

4. Kubernetes deploying and exposing using Jenkins

Objective

The objective of this setup is to configure a **Jenkins CI/CD pipeline** to deploy an application on **Minikube** running on an Ubuntu 24.04 system. This ensures an automated deployment process where code changes are fetched from a Git repository, deployed to Kubernetes, and made accessible as a service.

Procedure

1. Minikube Installation and Setup

Installing Minikube

1. Install Minikube:
2. curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
3. sudo install minikube-linux-amd64 /usr/local/bin/minikube
4. Start Minikube with Docker Driver:
5. minikube start --driver=docker

Common Fixes

- If you get an error related to running Minikube as root, make sure you're running as a normal user.
- If minikube start fails due to permissions, update permissions:
 - sudo chown -R \$USER \$HOME/.kube \$HOME/.minikube
 - sudo chmod -R 755 \$HOME/.kube \$HOME/.minikube

2. Configuring Kubernetes and Minikube

Set Up Kubeconfig for Minikube

1. Ensure the Kubernetes context is switched to Minikube:
2. kubectl config use-context minikube
3. Verify Minikube status:
4. kubectl get nodes

Fixing TLS Certificate Issues

If you see a TLS certificate verification error:

```
kubectl config set-cluster minikube --server=https://192.168.49.2:8443 \  
--certificate-authority=$HOME/.minikube/ca.crt --embed-certs=true
```

3. Jenkins Integration with Minikube

Installing and Configuring Jenkins

Install Jenkins:

```
sudo apt update && sudo apt install jenkins -y
```

Add Jenkins to Docker Group:

```
sudo usermod -aG docker jenkins
```

Apply changes:

```
sudo systemctl restart jenkins
```

Configure Kubernetes for Jenkins by setting environment variables in the Jenkins pipeline:

```
environment {  
    KUBECONFIG = "/var/lib/jenkins/.kube/config"  
    MINIKUBE_HOME = "/var/lib/jenkins/.minikube"  
}
```

Ensure Jenkins has access to the Kubeconfig file:

```
sudo chown -R jenkins:jenkins /var/lib/jenkins/.kube  
sudo chmod -R 755 /var/lib/jenkins/.kube
```

4. Deploying Application via Jenkins Pipeline

Jenkinsfile (Pipeline Configuration)

```
pipeline {
```

```
    agent any
```

```
    environment {
```

```
        KUBECONFIG = "/var/lib/jenkins/.kube/config"  
        MINIKUBE_HOME = "/var/lib/jenkins/.minikube"  
    }
```

```

stages {
    stage('Setup Minikube Context') {
        steps {
            script {
                sh ""
                set -e
                echo "🔄 Switching to Minikube context..."
                kubectl config use-context minikube
                echo "✅ Minikube context set successfully!"
                ""
            }
        }
    }

    stage('Deploy Application') {
        steps {
            script {
                sh ""
                set -e
                echo "🚀 Deploying Application..."
                kubectl apply -f deployment.yaml --validate=false
                ""
            }
        }
    }
}

```

Pipeline Execution Output (Successful Deployment)

✓ Minikube context set successfully!

🚀 Deploying Application...

deployment.apps/devopstask04 created

Finished: SUCCESS

Pipeline

Define your Pipeline using Groovy directly or pull it from source control.

Definition

Pipeline script from SCM

SCM ?

Git

Repositories ?

Repository URL ?

Credentials ?

- none -

+ Add

Advanced ▾

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

Add Branch

Repository browser ?

(Auto)

Additional Behaviours

Add ▾

Script Path ?

Lightweight checkout ?

[Pipeline Syntax](#)

Advanced

Advanced ▾

[Save](#) [Apply](#)

 DevOps_Task04 Public

[main](#) [Branch](#) [Tags](#) [Add file](#) [Code](#)

ATHIKOS Update Jenkinsfile

1b39723 · 2 hours ago  17 Commits

File	Description	Time Ago
build	Add files via upload	yesterday
Dockerfile	Add files via upload	yesterday
Jenkinsfile	Update Jenkinsfile	2 hours ago
deploy.sh	Update deploy.sh	3 hours ago
deployment.yaml	Update deployment.yaml	2 hours ago
service.yaml	Update service.yaml	2 hours ago

[README](#)



Add a README

Help people interested in this repository understand your project by adding a README.

About
No description, website, or topics provided.

 Activity
 0 stars
 1 watching
 0 forks

Releases
No releases published
[Create a new release](#)

Packages
No packages published
[Publish your first package](#)

Languages



Language	Percentage
CSS	47.2%
HTML	32.7%
Shell	13.3%
JavaScript	4.1%
Python	2.7%

Jenkins

Dashboard > DevOps_Task04 > #9

Console Output

Status Changes Console Output Edit Build Information Timings Pipeline Overview Pipeline Console Replay Pipeline Steps Workspaces Previous Build

```

Started by user MOHAMED ATHIK R
Obtained Jenkinsfile from git https://github.com/ATHIK05/DevOps_Task04.git
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/DevOps_Task04
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/DevOps_Task04/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/ATHIK05/DevOps_Task04.git # timeout=10
Fetching upstream changes from https://github.com/ATHIK05/DevOps_Task04.git
> git --version # timeout=10
> git --version # 'git version 2.43.0'
> git fetch --tags --force --progress -- https://github.com/ATHIK05/DevOps_Task04.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 1b397232e8213127385a4c1ccf8eb29edce613b9 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 1b397232e8213127385a4c1ccf8eb29edce613b9 # timeout=10
Commit message: "Update Jenkinsfile"
> git rev-list --no-walk 1b397232e8213127385a4c1ccf8eb29edce613b9 # timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Setup Minikube Context)
[Pipeline] script
[Pipeline] {
[Pipeline] sh
+ set -e
+ echo ✅ Switching to Minikube context...
✅ Switching to Minikube context...
+ kubectl config use-context minikube
Switched to context "minikube".
+ echo ✅ Minikube context set successfully!
✅ Minikube context set successfully!
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Deploy Application)
[Pipeline] script
[Pipeline] {
[Pipeline] sh
+ set -e
+ echo 🚀 Deploying Application...
🚀 Deploying Application...
+ kubectl apply -f deployment.yaml --validate=false
deployment.apps/devopstask04 created
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

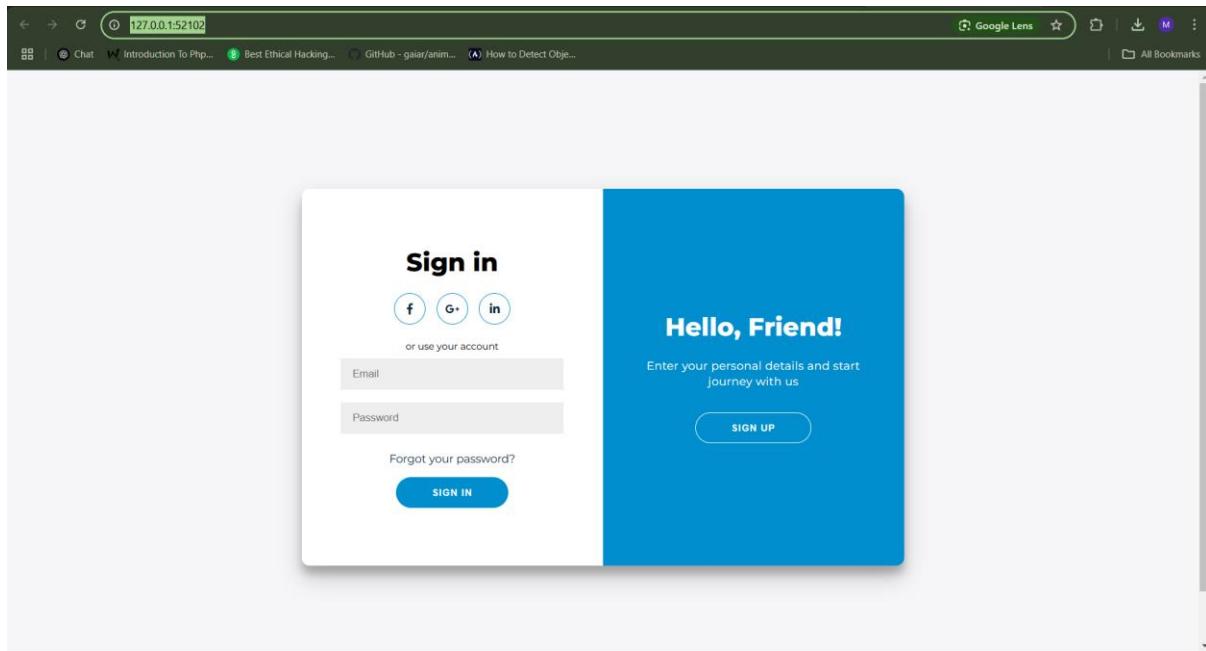
```

REST API Jenkins 2.495

```

jenkins@Athik:~$ kubectl get pods -n default
NAME                               READY   STATUS    RESTARTS   AGE
devopstask04-84dd496896-8b4jr   1/1     Running   0          4m24s
devopstask04-84dd496896-vftm6   1/1     Running   0          4m24s
jenkins@Athik:~$ kubectl get svc -n default
NAME      TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.96.0.1   <none>       443/TCP   43m

```



5. Running and Accessing the Deployed Service

Check Deployment and Service

```
kubectl get pods -n default
```

```
kubectl get svc -n default
```

Expected output:

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
devopstask04	NodePort	10.98.142.33	<none>	80:32233/TCP	10s

Get Minikube Service URL

```
minikube service devopstask04 --url -n default
```

Output:

<http://127.0.0.1:52102/>

Access it via browser or:

```
curl http://127.0.0.1:52102/
```

Expose Service (If Not Already Exposed)

```
kubectl expose deployment devopstask04 --type=NodePort --port=80 -n default
```

Conclusion

With this setup, Jenkins automatically deploys your application to Minikube, and the service is accessible through the provided Minikube service URL.

