1. **Setting up Terraform Server**

Created a terraform server instance

Connected to instance via ssh from remote desktop using Mobaxterm

-sudo yum update

- sudo yum-config-manager --add-repo <https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo>

-sudo yum -y install terraform

-terraform

-sudo hostnamectl set-hostname Terraform-Server

-sudo init 6(reboot the server)

**2. Setting up Jenkins server using Terraform**

-mkdir jenkins && cd Jenkins

- Create files – provider.tf, main.tf, security.tf, data.tf, variables.tf

-Create a terraform role(IAM) and attach to Terraform server.

-run terraform init

-terraform fmt

-terraform validate

-terraform plan

-terraform apply

**3. Install Jenkins**

**-**sudo su –

**-** **wget -O /etc/yum.repos.d/jenkins.repo** [**https://pkg.jenkins.io/redhat-stable/jenkins.repo**](https://pkg.jenkins.io/redhat-stable/jenkins.repo)

- rpm --import <https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key>

- yum upgrade

-amazon-linux-extras install epel

-amazon-linux-extras install java-openjdk11 -y

-yum install java-11-amazon-corretto -y

-yum install jenkins -y

-systemctl enable Jenkins

-systemctl start jenkins

- systemctl status jenkins

-hostnamectl set -hostname jenkins-server

-init 6

Sudo su –

-java –version

-javac --version

**4. Install & Configure Maven in Jenkins server**

-cd /opt

-wget <https://dlcdn.apache.org/maven/maven-3/3.9.6/binaries/apache-maven-3.9.6-bin.tar.gz>

- tar xzvf apache-maven-3.9.6-bin.tar.gz

- rm apache-maven-3.9.6-bin.tar.gz ? y

-mv apache-maven-3.9.6 maven

-cd maven/bin

- ./mvn -v (works only in bin folder)

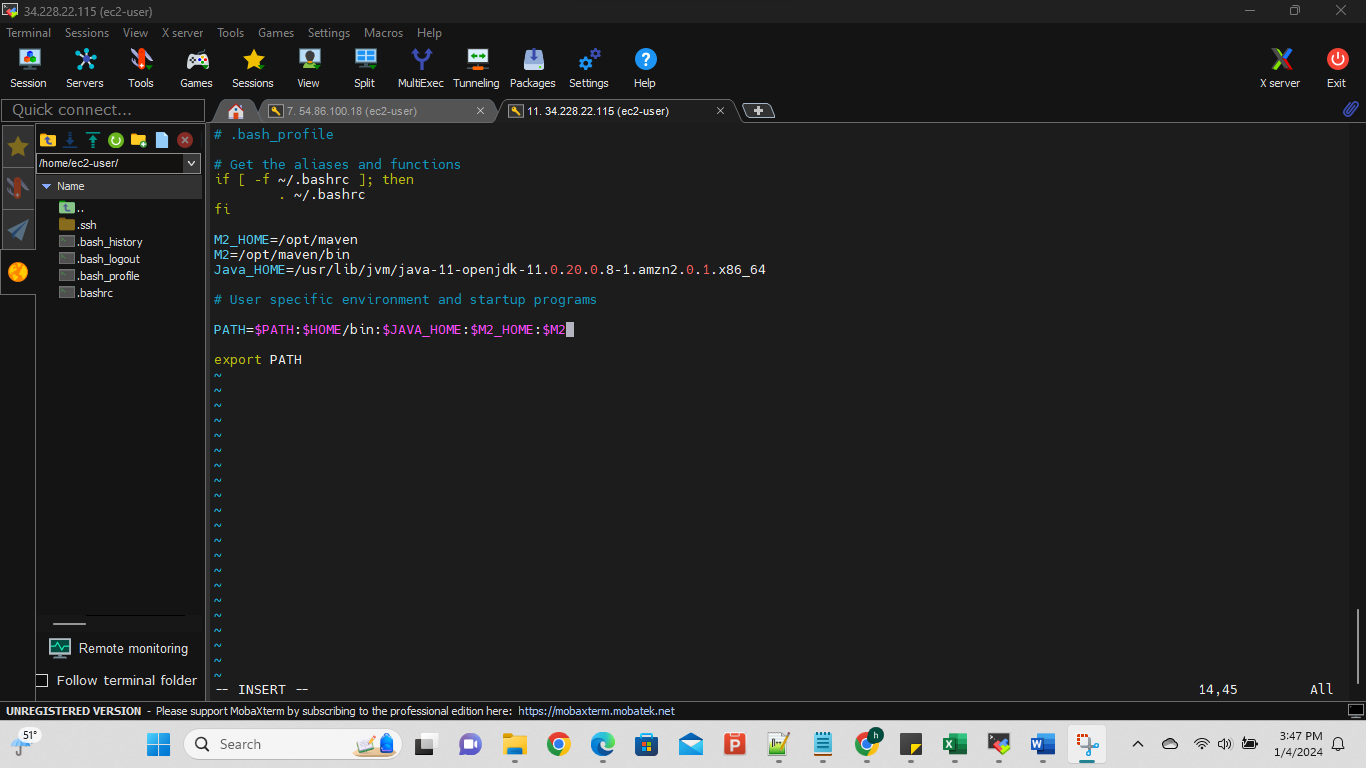
-cd ~

Ls -a

-find / -name java-11\*

-copy output-> /usr/lib/jvm/java-11-openjdk-11.0.20.0.8-1.amzn2.0.1.x86\_64 and paste in .bash\_profile

-vi .bash\_profile



* (Refresh path) -echo $PATH
* Mvn
* Mvn -v

**5. Configure Jenkins User Interface and Maven Integration**

Setup Jenkins with administrative pwd stored at cat /var/lib/jenkins/secrets/initialAdminPassword

Manage Jenkins-> plugins-> Available plugins-> Maven integration-> Install

Manage plugins-> Tools-> provide values for JAVA\_HOME and MAVEN\_HOME and save

Manage Plugins->installed plugins-> disable “Github Branch Source Plugin” and enable “Github plugin”

Restart Jenkins

In Jenkins server-> Install Git-> yum install git -y

**6. Create a test job**

**7. Provision Ansible server with terraform**

Go to terraform-Server

-cp -r Jenkins ansible

Modify all necessary files

-rm -rf .terraform .terraform.lock.hcl

-terraform init

-terraform fmt

-terraform validate

-terraform plan

-terraform apply

Now Ansible server is up and running. Connect via mobixterm

**8. Install and Configure Ansible**

sudo hostnamectl set-hostname ansible-server

-sudo su –

-useradd ansadmin

Passwd ansadmin

(gave admin)

-visudo

A screenshot of a computer

Description automatically generated

-cd /etc/ssh

-vi sshd\_config

Update password authentication to yes

A screenshot of a computer

Description automatically generated

-service sshd reload

-su ansadmin

-cd ~

-ssh key-gen

-ls .ssh/

(root user)- sudo su

-cd ~

-(installing ansible) amazon-linux-extras install ansible2

-ansible –version

**9. Integration Ansible with Jenkins**

Jenkins-> Manage Jenkins-> Available plugins->Install Publish over SSh & restart

Jenkins-> Manage Jenkins->System->Add SSh Servers

Name; Ansible-Server

Hostname: Ansible server ip addr

Username: ansadmin

Advanced-> check use password authentication->password of ansadmin-> apply-> save

**10. Install Docker in the Ansible Server**

**-**login into ansadmin

-cd /opt

-sudo mkdir docker

-sudo chown ansadmin:ansadmin docker

Go to Jenkins-> in test-> add post-build actions->Send build artifacts over SSH->Name: Ansible-Server

Source files: webapp/target/\*.war

Remove prefix: webapp/target

Remote dir: //opt//docker

Save

->Build now(test build)

🡪 u can see (\*.war) in ls docker

-cd docker

-sudo yum install docker

-systemctl start docker

**11. Create Project Dockerfile in the Ansible Server**

- vi Dockerfile

**12. Create Ansible Playbook for Docker Tasks**

-sudo vi /etc/ansible/hosts

(add [ansible] <private ip addr of ansible server>)

-sudo vi /etc/ansible/hosts

-ssh-copy-id <private ip addr of ansible server>

**-**vi register-ci.yml

-ansible-playbook register-ci.yml –check

-(Login to Docker) docker login -u heman12345

-pwd:\*\*\*\*\*\*

- ansible-playbook register-ci.yml

**13. Create CI Job**

Go to Jenkins-> create Maven project-> Give git-> Post build option->

A screenshot of a computer

Description automatically generated

**14. Provision EKS Server with Terraform**

-> in Terraform server

-cp -r ansible eks-server

Modify files

-rm -rf .terraform .terraform.lock.hcl

-terraform init

-terraform fmt

-terraform validate

-terraform plan

-terraform apply -auto-approve

Now server is available

Connect through mobaxterm

-sudo vi /etc/hostname-🡪 Change to EKS-Server & Restart the terminal

**15. Provision EKS Cluster using eksctl**

**-**sudo su –

(root) - curl -O <https://s3.us-west-2.amazonaws.com/amazon-eks/1.28.3/2023-11-14/bin/linux/amd64/kubectl>

- chmod +x ./kubectl

-mv kubectl /bin

**-**kubectl version

- curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl\_$(uname -s)\_amd64.tar.gz" | tar xz -C /tmp

-cd /tmp

-mv /tmp/eksctl /bin

-eksctl version

Add IAM Roles to EKS Server

Go to AWS Console-> Create new role with EC2FullAccess, AWSCloudFormationFullAccess, IAMFullAccess, AdminstratorAccess

-eksctl create cluster –name registerapp-cluster \

> -- region us-east-1 \

> --node-type t2.small

(Successful)

-Install AWS CLI

(From website <https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html> )

- curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip" -o "awscliv2.zip"

-unzip awscliv2.zip

-sudo ./aws/install

-aws

-aws eks update-kubeconfig –region us-east-1 –name registerapp-cluster

-kubectl get nodes

-kubectl get pods

-vi register-deplyment.yml

-vi register-service.yml

**16. Integrate EKS Server with Ansible**

- vi /etc/ssh/sshd\_config 🡪 PasswordAuth=yes

-passwd root

(gave root)

-service sshd reload

Copy private addr of EKS-Server and switch to Ansible-Server

-sudo su - ansadmin

-sudo vi /etc/ansible/hosts 🡪 Add priv addr of eks server here

-ssh-copy-id root@<public addr of EKS Server> 🡪 yes🡪 root password

-vi kube\_deploy.yml

-ansible-playbook kube\_deploy.yml –check 🡪 yes

- Ansible-playbook kube\_deploy.yml

Go to EKS-Server

-kubectl get pods

- kubectl get all

**17. Create Continuous Deployment Job on Jenkins**

Create new job -> Register-CD -> post build action-> exec cmd-> ansible-playbook /opt/docker/kube\_deploy.yml

-Build

**18. Integrate CI and CD Jobs**

Go to Register-CI job -> add poll SCM -> \* \* \* \* \*

-Add Post build action-> Build other projects-> Register-CD

Build and verify

**19. Deploy/test the CI-CD Configuration**

edit a readme file in git and verify

**20. Verify**

**-**kubectl get pods

-kubectl get svc

A computer screen with text

Description automatically generated

-copy link and paste in browser

-<link>:8080/register

**OUTPUTS:**

A screenshot of a computer

Description automatically generated

SUCCESSFUL

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

References:

1. <https://www.youtube.com/watch?v=f6AlNVui2yQ>
2. <https://www.youtube.com/watch?v=NKUOSc9pCfk&t=1853s>
3. Github repos:
4. <https://github.com/DinmaMerciBoi/MyProjectApp.git>
5. <https://github.com/jhawithu/hello-world>