

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

Dashboard > DevOps_Task04 >

Status

Changes

Build Now

Configure

Delete Pipeline

Stages

Rename

Pipeline Syntax

DevOps_Task04

Add description

Permalinks

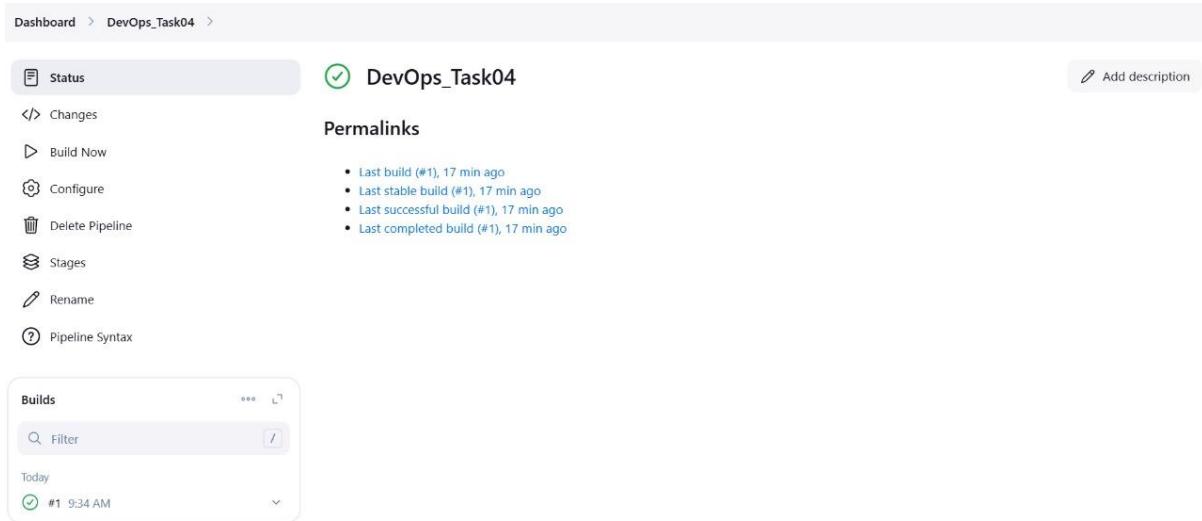
- Last build (#1), 17 min ago
- Last stable build (#1), 17 min ago
- Last successful build (#1), 17 min ago
- Last completed build (#1), 17 min ago

Builds

Filter

Today

#1 9:34 AM



* New More Docker, Easy Access, New Streamlined Plans. Learn more →

dockerhub Explore Repositories Organizations Usage

Search Docker Hub

Ragavi Edit profile ↗

Community User

Repositories Starred Contributed

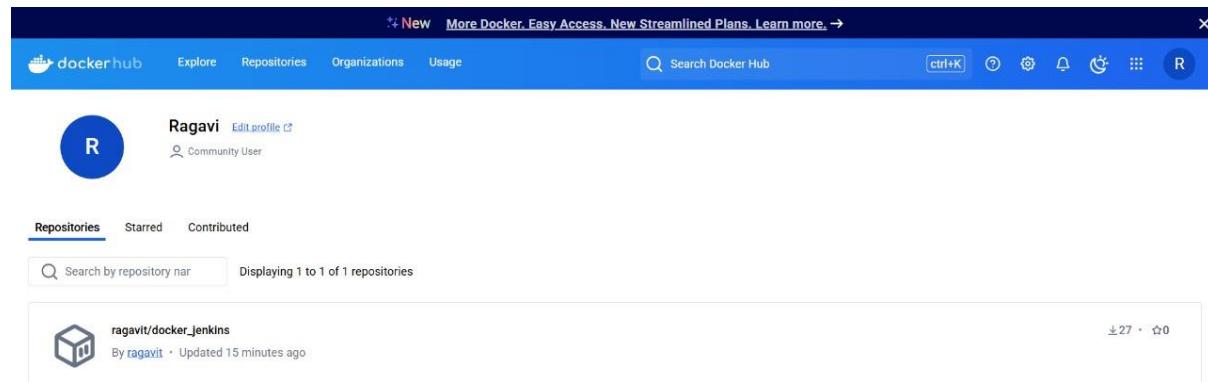
Search by repository name

Displaying 1 to 1 of 1 repositories

ragavit/docker_jenkins

By ragavit · Updated 15 minutes ago

27 · 0



ragavit/docker_jenkins 

Last pushed 16 minutes ago • Repository size: 260.6 MB

Add a description  

Add a category  

General Tags Builds Collaborators Webhooks Settings

Tags

This repository contains 2 tag(s).

Tag	OS	Type	Pulled	Pushed
ekart		Image	14 minutes ago	16 minutes ago
latest		Image	26 minutes ago	3 days ago

[See all](#)

Automated builds

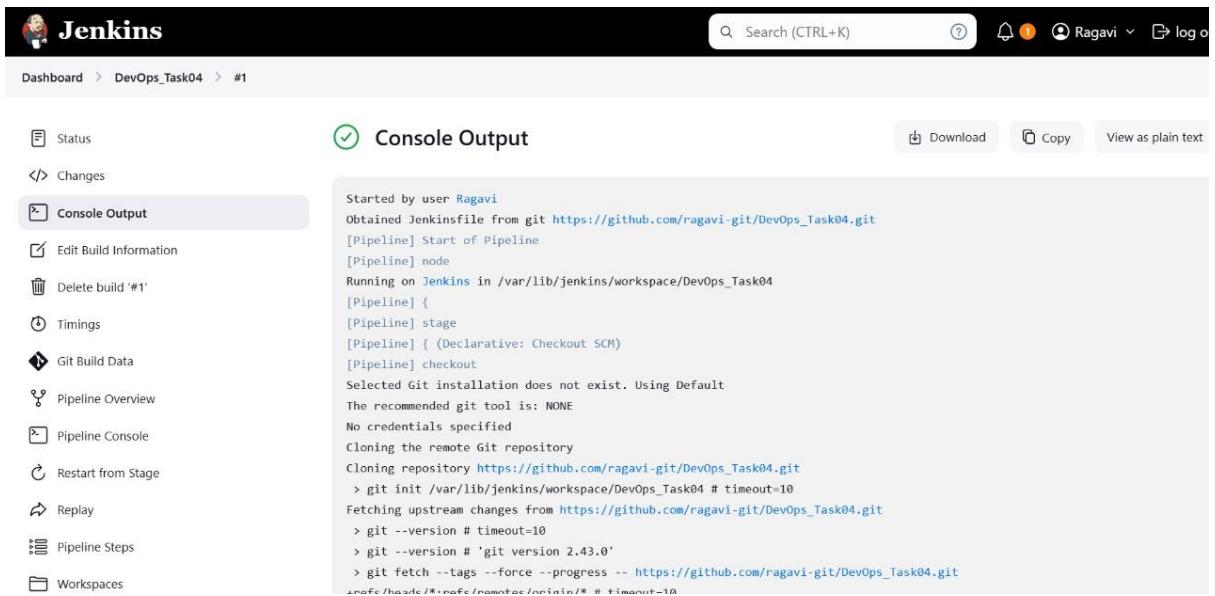
Manually pushing images to Docker Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions. [Read more about automated builds](#) .

[Upgrade](#)

Repository overview INCOMPLETE

An overview describes what your image does and how to run it. It displays in [the public view of your repository](#) once you have pushed some content.



The screenshot shows the Jenkins interface for a pipeline job named "DevOps_Task04". The left sidebar has a "Console Output" item selected. The main area shows the Jenkins pipeline logs:

```

Started by user Ragavi
Obtained Jenkinsfile from git https://github.com/ragavi-git/DevOps_Task04.git
[Pipeline] Start of Pipeline
[Pipeline] node
[Jenkins] node
Running on Jenkins in /var/lib/jenkins/workspace/DevOps_Task04
[Pipeline] {
[Pipeline] stage
[Pipeline] {
  (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/ragavi-git/DevOps_Task04.git
> git init /var/lib/jenkins/workspace/DevOps_Task04 # timeout=10
Fetching upstream changes from https://github.com/ragavi-git/DevOps_Task04.git
> git --version # timeout=10
> git --version # 'git version 2.43.0'
> git fetch --tags --force --progress -- https://github.com/ragavi-git/DevOps_Task04.git
+refs/heads/*:refs/remotes/origin/* # timeout=10

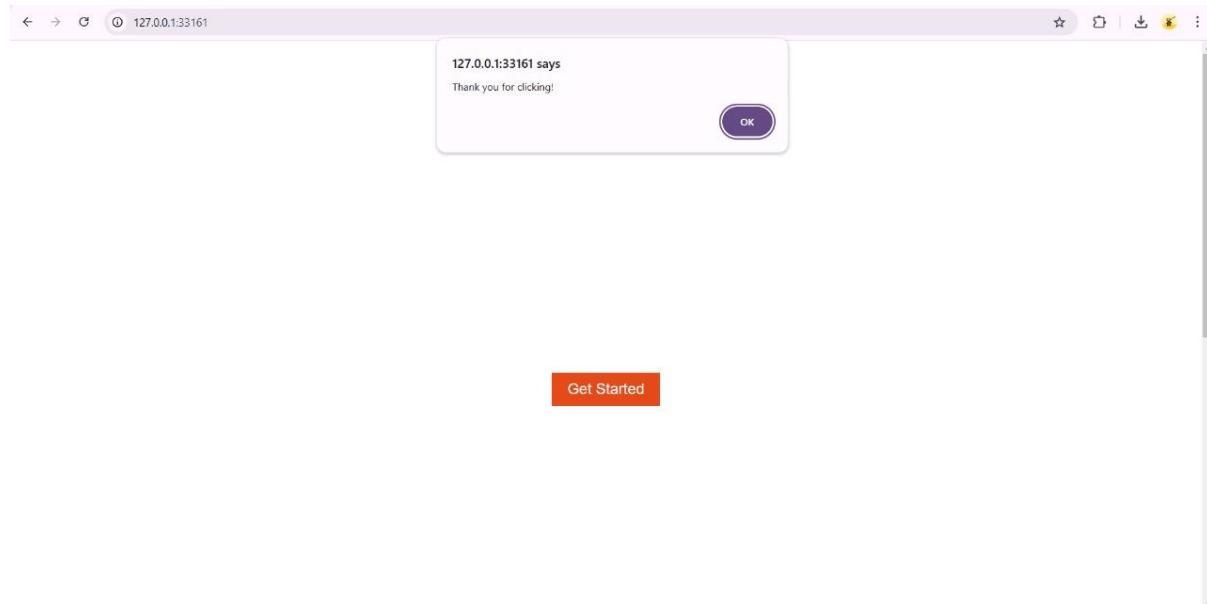
```

```

user036@LAPTOP-9EU7EI28:~$ kubectl get deployments
NAME        READY   UP-TO-DATE   AVAILABLE   AGE
devopstask04 1/1     1           1           31s
hello       1/1     1           1           2d1h
hi          1/1     1           1           8h
user036@LAPTOP-9EU7EI28:~$ minikube service devopstask04
✖ Exiting due to SVC_NOT_FOUND: Service 'devopstask04' was not found in 'default' namespace.
You may select another namespace by using 'minikube service devopstask04 -n <namespace>'. Or list out all the services using 'minikube service list'

user036@LAPTOP-9EU7EI28:~$ kubectl get services
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
hello     NodePort    10.108.177.25  <none>        80:32436/TCP  2d1h
hi        NodePort    10.101.131.213  <none>        80:32626/TCP  8h
kubernetes ClusterIP  10.96.0.1     <none>        443/TCP      2d1h
user036@LAPTOP-9EU7EI28:~$ kubectl expose deployment devopstask04 --type=NodePort --port=80
service/devopstask04 exposed
user036@LAPTOP-9EU7EI28:~$ kubectl get services
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
devopstask04  NodePort    10.109.76.65  <none>        80:32506/TCP  6s
hello     NodePort    10.108.177.25  <none>        80:32436/TCP  2d1h
hi        NodePort    10.101.131.213  <none>        80:32626/TCP  8h
kubernetes ClusterIP  10.96.0.1     <none>        443/TCP      2d1h
user036@LAPTOP-9EU7EI28:~$ minikube service devopstask04
|-----|-----|-----|-----|
| NAMESPACE | NAME | TARGET PORT | URL |
|-----|-----|-----|-----|
| default | devopstask04 | 80 | http://192.168.67.2:32506 |
|-----|-----|-----|-----|
✖ Starting tunnel for service devopstask04.
|-----|-----|-----|-----|
| NAMESPACE | NAME | TARGET PORT | URL |
|-----|-----|-----|-----|
| default | devopstask04 | | http://127.0.0.1:33161 |
|-----|-----|-----|-----|
✖ Opening service default/devopstask04 in default browser...
☞ http://127.0.0.1:33161
! Because you are using a Docker driver on linux, the terminal needs to be open to run it.

```



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.