1. Solution Architecture

Diagram

Description automatically generated with low confidence

1. Implementation

* I choose Terraform to create a Jenkins CI server.

git clone https://github.com/ahcankaya/Codingassignment

cd terraform/

terraform init

terraform plan

terraform apply -auto-approve

After run commands below Terraform will show the ip address of the Jenkins CI server

Terraform will run Jenkins installation sh file

react-and-spring-data-rest\resource\terraform-jenkins\install\_jenkins.sh

After run sh, the Jenkins page can open by using address below

<http://jenkins_CI_server_ip_address:8080>

* Configure Jenkins.
* (described in document (Documents\JenkinsInstallAndConfigurations.docx)
* Configure Jenkins job and pipeline.
* Create Docker
* Create deployments,services,volumes in Kubernates

**Ref:** <https://phoenixnap.com/kb/install-jenkins-ubuntu>

**Step 1: Firstly install JDK to the system:**

$sudo apt-get update

sudo apt-get install openjdk-8-jdk

**Step 2: Add the Jenkins Repository**

1. Start by importing the GPG key:

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

2. Next, add the Jenkins software repository to the sources list and provide the key for authentication:

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

To install Jenkins on Ubuntu, use the following commands:

sudo apt update

sudo apt install Jenkins

sudo systemctl status Jenkins

**Step 4: Modify Firewall to Allow Jenkins**

sudo ufw allow 8080

sudo ufw status

sudo ufw enable

**Step 5: Set up Jenkins**

1. To launch and set up Jenkins, open a web browser, and navigate to the IP address of your server:

http://ip\_address\_or\_domain:8080

Use the actual IP address or domain name for the server you’re using Jenkins on. For example, if you're running locally, use:

<http://localhost:8080>

2. You should see a page that prompts you to Unlock Jenkins. You’ll need the default password. You can get the default password by switching to a command line and entering the following:

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

3. The system returns an alphanumeric code. Enter that code, then click Continue.

4. Next, you are prompted to either Install suggested plugins or Select plugins to install. It’s fine to simply install the suggested plugins.

You can always install more plugins later. The system continues the initial Jenkins setup.

5. Then you will be prompted to Create First Admin User.

Enter the credentials you want to use for your Jenkins administrator, then Save and Continue.

6. After this, you should set up the Instance Configuration. This is the preferred network address for this Jenkins installation. Confirm the address you want to use for your server. This is most likely the same address you used to get to this configuration page. When you’re satisfied, click Save and Finish.

7. You should see a page that says Jenkins is ready!

**Step6: Connect Kubernetes from Jenkins UI I use “Kubernetes plugin for Jenkins”**

Ref: <https://plugins.jenkins.io/kubernetes/>

$ mkdir ~jenkins/.kube

$ cd /tmp

$ curl -LO https://storage.googleapis.com/kubernetes-release/release/`curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt`/bin/linux/amd64/kubectl

$ chmod +x ./kubectl

$ sudo mv ./kubectl /usr/local/bin/kubectl

$ sudo cp ~/.kube/config ~jenkins/.kube/

$ sudo chown -R jenkins: ~jenkins/.kube/

$ source <(kubectl completion bash)

$ kubectl get nodes

**Step7: Install Docker on Jenkins Server**

$ curl -fsSL https://get.docker.com -o get-docker.sh

$ sudo sh get-docker.sh

$ sudo groupadd docker

$ sudo usermod -aG docker jenkins

$ newgrp docker

**Step8: Create EKS environment**

Solution uses Docker container creation steps.

* Go to resources/terraform folder

$ cd resources/terraform

$ docker run -it --rm -v ${PWD}:/work -w /work --entrypoint /bin/sh amazon/aws-cli:2.0.43

$ yum install -y jq gzip nano tar git unzip wget

* Create new access key
* Install Terraform to container

$ curl -o /tmp/terraform.zip -LO https://releases.hashicorp.com/terraform/0.13.1/terraform\_0.13.1\_linux\_amd64.zip

$ unzip /tmp/terraform.zip

$ chmod +x terraform && mv terraform /usr/local/bin/

$ terraform

$ terraform init

$ terrafor plan

$ terraform apply

$ terraform state list

$ terraform output kubeconfig ~/.kube/config

$ aws eks update-kubeconfig --name eks-cluster --region ap-southeast-2

* Install Kubectl

$ curl -LO https://storage.googleapis.com/kubernetes-release/release/`curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt`/bin/linux/amd64/kubectl

$ chmod +x ./kubectl

$ mv ./kubectl /usr/local/bin/kubectl

$ kubectl get nodes

* Destroy Env(clean-up)

$ terraform destroy

$ terraform state list

**Step9: Use Jenkins in EC2 instance using Terraform**

Connect EC2 instance by using key pair(Jenkins-key)

$ cd resources/terraform-jenkins

$ terraform init

$ terraform plan

$ terraform apply

Terraform will show the Jenkin link address

Open browser and paste the link like : <http://jenkins_ip_addr:8080>

Using the AWS Cloud9 shell terminal, log in to the Jenkins CI server, find the Administrator password by running the following command:

$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword

Enter this password to wizard page and install suggested plugin(Docker and Git plugins)

After install plugins restart the Jenkins

From Jenkins console, click New item. Click Pipeline name it and click OK

Set git and docker credential

Jenkins build will check the git repo and find out the pipeline which is called JenkinsPipeline

**Step10: Elastic Cloud**

$ kubectl apply -f https://download.elastic.co/downloads/eck/1.2.1/all-in-one.yaml

$ kubectl get all -n elastic-system

Go to resource/Kubernetes path and run yaml files to create persistentvolume,persistentvolumeclaims and elasticsearch

$ cd resource/kubernetes

$ kubectl apply -f pves.yaml

$ kubectl apply -f pvces.yaml

$ kubectl apply -f elasticsearch.yaml