

Deployment #2

The goal of this deployment is to set up a CI/CD pipeline from start to finish. I used Elastic Beanstalk and customized the pipeline.

Step 1: Install Jenkins on an EC2

Step 2: Activate the Jenkins user on the EC2:

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

Step 3: Create a Jenkins user in my AWS account:

- I navigated to IAM in the AWS console. Next, I clicked on the Users option in the Access management.

EC2

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

User groups

Users

Roles

Policies

Identity providers

Account settings

Access reports

Access analyzer

Archive rules

Analyzers

Settings

Credential report

Organization activity

Service control policies (SCPs)

Related consoles

There is a better way to connect your existing directory and give your users access across AWS

AWS IAM Identity Center (successor to AWS Single Sign-On) offers a better way to connect or create a workforce directory, and to manage users' access to multiple AWS accounts, AWS applications, and SAML 2.0-based cloud applications. [Learn more](#)

[Go to IAM Identity Center](#)

IAM dashboard

Security recommendations

- Root user has MFA**
Having multi-factor authentication (MFA) for the root user improves security for this account.
- Root user has no active access keys**
Using access keys attached to an IAM user instead of the root user improves security.

IAM resources

User groups	Users	Roles	Policies	Identity providers
0	5	16	1	0

What's new

Updates for features in IAM

- Right-size permissions for more roles in your account using IAM Access Analyzer to generate 50 fine-grained IAM policies per day.** 9 months ago
- Amazon S3 Object Ownership can now disable access control lists to simplify access management for data in S3.** 10 months ago

[View all](#)

AWS Account

Account ID
266686430719

Account Alias
tech-t [Edit](#) [Delete](#)

Sign-in URL for IAM users in this account
<https://tech-t.signin.aws.amazon.com/console>

Quick Links

[My security credentials](#)
Manage your access keys, multi-factor authentication (MFA) and other credentials.

Tools

[Policy simulator](#)

- I selected add user. The username I added was EB-user. I then selected Programmatic access and then I clicked next.

EC2

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

User groups

Users

Roles

Policies

Identity providers

Account settings

Access reports

Access analyzer

Archive rules

Analyzers

Settings

Credential report

Organization activity

Service control policies (SCPs)

Introducing the new Users list experience

We've redesigned the Users list experience to make it easier to use. [Let us know what you think.](#)


IAM > Users

Managing human user access account by account? There's a better way.


Dismiss [Go to Identity Center](#)

Streamline human access to AWS and cloud apps when you enable Identity Center.


[Learn more](#) [Watch how it works](#)




One-time set up for workforce user access



Centrally manage access to multiple AWS accounts



Provide access centrally to the cloud applications your workforce uses



All with one-click access through a simple web portal

Users (5) [Info](#)

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

[Delete](#) [Add users](#)

< 1 >

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

Add user

1 2 3 4 5

Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name*

[Add another user](#)

Select AWS access type

Select how these users will primarily access AWS. If you choose only programmatic access, it does NOT prevent users from accessing the console using an assumed role. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Select AWS credential type* ☒ **Access key - Programmatic access**
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

☐ **Password - AWS Management Console access**
Enables a **password** that allows users to sign-in to the AWS Management Console.

* Required

[Cancel](#) [Next: Permissions](#)

- Next, I selected “Attach existing policies directly” and select administrator access. Then I selected next for the current page as well as the next pages.

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

Add user

1 2 3 4 5

Set permissions

[Add user to group](#) [Copy permissions from existing user](#) [Attach existing policies directly](#)

[Create policy](#)

Filter policies Search Showing 768 results

	Policy name	Type	Used as
<input type="checkbox"/>	AdministratorAccess	Job function	Permissions policy (5)
<input type="checkbox"/>	AdministratorAccess-Amplify	AWS managed	None
<input type="checkbox"/>	AdministratorAccess-AWSElasticBeanstalk	AWS managed	None
<input type="checkbox"/>	AlexaForBusinessDeviceSetup	AWS managed	None
<input type="checkbox"/>	AlexaForBusinessFullAccess	AWS managed	None
<input type="checkbox"/>	AlexaForBusinessGatewayExecution	AWS managed	None
<input type="checkbox"/>	AlexaForBusinessLifesizeDelegatedAccessPolicy	AWS managed	None
<input type="checkbox"/>	AlexaForBusinessPolyDelegatedAccessPolicy	AWS managed	None

[Cancel](#) [Previous](#) [Next: Tags](#)

- Finally, I created the user and then copied and saved the “access key ID” and the “secret access key” to a text file on my PC.

The screenshot shows the AWS IAM console 'Add user' page, specifically the 'Review' step (step 4 of 5). The page is titled 'Add user' and has a progress indicator with steps 1 through 5, where step 4 is highlighted. Below the title, there is a 'Review' section with the text: 'Review your choices. After you create the user, you can view and download the autogenerated password and access key.' This is followed by a 'User details' section with the following information:

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

Below this is a 'Permissions summary' section with the text: 'The following policies will be attached to the user shown above.' This is followed by a table showing the attached policies:

Type	Name
Managed policy	AdministratorAccess

Below this is a 'Tags' section with the text: 'No tags were added.'

At the bottom right of the page, there are three buttons: 'Cancel', 'Previous', and '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

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

- Using the curl command I downloaded AWS CLI files from the AWS CLI website
- I had difficulty with this step because the user was not part of the sudoers list and unzip was not installed on the jenkins user
- So I had to exit the jenkins user, and add jenkins to sudoers on the ubuntu EC2.
- Then as a Jenkins user I installed the unzip command and proceeded to configure.

```
$curl
"https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
$exit
$sudo usermod -aG sudo jenkins
sudo su - jenkins -s /bin/bash
$sudo apt install unzip
```

```
$sudo ./aws/install
$unzip awscliv2.zip

$aws --version
$aws configure
  - Set Access Key ID
  - Set Secret Access Key
  - Set region to: us-east-1
  - Set Output format: json
```

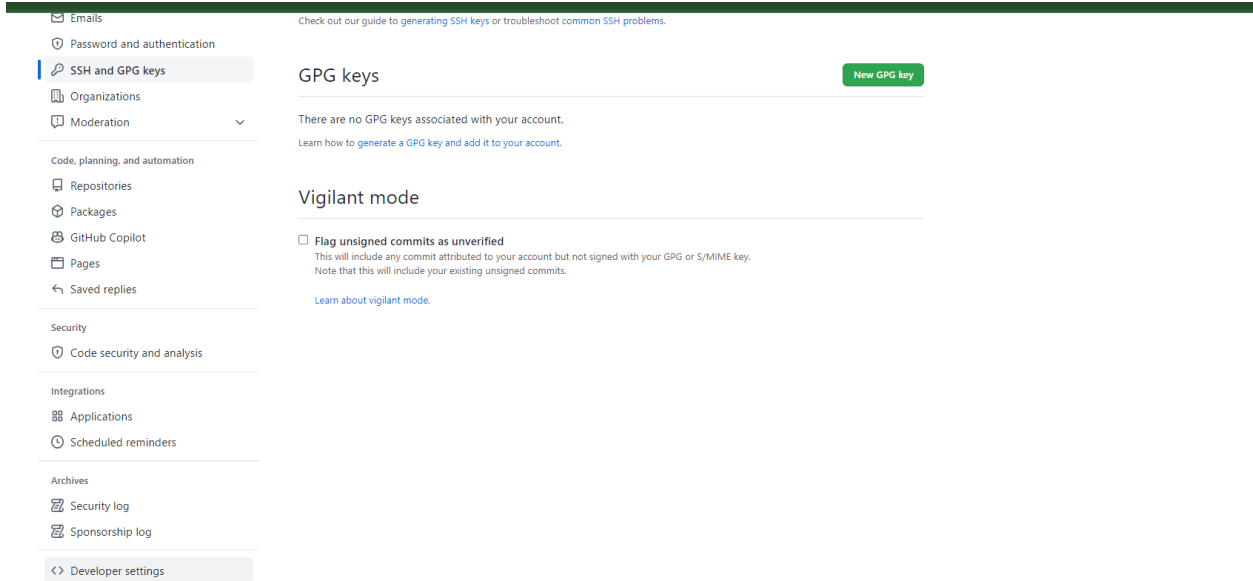
Step 5: Install EB CLI in the jenkins EC2 user:

- I tried to install EB CLI but my jenkins user did not have the pip command
- So I had to install the pip command using \$sudo apt install python3-pip
- Then, I used the below commands to install the EB CLI in the Jenkins EC2

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

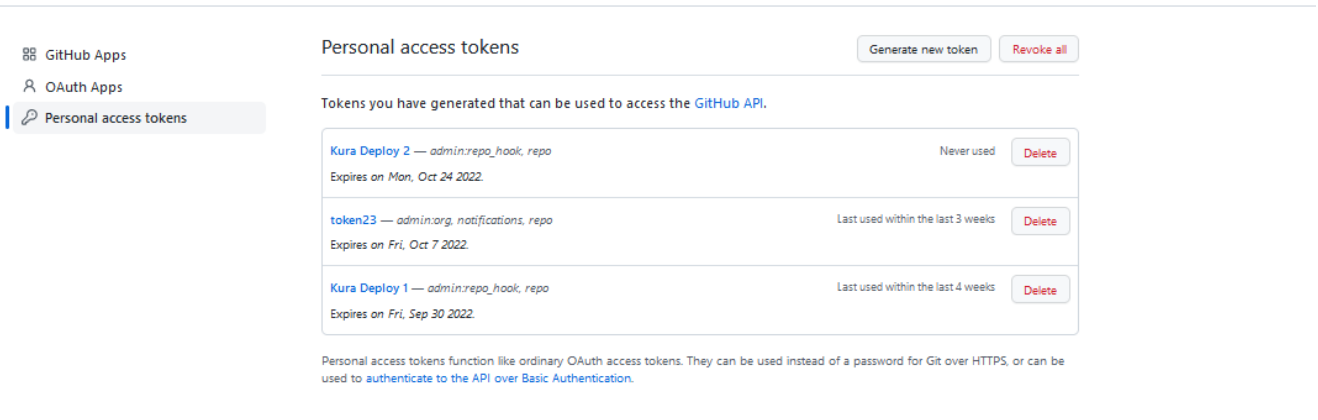
Step 6: Connect GitHub to Jenkins Server:

- I first forked the Deployment repo:
https://github.com/kura-labs-org/kuralabs_deployment_2.git
- Next, I created an access token from GitHub:
 - I navigated to my GitHub settings, and selected developer settings



- I then selected a personal access token and generated a new token.

Settings / Developer settings



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

Expiration *

30 days

The token will expire on Sun, Sep 25 2022

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events
<input type="checkbox"/> workflow	Update GitHub Action workflows
<input type="checkbox"/> write:packages	Upload packages to GitHub Package Registry
<input type="checkbox"/> read:packages	Download packages from GitHub Package Registry
<input type="checkbox"/> delete:packages	Delete packages from GitHub Package Registry
<input type="checkbox"/> admin:org	Full control of orgs and teams, read and write org projects
<input type="checkbox"/> write:org	Read and write org and team membership, read and write org projects
<input type="checkbox"/> read:org	Read org and team membership, read org projects
<input type="checkbox"/> manage_runners:org	Manage org runners and runner groups
<input type="checkbox"/> admin:public_key	Full control of user public keys
<input type="checkbox"/> write:public_key	Write user public keys
<input type="checkbox"/> read:public_key	Read user public keys
<input checked="" type="checkbox"/> admin:repo_hook	Full control of repository hooks
<input checked="" type="checkbox"/> write:repo_hook	Write repository hooks
<input checked="" type="checkbox"/> read:repo_hook	Read repository hooks
<input type="checkbox"/> admin:org_hook	Full control of organization hooks
<input type="checkbox"/> gist	Create gists
<input type="checkbox"/> notifications	Access notifications

Step 7: Create a multibranch build:


- I logged back into Jenkins and selected “New item”



Jenkins

Dashboard >


+ New Item


 People

 Build History

 Project Relationship

 Check File Fingerprint

 Manage Jenkins

 My Views

 New View

All

+

S

W

Na



Bui



tes



url.

Build Queue



No builds in the queue.

Build Executor Status



1 Idle

2 Idle

Icon:

S

M

L

- I entered the item name as "url-shortener".
- Then I selected a multibranch pipeline.

The screenshot shows the Jenkins 'New Item' page. At the top, the Jenkins logo and a search bar are visible. Below the header, the breadcrumb 'Dashboard > All' is shown. The main form is titled 'Enter an item name'. A text input field contains 'url-shortener', with a red underline indicating it is a required field. Below the input field, a list of item types is displayed: Freestyle project, Pipeline, Multi-configuration project, Folder, Multibranch Pipeline, and Organization Folder. The 'Multibranch Pipeline' option is highlighted with a blue border. At the bottom of the form, there is a section for 'Copy from' with a dropdown menu and an 'OK' button.

- I entered the display name as "Build Flask".
- Then I entered a brief description.

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

Display Name ?
Build Flask

Description
CI/CD pipeline deployment 1
[Plain text] [Preview](#)

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

Branch Sources

GitHub Credentials ? ✕

- I added a Branch source by selecting Add source and then selecting 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 ?

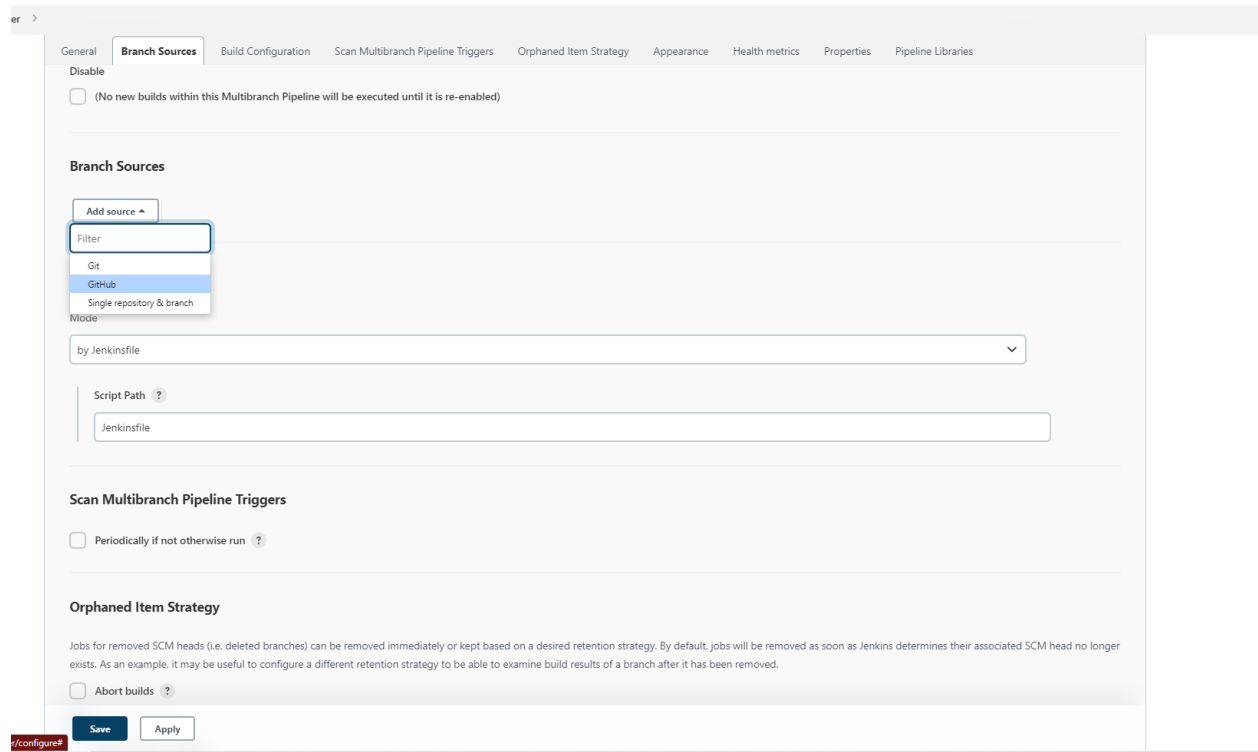
Orphaned Item Strategy

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](#)

- I then selected the Add button and selected GitHub.



- I clicked on Add and then selected Jenkins

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

GitHub ? ×

Credentials ?

- none - ▼

+ Add

- url-shortener Recommended
- Jenkins URL
- Jenkins Credentials Provider

Repository URL ?

Validate

☐ Repository Scan - Depreciated Visualization

Behaviors

Discover branches ? ×

Strategy ?

Exclude branches that are also filed as PRs ▼

Save Apply

- Under username I entered my GitHub username "AnjKura"
- Under password I entered my token from GitHub.

This is a screenshot of a Jenkins 'Add' dialog for creating a new credential. The dialog has a title bar 'Description' and a dropdown menu set to 'Global credentials (unrestricted)'. Below this, the 'Kind' is set to 'Username with password'. The 'Scope' is 'Global (Jenkins, nodes, items, all child items, etc)'. The 'Username' field contains 'kura-labs01'. There is an unchecked checkbox for 'Treat username as secret'. The 'Password' field is masked with dots. The 'ID' field is empty. The 'Description' field contains 'GitHub Token'. At the bottom are 'Add' and 'Cancel' buttons.

Description

Global credentials (unrestricted)

Kind

Username with password

Scope ?

Global (Jenkins, nodes, items, all child items, etc)

Username ?

kura-labs01

☐ Treat username as secret ?

Password ?

ID ?

Description ?

GitHub Token

Add Cancel

- I entered the URL to my repository "https://github.com/AnjKura/kuralabs_deployment_2.git".
- Then, I validated by selecting validate.

This is a screenshot of the Jenkins 'url-shortener' configuration page. The breadcrumb trail shows 'Dashboard > url-shortener >'. There are two radio buttons: 'Repository HTTPS URL' (selected) and 'Repository Scan - Deprecated Visualization'. Under the selected option, there is a text field with the URL 'https://github.com/AnjKura/kuralabs_deployment_2.git'. Below the field, it says 'Credentials ok. Connected to https://github.com/AnjKura/kuralabs_deployment_2.' and there is a 'Validate' button. At the bottom are 'Save' and 'Apply' buttons.

Dashboard > url-shortener >

☒ Repository HTTPS URL

☐ Repository Scan - Deprecated Visualization

Repository HTTPS URL ?

https://github.com/AnjKura/kuralabs_deployment_2.git

Credentials ok. Connected to https://github.com/AnjKura/kuralabs_deployment_2.

Validate

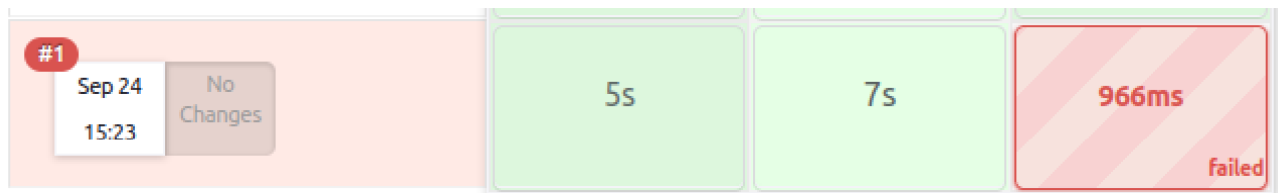
Save Apply

- I made sure that the Build Configuration says Jenkinsfile.

The screenshot shows the Jenkins configuration interface for a Multibranch Pipeline. The top navigation bar includes tabs for General, Branch Sources, Build Configuration, Scan Multibranch Pipeline Triggers, Orphaned Item Strategy, Appearance, Health metrics, Properties, and Pipeline Libraries. The 'Build Configuration' tab is currently selected.

Under the 'Build Configuration' tab, there is a 'Disable' section with a checkbox labeled '(No new builds within this Multibranch Pipeline will be executed until it is re-enabled)'. Below this is the 'Branch Sources' section with a 'Add source' button. The 'Build Configuration' section contains a 'Hilck' dropdown menu set to 'by Jenkinsfile' and a 'Script Path' text field containing 'Jenkinsfile'. The 'Scan Multibranch Pipeline Triggers' section has a checkbox for 'Periodically if not otherwise run'. The 'Orphaned Item Strategy' section includes a checkbox for 'Alert builds' and a descriptive paragraph about removing SCM heads. At the bottom, there are 'Save' and 'Apply' buttons.

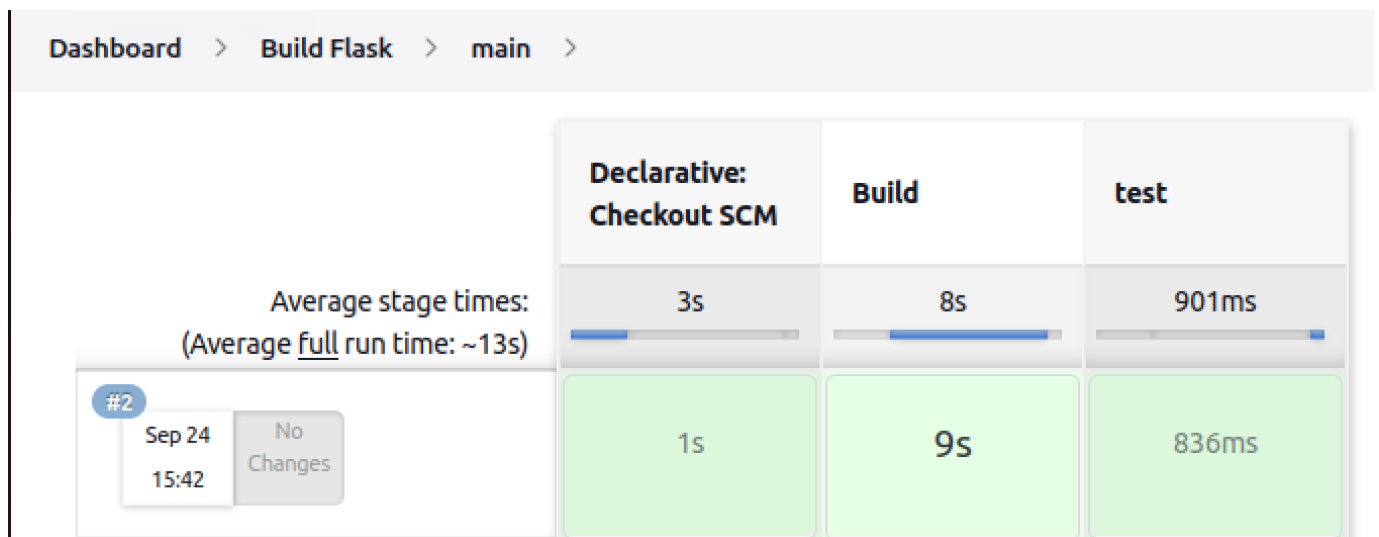
- I selected Apply and then Save.
- I did not see a Build happening right away, so I selected Scan Repository Now
- After I scanned the repository, I found out that my first build failed.



- I checked the logs and found that I needed to install python on my EC2.

```
ubuntu@ip-172-31-31-68:~$ sudo apt install python3.10-venv
```

- After I installed python, I ran the Build again and the Build was successful.



Step 8: Deploy the application from Elastic Beanstalk CLI:

- I used sudo to initialize the jenkins user again.
- I changed to directories in order to enter the workspace where the url-shortener_main is located
- I ran \$pip install awsebcli --upgrade --user
- Then, I ran export PATH="/var/lib/jenkins/.local/bin:\$PATH"

●

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

```
$cd workspace/url-shortener_main/
```

```
$eb init
```

- Select: us-east-1
- Press enter
- Select application to use: url-shortener3
- Select: Python
- Select: (The latest version of python available)
- Select: N (for CodeCommit)
- Set up SSH for instance: Yes
- Select keypair: I selected the keypair ssh2

```
$eb create
```

- I entered the default for the next 3 questions by hitting enter
- My environment name was url-shortener3-dev

- Spot Fleet: No
- I waited for the environment to be made.
- When I checked the environment on elastic beanstalk it was degraded.

Elastic Beanstalk > Environments

All environments Actions Create a new environment

Filter results matching the display values

Environment name	Health	Application name	Date created	Last modified	URL	Run ver
<input type="radio"/> url-shortener3-dev	Degraded	url-shortener3	2022-09-24 16:39:02 UTC-0400	2022-09-24 16:40:58 UTC-0400	url-shortener3-dev.us-east-1.elasticbeanstalk.com	

Step 9 I attempted to add a deployment stage to the pipeline in the 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-shortener3-dev5'

    }
}
}
}
}

```

- I could not successfully deploy the application because my url shortener environment was degraded.
 - I received the following error message: "Environment health has transitioned from Pending to Degraded. Auto Scaling activity failed 35 seconds ago with error: You have requested more vCPU capacity than your current vCPU limit of 1 allows for the instance bucket that the specified instance type belongs to. Please visit <http://aws.amazon.com/contact-us/ec2-request> to request an adjustment to this limit. Launching EC2 instance failed. At 2022-09-25T01:08:00Z a user request update of AutoScalingGroup constraints to min: 1, max: 4, desired: 1 changing the desired capacity from 0 to 1. At 2022-09-25T01:08:09Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 1. Initialization in progress (running for 87 seconds). There are no instances. Auto Scaling group desired capacity is set to 1."
-
- Richard Deodutt from class worked with me for over 2 hours to try to find the reason for this capacity issue .
 - We found that since my vCPU capacity is only one, I was not able to run elastic beanstalk and Jenkins EC2 at the same time.
 - I contacted AWS support to request assistance for increasing my instance quota. I am awaiting Amazon's response.

<p>Amazon Web Services</p> <p>Sat Sep 24 2022 21:46:38 GMT-0400 (Eastern Daylight Time)</p>	<p>was this response helpful? Click here to rate: ★ ★ ★ ★ ★</p> <p>Hello,</p> <p>Thank you contacting Amazon Web Services.</p> <p>I understand that you are contacting us regarding Service Quota increase for EC2.</p> <p>We've received your EC2 All Standard (A, C, D, H, I, M, R, T, Z) Instances quota increase request for 20 in the US East (Northern Virginia) region.</p> <p>For a quota increase of this type, I will need to collaborate with our service team to get approval. Please note that it can take some time for the service team to review your request. This is to ensure that we can meet your needs while keeping existing infrastructure safe.</p> <p>I will follow up with you as soon as I receive an update from the service team.</p> <p>If you need any further clarifications, please feel free to reach out to me and it'll be my pleasure to assist you.</p> <p>Have a great day ahead!</p> <p>We value your feedback. Please share your experience by rating this and other correspondences in the AWS Support Center. You can rate a correspondence by selecting the stars in the top right corner of the correspondence.</p> <p>Best regards, Dushyant S. Amazon Web Services</p>
<p>Anj-Kura</p> <p>Sat Sep 24 2022 21:27:55 GMT-0400 (Eastern Daylight Time)</p>	<p>01:31:32 AM Dushyant: Hello,</p> <p>01:31:40 AM Customer: could you please increase my limit?</p> <p>01:31:52 AM Customer: Hi thanks for chatting</p> <p>01:32:03 AM Dushyant: Please allow me a moment to check on this for you.</p> <p>01:33:01 AM Customer: Thanks! I really need the limit increase for class. All my elastic beanstalks are degrading because my limit is set to one.</p> <p>01:34:44 AM Dushyant: To get this addressed, I will need to reach out to the service team as they have the necessary resources and tools to review this further. I will follow up with you via email as soon as I receive an update from them.</p> <p>01:35:05 AM Customer: ok thank you Dushyant</p> <p>01:35:15 AM Dushyant: You're welcome</p> <p>01:35:25 AM Dushyant: Is there anything else, I can help you with?</p> <p>01:36:03 AM Customer: I look forward to hearing back from you. Nothing else I need help with. Should I exit the chat now?</p> <p>01:36:14 AM Dushyant: Yes we can end the chat.</p> <p>01:36:23 AM Customer: ok goodbye</p>
<p>Anj-Kura</p> <p>Sat Sep 24 2022 21:27:52 GMT-0400 (Eastern Daylight Time)</p>	<p>Limit increase request 1</p> <p>Service: EC2 Instances</p> <p>Region: US East (Northern Virginia)</p> <p>Primary Instance Type: All Standard (A, C, D, H, I, M, R, T, Z) Instances</p> <p>Limit name: Instance Limit</p> <p>New limit value: 20</p> <p>-----</p> <p>Use case description: I am in a coding/ cloud computing course and need more instances/capacity to complete assignments</p>

Step 10: I added an extra test to the pipeline.

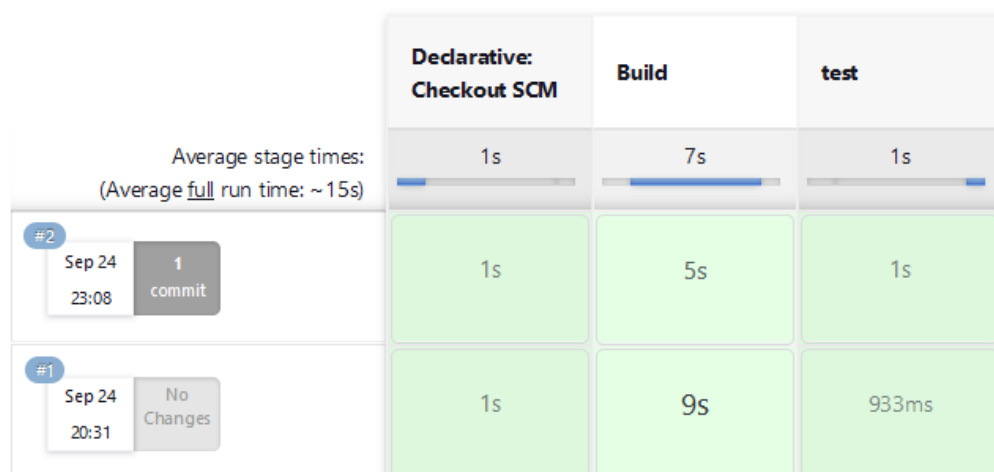
- I added the below code to the test_app.py file

```
# def test_fast():
#     b = "Bobby"
#     new_greeting = greet(b)
#     assert new_greeting == "Good Evening Bobby"
```

```
def test_home_page():
    response = app.test_client().get('/')
    assert response.status_code == 200
```

- I built the pipeline again and the test ran successfully.

Stage View



What could have been done differently:

- To install the eb command I could have added the path to the bash.rc file so that eb would have been added to my path. Since I only installed the eb on the Jenkins user, it needed to be reinstalled every time I stopped and restarted the instance.
- To prevent my initial build from failing, I should have installed python on my EC2, especially because the url-shortener is a python application.
- I could have also made sure that I had adequate vCPU capacity before I began the deployment.