Project 6 - CI/CD

A diagram of a computer system

Description automatically generated with medium confidence

In this project, students will work in a team to deploy a CI/CD pipeline with Jenkins with build, test and deploy stages for a web app. The web app is also deployed as a container in a environment - e.g. via Docker, Docker Compose, AWS ECS, Kubernetes/AWS EKS, etc. The app will be updated on the developer’s local machine, and these changes to the app will be commited in Git and pushed to the main branch on a remote GitHub repo. This triggers a Jenkins CI pipeline to build the app as a new Docker image, test the new build and push the Docker image to Docker Hub. Finally, the Jenkins pipeline will then deploy the updated app as a container into a the container environment of choice with appropriate Dockerfiles, Docker compose, AWS ECS or Kubernetes manifest and configuration files.

* Warning:
  + You should use this scripts in production environment, you must read each script and try to have a deep understanding about these scripts. The permissions and AWS policies, GitHub Credentials, and Docker Hub are temporals.
  + The docker image is an Ubuntu Server with a Jenkins services running in background.

**Project Solution**

* The following are the steps to deploy any Python project in a K8S environment, thanks to EKS, EKSCTL, Terraform, and KubeCtl.
  + Download the project from <https://github.com/maxiplux/project-7-jenkins-to-eks> in your local machine
  + **Mandatory Preconditions**
    - You should have Terraform in your PATH environment.
    - You should have AWS cli in your local environment.
    - You should have a Git Account and a its Auth Token.
    - You should have a DockerHub Account and a its Auth Token.
    - You should work always in /tmp
    - All the commands must be run as ubuntu user ( default user)
    - You should try to understand each folder in project-7-jenkins-to-eks/ (<https://github.com/maxiplux/project-7-jenkins-to-eks>)
      * A screenshot of a computer

        Description automatically generated
  + Actions to deploy this project
    - Open your Terminal with aws cli ready in your .aws local configuration folder.
    - git clone <https://github.com/maxiplux/project-7-jenkins-to-eks.git>
    - cd aws-eks-permissions
      * terraform init
      * terraform apply -auto-approve
      * A screenshot of a computer program

        Description automatically generated
      * A computer screen with white text

        Description automatically generated
      * Copy in a notepad the aws\_access\_key\_id & aws\_access\_key\_id, you will need them to setup your Jenkins machine.
    - Go to the folder aws-jenkins-machine using your terminal.
      * This is your root project, and you should be able to see something like the image below.
      * A screenshot of a computer

        Description automatically generated
      * cd aws-jenkins-machine/terraform
        + Create your Jenkins machine.

Execute the command below

terraform apply -auto-approve

* + - * A black screen with white text

        Description automatically generated
        + This command will wait until to have Jenkins running, to achive that I have a validator “check\_status.sh” waiting for 403 status from the Jenkins Machine server.
      * Go to your Jenkins Machine console by your local terminal or by AWS console .
        + The PEM file to connect to this machine is your current folder, and the file name is “terraform-pem-ansible-dec04.pem”
      * The images below show how to connect by Aws console to your Jenkins Machine EC2.
  + A screenshot of a computer

    Description automatically generated
    - A screenshot of a computer

      Description automatically generated
  + Setup the docker Hub Token
    - Go to /tmp in your docker-machine by terminal, and edit the docker-compose.yml and then update your docker hub token with your token and change the username to your docker hub username.
    - You should change the variables below.
      * DOCKER\_PASSWORD=TOKEN\_HERE
      * DOCKER\_USERNAME=maxiplux
      * The code about this Jenkins image is in <https://github.com/maxiplux/project-7-jenkins-to-eks/blob/main/jenkins-image/Dockerfile>
  + A screenshot of a computer

    Description automatically generated
* Restart your container image
  + docker compose down
  + docker compose up -d
  + docker exec -it jenkins /etc/init.d/jenkins start
    - **This final command is mandatory to start the Jenkins Server.**
  + A screenshot of a computer program

    Description automatically generated
  + Setup your EKS CLUSTER
    - Download eksctl installer using your terminal
      * wget <https://raw.githubusercontent.com/maxiplux/project-7-jenkins-to-eks/main/aws-eks-cluster/eks-installer.sh>
    - set permissions.
      * chmod +x eks-installer.sh
* A screenshot of a computer

  Description automatically generated
  + - Install the eksctl
      * sh ./ eks-installer.sh
    - Create your EKS cluster in AWS
      * All these instructions should be run in your terminal.
      * Configure your AWS cli
        + *Update your AWS cli ( This is mandatory, because the default version is not working well)*

Execute this in your terminal with the user ubuntu

pip3 install awscli --upgrade –user

* + - * + Configure

Execute in your terminal

aws configure

with the secrets that you got thanks to the permissions script.

aws\_access\_key\_id = AKIA5I57XN36GP5J2XXA (You got this credentials thanks to the step aws-eks-permissions )

aws\_secret\_access\_key = Uc2Ft/FTwE4P4I75Ao/to8mV8RATGxRQW8s1ImDI(You got this credentials thanks to the step aws-eks-permissions )

region = us-east-1

* + - * Create your EKS cluster
        + Execute the command below in your terminal. You should wait between 10 minutes and 20 minutes until to finish it.

wget <https://github.com/maxiplux/project-7-jenkins-to-eks/blob/main/aws-eks-cluster/cluster.yaml>

eksctl create cluster -f cluster.yaml

* Deploy the first version of this project in your in K8S cluster.
  + Go project-7-jenkins-to-eks/ ( You should have this, because you are following the previous steps)
    - git clone <https://github.com/maxiplux/project-7-jenkins-to-eks.git>
    - cd project-7-jenkins-to-eks/
    - cd k8s/
  + A screenshot of a computer program

    Description automatically generated
  + Execute the following commands to deploy the first version of the container image in your cluster
    - kubectl apply -f namespace.yml
    - kubectl apply -f deployment.yml
    - kubectl apply -f service.yaml
  + A screen shot of a computer

    Description automatically generated

A screenshot of a computer

Description automatically generated

* + - If your Jenkins Container is running, please start your Jenkins container, following the commands below.
      * Execute the commands below.
        + cd /tmp
        + docker compose up -d
* A screenshot of a computer

  Description automatically generated
  + - * docker exec -it jenkins /etc/init.d/jenkins start
* 
  + Go to your container and setup the EKS cluster.
    - Do login as Jenkins user.
      * In your EC2 Jenkins Machine, execute the commands below
      * docker exec -it jenkins bash
      * su Jenkins
      * Execute this in your terminal with the user Jenkins in your Jenkins Container

pip3 install awscli --upgrade –user

This is mandatory to avoid AWS CLI errors.

* + - * 
    - Execute the following commands as Jenkins user, in your Jenkins Container Server
      * aws configure
        + aws\_access\_key\_id = AKIA5I57XN36GP5J2XXA
        + aws\_secret\_access\_key = Uc2Ft/FTwE4P4I75Ao/to8mV8RATGxRQW8s1ImDI
        + region = us-east-1
      * A screen shot of a computer code

        Description automatically generated
    - Setup your EKS cluster in your K8S config file in your Jenkins Container Server
      * aws eks update-kubeconfig --region us-east-1 --name weclouddata
      * test you connection to EKS
        + kubectl version
    - A screenshot of a computer code

      Description automatically generated
  + Open your browser targeting to the public Ip of your Jenkins machine and setup with default settings.
  + Clone the code in <https://github.com/maxiplux/api-python-project-devops-fast-api>
    - The branch for this project is project-6
    - <https://github.com/maxiplux/api-python-project-devops-fast-api/tree/project-6>
  + Create a multi pipeline and setup this project to read the branch project-6 and its Jenkins File.
  + A screenshot of a computer

    Description automatically generated
  + Install the Blue Ocean plugins for Jenkins, this is optional.  
    A screenshot of a computer

    Description automatically generated
  + You can find a full video about this project in <https://youtu.be/QwN1tP6YZs4>
  + Delete your EKS CLUSTER
    - In your EC2 MACHINE execute the command below.
      * eksctl delete cluster --region=us-east-1 --name=weclouddata --disable-nodegroup-eviction
  + Delete your aws credentials
    - In your EC2 MACHINE execute the command below.
    - Go to project-7-jenkins-to-eks/aws-eks-permissions
      * cd /tmp/project-7-jenkins-to-eks/aws-eks-permissions
      * terraform destroy -auto-approve
  + Delete your EC2 MACHINE
    - Go to project-7-jenkins-to-eks/aws-jenkins-machine
      * cd /tmp/project-7-jenkins-to-eks/aws-jenkins-machine
      * terraform destroy -auto-approve