**Creating an EC2**

Graphical user interface, text, application

Description automatically generated

Table

Description automatically generated

Selected Ubuntu as the AMI and opened ports 22, 80, and 8080

**Jenkins Setup**

Graphical user interface, text

Description automatically generatedSSH into the EC2 then run **setup\_jenkins.sh** to install Jenkins on the EC2, create a jenkins user, and activate the jenkins user by switching to it in Bash shell

Go to http://<public-ip>:8080

Graphical user interface, text, application

Description automatically generated

Use sudo cat to read the file on EC2 server that has the administrator password

Graphical user interface, application, Word

Description automatically generated

Install suggested plugins

Graphical user interface, text, application

Description automatically generated

Save and finish

**Creating a Jenkins user on my AWS Account**

Navigate to AWS > IAM > Access Management > Users > Add users

Graphical user interface, text, application, email

Description automatically generated

Enter a user name (i.e. EB-user) and select **Access key - Programmatic access** for “AWS access type”

Graphical user interface, application

Description automatically generated

Click on “Attach existing policies directly” and select **AdministratorAccess**

Graphical user interface, text, application, email

Description automatically generated

Review the user info then create user

Graphical user interface, text, application, Teams

Description automatically generated

Download the CSV generated – it contains the Access key ID and the Secret access key

**AWS CLI setup on EC2**

As the Ubuntu user (NOT Jenkins user) on the EC2, run **setup\_awscli.sh**

Text

Description automatically generated

\*\*\* had to make sure we sudo apt install unzip to be able to unzip the downloaded file \*\*\*

Text

Description automatically generated

Graphical user interface, text

Description automatically generated

Text

Description automatically generated

You can cd into .aws directory and confirm the config and credentials files

**Installing EB CLI on jenkins user**

First switch to the jenkins user’s bash terminal, then run the command

Text

Description automatically generated

Text

Description automatically generated

Installation completed, but warning messages about PATH variable not including the bin folder it’s installed in – change the PATH variable as shown in warning message:

Text

Description automatically generated with low confidence

\*\*\* eb --version command did NOT work prior to setting the PATH \*\*\*

**Deploying url-shortener on EB CLIText

Description automatically generated**

Ran eb init command

**Text

Description automatically generated**

Continue with configuration as described below:

**Graphical user interface, text, application

Description automatically generated**

**Text

Description automatically generated**

Ran eb create command with configurations listed above to create the environment and deploy

A screenshot of a computer

Description automatically generated with medium confidence

Elastic Beanstalk creating the environment and application

**Graphical user interface, text, application

Description automatically generated**

**Application deployment successful**

**Editing Jenkinsfile to include Deploy stage**

**Graphical user interface, text, application, email

Description automatically generated**

Add deploy stage

**Table, calendar

Description automatically generated**

Build successful on Jenkins pipeline

**Connecting GitHub to Jenkins Server**

Graphical user interface, text, application

Description automatically generated

Github > Settings > Developer settings > Personal access tokens > Generate new token

Under “Select scopes” - select **repo** and **admin:repo\_hook**

**Graphical user interface, text, application

Description automatically generated**

Generate and copy the personal access token

Go to http://<public-ip>:8080 and log in as admin

Graphical user interface, text, application, Teams

Description automatically generated

Dashboard > New Item > Multibranch pipeline

Graphical user interface, application

Description automatically generated

Under Branch Sources, add GitHub credentials

Graphical user interface, text, application, email

Description automatically generated

Enter GitHub username and generated access token as the password

Graphical user interface, text, application, Teams

Description automatically generated

Select the entered credentials and enter the forked repo URL, then validate connection

Graphical user interface, application

Description automatically generated

After clicking Apply then Save, there will be a build happening

**\*\*\* test stage kept failing, so I had to go into Jenkinsfile and change line 20 to activate the pytest module for it to recognize the test file\*\*\***

Graphical user interface, text

Description automatically generated with medium confidence

BEFORE

Timeline

Description automatically generated with medium confidence

AFTER

**Deploy the url-shortener application using Elastic Beanstalk CLI (as a Jenkins user)**

**Adding Slack notification on Jenkins**

**Graphical user interface, text, application, email

Description automatically generated**

On Slack app directory, search Jenkins CI and add to Slack

Graphical user interface, text, application, Teams

Description automatically generated

Follow instructions provided:

**Graphical user interface, application

Description automatically generated**

Go to Jenkins > Dashboard > Manage Jenkins > Plugin Manager > Available > Search “Slack Notification” > Install the plugin – it should show up under Installed after completed

**Graphical user interface, text, application, email

Description automatically generated**

Dashboard > Manage Jenkins > Configure System > Slack > Add workspace name > Add credential as “Secret Text” – enter the provided Integration Token Credential ID

A picture containing logo

Description automatically generated

You will receive a Slack notification once it is all set

**To add slack notification based on status of Jenkins pipeline:**

Graphical user interface, text, application, email

Description automatically generated

Add slackSend function inside Jenkinsfile

Application

Description automatically generated with low confidence

Notification will be sent each time the repository scans