## Install Jenkins on an EC2 from the Default VPC

* Create an EC2 with Ubuntu AMI and ports 22, 80, and 8080 open:

Graphical user interface, table

Description automatically generated

* SSH into the EC2 and run setup\_jenkins.sh to install, run, and check status of Jenkins:

Text

Description automatically generated

* Go to http://<ec2-public-ip>:8080 to set up Jenkins admin role – retrieve password by running the command **sudo cat /var/lib/Jenkins/secrets/initialAdminPassword**
* Install suggested plugins
* Reset admin password then save

Graphical user interface, text, application

Description automatically generated

## Create an EC2 in the Public Subnet of Kura VPC

* Select Ubuntu AMI and follow the configurations:
  + Kura VPC
  + Public Subnet
  + Auto-Assign Public IP -> Enable

Graphical user interface, text, application, email

Description automatically generated

* + Ports 22 and 5000:

Graphical user interface, application

Description automatically generated

* + Install necessary packages: default-jre, python3-pip, python3.10-venv, nginx > Use **setup\_VPC\_pub\_ec2.sh** or Include under User Data as a bootstrap script:

Text, letter

Description automatically generated

## Configure and Connect a Jenkins Agent to Jenkins

* Inside Jenkins server (EC2 from Default VPC), click on Build Executor Status > + New Node > Enter node name and select Permanent Agent:

Graphical user interface, application, Teams

Description automatically generated

* Enter following configurations:
  + **Name:** awsDeploy
  + **Description:** Deployment Server
  + **Number of Executors:** 1
  + **Remote Root Directory:** /home/ubuntu/agent
  + **Labels:** awsDeploy
  + **Usage:** Only build jobs with label…
  + **Launch Method:** Launch Agents via SSH
  + **Host:** Public IP of EC2 from Kura VPC
  + **Host Key Verification Strategy:** Non-verifying verification strategy
  + **Availability:** Keep this agent online as much as possible

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

* + **Credentials:**

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated

***\*\*\*PRIVATE KEY IS THE CONTENT OF THE PEM FILE USED TO SSH INTO EC2 INSTANCES\*\*\****

* Once Agent configuration info is saved, it will be created and can be viewed from Dashboard > Build Executor Status:

Table

Description automatically generated

* You can check the log as well:

Graphical user interface, text

Description automatically generated

## Create a Pipeline Build in Jenkins

* Prior to building a pipeline – SSH into the EC2 in Kura VPC and change the **/etc/nginx/sites-enabled/default** file:



1. Change port from 80 to 5000:

Text

Description automatically generated

1. Replace contents of location as below: Text

   Description automatically generated

* Go back to Jenkins server on EC2 in Default VPC and configure a multibranch pipeline by navigating to Dashboard > New Item > Multibranch Pipeline
* Under Branch Sources > GitHub > Credentials > + Add > Enter GitHub username and generated access token as password:

Graphical user interface, text, application, email

Description automatically generated

* Once entered, validate connection to ensure Jenkins can access GitHub repo:

Graphical user interface, text, application, email

Description automatically generated

* Then, navigate to Dashboard > Manage Jenkins > Plugin Manager and install Pipeline Keep Running Step plugin:

Graphical user interface, text, application, email

Description automatically generated

* Once completed, edit the Jenkinsfile in deployment repo with the following code:

Graphical user interface, text, application

Description automatically generated

### **Test Stage Issues**

* Test failed initially due to an extra space on line 6 – removed the extra space and did another build:

Graphical user interface, text, application

Description automatically generated

* Successful Build after edits made in the Test Stage:

Graphical user interface, application

Description automatically generated

## **Deployment Issues**

### Initial Jenkinsfile

* Even with a successful Deploy stage on Jenkins, there was 502 Bad Gateway error:

Graphical user interface, text, application

Description automatically generated

* To resolve this, I manually went inside the VPC EC2, activated the venv and ran the last command found in the Jenkins Deploy stage:



* Then, I was able to access the url-shortener website using the VPC EC2’s IP and port 5000:

Graphical user interface, text, application

Description automatically generated

### Updated Jenkinsfile

* That said, Tyrone informed us that there was a bug within the original Jenkinsfile, so he provided us with new instructions – upon installing the Jenkins Pipeline Keep Running Step and editing the Jenkinsfile, the Deploy stage was successful:

Graphical user interface, application

Description automatically generated

## Additions from Deployment 2

### Webhook

* Navigate to GitHub repo > Settings > Webhooks
  + Payload URL: http://<ec2-public-ip>:8080/github-webhook/
  + Content Type: application/json

Graphical user interface, text, application, email

Description automatically generated

### Slack Notifications

* Navigate to <https://kura-labs.slack.com/apps/new/A0F7VRFKN-jenkins-ci>:

Graphical user interface, text, application

Description automatically generated

* Once added, follow the steps provided to set up – take note of the Integration Token Credential ID to enter in Jenkins:

Graphical user interface, text, application, email

Description automatically generated

* In Jenkins, go to Dashboard > Manage Jenkins > Plugin Manager > Available > Search for “Slack” and Install

Graphical user interface, application

Description automatically generated

* Once completed, navigate to Dashboard > Manage Jenkins > Configure System > Slack > Add workspace name > Add credentials as “Secret Text” > Enter the Integration Token Credential ID provided earlier:

Graphical user interface, text, application, email

Description automatically generated

* Make sure to enter workspace name (i.e. kura-labs) and select the credential entered:

Application

Description automatically generated with medium confidence

* Test connection to ensure it is successful:



* Once you Apply and Save, you will receive a confirmation Slack notification:

Graphical user interface, text, application, email

Description automatically generated

* On GitHub, go to the project repository and edit the Jenkinsfile to include Slack notifications based on build status on each run:

Graphical user interface, text, application

Description automatically generated

* As soon as the change is committed, Jenkins will detect the changes (thanks to the webhook) and will automatically trigger a new build:

Graphical user interface, application, table

Description automatically generated

* As Jenkins builds each of the stages, a Slack notification will be sent as below:

Graphical user interface, text, application, email

Description automatically generated

## Software Stack

My guess of how the **nginx-gunicorn-Flask** stack works:

Timeline

Description automatically generatedGraphical user interface, application

Description automatically generated