**Deployment 5**

**Deploying Terraform & Docker with Jenkins Agents and Containerizing a Python Application**

# Step 1

**Create 2 EC2**

* Log into AWS account:
  + Create EC2 and install Terraform:
    - Link to Terraform [Install | Terraform | HashiCorp Developer](https://developer.hashicorp.com/terraform/downloads)
  + Create EC2 and install Docker:
    - Link to Docker [Install Docker Engine on Ubuntu | Docker Documentation](https://docs.docker.com/engine/install/ubuntu/)

# Step 2

**ssh into Jenkins Server**

* ssh to the current Jenkins server
* Check to see if GitHub token is still active
* Set up Jenkins agents:
  + Docker
  + Terraform

Application

Description automatically generated with low confidence

# Step 3

Create your Jenkinsfile:

pipeline {

  agent any

   stages {

    stage ('Build') {

      steps {

        sh '''#!/bin/bash

        python3 -m venv test3

        source test3/bin/activate

        pip install pip --upgrade

        pip install -r requirements.txt

        export FLASK\_APP=application

        flask run &

        '''

     }

   }

    stage ('test') {

      steps {

        sh '''#!/bin/bash

        source test3/bin/activate

        py.test --verbose --junit-xml test-reports/results.xml

        '''

      }

      post{

        always {

          junit 'test-reports/results.xml'

        }

      }

    }

    stage ('Image Build/Push') {

       agent {label 'DockerAgent'}

       steps {

         sh '''#!/bin/bash

         docker build -t mallahdiv/url-shortener "https://raw.githubusercontent.com/mallahdiv/kuralabs\_deployment\_5/main/dockerfile"

         docker login --username mallahdiv -p Ankobia1#

         docker push mallahdiv/url-shortener

         '''

      }

    }

    stage('Init') {

       agent {label 'Terra Agent'}

       steps {

        withCredentials([string(credentialsId: 'AWS\_ACCESS\_KEY', variable: 'aws\_access\_key'),

                        string(credentialsId: 'AWS\_SECRET\_KEY', variable: 'aws\_secret\_key')]) {

                            dir('intTerraform') {

                              sh 'terraform init'

                            }

         }

    }

   }

      stage('Plan') {

       agent {label 'Terra Agent'}

       steps {

        withCredentials([string(credentialsId: 'AWS\_ACCESS\_KEY', variable: 'aws\_access\_key'),

                        string(credentialsId: 'AWS\_SECRET\_KEY', variable: 'aws\_secret\_key')]) {

                            dir('intTerraform') {

                              sh 'terraform plan -out plan.tfplan -var="aws\_access\_key=$aws\_access\_key" -var="aws\_secret\_key=$aws\_secret\_key"'

                            }

         }

    }

   }

      stage('Apply') {

       agent {label 'Terra Agent'}

       steps {

        withCredentials([string(credentialsId: 'AWS\_ACCESS\_KEY', variable: 'aws\_access\_key'),

                        string(credentialsId: 'AWS\_SECRET\_KEY', variable: 'aws\_secret\_key')]) {

                            dir('intTerraform') {

                              sh 'terraform apply plan.tfplan'

                            }

         }

    }

   }

    stage('Destroy') {

      agent {label 'Terra Agent'}

      steps {

        withCredentials([string(credentialsId: 'AWS\_ACCESS\_KEY', variable: 'aws\_access\_key'),

                        string(credentialsId: 'AWS\_SECRET\_KEY', variable: 'aws\_secret\_key')]) {

                            dir('intTerraform') {

                              sh 'terraform destroy -auto-approve -var="aws\_access\_key=$aws\_access\_key" -var="aws\_secret\_key=$aws\_secret\_key"'

                            }

          }

    }

   }

   }

}

# Step 4

**The Pipeline**

**Graphical user interface, application, table

Description automatically generated**

# Trouble Shooting

* The pipeline did not run at first because I tried to run the Terraform from the server that had Jenkins on it
* Had to create a third EC2 for Terraform
* Added Dockerhub credentials to Jenkins. Thought this was the way to use the Docker plugin. The build failed many times before I just decided to create a Jenkins agent
* Build failed more times. I decided to just use:
  + Command in the Jenkins file docker login --username mallahdiv -p Ankobia1#

# Diagram

Diagram

Description automatically generated

