Deployment #5

Name: Biki Gurung

Date: 11/14/2022

**Description:**

Demonstrate your ability to deploy a containerized application.

**Deployment Instructions:**

<https://github.com/kura-labs-org/kuralabs_deployment_5/blob/main/Deployment-5_Assignment.pdf>

Table of Contents

[Install Jenkins on EC2 3](#_Toc119355758)

[Install Terraform on a EC2 Server: 3](#_Toc119355759)

[Configure Credentials on Jenkins GUI: 3](#_Toc119355760)

[Install Docker on EC2 4](#_Toc119355761)

[Configure and connect a Jenkins agent to Jenkins: 5](#_Toc119355762)

[Create your Jenkinsfile: 8](#_Toc119355763)

[Issues faced during working on this deployment: 10](#_Toc119355764)

[Modifications to add: 10](#_Toc119355765)

[Pipeline Diagram: 10](#_Toc119355766)

|  |
| --- |
| Install Jenkins on EC2 |
| Install Terraform on a EC2 Server:   wget -O- https://apt.releases.hashicorp.com/gpg | gpg --dearmor | sudo tee /usr/share/keyrings/hashicorp-archive-keyring.gpg  echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg]  https://apt.releases.hashicorp.com $(lsb\_release -cs) main" | sudo tee /etc/apt/sources.list.d/hashicorp.list  sudo apt update && sudo apt install terraform  Link: <https://www.terraform.io/downloads> |
| Configure Credentials on Jenkins GUI: In the Jenkins Dashboard, click on manage Jenkins and then select Manage Credentials:  Graphical user interface, text, application  Description automatically generated with medium confidence  Now Select global:  Graphical user interface, application  Description automatically generated  On the right-hand side, select Add Credentials:  A picture containing logo  Description automatically generated |
| Then enter the First Credentials:   * Select “Secret text” for kind * Scope should be Global * Secret: Copy and Paste your AWS access Key * ID: AWS\_ACCESS\_KEY * Select Create   Then enter the Second Credentials:   * Select “Secret text” for kind * Scope should be Global * Secret: Copy and Paste your AWS secret Key * ID: AWS\_SECRET\_KEY * Select Create |
| Install Docker on EC2 Link: <https://docs.docker.com/engine/install/ubuntu/>  $ sudo apt-get update  $ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin  Verify that the Docker Engine installation is successful by running the hello-world image:  $ sudo docker run hello-world |

|  |  |
| --- | --- |
| Configure and connect a Jenkins agent to Jenkins: For example, below: Create 2 Agent:   1. Docker Agent 2. Terraform Agent | |
| page2image57780176 | Enter your Jenkins server  Select the Build Executor Status | |
| page3image57721040 | - Next Select “+ New Node” to configure and add the agent. | |
| page4image57690768 | | - Enter the node name “awsDeploy” and select “Permanent Agent” and then create. |
| page6image41184752 | | Now enter the configurations below:  ○  Name: awsDeploy  ○  Description: Deployment server  ○  Number of executors: 1  ○  Remote root directory: /home/ubuntu/agent  ○  Labels: aweDeploy  ○  Usage: only build jobs with label....  ○  Launch method: launch agents via ssh  ○  Host: {Enter the public IP of your EC2 in the Public subnet and not this text}  ○  Credentials: see below  ○  Host key verification strategy: non verifying verification strategy  Availability: keep this agent online as much as possible |
| page7image57759424 | | Credential steps:  ○  Select “Add” => “Jenkins”=>Kind:”SSH username  with private key”  ○  Enter the ID, Description, username  ○  To add the key, select “Enter Directly” => select  “add” => paste the private key into the white box and save. |
| page9image57750528 | | Save the configurations and wait for Jenkins to connect to the agent. It should look like what you see below: |
| Create your Jenkinsfile: Your Jenkinsfile should have the following stages:  ○ Build  ○ Test ○ Create - container (Docker agent) ○ Push - to Dockerhub (Docker agent)  ○ Deploy - to ECS (Terraform agent)  Using the Docker Agent:  We will create an image of our URL\_SHORTENER using docker build and using the docker push we will push the image to the DockerHub where the repository. Has been created.  Graphical user interface, text, application, email  Description automatically generated | | |
| Using the TerraformAgent we will pull the image and deploy the application to the AWS ECS hosted inside the AWS Fargate.  Graphical user interface, application  Description automatically generated | | |
| Issues faced during working on this deployment:  1. I couldn’t run any of the docker commands and push the image to the dockerhub using the jenkinsfile. Got Error message below:   Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post  To address this problem all the commands had to be run using sudo. | | |
| Modifications to add:  * Duplicate the env from one AZ to another without doing it manually. * Use auto scale to add more resources if required | | |
| Pipeline Diagram: Diagram  Description automatically generated | | |