikube/kicbase...: 398.08 MiB / 452.84 MiB 87.91% 1 B/s ^R[C

> gcr.io/k8s-minikube/kicbase...: 398.08 MiB / 452.84 MiB 87.91% 1 B/s ^R

> gcr.io/k8s-minikube/kicbase...: 398.08 MiB / 452.84 MiB 87.91% 0 B/s
> gcr.io/k8s-minikube/kicbase...: 398.08 MiB / 452.84 MiB 87.91% 1.20 MiB/s
> index.docker.io/kicbase/sta...: 0 B [ ] 0% ? p/s ?
> index.docker.io/kicbase/sta...: 452.84 MiB / 452.84 MiB 100.00% 4.12 MiB/s
! minikube was unable to download gcr.io/k8s-minikube/kicbase:v0.0.46, but successfully downloaded docker.io/kicbase/stable:v0.0.46@sha256:fd2d445ddcc33ebc5c6b68a17e6219ea207ce63c005095ea1525296da2d1a279 as a fallback image
🔧 Creating docker container (CPUs=2, Memory=2200MB) ...
🔧 Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
  ▪ Generating certificates and keys ...
  ▪ Booting up control plane ...
  ▪ Configuring RBAC rules ...
🔧 Configuring bridge CNI (Container Networking Interface) ...
🔧 Verifying Kubernetes components...
  ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
Enabled addons: storage-provisioner, default-storageclass
🌟 Done! kubect1 is now configured to use "minikube" cluster and "default" namespace by default
Raswanths-MacBook-Air:~ raswanthasabarish$
Raswanths-MacBook-Air:~ raswanthasabarish$
```

WSL tool. (only for windows)

Installing java- `sudo apt install fontconfig openjdk-17-jre java -version`

Installing Jenkins on Ubuntu/Debian

- Follow the official Jenkins installation guid [Jenkins Installation Guide](#)
- Restart and check Jenkins service status

```
sudo service jenkins restart
```

```
sudo service jenkins status
```

Installing Docker

```
sudo apt install docker.io -y
```

```
sudo service docker restart
```

```
sudo service docker status
```

1. Add user to the Docker group
 - a.

```
sudo usermod -aG docker $USER
```
2. Check Docker images and running containers.
 - a.

```
sudo chmod 666 /var/run/docker.sock
```

Installing Kubernetes (kubectl)

Download and install kubectl.

```
curl -LO https://dl.k8s.io/release/v1.32.0/bin/linux/amd64/kubectl
```

```
sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
```

```
chmod +x kubectl
```

```
mkdir -p ~/.local/bin
```

```
mv ./kubectl ~/.local/bin/kubectl
```

```
kubectl version --client
```

Installing Minikube (Kubernetes)

Download and install Minikube

```
curl -LO
```

```
https://github.com/kubernetes/minikube/releases/latest/download/minikube-linux-amd64
```

```
sudo install minikube-linux-amd64 /usr/local/bin/minikube && rm minikube-linux-amd64
```

Start Minikube and check status.

```
minikube start
```

```
minikube status
```

Check Kubernetes resources.

```
kubectl get pod
```

```
kubectl get deploy
```

```
kubectl get replica
```

```
kubectl get pod -o wide
```

Docker Compose (Managing Multi-Container Applications)

Install Docker Compose.

```
sudo apt install docker-compose -y
```

Download the latest Docker Compose binary.

```
sudo curl -L
```

```
"https://github.com/docker/compose/releases/latest/download/docker-compose-\$\(uname -s\)-\$\(uname -m\)" -o /usr/local/bin/docker-compose
```

Example docker-compose.yml file for running **NGINX** and **MySQL**.

yaml code:

```
version: '3'
```

```
services:
```

```
web:
```

```
image: nginx:latest
```

```
ports:
```

```
- 80:80
```

```
db:
image: mysql:latest
environment:
- MYSQL_ROOT_PASSWORD=secret
```

Running MySQL Inside Docker Container

Enter the MySQL container shell.

```
docker exec -it david-db-1 /bin/bash
```

Login to MySQL

```
mysql -u root -p
```

Jenkins Workspace and Maven Build Location

Path where Jenkins builds and stores the .war file.

```
/var/lib/jenkins/workspace/maven/target/my-app.war
```

Pipelining code for Tomcat

```
pipeline {
agent any

environment {
    DOCKER_CREDENTIALS = credentials('docker-hub-cred') // Docker Hub
    Credentials ID
}

stages {
    stage('SCM') {
        steps {
            git branch: 'main', url: '<https://github.com/MugeshS-04/guvidevopsday1.git>'
        }
    }
    stage('Build') {
        steps {
```

```
        sh "mvn clean"
        sh "mvn install"
    }
}
stage('Build Docker Image') {
    steps {
        script {
            sh 'docker build -t mugeshs04/guvidevopsday1 .'
        }
    }
}
stage('Push to Docker Hub') {
    steps {
        script {
            docker.withRegistry('<https://index.docker.io/v1/>', 'docker-hub-
cred') {
                sh 'docker push mugeshs04/guvidevopsday1'
            }
        }
    }
}
}}
```