

Project Description: Banking Application Deployment using Kubernetes, Ansible, and Jenkins

In this project, I worked on deploying a banking application using a combination of Kubernetes, Ansible, and Jenkins. The goal was to automate the deployment process and ensure scalability and reliability.

Step 1: Access Jenkins application

I utilized Jenkins, a popular open-source automation server, as the foundation for our deployment pipeline. I gained access to the Jenkins application and configured it to suit our project's requirements.

Step 2: Prepare Jenkins pipeline script

I created a Jenkins pipeline script that defined the deployment process. The pipeline started with basic stages such as code checkout and Maven build automation. This ensured that the application's source code was retrieved from the repository and compiled correctly.

Step 3: Integrate Docker image build stage

To containerize the application, I integrated a Docker image build stage into the Jenkins pipeline. This stage used a Dockerfile to build the application into a container image, which encapsulates the application and its dependencies.

Step 4: Create Docker Hub Credentials

To publish the Docker image, I created Docker Hub credentials within Jenkins. This involved providing the necessary authentication details to Jenkins, enabling it to push the built Docker image to a Docker Hub repository.

Step 5: Integrate Docker Image publish

Within the Jenkins pipeline, I integrated a stage to publish the Docker image to the Docker Hub repository. This ensured that the built image was available for deployment onto the target environment.

Step 6: Setting up Kubernetes Cluster and Configure Deploy stage

To deploy the application, I set up a Kubernetes cluster. This involved provisioning and configuring the necessary nodes and components to create a scalable and reliable environment for running containers. I then configured the deploy stage in the Jenkins pipeline to interact with the Kubernetes cluster.

Step 7: Configure Ansible Playbook for Deployment

To facilitate the deployment of the containerized application onto the Docker host, I configured an Ansible playbook. The playbook contained the necessary instructions to set up the Docker host, pull the Docker image from the Docker Hub repository, and run the containerized application.

Step 8: Execute Jenkins Build

With all the pipeline stages and configurations in place, I executed the Jenkins build. Jenkins automatically triggered the deployment process, pulling the latest code, building the Docker image, publishing it to Docker Hub, and deploying the application onto the Kubernetes cluster using Ansible.

Step 9: Access Deployed Application on Docker Host

Upon successful completion of the Jenkins build, the banking application was deployed onto the Docker host within the Kubernetes cluster. I verified the deployment by accessing the application using the appropriate URL or IP address associated with the Docker host.

Throughout this project, I demonstrated proficiency in Jenkins, Docker, Kubernetes, and Ansible. I successfully automated the deployment process, ensuring a streamlined and efficient workflow for the banking application. This experience showcases my skills in DevOps practices, containerization, and infrastructure automation.