Chanesh Mahadeo
10/14/22
Kura Labs

Deployment 3

Intro

This deployment demonstrates how we can use jenkins and a jenkins agent to remotely deploy services to another VPC. To begin we must first create our jenkins server and then we will set up the agent in order to orchestrate the deployment.

Setup

For this section we will be configuring a jenkins server and installing dependencies on our AWS EC2 before configuring our AWS settings. To further consolidate this part and speed up our setup process I have created a script designed to automatically do this.

NOTE THAT SECURITY GROUP FOR EC2 MUST HAVE 22 80 AND 8080 OPEN. OPTIONALLY ALSO 5000 TO ALLOW HOSTING OF THE FLASK SERVER

```
#!/bin/bash

# This script is meant to be used as userdata for launching a jenkins ec2
for deployment 2

if [ $UID != 0 ]; then
   echo "Run again with admin permissions"
   exit 1
fi
wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | gpg
--batch --yes --dearmor -o /usr/share/keyrings/jenkins.gpg
sh -c 'echo deb [signed-by=/usr/share/keyrings/jenkins.gpg]
http://pkg.jenkins.io/debian-stable binary/ >
/etc/apt/sources.list.d/jenkins.list' && echo "Jenkins Repo Added"
apt-get update
apt-get install default-jre -y && apt-get install jenkins -y && apt-get
install python3-pip -y && apt-get install python3.10-venv -y
apt-get install unzip -y
systemctl start jenkins && sleep 5s
usermod -aG sudo jenkins
sudo su jenkins
```

```
cd ~
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o
"awscliv2.zip"
unzip awscliv2.zip && sudo ./aws/install
pip install awsebcli --upgrade --user
echo "export PATH=~/.local/bin/:$PATH" > .bashrc
source ./.bashrc



exit 0
```

This code is responsible for:

- Creating and starting the Jenkins server
- Installing aws and aws cli
- Configuring the path to allow jenkins user to use “eb”

** Please note that this is a previous script that also installs awsebcli which also will allow us to deploy using elastic beanstalk but for this deployment we will omit this step. This script will still initialize our Jenkins setup for preparation of this deployment.

You must now create an ubuntu EC2 instance within a separate VPC on a public subnet. This EC2 must have :

- Ports 22 and 5000 open
- Must be on a separate VPC than your jenkins server and within a PUBLIC subnet
- Must have a public IP
- Must install the dependencies: default-jre, python3-pip, python3.10-venv, nginx

Configuring Jenkins

To configure jenkins is a straightforward and simple process. And will allow us to set up our pipeline.

Steps:

- Download the plugin Pipeline keep running step
- Fork the deployment repo from https://github.com/kura-labs-org/kuralabs_deployment_3
- Go to your jenkins url with {Public IP of your ec2}:8080
- Ssh into your ec2 instance or use connect on aws site
- Run sudo cat with the password location given
- Input the password into the bar to unlock jenkins
- Install recommended plugins
- Create your user
- Select New item and create a multibranch pipeline
- For source select github and add your forked repo link

- For this example specifically I did not make my repo private so I do not need to set up a github access key.
- Once you validate and it says OK, complete the creation of the repo.
- You should see a build happening

We must now create an agent for the separate EC2 instance.

- Enter your Jenkins server and Select the Build Executor Status
- Next Select “+ New Node” to configure and add the agent. Enter the node name “awsDeploy” and select “Permanent Agent” and then create
- Now enter the configurations:
    - 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 running if possible
- 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
- Save the configurations and wait for Jenkins to connect to the agent.

Before you build your pipeline, SSH into the EC2 in your VPC and then nano into the “/etc/nginx/sites-enabled/default” file.

- server { listen 5000 default_server; listen [::]:5000 default_server;
- location / { proxy_pass http://127.0.0.1:8000; proxy_set_header Host $host; proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for; }

You should now be able to start your building and access your deployment. Congrats!

Issues

The initial documentation had some issues in keeping the deployment running due to Jenkins lacking a plugin which I corrected on my end. The jenkinsfile also had small tweaks which had to be done which was just terminating the scopes of certain functions.

Additions

I added another test in order to test whether the database as well as another to test that the routes were working.

Pipeline Diagram