**Deployment 2**

Goal to setup & CI/CD pipeline

## Step 1

**Activate the Jenkins user on the EC2:**

$sudo passwd Jenkins

$sudo su – Jenkins -s /bin/bash

## Step 2

**Create a Jenkins users in aws**

* Name the username EB-user
* Select programmatic access
* Attach existing policies: administrator access
* Copy and save access key ID and secret access key **(warning it must be done close out set-up for user)**

## Step 3

**Install AWS CLI on the Jenkins EC2 and configure**

$curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_6 4.zip" -o "awscliv2.zip"

$unzip awscliv2.zip

$sudo ./aws/install

$aws –version

$sudo su - jenkins -s /bin/bash

$aws configure

* + - - Set Access Key ID
    - - Set Secret Access Key
    - - Set region to: us-east-1
    - - Set Output format: json

## Step 4

**Install EB CLI in the Jenkins EC2 user**

$pip install awsebcli --upgrade –user

$eb –-version

## Step 5

**Connect Github to Jenkins Server**

* Fork the Deployment repo: https://github.com/kura-labs-org/kuralabs\_deployment\_2. git
* create an access token from GitHub:
  + - Navigate to your GitHub settings, select developer settings
    - Select personal access token and create a new token
    - Select the settings you see below for access token permissions

## Step 6

**Create a multibranch build**

* Log back into Jenkins and select “New item”
* Select multibranch pipeline
* Add a Branch source by selecting and source and select GitHub
* Click on Add and then select Jenkins
* Under username enter your GitHub username
* under password enter your token
* Table

  Description automatically generatedEnter your URL to the repository and you can validate by selecting validate
* Make sure this says Jenkinsfile
* Select Apply and then Save
* If you do not see a build happening, select Scan Repository

## Step 7

**Deploy application for Elastic Beanstalk CLI**

$sudo su - jenkins -s /bin/bash

$cd /var/workspace/url-shortner

$eb init

* Select: us-east-1
* Press enter
* Select: Python
* Select: (The latest version of python available
* Select: N (for CodeCommit)

$eb create

* Take the default for the next 3 questions by hitting enter (remember the environment name)
* Spot Fleet: No
* Wait for the environment to be made!! And then check it

## Step 8

**Add a deployment stage to the pipeline in 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 ('Deploy') {

steps {

sh '/var/lib/jenkins/.local/bin/eb deploy url-shortner-dev'

}

}

}

}

## Step 9

**Added CloudWatch**

* Select create alarm
* Select metric
* Select ElasticBeanstalk
* Select Environment Metric
* Select url-shortner-dev
* A screenshot of a computer

  Description automatically generatedSelect metric to create