Mastering AWS and DevOps

Day 23: Continuous Deployment with Jenkins and Kubernetes

- **Topics**: Jenkins Pipelines, Deploying to Kubernetes

Continuous Deployment with Jenkins and Kubernetes

Continuous Deployment (CD) automates the deployment of applications to production environments, ensuring rapid and reliable software delivery. This documentation provides a clear and concise guide to implementing CD using Jenkins Pipelines and deploying to Kubernetes.

Jenkins Pipelines

What is Jenkins?

Jenkins is an open-source automation server that facilitates continuous integration and continuous delivery (CI/CD). It automates the parts of software development related to building, testing, and deploying, allowing for reliable delivery of applications.

Setting Up Jenkins

1. **Install Jenkins**:

   - Download and install Jenkins from the [official website](https://www.jenkins.io/download/).

   - Start the Jenkins service and complete the initial setup wizard.

2. **Install Necessary Plugins**:

   - Navigate to `Manage Jenkins` > `Manage Plugins`.

- Install `Pipeline`, `Kubernetes`, and `GitHub` plugins.

3. **Create a Pipeline Job**:

- Go to `New Item`, enter a name, select `Pipeline`, and click `OK`.

- Configure the pipeline by defining the source code repository and the pipeline script.

Writing a Jenkins Pipeline

A Jenkins Pipeline defines the stages of your CI/CD process in code. Below is an example of a simple pipeline script in Groovy:

```groovy
pipeline {
    agent any
    stages {
    stage('Checkout') {
    steps {
        git 'https://github.com/your-repo.git'
    }
    }
    stage('Build') {
    steps {
        sh 'mvn clean package'
        }
    }
```

```
        stage('Test') {
        steps {
                sh 'mvn test'
        }
        }
        stage('Deploy to Kubernetes') {
        steps {
                script {
                kubernetesDeploy(
                        configs: 'k8s-deployment.yaml',
                        kubeConfig: [path: 'path/to/kubeconfig']
                )
                }
        }
        }
        }
}
```

Key Pipeline Concepts

- **Stages**: Define different steps of the CI/CD process (e.g., build, test, deploy).

- **Steps**: Individual tasks performed in each stage.

- **Agent**: Defines where the pipeline or a specific stage will run.

Mastering AWS and DevOps

Deploying to Kubernetes

What is Kubernetes?

Kubernetes is an open-source platform designed to automate deploying, scaling, and operating application containers.

Preparing Kubernetes Cluster

1. **Set Up a Kubernetes Cluster**:

   - Use a managed Kubernetes service like Amazon EKS, Google Kubernetes Engine (GKE), or set up a local cluster with Minikube or Kind.

2**. Configure kubectl**:

   - Ensure `kubectl` is installed and configured to interact with your Kubernetes cluster.

Kubernetes Deployment Configuration

Create a Kubernetes deployment YAML file (`k8s-deployment.yaml`) to define the application deployment:

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-app
```

```
spec:
  replicas: 3
  selector:
        matchLabels:
        app: my-app
  template:
        metadata:
        labels:
        app: my-app
        spec:
        containers:
        - name: my-app
        image: my-app:latest
        ports:
        - containerPort: 80
---
apiVersion: v1
kind: Service
metadata:
  name: my-app-service
spec:
  type: LoadBalancer
  ports:
  - port: 80
```

```
    targetPort: 80
  selector:
    app: my-app
```

Deploying via Jenkins

In the Jenkins pipeline, the `kubernetesDeploy` step deploys the application to Kubernetes:

1. Add Kubernetes Credentials:

- Go to `Manage Jenkins` > `Manage Credentials` and add Kubernetes credentials.

2. Use kubernetesDeploy in Pipeline:

- Ensure the pipeline script references the correct kubeconfig path and deployment YAML file.

Monitoring and Rollbacks

- **Monitor Deployments**: Use `kubectl get pods` and `kubectl get services` to monitor the deployment status.

- **Rollback**: If necessary, rollback to a previous deployment version using `kubectl rollout undo deployment/my-app`.

- **Tasks**: Create a Jenkins pipeline for continuous deployment

- **Reference**: [Jenkins X Documentation](https://jenkins-x.io/docs/)