JAVA APPLICATION DEPLOYMENT IN MINIKUBE

Linux System Setup and Java Installation

sudo apt install fontconfig openjdk-17-jre

java -version

Jenkins Installation and Management

sudo service jenkins restart

sudo service jenkins status

For installation instructions: Jenkins Installation Guide

_____

Docker Installation and Commands

sudo apt install docker.io -y

sudo service docker restart

sudo service docker status

sudo usermod -aG docker $USER

Checking Docker Images and Containers

docker images

docker ps

Fixing permission issues

sudo chmod 666 /var/run/docker.sock

Docker Compose Installation

sudo apt install docker-compose -y

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

_____

Kubernetes (K8s) Installation and Commands

Installing 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

More details: Install kubectl

Installing 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

minikube start

minikube status

_____

Kubernetes Commands

Pod Management

Create a pod

kubectl run my-pod --image=nginx --port=80

View all pods

kubectl get pods

kubectl get pods -A

kubectl get pods -n kube-system

View pod details

describe pod <pod-name>

kubectl logs <pod-name>

kubectl exec <pod-name> -- <command>

YAML Configuration for a Pod

```
apiVersion: v1
kind: Pod
metadata:
  name: my-pod
  labels:
    app: my-web-app
    type: backend
spec:
  containers:
    - name: nginx-container
      image: nginx
```

```
    ports:
     - containerPort: 80
```

ReplicaSet Management

```
# Create a ReplicaSet
kubectl create -f rs-test.yml
kubectl apply -f rs-test.yml

# View ReplicaSets
kubectl get replicasets
kubectl get rs -o wide

# Scale a ReplicaSet
kubectl scale replicaset <replicaset-name> --replicas=<desired-replica-count>

# Delete a ReplicaSet
kubectl delete rs <replicaset-name>
kubectl delete -f rs-test.yml
```

ReplicaSet YAML Configuration

```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: my-rs
  labels:
    name: my-rs
```

```
spec:
  replicas: 4
  selector:
    matchLabels:
      apptype: web-backend
  template:
    metadata:
      labels:
        apptype: web-backend
    spec:
      containers:
      - name: my-app
        image: nginx
        ports:
          - containerPort: 8080
```

_____

Deployment Management

```
# Create a deployment
kubectl create deployment webnginx2 --image=nginx:latest --replicas=1

# View deployments
kubectl get deployments
kubectl describe deploy <deployment-name>

# Scale a deployment
kubectl scale deploy <deployment-name> --replicas=<desired-replica-count>
```

```
# Delete a deployment
kubectl delete deploy <deployment-name>
kubectl delete -f web-deploy.yml
```

Deployment YAML Configuration

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-deploy
  labels:
    name: my-deploy
spec:
  replicas: 4
  selector:
    matchLabels:
      apptype: web-backend
  strategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        apptype: web-backend
    spec:
      containers:
      - name: my-app
        image: nginx
```

```
      ports:
       - containerPort: 7070
```

_____

Service Management

View services

kubectl get svc

Create a service from YAML

kubectl create -f service.yml

Delete a service

kubectl delete svc <service-name>

Service YAML Configuration

```
apiVersion: v1
kind: Service
metadata:
  name: my-service
  labels:
    app: my-service
spec:
  type: NodePort
  ports:
    - port: 9000
      targetPort: 8080
      nodePort: 30002
  selector:
```

```
  apptype: web-backend
```

_____

Namespace Management

Create a namespace

kubectl create namespace <namespace-name>

kubectl create ns my-bank

View namespaces

kubectl get ns

Switch to a namespace

kubectl config set-context --current --namespace=<namespace-name>

PIPELINE

```
pipeline {
  agent any
  tools {maven "maven"}
  stages {
    stage('SCM') {
      steps {
        git branch: 'master', url: 'https://github.com/Saran-Avinash/DevOps.git'
      }
    }
    stage('Build-clean') {
      steps {
        sh 'mvn clean'
      }
```

```
}
     stage('Build-validate') {
    steps {
       sh 'mvn validate'
    }
}
     stage('Build-com') {
    steps {
       sh 'mvn compile'
    }
}
     stage('Build-test') {
    steps {
       sh 'mvn test'
    }
}
     stage('Build-pac') {
    steps {
       sh 'mvn package'
    }
}
stage('build to images') {
    steps {
       script {
          sh 'docker build -t saranavinashb/webapp1 .'
       }
```

```
        }
      }
      stage('push to hub') {
        steps {
          script {
            withDockerRegistry(credentialsId: 'docker_cred', toolName: 'docker', url:
'https://index.docker.io/v1/') {
                sh 'docker push saranavinashb/webapp1'
            }
          }
        }
      }

        stage('Deploy App') {
        steps {
                withKubeConfig(caCertificate: '', clusterName: 'minikube', contextName:
'minikube', credentialsId: 'minikube_id', namespace: '', restrictKubeConfigAccess: false,
serverUrl:          'https://192.168.39.226:8443') {
                sh 'kubectl apply -f deployment.yml --validate=false'
                }
            }
         }


   }
}
```

Deployment.yml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-deploy
  labels:
    name: my-deploy
spec:
  replicas: 1
  selector:
    matchLabels:
      apptype: web-backend
  strategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        apptype: web-backend
    spec:
      containers:
      - name: my-app
```

```
      image: saranavinashb/webapp1:latest
      ports:
      - containerPort: 9000

---

apiVersion: v1
kind: Service
metadata:
  name: my-service
  labels:
    app: my-service
spec:
  type: NodePort
  ports:
    - port: 9000
      targetPort: 8080
      nodePort: 30002
  selector:
    apptype: web-backend
```

kubectl run <pod-name> --image=<image-name> --port=<container-port>

$ kubectl run my-pod --image=nginx --port=80

2. View all the pods

(In default namespace)

$ kubectl get pods

(In All namespace)

$ kubectl get pods -A

For a specific namespace

$ kubectl get pods -n kube-system

For a specific type

$ kubectl get pods <pod-name>

$ kubectl get pods <pod-name> -o wide

$ kubectl get pods <pod-name> -o yaml

$ kubectl get pods <pod-name> -o json

3. Describe a pod (View Pod details)

$ kubectl describe pod <pod-name>

$ kubectl describe pod my-pod

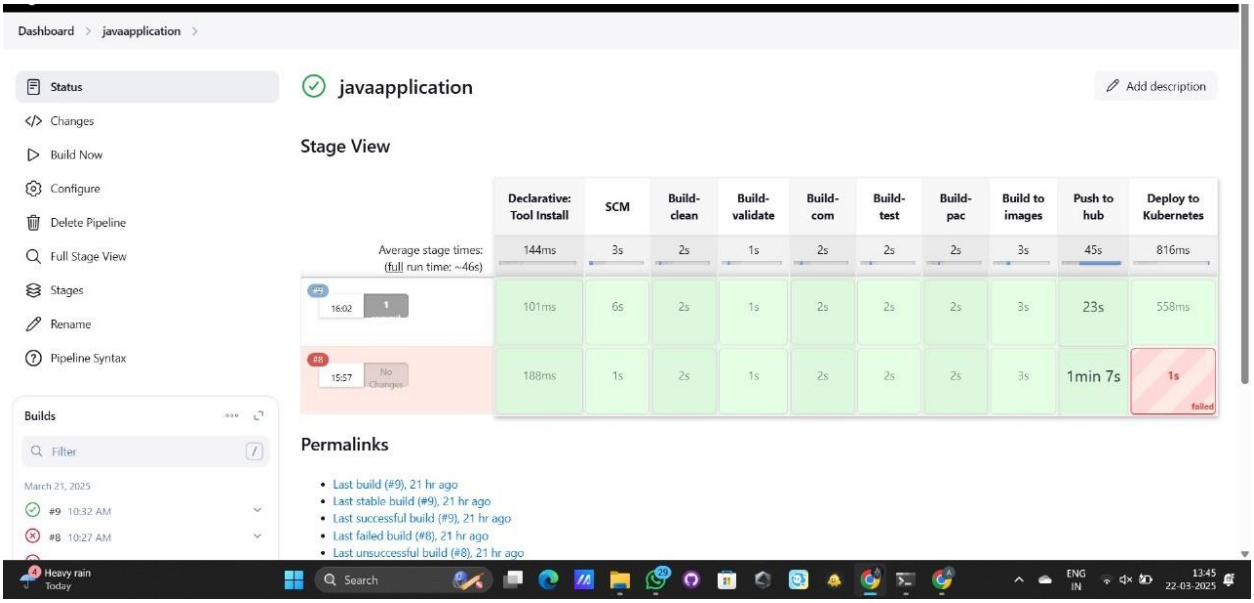
4. View Logs of a pod

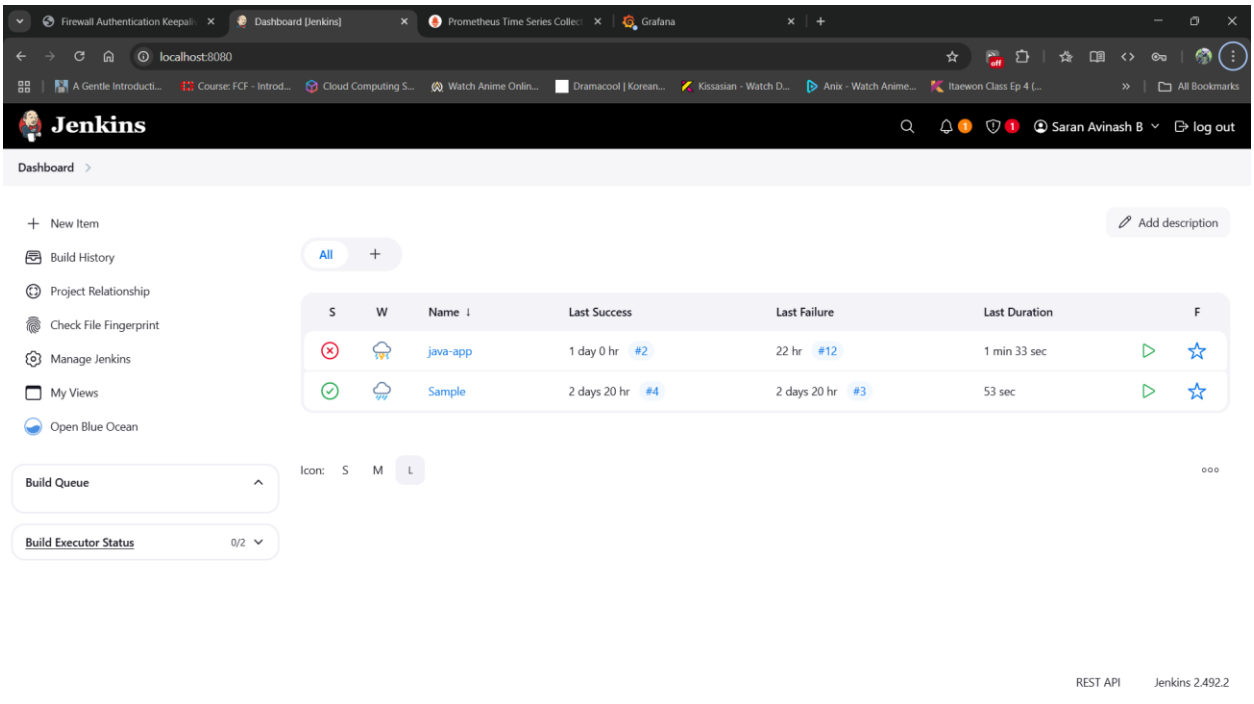
$ kubectl logs <pod-name>

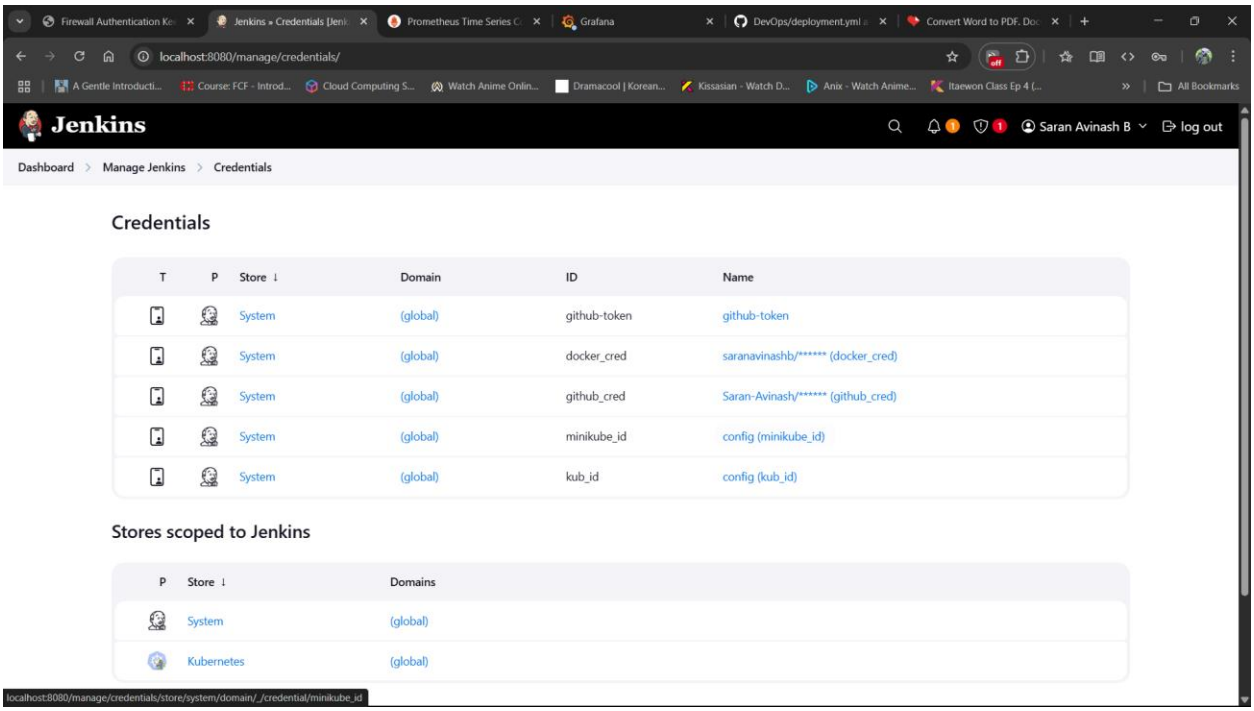
$ kubectl logs my-pod

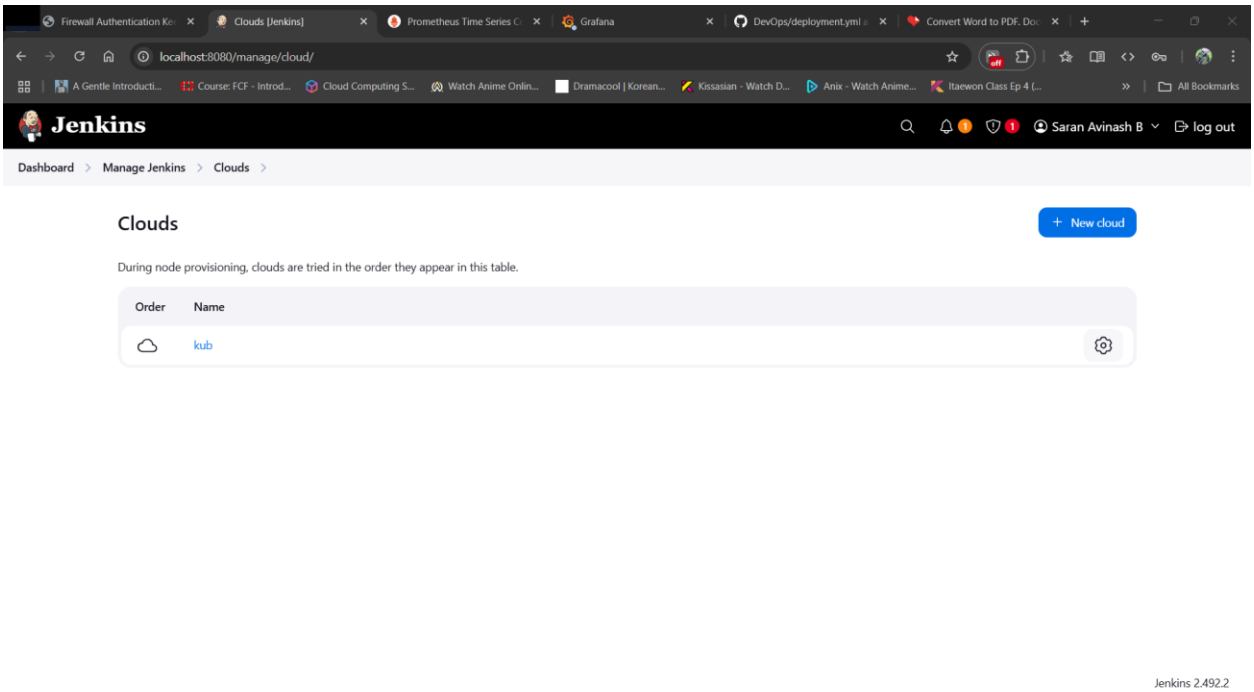
5. Execute any command inside Pod (Inside Pod OS)

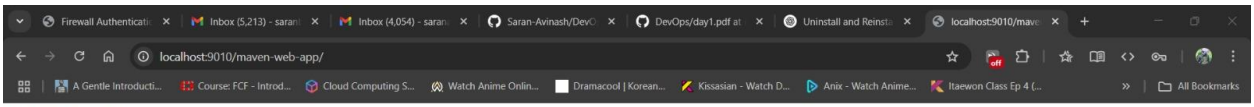
$ kubectl exec <pod-name> -- <command>


Dashboard > javaapplication >
Status
Changes
Build Now
Configure
Delete Pipeline
Full Stage View
Stages
Rename
Pipeline Syntax
javaapplication
Add description
Stage View
Declarative: Tool Install
SCM
Build-clean
Build-validate
Build-com
Build-test
Build-pac
Build to images
Push to hub
Deploy to Kubernetes
Average stage times: (full run time: ~46s)
144ms
3s
2s
1s
2s
2s
2s
3s
45s
816ms
#9 16:02
101ms
6s
2s
1s
2s
2s
2s
3s
23s
558ms
#8 15:57 No Changes
188ms
1s
2s
1s
2s
2s
2s
3s
1min 7s
1s
failed
Builds
Filter
March 21, 2025
#9 10:32 AM
#8 10:27 AM
Permalinks
Last build (#9), 21 hr ago
Last stable build (#9), 21 hr ago
Last successful build (#9), 21 hr ago
Last failed build (#8), 21 hr ago
Last unsuccessful build (#8), 21 hr ago
Heavy rain
Today
Search
ENG
IN
13:45
22-03-2025

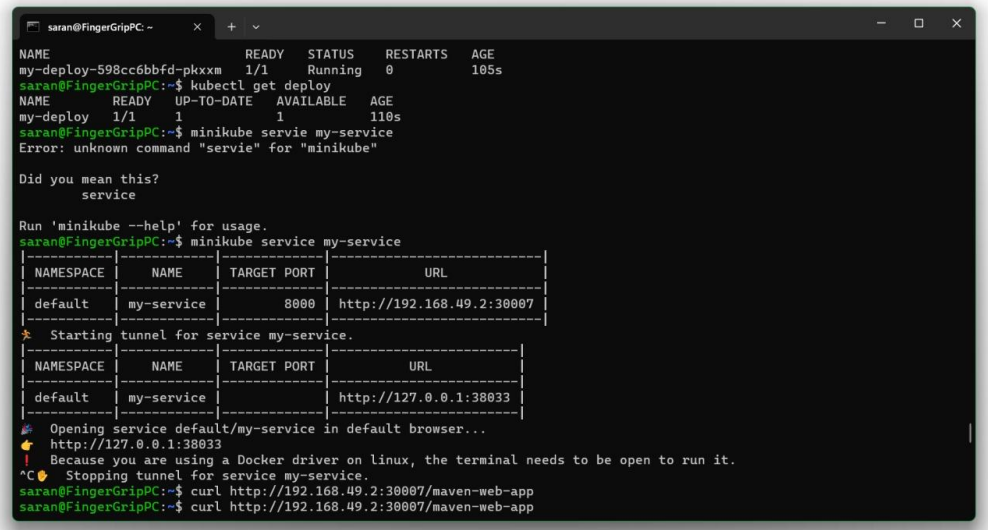

Firewall Authentication Keepalive
Dashboard [Jenkins]
Prometheus Time Series Collection
Grafana
localhost:8080
A Gentle Introducti...
Course: FCF - Introd...
Cloud Computing S...
Watch Anime Onlin...
Dramacool | Korean...
Kissasian - Watch D...
Anix - Watch Anime...
Itaewon Class Ep 4 (...
All Bookmarks
Jenkins
Saran Avinash B
log out
Dashboard
New Item
Build History
Project Relationship
Check File Fingerprint
Manage Jenkins
My Views
Open Blue Ocean
Build Queue
Build Executor Status 0/2
Add description
All
S
W
Name
Last Success
Last Failure
Last Duration
F
java-app
1 day 0 hr #2
22 hr #12
1 min 33 sec
Sample
2 days 20 hr #4
2 days 20 hr #3
53 sec
Icon: S M L
REST API
Jenkins 2.492.2


Jenkins
Saran Avinash B
log out
Dashboard > Manage Jenkins > Credentials
Credentials
T P Store ↓ Domain ID Name
System (global) github-token github-token
System (global) docker_cred saranavinashb/****** (docker_cred)
System (global) github_cred Saran-Avinash/****** (github_cred)
System (global) minikube_id config (minikube_id)
System (global) kub_id config (kub_id)
Stores scoped to Jenkins
P Store ↓ Domains
System (global)
Kubernetes (global)
localhost:8080/manage/credentials/store/system/domain/_/credential/minikube_id

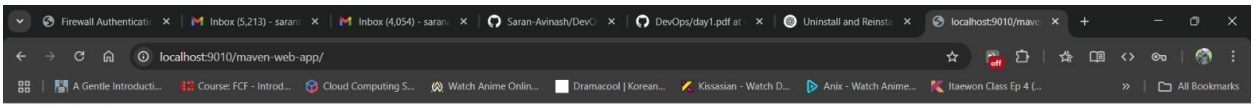

Jenkins
Saran Avinash B
log out
Dashboard > Manage Jenkins > Clouds
Clouds
+ New cloud
During node provisioning, clouds are tried in the order they appear in this table.
Order Name
kub



Hello World!



OUTPUT:



Hello World!