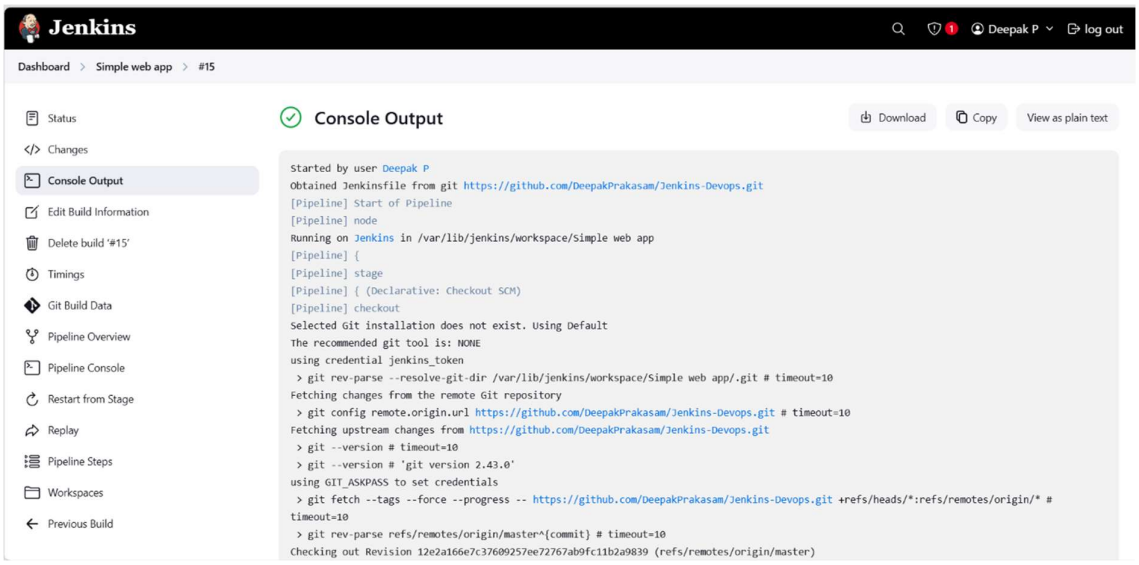
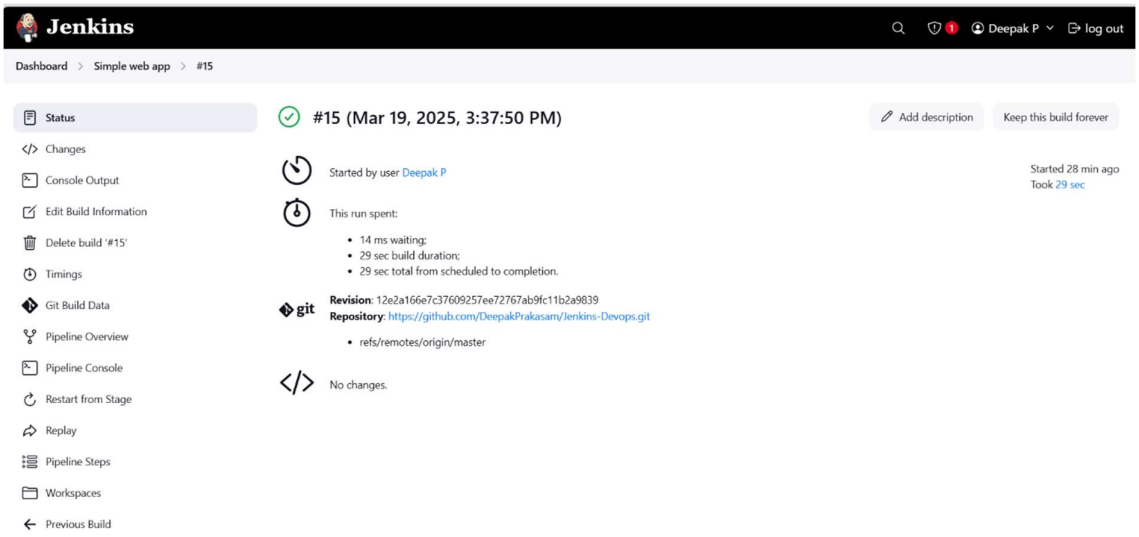
Day 3: Minikube

- *Installed Minikube*
- *Practiced essential Minikube commands*

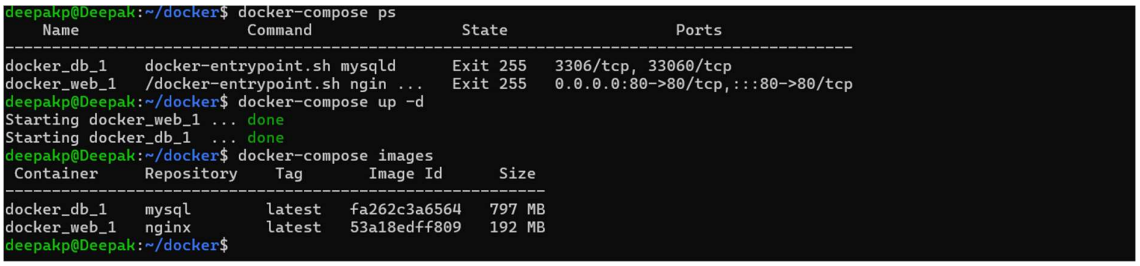
Console output:



Pipeline output:



Terminal output:



Docker Commands:

 # Download Minikube

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

 # Install Minikube

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

 # Start Minikube

minikube start

 # Check Minikube status

minikube status

 # Get running pods

kubectl get pod

 # Get deployments

kubectl get deploy

 # Get replicas

kubectl get replica

 # Get detailed pod information

kubectl get pod -o wide

Docker Compose Commands:

 # Install Docker Compose

sudo apt install docker-compose -y

 # Download the latest version of Docker Compose

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

 # Make Docker Compose executable

sudo chmod +x /usr/local/bin/docker-compose

 # Check Docker Compose version

docker-compose --version

Example docker-compose.yml file:

```
version: '3'

services:
  web:
    image: nginx:latest
    ports:
      - "80:80"
  db:
    image: mysql:latest
    environment:
      - MYSQL_ROOT_PASSWORD=secret

# Start services using Docker Compose
docker-compose up -d

# Execute a shell inside the database container
docker exec -it david-db-1 /bin/bash

# Access MySQL inside the container
mysql -u root -p
```

Kubernetes Commands:

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

```
# Install kubectl with correct permissions
sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl

# Make kubectl executable
chmod +x kubectl

# Create a local bin directory if it doesn't exist
mkdir -p ~/.local/bin

# Move kubectl to the local bin directory
mv ./kubectl ~/.local/bin/kubectl

# Verify kubectl installation
kubectl version --client
```

Jenkins Pipeline Script:

```
pipeline {
    agent any

    stages {
        stage('SCM') {
            steps {
                git 'https://github.com/DeepakPrakasam/Maven.git'
            }
        }
```

```
        stage('Build'){
            steps{
                bat 'mvn clean install'
            }
        }
        stage('Build Docker Image'){
            steps{
                script{
                    bat 'docker build -t deepakp2003/simplewebapp .'
                }
            }
        }
        stage('Push Docker Image'){
            steps{
                script{
                    withDockerRegistry(credentialsId: 'Dockercred', url:
'https://index.docker.io/v1/') {
                        bat 'docker push deepakp2003/simplewebapp'
                    }
                }
            }
        }
    }
}
```