**DEVOPS PROJECT – CI/CD**

**Requirements:**

Implement a Devops lifecycle using the following specifications

1. Install the necessary software on the machines using a configuration management tool

2. Git workflow has to be implemented

3. CodeBuild should automatically be triggered once a commit is made to master branch or develop branch.

a. If a commit is made to master branch, test and push to prod

b. If a commit is made to develop branch, just test it, do not push to prod

4. The code should be containerized with the help of a Dockerfile. The Dockerfile should be built every time there is a push to GitHub. The code should reside in '/var/www/html'

5. The above tasks should be defined in a Jenkins Pipeline with the following jobs:

Job1 : Build

Job2 : Test

Job3 : Prod

**Implementation:**

Softwares to be installed - Jenkins, Java, Docker

1. Create 3 instances Master, slave1, slave2. On Master, install ansible manually using below commands

A black background with white text

Description automatically generated

1. Update the slave machines using sudo apt update
2. To establish connection between master and slaves, create a keypair using ssh-keygen

A computer screen shot of a black screen

Description automatically generated

1. Copy the contents of id\_rsa.pub from master and paste it in authorized\_keys file in .ssh folder of slave nodes

A black rectangular object with white text

Description automatically generated

1. Add the private IP address of slave nodes in the master node’s hosts file

Sudo nano /etc/ansible/hosts

A computer screen with blue text

Description automatically generated

1. Ping slave machines to see whether connection is successful

A screenshot of a computer screen

Description automatically generated

1. Create an ansible playbook to install Jenkins and java on the master node. Write 2 scripts – one to install softwares on master machine and the other to install softwares on slave machines.

A screenshot of a computer program

Description automatically generated

A screen shot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

1. Execute the ansible playbook after syntax check

A screenshot of a computer

Description automatically generated

1. Verify that Jenkins has been installed on the master node. (public IP of master with port 8080)

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. In the Jenkins dashboard, configure slave nodes

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Add credentials for the slave node using pem file

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Configure another node as slave2 for prod instance

A screenshot of a computer

Description automatically generated

A person standing in front of a white background

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Go to GitHub and fork the repository - <https://github.com/hshar/website.git>

A screenshot of a computer

Description automatically generated

1. Create a docker file in Github and commit changes – Add file -> create new file

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Create 2 branches master and develop (master already exists)

A screenshot of a computer

Description automatically generated

1. Create jobs in the Jenkins dashboard

A screenshot of a computer

Description automatically generated

Job1 – slave1 – develop branch

Job2 – slave1 – master branch

Job3 – slave2 – master branch

A screenshot of a web page

Description automatically generated

A close-up of a line

Description automatically generated

1. Once it is saved, click on build now and verify that the repository has been copied to test machine

A screenshot of a computer

Description automatically generated

A screen shot of a computer code

Description automatically generated

1. Build the Docker file in Jenkins dashboard using shell script. Go to job1 and click configure

A screenshot of a computer

Description automatically generated

1. Click on build now and verify the webpage on port 83 of the test machine

A screenshot of a computer program

Description automatically generated

A cartoon of a cat

Description automatically generated

1. Configure webhooks in GtHub for automated deployment. Payload URL will be Jenkins IP address. Also enable ‘GitHub hook trigger for GITSCM polling’ in Jenkins dashboard under Build triggers

A screenshot of a webhook

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Verify whether the job gets triggered automatically when changes are made to develop branch

A screenshot of a computer code

Description automatically generated

A cartoon of a cat

Description automatically generated

1. Now create Job2 similar to Job1

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Build job2 to verify if it is successful on port 82

A screenshot of a computer

Description automatically generated

1. Edit the index.html file on master branch to see if the changes are reflected automatically

A screenshot of a computer

Description automatically generated

A cartoon of a cat

Description automatically generated

1. Configure Job3 similar to Job2

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Build job3 and verify it on port 80

A cartoon of a cat

Description automatically generated

1. **Project statement** - If a commit is made to master branch, test and push to prod.

Verified that the master branch commits are reflected on both test and prod machines.

1. **Project statement** - If a commit is made to develop branch, just test the product, do not push to prod.

Verified that the develop branch commits only affect the test machine. Changes are not reflected on prod machine.