Chanesh Mahadeo
Kura Labs
11/13/2022

Deployment 5

Intro

Welcome to Deployment 5. In this deployment we will be deploying a container to Amazon ECS. This will be done using our Jenkins server to command two amazon EC2 which serve as agents. One agent will be the Terraform server and the other will be our Docker Server. We will first be creating our image within the Docker Server and then pushing the image to dockerhub. We will then tell our terraform agent to pull that image down and deploy it to ECS.

Steps

Jenkins

This deployment assumes you have an active Jenkins server already running. If not, please defer to previous documentation for steps on how to do this: https://github.com/Chaneshm/kuralabs_deployment_3/blob/main/KuraDeployment3Documentation.pdf

**** Note that previous documentation will have extra steps not related to this deployment and this is only a reference to initializing the Jenkins server. ****

Terraform

Our next server we must have running is Terraform. This will be used for deploying our image to Amazon ECS. Below I have provided a userdata script for launching this server within an Amazon EC2. This script will also be located within this github repo.

UserData for UBUNTU EC2:

```
#!/bin/bash

sudo apt-get update && sudo apt-get install -y gnupg
software-properties-common
wget -O- https://apt.releases.hashicorp.com/gpg | \
    gpg --dearmor | \
    sudo tee /usr/share/keyrings/hashicorp-archive-keyring.gpg
gpg --no-default-keyring \
    --keyring /usr/share/keyrings/hashicorp-archive-keyring.gpg \
    --fingerprint
```

```
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-get install terraform -y
```

****IF THIS USERDATA DOES NOT WORK PLEASE REFER TO HASHICORP DOCUMENTATION ON INSTALLATION STEPS: https://developer.hashicorp.com/terraform/tutorials/aws-get-started/install-cli ****

Docker

Our next step is to install docker onto another EC2 as we will need to build the image using a dockerfile. This will then be pushed to Dockerhub so that our terraform server can pull and deploy it to ECS.

Userdata for Ubuntu EC2:

```
#!/bin/bash

sudo apt-get update
sudo apt-get install \
    ca-certificates \
    curl \
    gnupg \
    lsb-release
sudo mkdir -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg
--dearmor -o /etc/apt/keyrings/docker.gpg
echo \
  "deb [arch=$(dpkg --print-architecture)
signed-by=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/ubuntu \
  $(lsb_release -cs) stable" | sudo tee
/etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
sudo apt-get install docker-ce docker-ce-cli containerd.io
docker-compose-plugin -y
```

**** Please note that if this does not work you can refer to dockers documentation on installation steps: https://docs.docker.com/engine/install/ubuntu/ ****

IAM User

For this deployment we will be using the same IAM User we created in previous deployments, as this already has administrative access, but note that you should create one with just permissions to deploy onto ECS in a production environment.

Deploy

This repository should have already been formatted with the correct files for this procedure. To now test this deployment our yourself you must configure your Jenkins agents to your Terraform server and your Docker server. You will also need to install the docker pipeline plugin to your Jenkins server.

Steps:

- Install Docker Pipeline Plugin
- Add a new Node to create Docker agent
- Configure the following information:
- Name: dockerDeploy
- Description: Docker deployment server
- Remote root directory: /home/ubuntu/agent
- Labels: dockerDeploy
- Usage: Only build jobs with label…
- Launch method: Launch agent via SSH
- Host: {IP of Docker EC2}
- Credentials: {Global Credential with AWS Secret and Access Key}
- Host Key Verification Strategy: Non verifying…
- Availability: Keep this agent online as much as possible
- **Repeat this process for a terraform agent but change the IP**

Configure TF files

The terraform files will need configuring in order to deploy your container on to ecs. These lines are lines 31,48,70,71 in the main.tf file within the intTerraform folder of this repo.

The next step would be to clone into this repo on your Jenkins server with git clone https://github.com/Chaneshm/kuralabs_deployment_5. You can then do all the previous configs and do your Jenkins build. During the 60s sleep of the Jenkins step, you can check to see your app running!

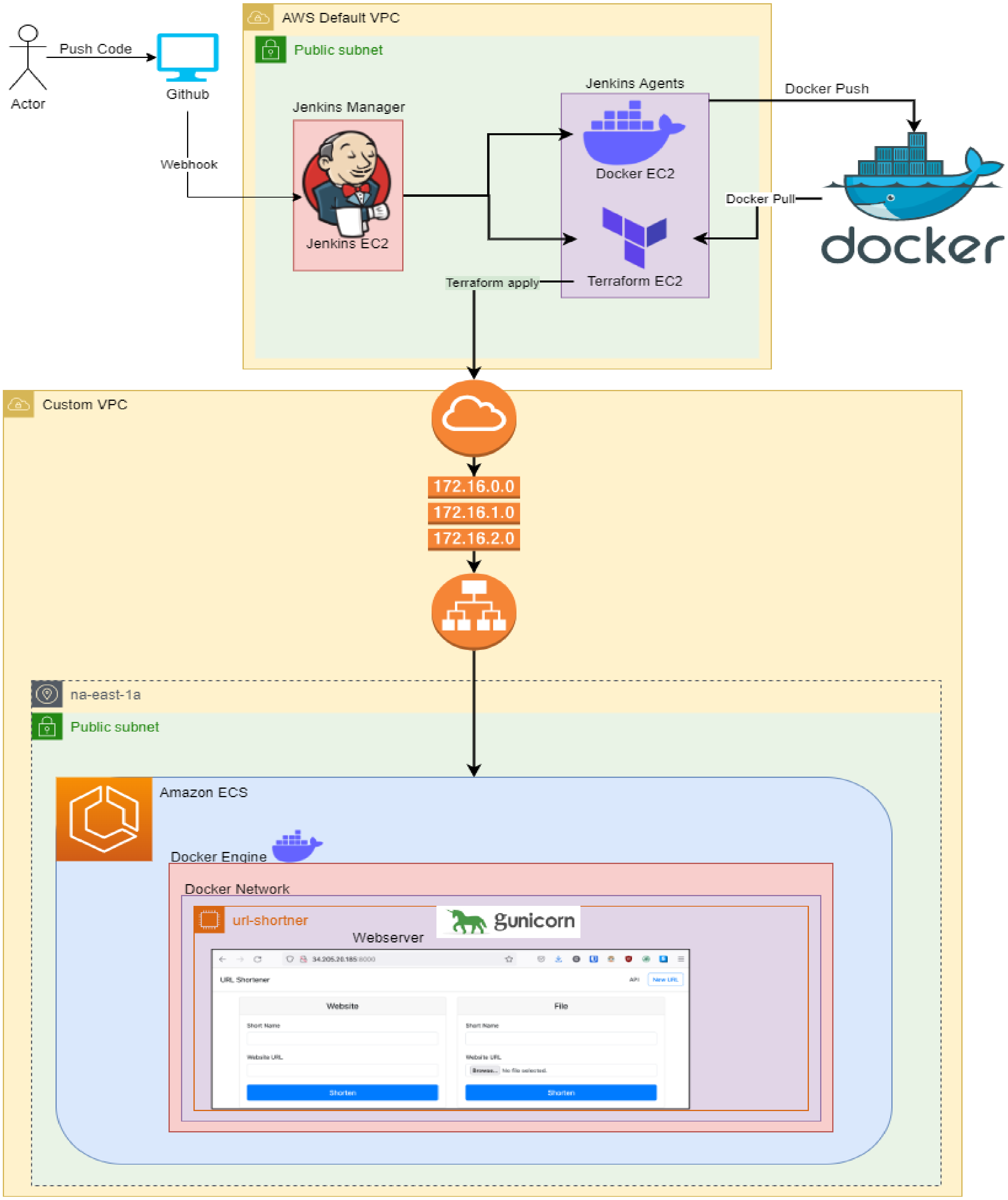
Changes

Reformatted the repo and also created the images using a compose yaml instead of using a dockerfile. All other previous changes also applied to this one. Needed to install docker pipeline plugin.

Issues

Couldn’t figure out how to have two containers communicate in a cloud setting. Once resolved, will deploy fullstack application onto a cloud service using same pipeline.

Diagram


Push Code
Actor
Github
Webhook
AWS Default VPC
Public subnet
Jenkins Manager
Jenkins EC2
Jenkins Agents
Docker EC2
Terraform EC2
Docker Push
Docker Pull
docker
Terraform apply
Custom VPC
172.16.0.0
172.16.1.0
172.16.2.0
na-east-1a
Public subnet
Amazon ECS
Docker Engine
Docker Network
url-shortner
gunicorn
Webserver
34.205.20.185:8000
URL Shortener
API
New URL
Website
File
Short Name
Website URL
Browse... No file selected.
Shorten
Shorten