Deployment #3

Welcome to Deployment 3!! Time to step it up for this deployment. You will need to follow the steps below to set up a CI/CD pipeline from start to finish. We will still use Elastic Beanstalk and you will be in charge of adding to your pipeline!!

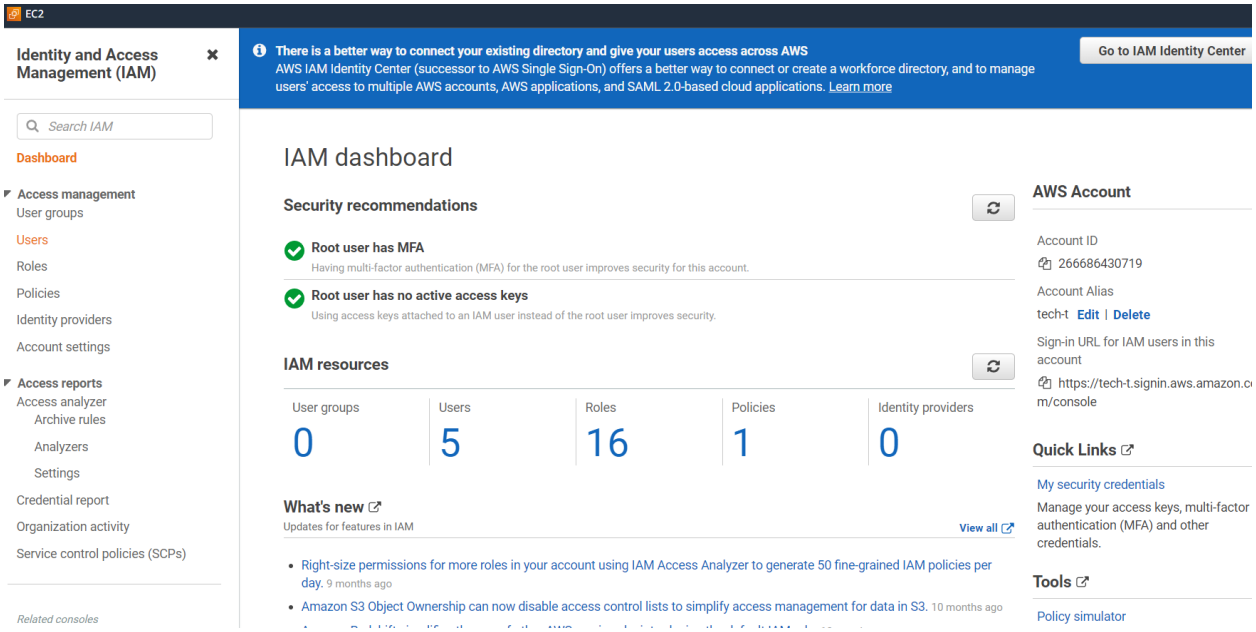
1. Install Jenkins on an EC2 if you haven't already.

2. Activate the Jenkins user on the EC2 by running the commands below.

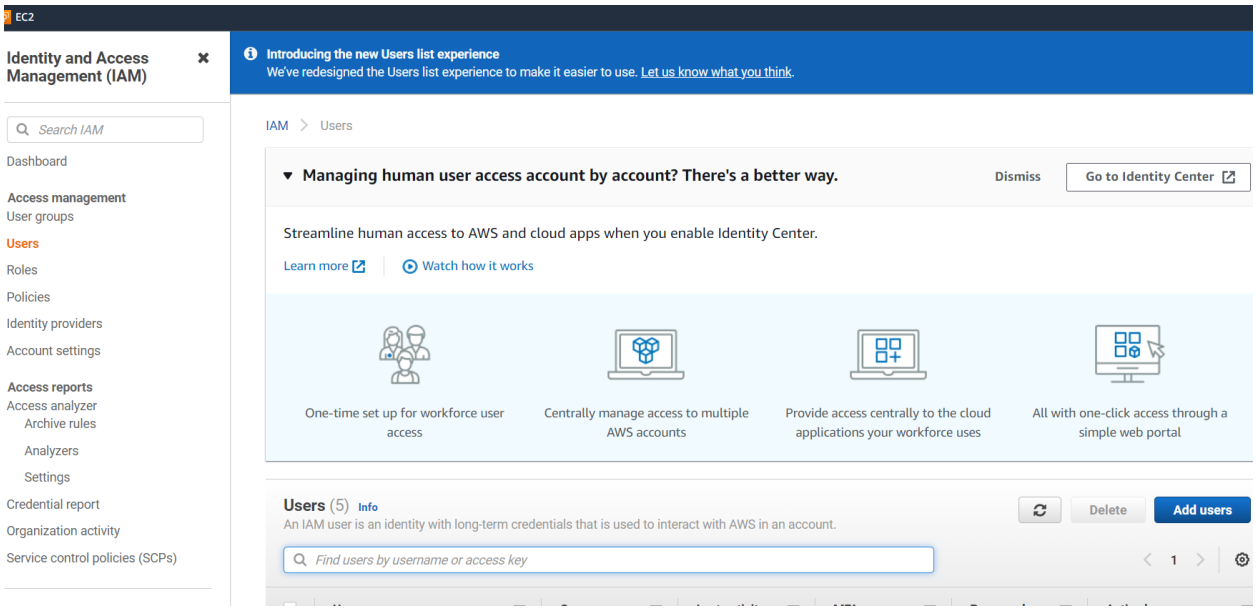
```
$sudo passwd jenkins
$sudo su - jenkins -s /bin/bash
```

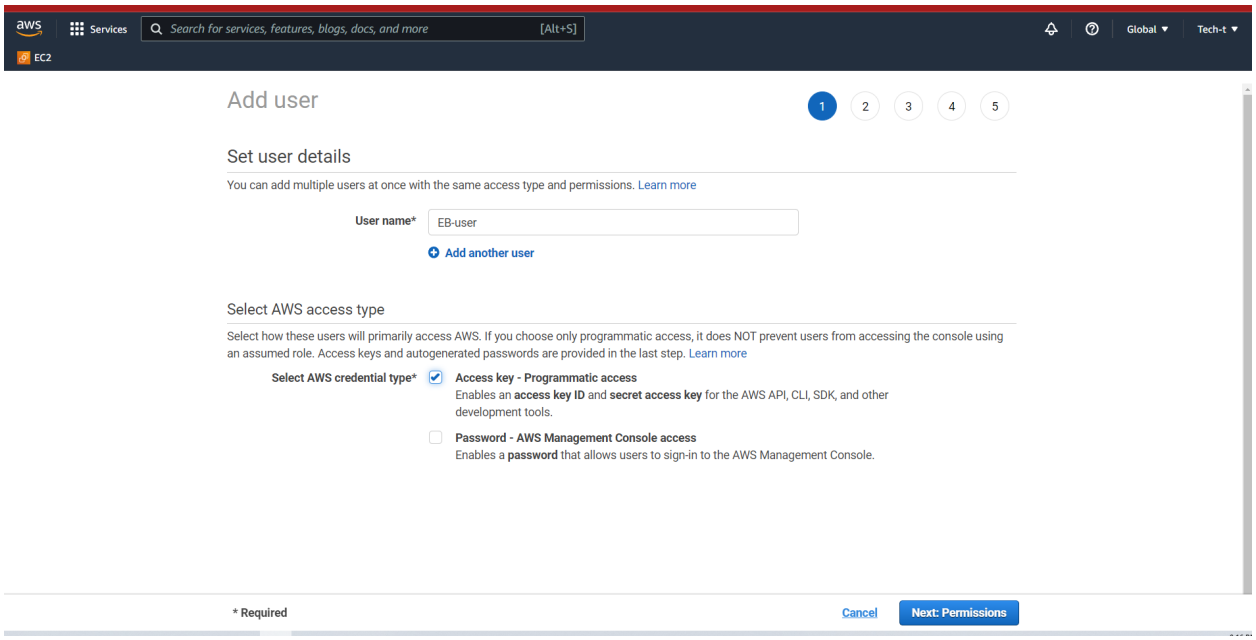
3. Create a Jenkins user in your AWS account:

- Navigate to IAM in the AWS console. Next click on the Users option in Access management:

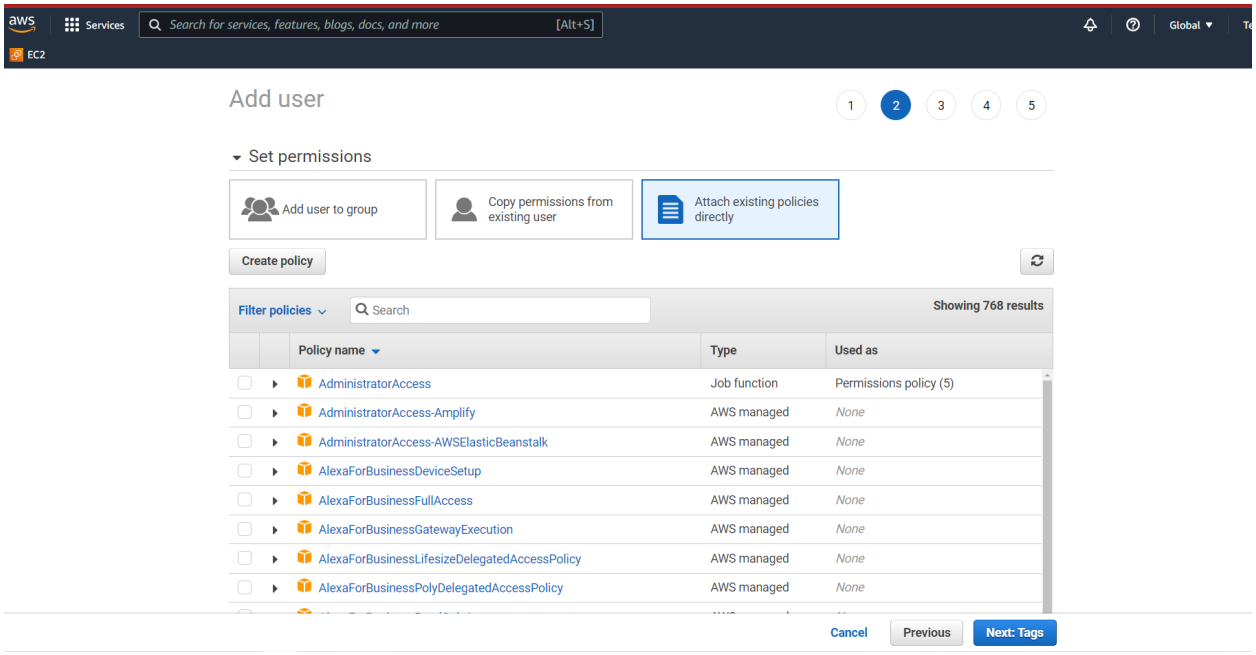


- Select add Users. Next, the username can be EB-user. Then select Programmatic access and then next:





- Select “Attach existing policies directly” and select administrator access. Now choose Next for this page and the following page:



- Finally, create the user and then copy and save the “access key ID” and the “secret access key”:

aws Services Search for services, features, blogs, docs, and more [Alt+S]

EC2

Add user

1 2 3 4 5

Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

User details

User name	EB-user
AWS access type	Programmatic access - with an access key
Permissions boundary	Permissions boundary is not set

Permissions summary

The following policies will be attached to the user shown above.

Type	Name
Managed policy	AdministratorAccess

Tags

No tags were added.

Cancel Previous Create user

aws Services Search for services, features, blogs, docs, and more [Alt+S]

EC2

Add user

1 2 3 4 5

Success

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: https://tech-t.signin.aws.amazon.com/console

Download .csv

User	Access key ID	Secret access key
EB-user	AKIAT4F577X7WOLMS673	********* Show

Close

4. Install AWS CLI on the Jenkins EC2 and configure:

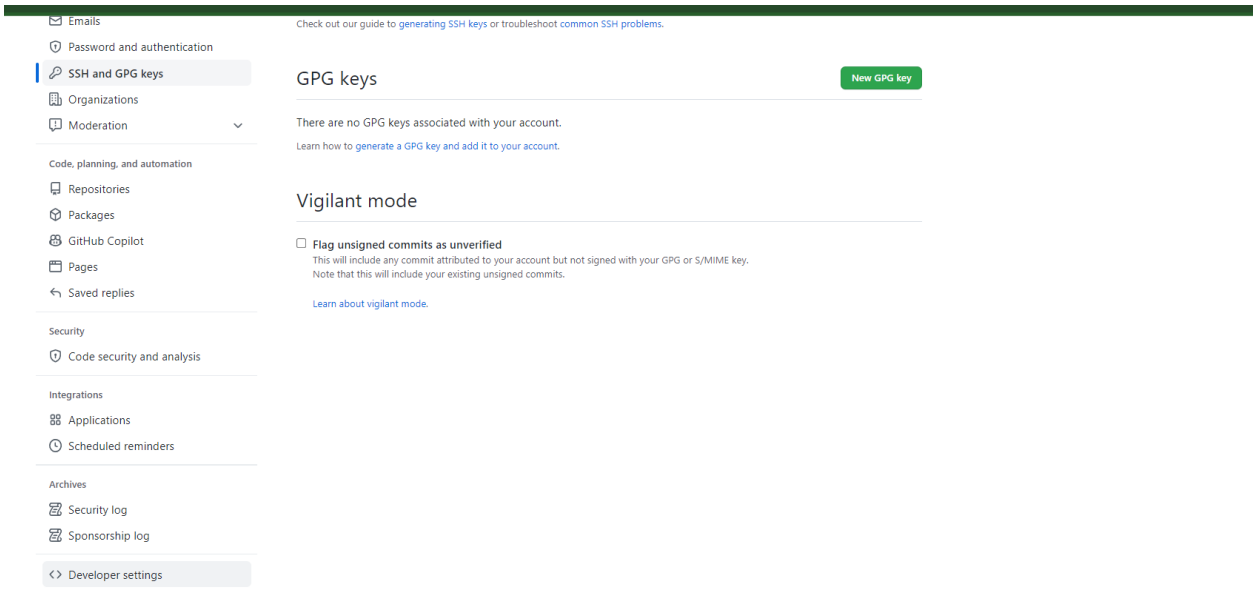
```
$curl \
"https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" \
-o "awscliv2.zip"

$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
```

5. Install EB CLI in the Jenkins EC2 user:

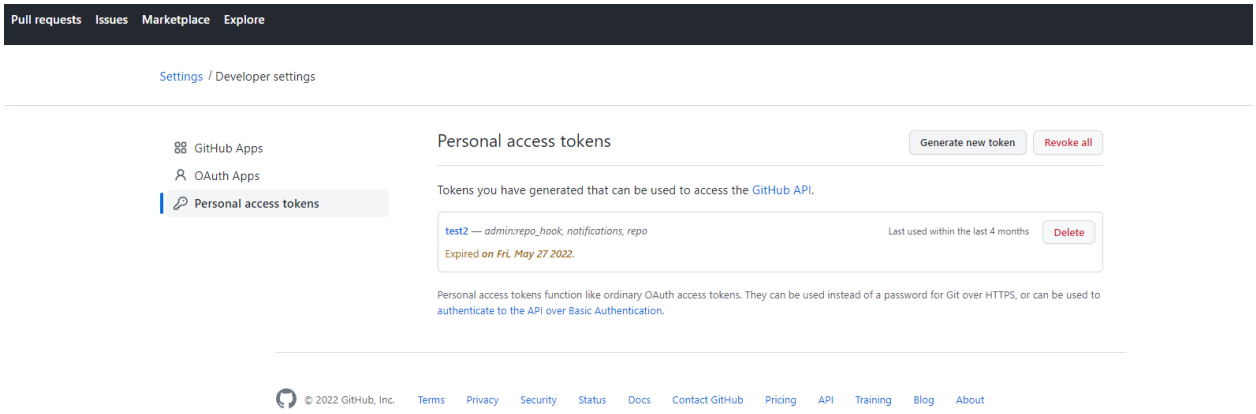
```
$pip install awsebcli --upgrade --user
$eb –-version
```

6. Connect GitHub to Jenkins Server:

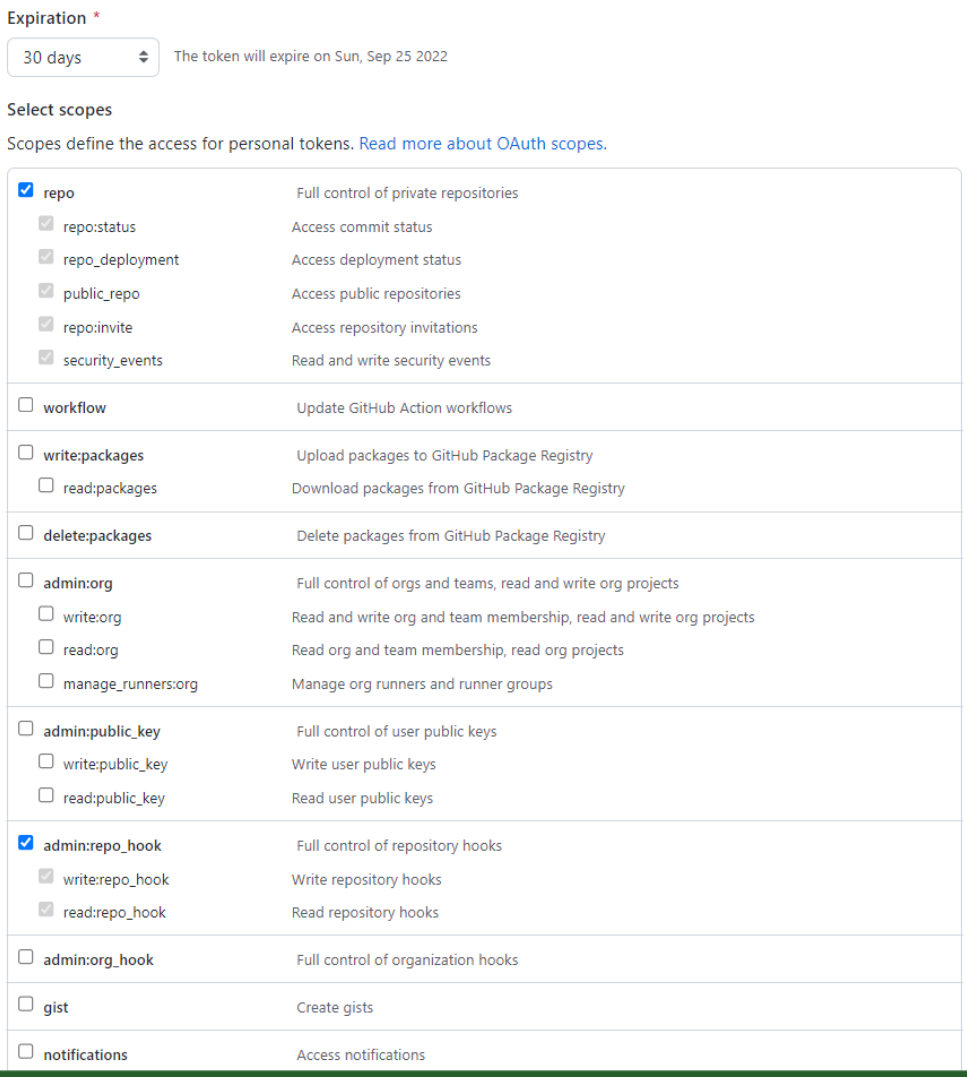
- First Fork the Deployment repo: https://github.com/kura-labs-org/kuralabs_deployment_2.git
- Next, create an access token from GitHub:
  - Navigate to your GitHub settings, select developer settings



  - Select personal access token and create a new token.

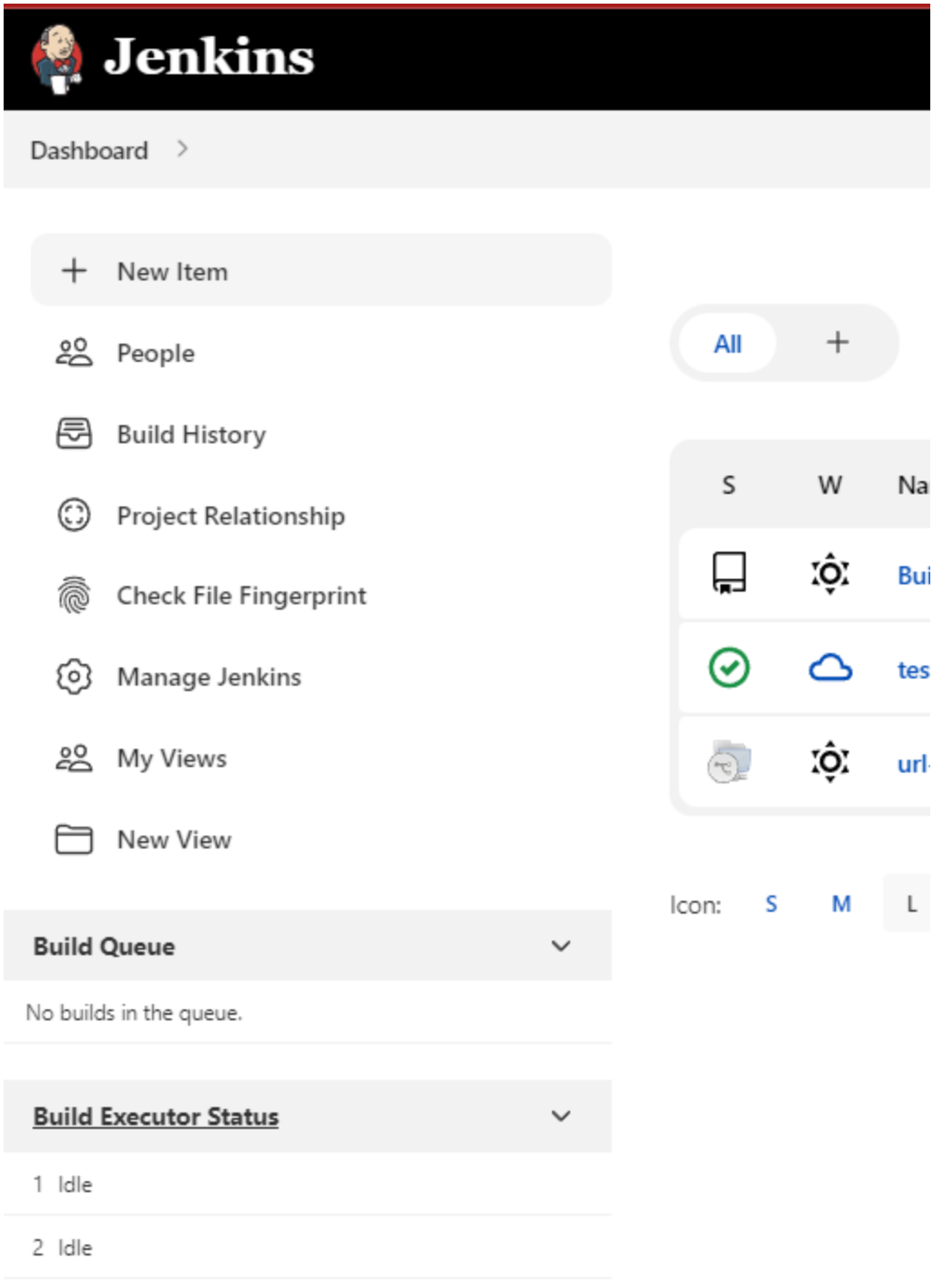


- Select the settings you see below for access token permissions.

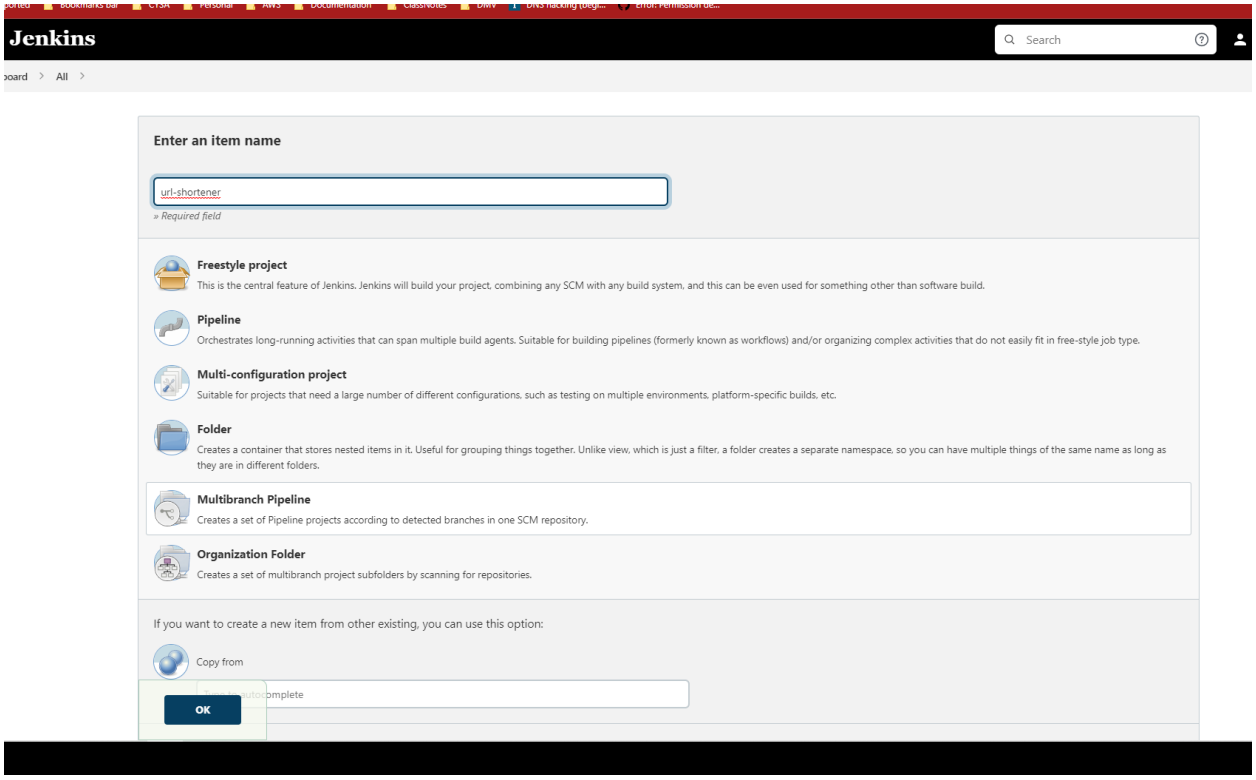


7. Create a multibranch build:

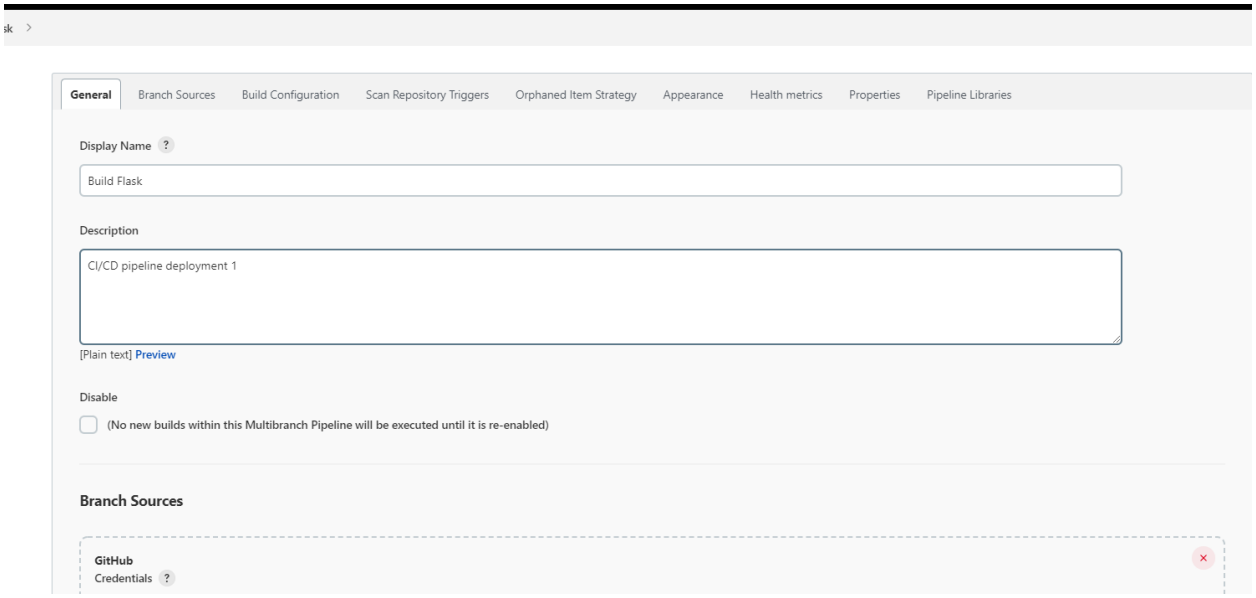
- Log back into Jenkins and select “New item”


Jenkins
Dashboard
New Item
People
Build History
Project Relationship
Check File Fingerprint
Manage Jenkins
My Views
New View
Build Queue
No builds in the queue.
Build Executor Status
1 Idle
2 Idle
All
S
W
Na
Bui
tes
url
Icon:
S
M
L

- Select multibranch pipeline



- Enter a display name and brief description



- Add a Branch source by selecting Add source and select GitHub

General | **Branch Sources** | Build Configuration | Scan Multibranch Pipeline Triggers | Orphaned Item Strategy | Appearance | Health metrics | Properties | Pipeline Libraries

Disable

☐ (No new builds within this Multibranch Pipeline will be executed until it is re-enabled)

Branch Sources

Add source

Build Configuration

Mode

by Jenkinsfile

Script Path ?

Jenkinsfile

Scan Multibranch Pipeline Triggers

☐ Periodically if not otherwise run ?

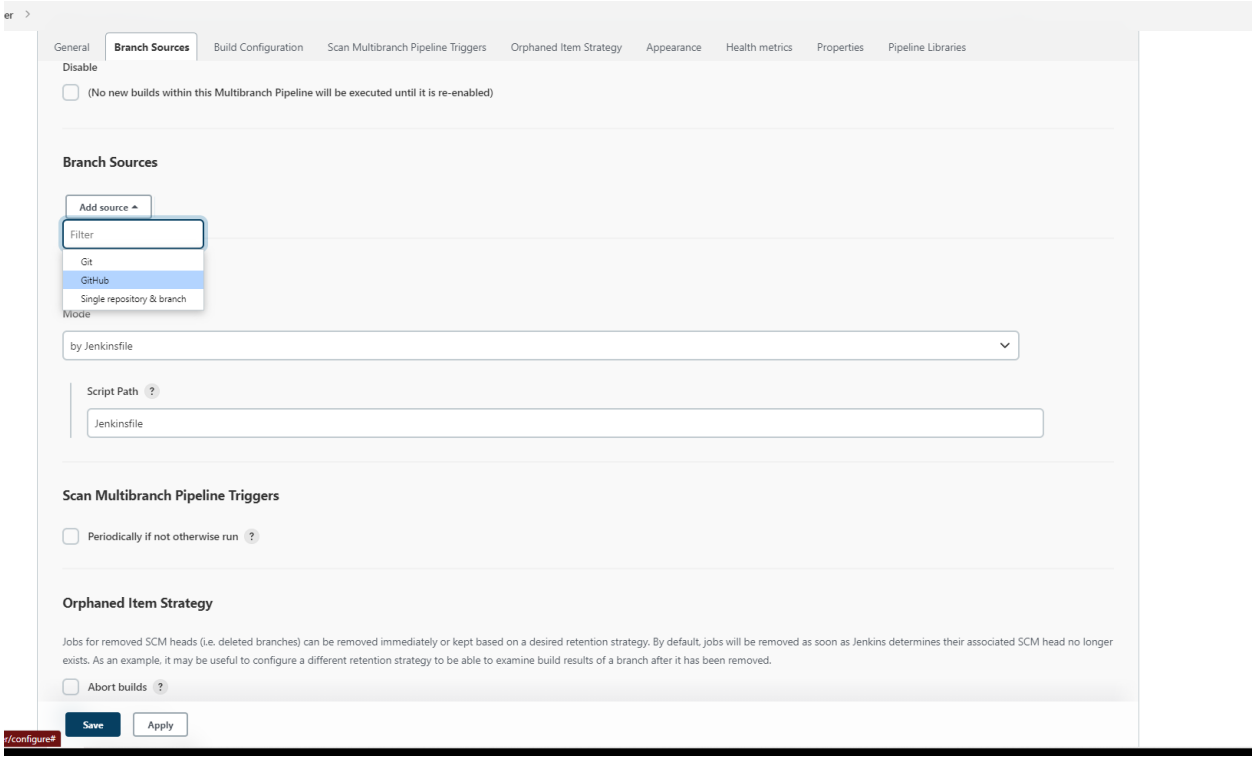
Orphaned Item Strategy

Jobs for removed SCM heads (i.e. deleted branches) can be removed immediately or kept based on a desired retention strategy. By default, jobs will be removed as soon as Jenkins determines their associated SCM head no longer exists. As an example, it may be useful to configure a different retention strategy to be able to examine build results of a branch after it has been removed.

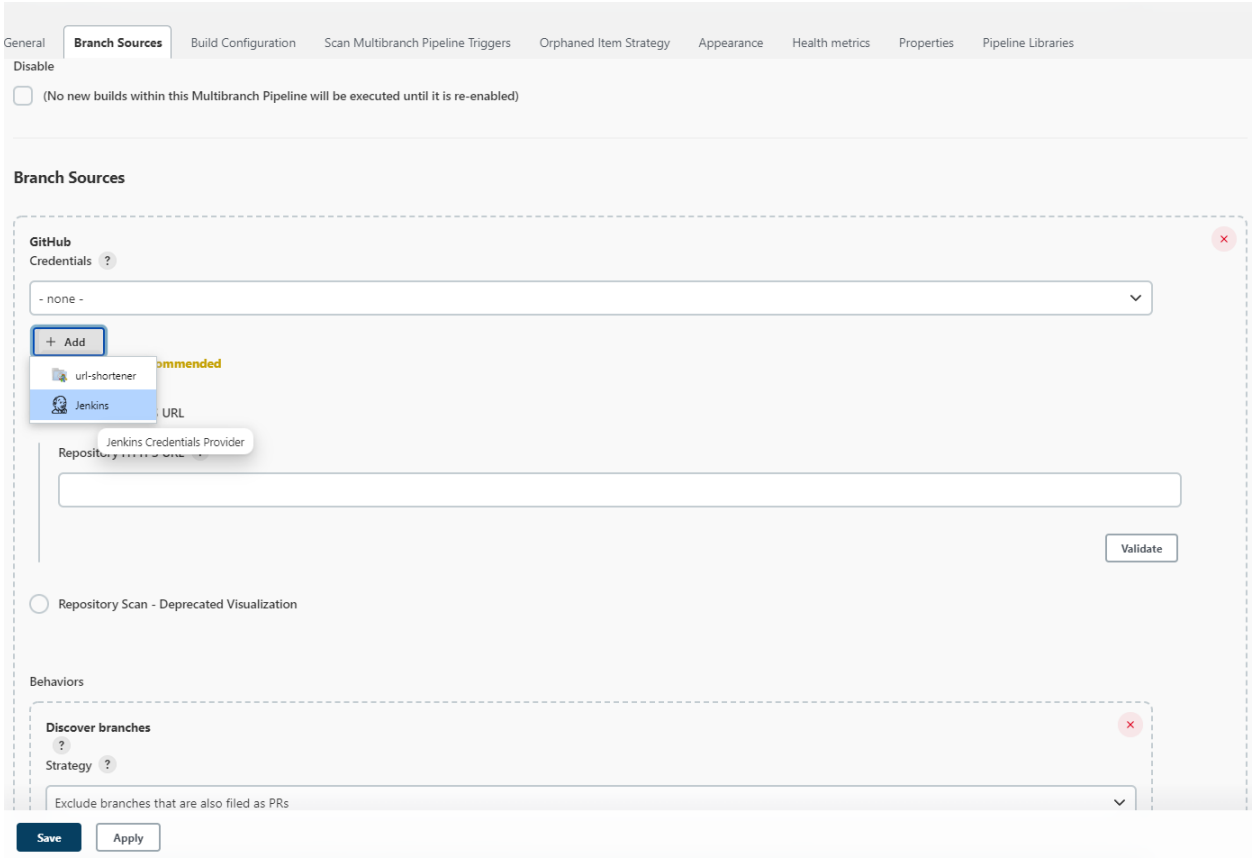
☐ Abort builds ?

Save | Apply

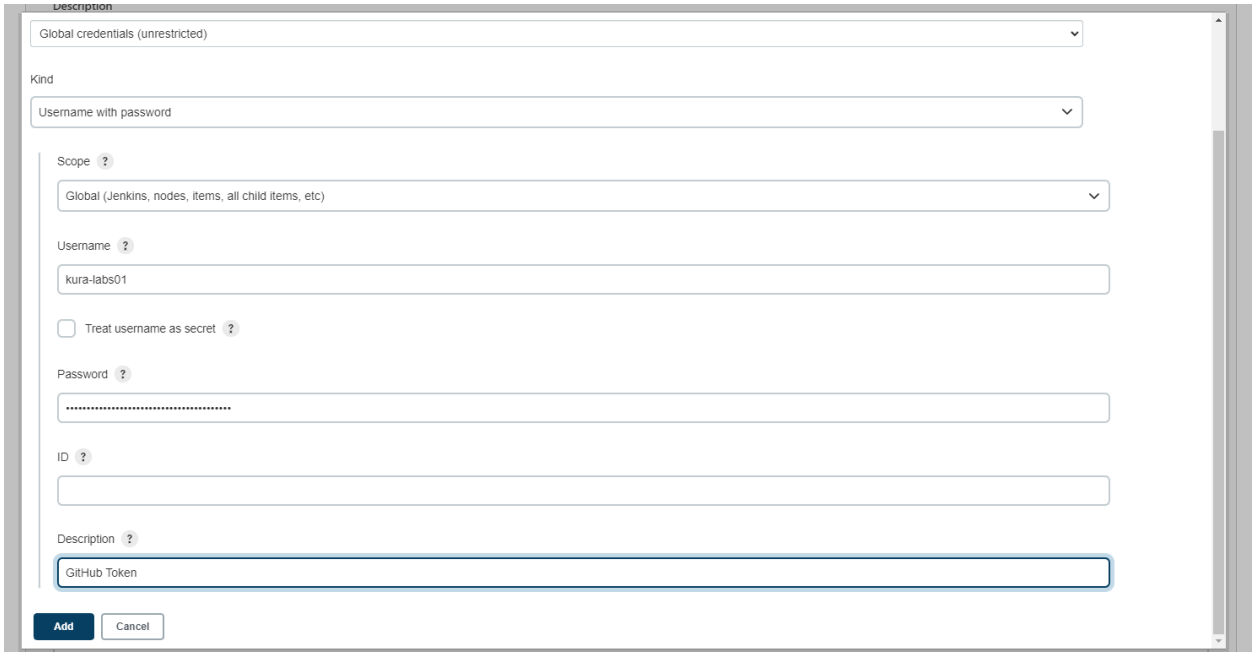
- Select the Add button and select GitHub



- Click on Add and then select Jenkins

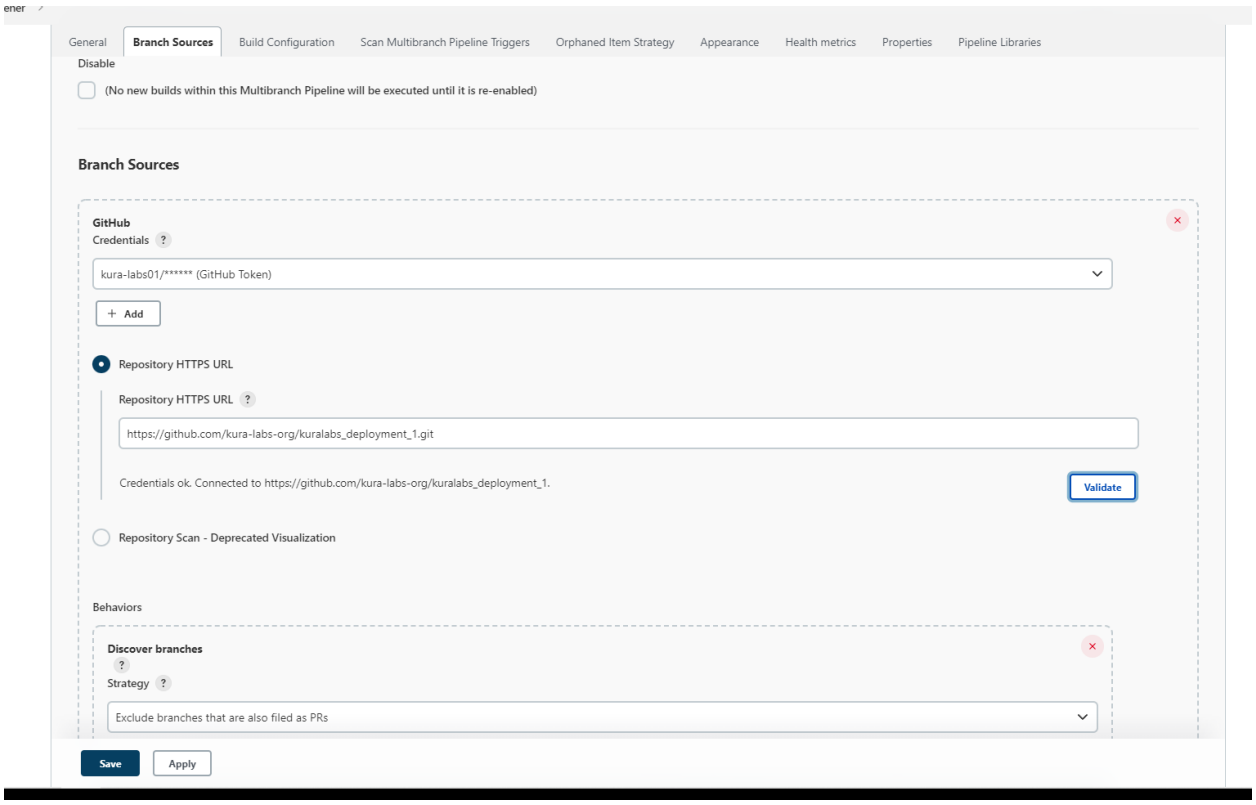


- Under username enter your GitHub username
- Under password enter your token

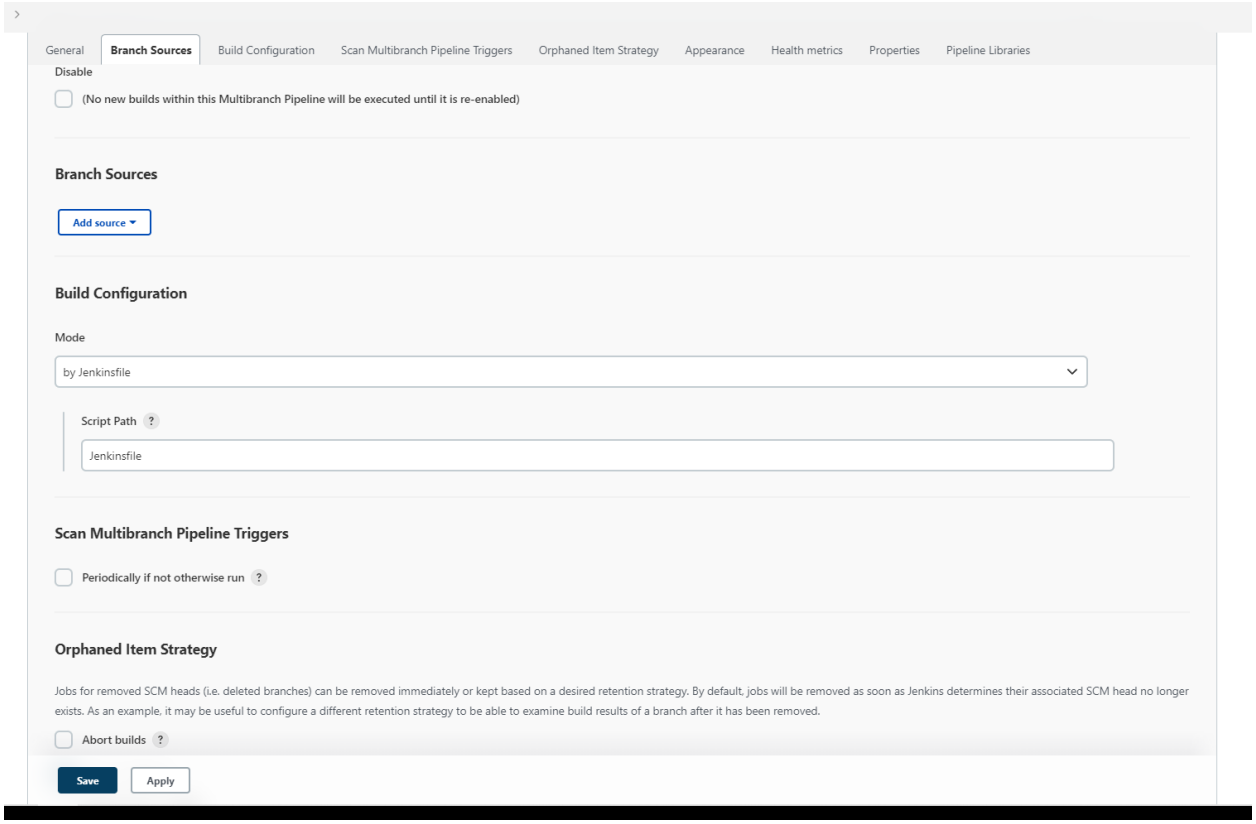


- **(Optional)** under ID and Description enter GitHub repo

- Enter your URL to the repository and you can validate by selecting validate.

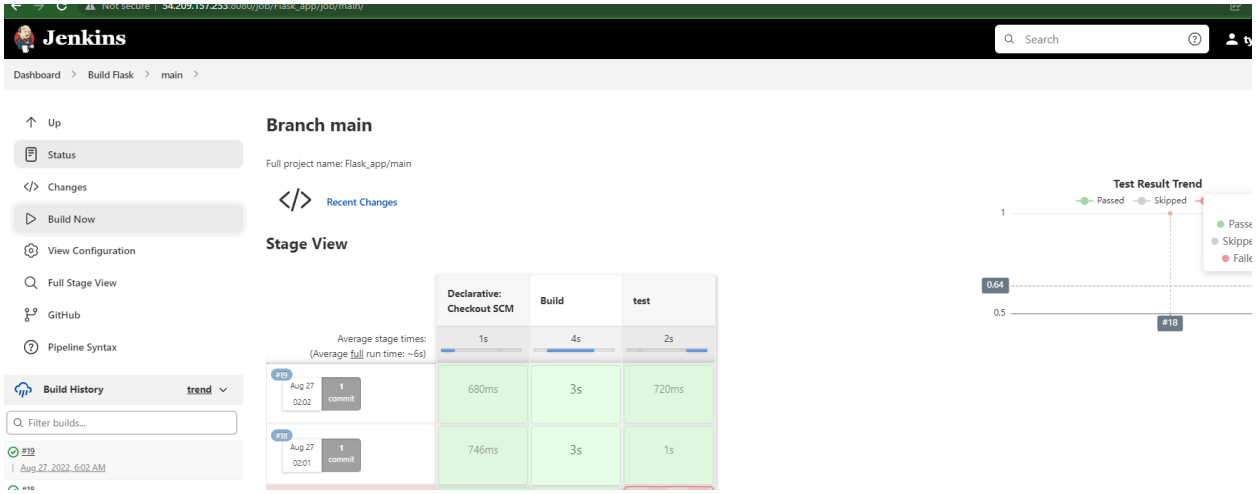


- Make sure this says “Jenkinsfile”



- Select Apply and then Save

- You should see a build happening. If you don’t, select Scan Repository.



8. Now deploy the application from Elastic Beanstalk CLI:

```
$sudo su - jenkins -s /bin/bash
$cd /var/workspace/{{The name of your project}}/
$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

9. Now 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 {{Your environment
name}}'
      }
    }
  }
}
```

Now add to the pipeline:

- Add a Webhook to the pipeline
- Create email notification to the pipeline

Once you have done that, apply your changes by building the pipeline again

Create documentation:

Document your process for adding a webhook and email notification. Also, include a summary of what you configured in each part of your setup. Lastly, include any issues you ran into and what you did to fix them.

> **Note:** Submit your work by uploading your work to a repo or the forked repo. Then submit the link to the repo via LMS.