**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.
* 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>
  + Preconditions
    - You should have Terraform in your PATH environment
    - You should have AWS cli in your local environment
  + Actions
    - Open your Terminal
    - 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
      * This is your root project
      * A screenshot of a computer

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

This is

terraform apply -auto-approve

* + - * A black screen with white text

        Description automatically generated
      * Go to AWS console by your local terminal or by AWS console
        + The PEM file to connect to this machine is yout current folder terraform-pem-ansible-dec04.pem
  + 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 and edit the docker-compose.yml and adding your docker hub token to this project and change the username to your docker hub username.
* 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
  + A screenshot of a computer program

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

  Description automatically generated
  + - Create your EKS cluster in AWS
      * All these instructions should we run in your terminal.
      * Configure your AWS cli
        + Update your AWS cli

pip3 install awscli --upgrade –user

* + - * + configure

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

aws\_access\_key\_id = AKIA5I57XN36GP5J2XXA

aws\_secret\_access\_key = Uc2Ft/FTwE4P4I75Ao/to8mV8RATGxRQW8s1ImDI

region = us-east-1

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

eksctl create cluster -f cluster.yaml

* + - Start your Jenkins container
      * 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 loging as Jenkins user.
      * su Jenkins
      * 
    - Execute the following commands as Jenkins user .
      * aws configure
        + aws\_access\_key\_id = AKIA5I57XN36GP5J2XXA
        + aws\_secret\_access\_key = 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
        + region = us-east-1
      * A screen shot of a computer code

        Description automatically generated
    - Setup your EKS cluster in your K8S config file
      * 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 JenkinsFile.
  + A screenshot of a computer

    Description automatically generated
  + Install the BlueOcean plugins for Jenkins, this is optional.