Task 1: Clone the project from the GitHub link shared in resources to your local machine. Build the code using Maven commands.

Solution: I have taken the code from the Eureka link to my local windows machine where I installed git along with git bash.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

Task 2: Set up the Git repository and push the source code.

Then, log in to Jenkins

1. Create three jobs: • One for compiling source code • Second for testing source code • Third for packing the code

2. Set up CI/CD pipeline to execute the jobs created in the above step

3. Set up master-slave node to distribute the tasks in the pipeline.

Solution: I have set up the Git repository and pushed the source code

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

Created the cicd pipeline to compile,test and package the code by using command clean install package and build tool maven.

A computer screen shot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

CI PIPELINE GENERATED REPORTS:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

I HAVE MADE MASTER-SLAVE ARCHITECTURE BY USING THE LAUNCH METHOD AS LAUNCH AGENTS VIA SSH.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Task 3: Write a Docket file Create an Image and container on the Docker host. Integrate docker host with Jenkins. Create CI/CD job on Jenkins to build and deploy on a container

1. Enhance the package job created in step 1 of task 2 to create a docker image

2. In the Docker image add code to move the war file to the Tomcat server and build the image.

SOLUTION:

Integrated Dockerhost with Jenkins using command sudo usermod -aG docker Jenkins. So that Docker command cab be executed on Jenkins user. Created DOCKER FILE in git hub and build the dockerfile using jenkins CICD to build a docker image.

Dockerfile:

A screenshot of a computer

Description automatically generated

Docker hub password configured and encrypted:

A screenshot of a computer

Description automatically generated

Dockerfile is build using Jenkins Execute shell in Build-steps:

A screenshot of a computer

Description automatically generated

After the docker image is build it is run as container with host port as 1234 and container port is 8080 and able to access the application from the browser using the serverip and host port.

A screen shot of a computer

Description automatically generated

OUTPUT AFTER DOCKER IMAGE IS RUN AS A CONTAINER USING HOSTPORT:

A screenshot of a computer

Description automatically generated

And the war file is also deployed to the slave server on which tomcat is installed,using the deploy to container plugin.

BUILD RESTRICTED TO SLAVE MACHINE:

A screenshot of a computer

Description automatically generated

DEPLOY TO CONTAINER CONFIGURATION:A white paper with black lines

Description automatically generated with medium confidence

A screenshot of a login page

Description automatically generated

A screenshot of a computer

Description automatically generated

OUTPUT USING TOMCAT AFTER DEPLOYING WARFILE USING DEPLOY TO CONTAINER PLUGIN:

A screenshot of a computer

Description automatically generated

Task 4: Integrate Docker host with Ansible. Write an Ansible playbook to create an image and create a continuer. Integrate Ansible with Jenkins. Deploy Ansible-playbook. CI/CD job to build code on ansible and deploy it on docker container

Deploy Artifacts on Kubernetes

a. Write pod, service, and deployment manifest file

b. Integrate Kubernetes with Ansible

c. Ansible playbook to create deployment and service

SOLUTION: Ansible-playbook is written to create a docker image and Kubernetes deployment. Kubernetes deployment and service manifest files are written and placed in github repository along with source code and ansible-playbook.

DockerHub link: <https://github.com/NNUTAN07/EdurekaProject/>

Ansible-plugin is installed in Jenkins to execute the ansible-playbook through Jenkins

INVOKE ANSIBLE-PLAYBOOK AND DETERMINE PATH IN BUILD-STEPS OF JENKINS

A screenshot of a computer

Description automatically generated

ANSIBLE INSTALLATION PATH DETERMINED IN MANAGE JENKINS TOOLS

A screenshot of a computer

Description automatically generated

A black screen with many small colored text

Description automatically generated with medium confidence

OUTPUT:

A screenshot of a computer

Description automatically generated

Task 5: Using Prometheus, monitor the resources like CPU utilization: Total usage, usage per core, usage breakdown, memory, and network on the instance by providing the endpoints on the local host. Install node exporter and add URL to target in Prometheus.

Solution: I have installed the Prometheus and node exporter using the script got by the trainer.

Script link: <https://github.com/NNUTAN07/cloud-devops/tree/main/Prometheus/scripts>

Modified the Prometheus.yml by adding the node exporter and then restarted the Prometheus.

A black screen with many small white text

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated