

Writeup

DESCRIPTION

Project objective:

You have to develop an environment for Docker networking.

Background of the problem statement:

As you have worked on Docker containers previously, your manager has asked you to perform container scheduling over multiple hosts using Docker CLI and connect multiple hosts with Docker containers.

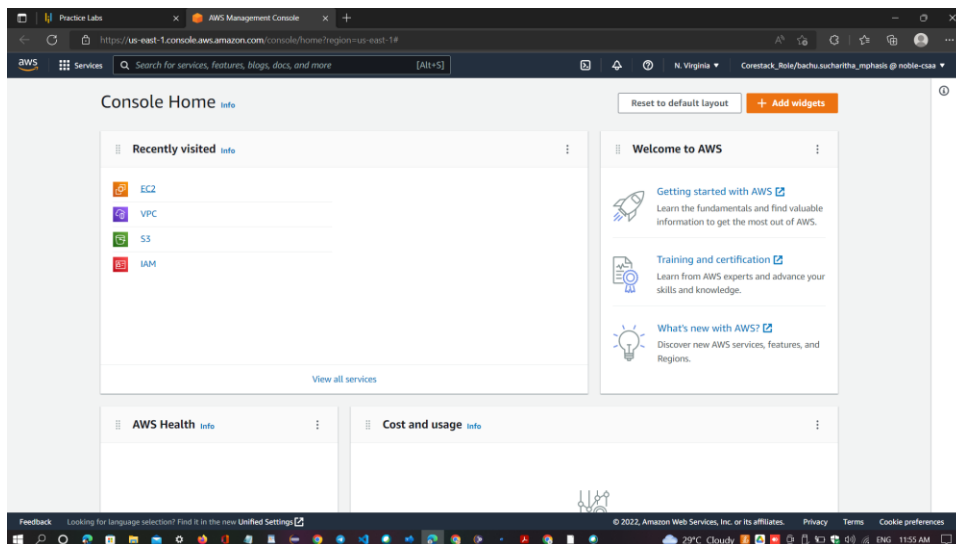
You must use the following:

- Jenkins: To create a pipeline to deploy Docker Swarm
- Docker Swarm: To implement container networking
- Git: To connect and push files from the local system to GitHub
- GitHub: To store the Angular application

Following requirements should be met:

- A few of the source code should be tracked on GitHub repositories. You need to document the tracked files that are ignored during the final push to the GitHub repository.
- Submission of your GitHub repository link is mandatory. In order to track your task, you need to share the link of the repository in the document.
- The step-by-step process involved in completing this task should be documented.

Steps:



The screenshot displays the AWS Management Console's 'Create new EC2 instance' wizard. The 'Name and tags' step is currently selected, showing a search bar for AMIs and a list of operating systems including Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE. The 'Summary' panel on the right provides a overview of the configuration: 1 instance, Canonical Ubuntu 22.04 LTS, t2.micro instance type, and free tier pricing. The 'Launch Instance' button is visible at the bottom right of the summary panel.

```
terminal@ip:172.31.87.204 - ssh
terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split Multitask Tunneling Packages Settings Help
Quick connect...
[+] 2. ubuntu@ip-172.31.87.204 - [x]
/home/ubuntu#
Name
bin
boot
cache
backlog
Audio
audio
Shoutcast
Selecting previously unselected package docker-scan-plugin.
Preparing to unpack .../docker-scan-plugin_0.17.0-ubuntu-jammy_amd64.deb ...
Unpacking docker-scan-plugin (0.17.0-ubuntu-jammy) ...
Selecting previously unselected package libltdl7:amd64.
Preparing to unpack .../libltdl7_2.4.6-15build2_amd64.deb ...
Unpacking libltdl7:amd64 (2.4.6-15build2) ...
Selecting previously unselected package libslrpdm4.
Preparing to unpack .../libslrpdm4_4.6.1-1build1_amd64.deb ...
Unpacking libslrpdm4 (4.6.1-1build1) ...
Selecting previously unselected package slirpnetns.
Preparing to unpack .../slirpnetns_1.0-1.2-build1_amd64.deb ...
Unpacking slirpnetns (1.0-1.2) ...
Setting up docker-scan-plugin (0.17.0-ubuntu-jammy) ...
Setting up contained (1.6.8-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /lib/systemd/system/containerd.service.
Setting up docker-compose-plugin (2.10.2-ubuntu-jammy) ...
Setting up libltdl7:amd64 (2.4.6-15build2) ...
Setting up docker-cell (5/20.10.18-3-0-ubuntu-jammy) ...
Setting up libslrpdm4 (4.6.1-1build1) ...
Setting up piz (2.6-1) ...
Setting up docker-ce-rootless-extras (5/20.10.18-3-0-ubuntu-jammy) ...
Setting up slirpnetns (1.0-1.2) ...
Setting up docker-ce (5/20.10.18-3-0-ubuntu-jammy) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /lib/systemd/system/docker.socket.
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu1.1) ...
Scanning processes ...
Scanning Linux images ...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172.31.87.204:~$ sudo docker -v
Docker version 20.10.18, build 4dc22fe
ubuntu@ip-172.31.87.204:~$
```

```
->sudo apt-get update
->sudo apt install default-jdk -y
```

INSTALL MAVEN

```
->sudo apt-get update
->sudo apt install maven -y
```

INSTALL JENKINS

->curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo tee /usr/share/keyrings/jenkins-keyring.asc > /dev/null

->echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] https://pkg.jenkins.io/debian-stable binary/ | sudo tee /etc/apt/sources.list.d/jenkins.list > /dev/null

->sudo apt-get update

->sudo apt-get install jenkins

TO START WITH JENKINS

->sudo service jenkins start

->sudo service jenkins status

CONNECT: goto> AWS>EC2>Copy Public IP:8080 on browser hit enter

TO GENERATE SECRET PASSWORD

->sudo cat /var/lib/jenkins/secrets/initialAdminPassword

To deal with permission denied error

->sudo chmod 777 /var/run/docker.sock

Now this will enable all the permissions

->copy the secret password to jenkins and start installation

FOR CONTINUOUS INTEGRATION (CI)

- Back to Dashboard and click on new item
- Go to newfile and enter pipeline
- And click ok
- Now we get a pipeline script
- Click on apply and save
- Now im going to build
- Console output:It shows Build Success

