

Assignment 01 : AWS & Docker

Steps to be Followed:

GIT-Hub : <https://github.com/akshaykumart/AWS-docker>

FILES AND COMMANDS USED:

1. main.tf
2. script.sh
3. jenkins.sh

STEPS :

1. Create a IAM user .
 - Login to AWS Console.
 - Create IAM user and add it to group with access permissions.
 - Note the Access key ,Secret key of the User you created.
2. Create a EC2 Instance .
 - Create a terraform script (main.tf).
 - Mention user credentials and script to create a Ec2 Instance.
 - Run the following commands to create .
\$ terraform init
\$ terraform fmt
\$ terraform plan
\$ terraform apply
3. Install Jenkins and configure it.
 - Run the following commands .
\$ sudo apt-get update
\$ sudo apt install openjdk-11-jdk -y
\$ sudo java -version
\$ wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add
\$ sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ >
/etc/apt/sources.list.d/jenkins.list'
\$ sudo apt update
\$ sudo apt install jenkins -y
\$ sudo systemctl start jenkins
\$ sudo systemctl enable jenkins
4. Create a Jenkins Job .

- Login to Jenkins by <publicip of instance>:8080
- Install all the related plugins from manage jenkins of dashboard.
- Go to new job -> create a new job
- In SCM mention the URL of GIT-Hub repository.
- In Build phase ,execute using shell scripting (use shell.sh from git hub repo).
- Apply and Save
- Build the job and validate the console output to success

5. Validate the Docker Images and Docker Container.

- Login to the instance.
- Create the script.sh file and paste the script
\$ nano script.sh
- Modify the access to execute for the file
\$ chmod +x script.sh
- Run the script
\$./script.sh
- Validate Docker images
\$ sudo docker images
- Validate Docker container
\$ sudo docker ps

FAQ / Troubleshooting :

- I. Mention the port in Inbound rules of Security Group of AWS if u face problem In launching Jenkins dashboard from Ec2-instance
Add ssh , HTTP, HTTPS , 8080
- II. If you get error in Jenkins job saying unable to get access to root user.
Add Jenkins user to Root access

```
#Defaults: sudo env_keep += "http_proxy https_proxy ftp_proxy all_proxy no_proxy"
# This allows running arbitrary commands, but so does ALL, and it means
# different sudoers have their choice of editor respected.
#Defaults: sudo env_keep += "EDITOR"
# Completely harmless preservation of a user preference.
#Defaults: sudo env_keep += "GREP_COLOR"
# While you shouldn't normally run git as root, you need to with etckeeper
#Defaults: sudo env_keep += "GIT_AUTHOR.* GIT_COMMITTER.*"
# Per-user preferences; root won't have sensible values for them.
#Defaults: sudo env_keep += "EMAIL DEEMAIL DEFULLNAME"
# "sudo scp" or "sudo rsync" should be able to use your SSH agent.
#Defaults: sudo env_keep += "SSH_AGENT_PID SSH_AUTH_SOCK"
# Ditto for GPG agent
#Defaults: sudo env_keep += "GPG_AGENT_INFO"
# Host alias specification
# User alias specification
# Cmnd alias specification
# User privilege specification
root    ALL=(ALL:ALL) ALL
# Members of the admin group may gain root privileges
%admin   ALL=(ALL) ALL
# Allow members of group sudo to execute any command
sudo    ALL=(ALL:ALL) ALL
jenkins ALL=(ALL) NOPASSWD: ALL
# See sudoers(5) for more information on "#include" directives:
@include /etc/sudoers.d
```

