Java Application Deployment In Kubernetes With Jenkins CI/CD Pipeline

Last Updated : 20 Mar, 2024

In modern software development, deploying applications to Kubernetes clusters has become a common practice due to their scalability and reliability. Automating this deployment process using continuous integration and continuous deployment (CI/CD) pipelines with Jenkins streamlines the development workflow and ensures consistent, reliable deployments. This article demonstrates deploying a sample Java application on a Kubernetes cluster using Minikube.

1. Creating a sample Java application

We'll start by creating a simple Java application. Compile the Java code using the javac command.

```
class Main {
    public static void main(String[] args) {
        System.out.println("Hello from App");
    }
}
```

2. Creating a Docker image

We'll create a Dockerfile to package the Java application into a Docker image. The Dockerfile will copy the compiled Java class file into the image and specify the command to run the application.

```
FROM openjdk:11-jre-slim
WORKDIR /app
COPY Main.class .
CMD ["java", "Main"]
```

```
PS                                    \deployJavaApp> docker images
REPOSITORY                  TAG       IMAGE ID       CREATED          SIZE
java-app                    latest    4113a4c5bd41   6 seconds ago    223MB
```

3. Defining Kubernetes Deployment YAML File

The deployment YAML file specifies the desired state of the application, including the number of replicas, container image, and other configurations.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: java-app-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: java-app
  template:
    metadata:
      labels:
        app: java-app
    spec:
      containers:
      - name: java-app-container
        image: java-app
```

4. Creating Jenkins Role YAML File (Optional)

If Jenkins needs permission to interact with Kubernetes resources, you can create a Kubernetes role YAML file to define the required permissions.

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: jenkins-role
rules:
- apiGroups: [""]
  resources: ["pods", "services", "deployments"]
  verbs: ["get", "list", "watch", "create", "update",
"delete"]
```

5. Creating a Service Account in Kubernetes

Start the minikube.

```
minikube start
```

Use the kubectl create serviceaccount command to create a service account in your Kubernetes cluster.

```
kubectl create serviceaccount jenkins
```

```
PS                                    \deployJavaApp> kubectl get serviceaccoun
t jenkins
>>
NAME      SECRETS   AGE
jenkins   0         87m
```

6. Generating Token using OpenSSl and setting it for the service account

This command generates a random 32-byte token encoded in base64. Copy the generated token as we need it in the next step.

```
openssl rand -base64 32
example output: vBM3jD+vKxWlwm+Bd9y2tnoKZdEh8Vc9nutV7LAZ/AE+
```

We can now set it for the desired service account in Kubernetes. Use the kubectl command to create a secret containing the token and associate it with the service account.

```
kubectl create secret generic jenkins-token --from-
literal=token=vBM3jD+vKxWlwm+Bd9y2tnoKZdEh8Vc9nutV7LAZ/AE+
```

Make sure you replace **vBM3jD+vKxWlwm+Bd9y2tnoKZdEh8Vc9nutV7LAZ/AE+** with your own generated token.

After creating the secret in above step, we need to associate it with the service account.

This command patches the jenkins service account to use the secret jenkins-token. It adds the secret to the list of image pull secrets for the service account.

```
kubectl patch serviceaccount jenkins -p '{"imagePullSecrets":
[{"name": "jenkins-token"}]}'
```

7. Finding the URL and port of the Minikube cluster

The **minikube dashboard --url** command is used to retrieve the URL of the Kubernetes dashboard running on your Minikube cluster.

```
minikube dashboard --url
```

```
PS                                                    > minikube dashboard --url
>>

🤔  Verifying dashboard health ...
🚀  Launching proxy ...
🤔  Verifying proxy health ...
http://127.0.         /api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/
```

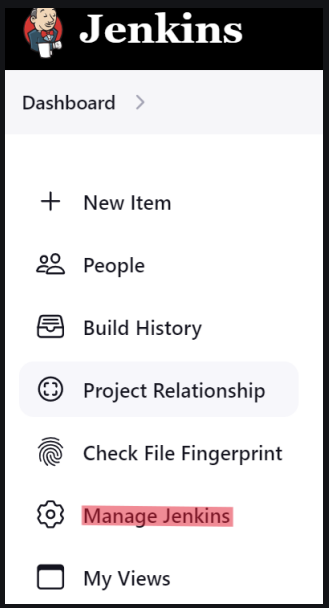
Note down this url and port number as we need it for the jenkins script.

8. Configuring Jenkins

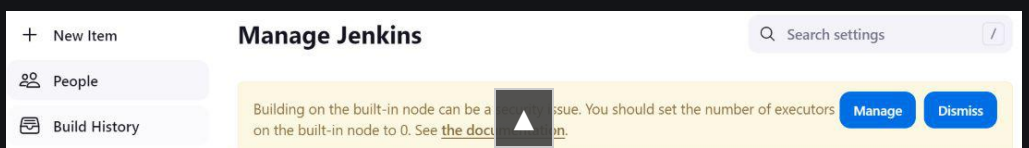
Install Jenkins LTS version for your Operating System and complete the on screen setup. On new browser tab visit **http://localhost:8080/**

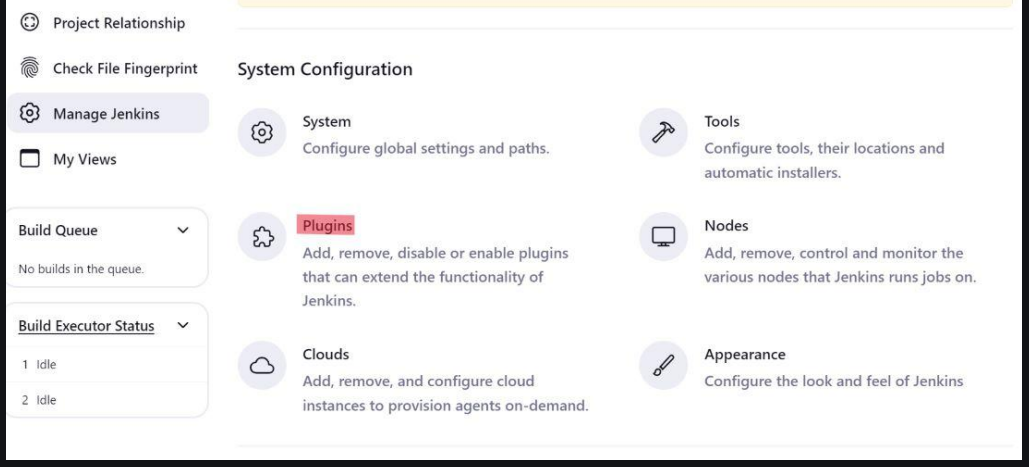
1. Installing Plugins

- On dashboard click on Manage Jenkins.

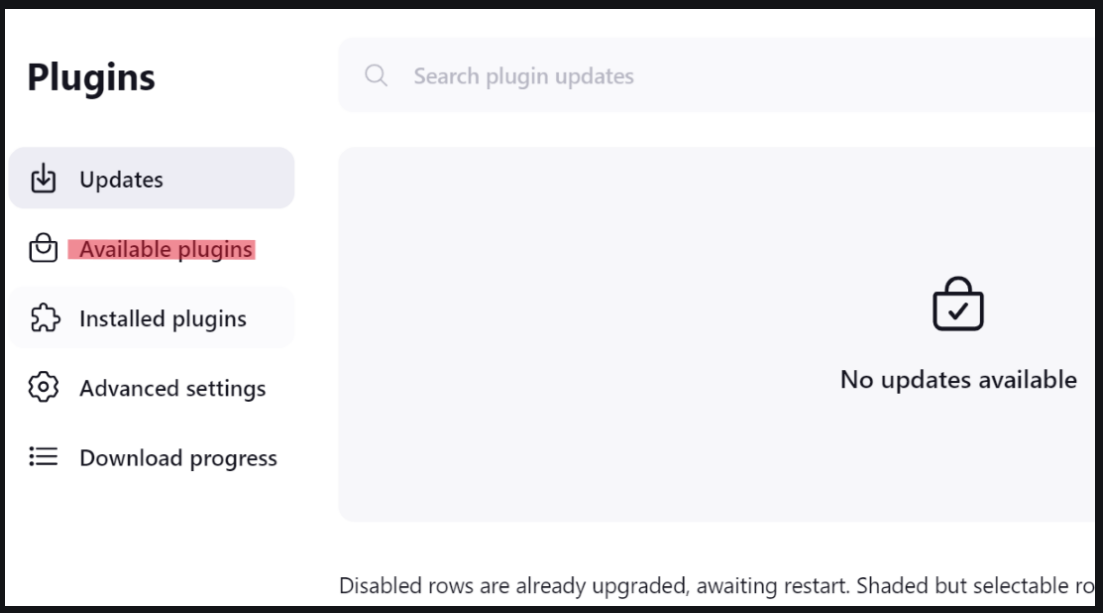


- Then click on Plugins.

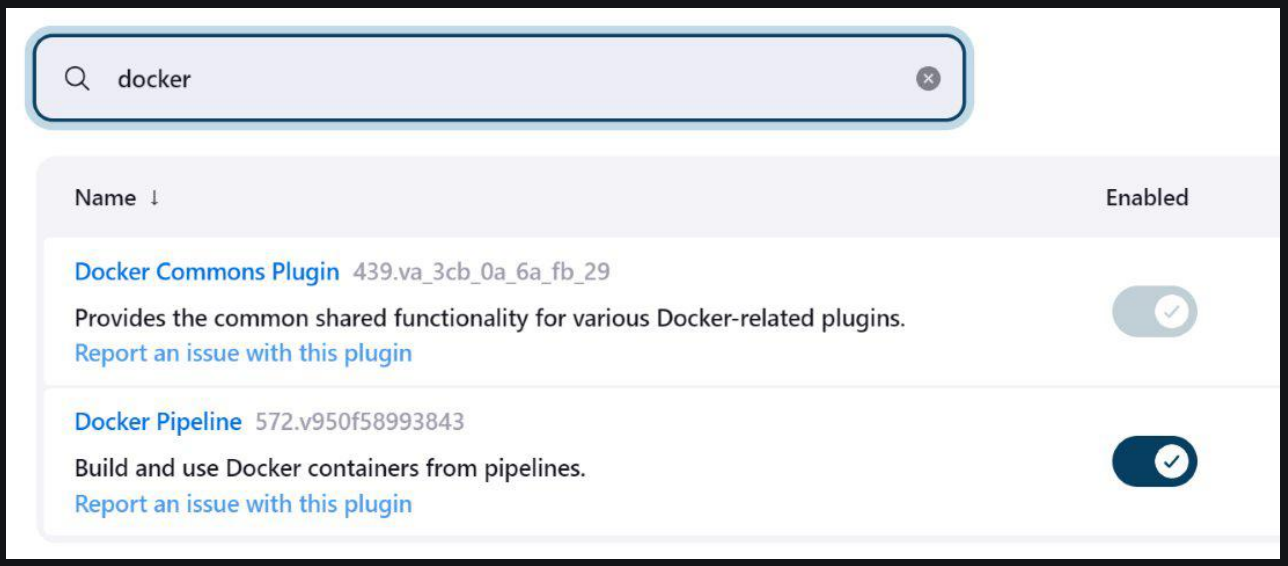




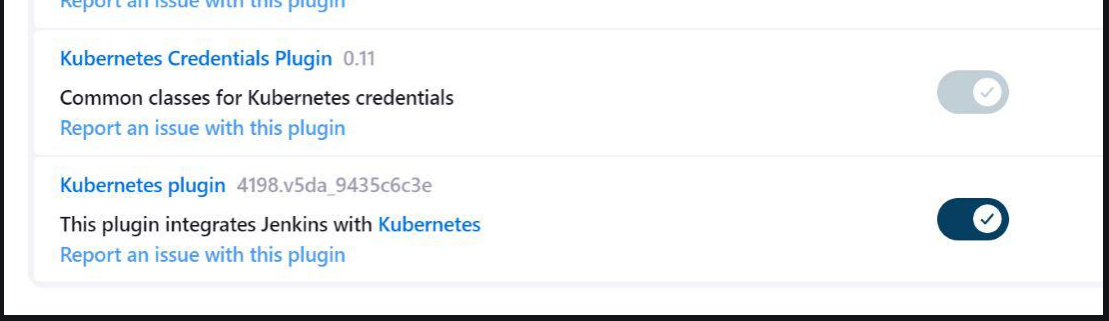
- Click on Available plugins.



- Install Docker Pipeline plugin and Kubernetes plugin.

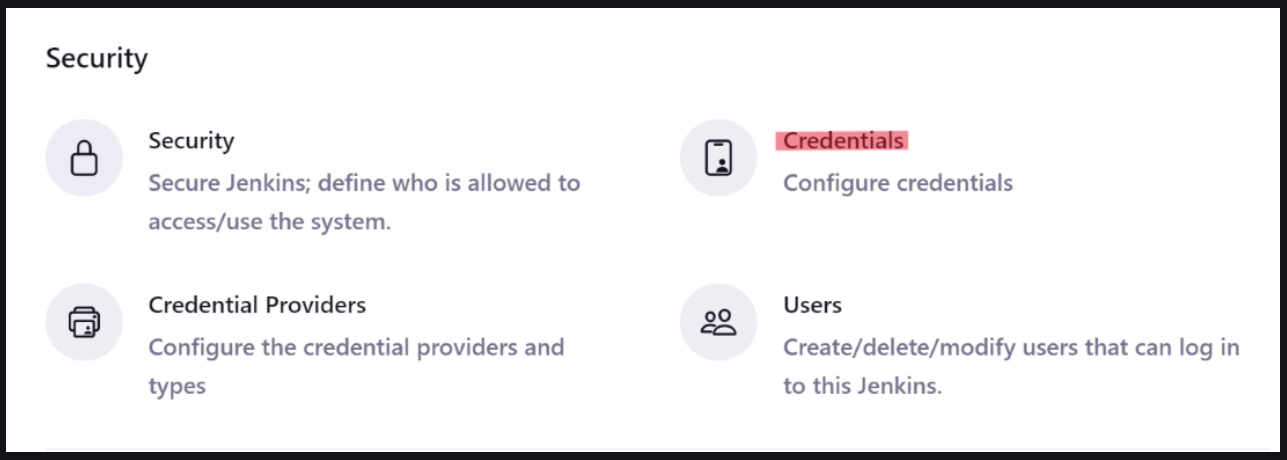


- Searching For kubernetes Plugin

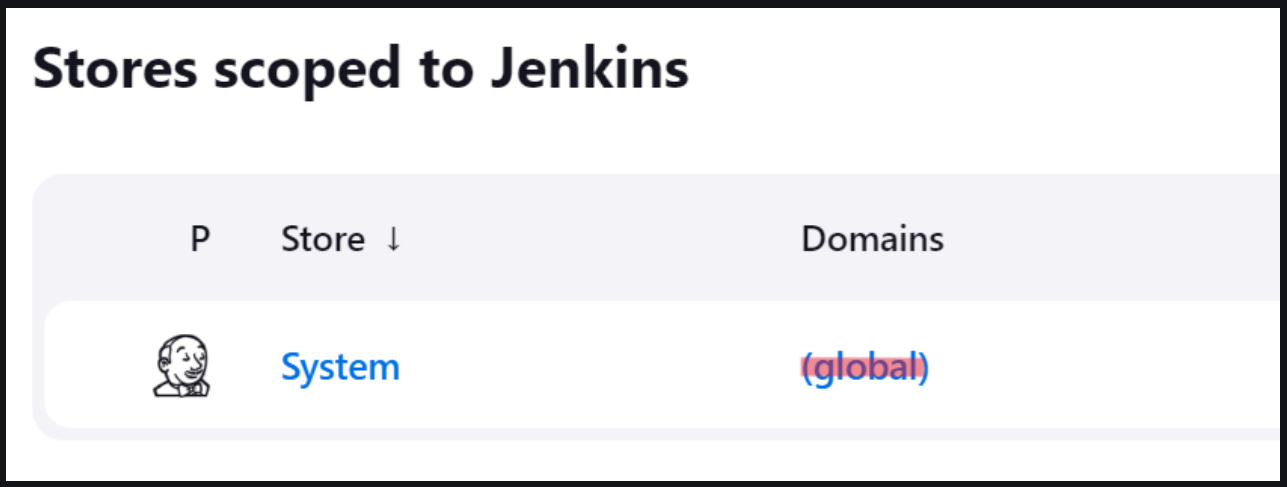


2. Configure Credentials

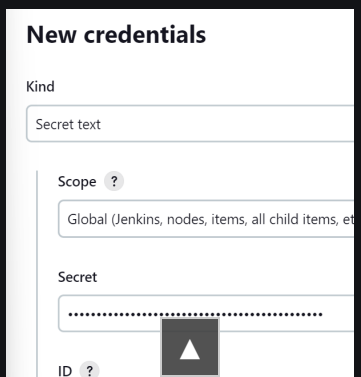
- On dashboard click on Manage Jenkins and under Security click on Credentials.

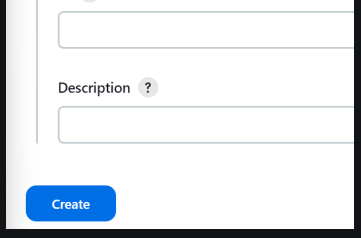


- Click on global and then add credentials.

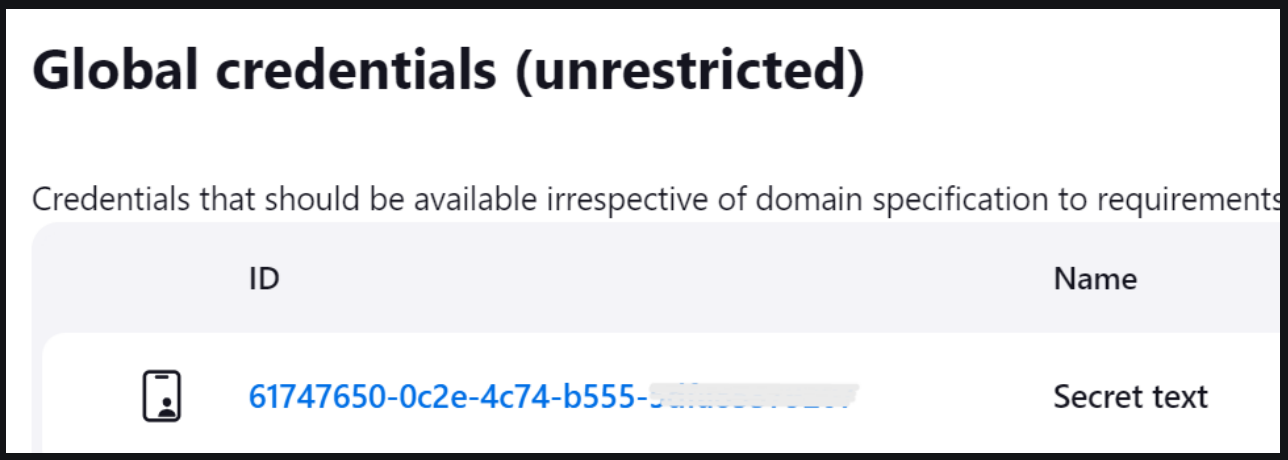


- Select Kind as Secret Text and then add the copied key generated in step 6 in Secret Tab. Keep the ID and Description as empty.



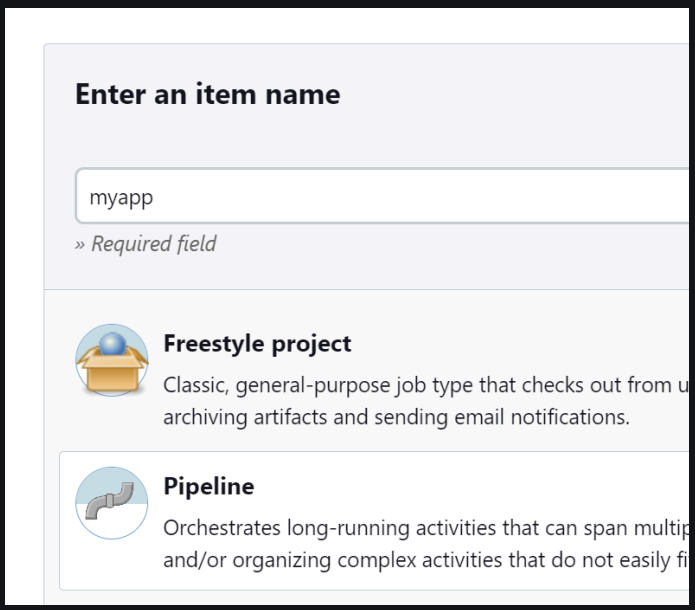


- Now you can see the added credentials and the auto generate ID. Copy this ID as this will be required in further steps.

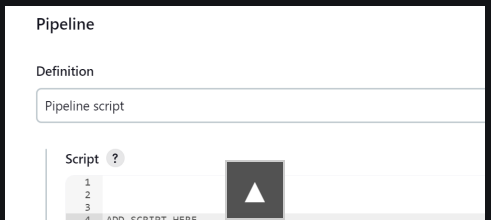


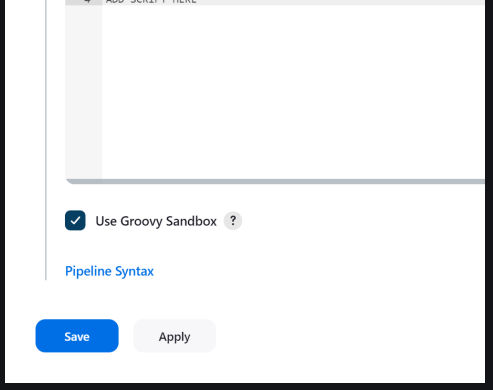
9. Creating Jenkins Pipeline

- On Dashboard click on New Item. Enter the item name, example myapp, and select Pipeline Option.



- Scroll down to Pipeline and add below Pipeline script under script section. Make sure you **replace your-credential-id** (obtained in step 8-2), **http://link:port** (obtained in step 7) and the **paths** with your actual values. Then click on Save.



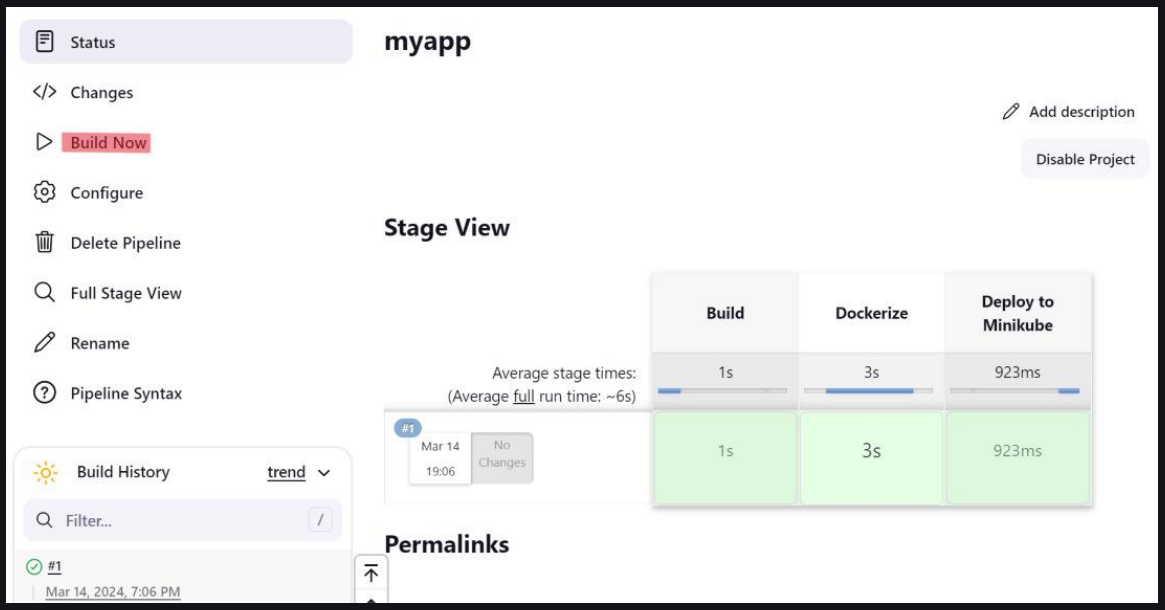


This Jenkins pipeline script automates the deployment of a Java application to a Kubernetes cluster. It comprises three stages: **Build** compiles the Java code, **Dockerize** builds a Docker image, and **Deploy to Minikube** applies the Kubernetes deployment using a specified service account token and server URL with TLS verification skipped. The script integrates with Jenkins credentials to securely access the Kubernetes service account token and executes commands to deploy the application to the specified Minikube cluster.

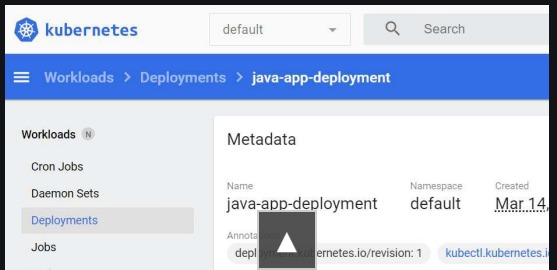
```
pipeline {
    agent any

    stages {
        stage('Build') {
            steps {
                // Compile Java code
                bat "javac \"C:\\Users\
\deployJavaAppMain.java\""
            }
        }

        stage('Dockerize') {
            steps {
                script {
                    // Disable BuildKit
                    bat 'SET DOCKER_BUILDKIT=0'

                    // Build Docker image
                    bat 'docker build -t java-app "C:\\Users\
\deployJavaApp"'
                }
            }
        }
```

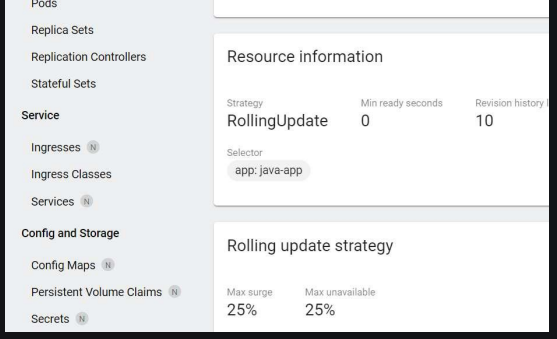
```
        stage('Deploy to Minikube') {
            steps {
                // Apply Kubernetes deployment using the
Kubernetes service account
                withCredentials([string(credentialsId: 'your-
credential-id', variable: 'KUBE_SA_TOKEN')]) {
                    bat """
                        kubectl apply -f "C:\\Users\
\deployJavaApp\\kubernetes-deployment.yaml" \
                        --token="$KUBE_SA_TOKEN" \
                        --server=http://link:port \
                        --insecure-skip-tls-verify
                    """
                }
            }
        }
    }
}
```

- Then click on Build Now which starts the pipeline to Build and automates the deployment process.



- Now deployment is successful and you can access the deployed application through exposed service or by accessing the pods directly or you can visit the Kubernetes dashboard to manage your deployment.





Java application deployment in Kubernetes with Jenkins CI/CD pipeline - FAQ's

How do I deploy an application in Kubernetes using Jenkins?

To deploy an application in Kubernetes using Jenkins:

1. *Set up Jenkins pipelines to automate the build, test, and deployment process.*
2. *Configure Jenkins to interact with Kubernetes clusters using plugins like Kubernetes Continuous Deploy.*

How do I deploy a Java application in Kubernetes?

To deploy a Java application in Kubernetes:

1. *Create Docker images of your Java application.*
2. *Define Kubernetes deployment manifests specifying image, ports, and resources for scaling and management.*

Are you ready to unleash the power of **DevOps** to streamline your **Software Development and Deployment**? Learn about our DevOps Live Course at GeeksforGeeks, created for all professionals in practice with continuous integration, delivery, and deployment. Learn about leading tools, industry best practices, and techniques for **automation** through an interactive session with hands-on **live projects**. Whether you are new to DevOps or looking to improve your skills, this course equips you with everything needed to streamline workflows and deliver excellent quality software in the least

amount of time. Learn to take your skills in DevOps to the next level now, and harness the power of streamlined software development!

katharv + Follow

Next Article

Node.Js Application Deployment in Kubernetes with Jenkins CI/CD Pipeline

Similar Reads

Python Application Deployment In Kubernetes With Jenkins CI/CD Pipeline

In modern software development, deploying applications to Kubernetes clusters has become a common practice due to their scalability and reliability....

6 min read

Spring Boot Application Deployment in Kubernetes with Jenkins CI/CD...

As we know in the modern world we use Continuous Integration and Continuous Deployment for fast and and efficient delivery of our application to...

14 min read

Node.Js Application Deployment in Kubernetes with Jenkins CI/CD Pipeline

As we know in the modern world we use Continuous Integration and Continuous Deployment for fast and and efficient delivery of our application to...

9 min read

Build, Test and Deploy a Flask REST API Application from GitHub using...

Nowadays even for small web applications or microservices, we need an easier and faster way to deploy applications that is reliable and safe. These...

4 min read

Kubernetes - Creating Deployment and Services using Helm in Kubernetes

Prerequisite: Kubernetes Helm is used for managing your Kubernetes Deployment. With helm, we can tear down and create a deployment with a...

4 min read

Deployment Of Spring Boot Application In Jenkins

In this article, we will be exploring how to automate the process of building Spring Boot applications by employing a polling trigger. We will also learn ho...

5 min read

Understanding Jenkins CI/CD Pipeline And Its Stages

Jenkins is an open-source automation server that enables developers to reliably build, test, and deploy applications. It supports continuous integration and...

15 min read

How To Create Methods In Jenkins Declarative Pipeline?

Jenkins is an open-source automation tool used to automate the fetch, build, test, and deploy stages of a software application. Here in this guide, we will fir...

4 min read

Setting Up A CI/CD Pipeline For PHP Applications Using Jenkins

In contemporary software development, Continuous Integration and Continuous Deployment (CI/CD) pipelines have arisen as key devices for smoothing out the...

6 min read

What Is Jenkins Declarative Pipeline ?

Jenkins Declarative Pipeline is a more recent addition to Jenkins as a code pipeline definition approach. It offers a more simplified and structured way of...

4 min read

How to use AWS Credentials in Jenkins Pipeline

Jenkins, an industry-standard automation server, assumes a pivotal role in present-day software development practices, especially in continuous...

8 min read

Introduction To Jenkins Pipeline As Code (Jenkinsfile)

Jenkins is an automating web server that build using java programming language. It come with support large number of plugins that made it as power...

5 min read

How to Make a CI-CD Pipeline in Jenkins?

Pre-requisites: Jenkins DevOps professionals mostly work with pipelines because pipelines can automate the processes like building, testing, and...

5 min read

What is Kubernetes Deployment?

Kubernetes is an open-source Container Management tool that automates container deployment, container scaling, descaling, and container load balanci...

10 min read

kubernetes Replicaset VS Deployment

In Kubernetes, both Deployment and ReplicaSet are used to manage the lifecycle of pods. In order to Manage the life cycle of applications in a...

6 min read

What Are The Kubernetes Deployment Stratagies ?

Traditionally, applications used to be deployed on the Physical Servers. However, it was not very efficient as multiple instances of the applications...

9 min read

What is Kubernetes Blue Green Deployment?

Blue-green deployment is a software deployment strategy that involves running two identical production environments, known as "blue" and "green." At any...

6 min read

Continuous Deployment to Kubernetes with GitOps

GitOps is the mechanism that a few companies like Google, Lyft, etc., has chosen to manage their workflows with. It is a modern approach to the implementatio...

14 min read

Kubernetes Deployment Strategies

Kubernetes is an open-source container orchestration tool. It helps to provide run and manage the thousands of containerized applications with different...

8 min read

Setup Jenkins On Kubernetes Cluster

Setting up Jenkins on a Kubernetes cluster involves deploying Jenkins as a containerized application within the cluster. Kubernetes is Born in Google,...

4 min read

Article Tags : DevOps | Dev Scripter | Kubernetes | Dev Scripter 2024 | +1 More

GeeksforGeeks
Sanchhaya Education Private Limited

Corporate & Communications Address:- A-143, 9th Floor, Sovereign Corporate Tower, Sector- 136, Noida, Uttar Pradesh (201305) | Registered Address:- K 061, Tower K, Gulshan Vivante Apartment, Sector 137, Noida, Gautam Buddh Nagar, Uttar Pradesh, 201305

Software
Development
Software Testing