

Deployment #1

Welcome to Deployment 1!! This deployment will walk you through setting up your pipeline. Take notes of each main step of the pipeline. Observe all the tools being used in the pipeline. There will be more in future deployments!!

Install Jenkins on an EC2:

- First create an Ubuntu EC2
- The EC2 will need port 80, 8080, and 22 open
- Once you've created the EC2, log into the EC2 and then enter the commands below to install Jenkins:

```
$sudo apt update && sudo apt install default-jre
```

```
$wget -q -O -
```

```
https://pkg.jenkins.io/debian-stable/jenkins.io.  
key |sudo gpg --dearmor -o  
/usr/share/keyrings/jenkins.gpg
```

```
$sudo sh -c 'echo deb
```

```
[signed-by=/usr/share/keyrings/jenkins.gpg]
```

```
http://pkg.jenkins.io/debian-stable binary/ >
```

```
/etc/apt/sources.list.d/jenkins.list'
```

```
$sudo apt update && sudo apt install jenkins -y
```

```
$sudo systemctl start jenkins
```

```
$sudo systemctl status jenkins
```

- Follow instructions via link below to setup Jenkins:
[https://www.jenkins.io/doc/tutorials/tutorial-for-installing-jenkins-on-AWS/](https://www.jenkins.io/doc/tutorials/tutorial-for-installing-jenkins-on-aws/)

Install Virtual Environment:

- Remote into the EC2 and install the two packages via the apt command:
 - python3-pip
 - python3-10-venv

Connect GitHub to Jenkins Server:

- First Fork the Deployment repo:
https://github.com/kura-labs-org/kuralabs_deployment_1.git
- Next, create an access token from GitHub:
 - Navigate to your GitHub settings, select developer settings

Emails

Password and authentication

SSH and GPG keys

Organizations

Moderation

Code, planning, and automation

Repositories

Packages

GitHub Copilot

Pages

Saved replies

Security

Code security and analysis

Integrations

Applications

Scheduled reminders

Archives

Security log

Sponsorship log

Developer settings

Check out our guide to [generating SSH keys](#) or [troubleshoot common SSH problems](#).

GPG keys

New GPG key

There are no GPG keys associated with your account.

Learn how to [generate a GPG key and add it to your account](#).

Vigilant mode

☐ **Flag unsigned commits as unverified**

This will include any commit attributed to your account but not signed with your GPG or S/MIME key. Note that this will include your existing unsigned commits.

[Learn about vigilant mode.](#)

- Select personal access token and create a new token.

Pull requests

Issues

Marketplace

Explore

Settings / Developer settings

GitHub Apps

OAuth Apps

Personal access tokens

Personal access tokens

Generate new tokenRevoke all

Tokens you have generated that can be used to access the [GitHub API](#).

test2 — admin:repo_hook, notifications, repo

Last used within the last 4 months

Delete

Expired on **Fri, May 27 2022**.

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

© 2022 GitHub, Inc.

[Terms](#)

[Privacy](#)

[Security](#)

[Status](#)

[Docs](#)

[Contact GitHub](#)

[Pricing](#)

[API](#)

[Training](#)

[Blog](#)

[About](#)

- 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

Create a multibranch build:

- Log back into Jenkins and select “New item”




+ 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

- Select multibranch pipeline

The screenshot shows the Jenkins 'New Item' page. At the top, there's a navigation bar with the Jenkins logo and a search bar. Below the navigation bar, there's a breadcrumb trail: 'board > All >'. The main content area is titled 'Enter an item name' and contains a text input field with the value 'url-shortener'. Below the input field, there's a link that says '» Required field'. The main content area also lists several item types with icons and descriptions:

- Freestyle project**: This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.
- Pipeline**: Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
- Multi-configuration project**: Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
- Folder**: Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.
- Multibranch Pipeline**: Creates a set of Pipeline projects according to detected branches in one SCM repository. (This option is highlighted with a blue border.)
- Organization Folder**: Creates a set of multibranch project subfolders by scanning for repositories.

Below the list of item types, there's a section titled 'If you want to create a new item from other existing, you can use this option:'. It contains a 'Copy from' section with a text input field and an 'OK' button.

- Enter a display name and 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 ? ✕

- 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

er >

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

Filter

- Git
- GitHub
- Single repository & branch
- Wolfram

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 ?

://configure#

- Click on Add and then select 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

Jenkins

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 enter your GitHub username
- Under password enter your token

The image shows a screenshot of the Jenkins 'Global Credentials' configuration page. The form is titled 'Description' and contains the following fields and options:

- Global credentials (unrestricted)**: A dropdown menu at the top.
- Kind**: A dropdown menu set to 'Username with password'.
- Scope**: A dropdown menu set to 'Global (Jenkins, nodes, items, all child items, etc)'.
- Username**: A text input field containing 'kura-labs01'.
- Treat username as secret**: An unchecked checkbox.
- Password**: A text input field filled with asterisks.
- ID**: An empty text input field.
- Description**: A text input field containing 'GitHub Token'.
- Buttons**: 'Add' and 'Cancel' buttons at the bottom left.

- (Optional) under ID and Description enter GitHub repo
- Enter your URL to the repository and you can validate by selecting validate.

ener

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 ?

kura-labs01/***** (GitHub Token) ▼

+ Add

☒ Repository HTTPS URL

Repository HTTPS URL ?

https://github.com/kura-labs-org/kuralabs_deployment_1.git

Credentials ok. Connected to https://github.com/kura-labs-org/kuralabs_deployment_1. Validate

☐ Repository Scan - Depreciated Visualization

Behaviors

Discover branches ?

Strategy ?

Exclude branches that are also filed as PRs ▼

Save Apply

- Make sure this says Jenkinsfile

>

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 ?

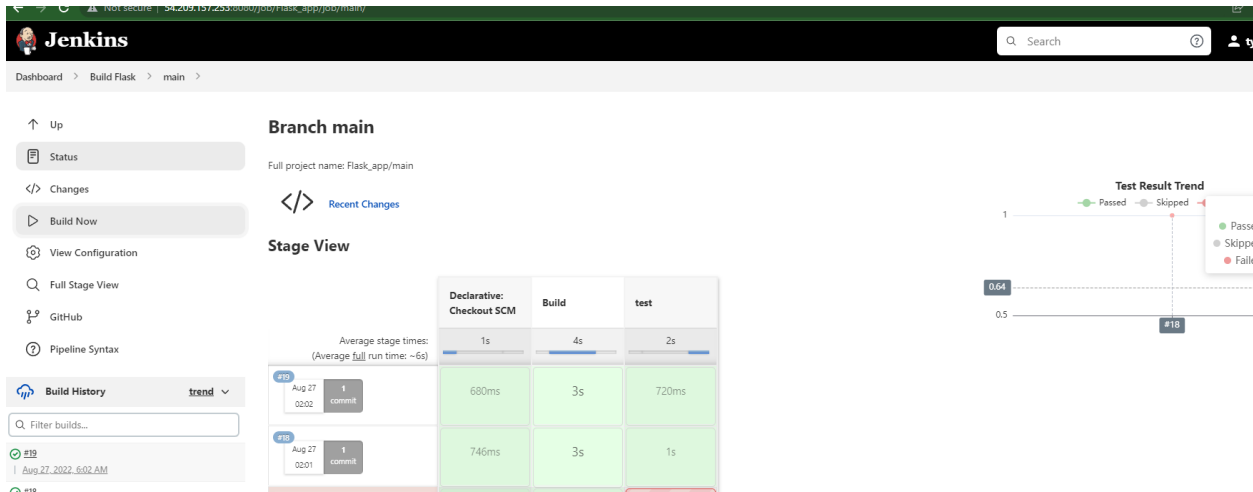
Orphaned Item Strategy

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 Apply and then Save
- You should see a build happening. If you don't, select Scan Repository.



Download application files from GitHub and deploy to Elastic Beanstalk:

- Use git clone to copy deployment1 repo files of the flask application to your local computer.
- To compress the files from the repo, follow the how to from aws documentation:
<https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/applications-sourcebundle.html>
- F
- After you have compressed/zipped your files, head over to AWS Elastic Beanstalk
- Select create a new environment

AWS Graviton now supported

AWS Graviton, an arm64-based processor, can offer up to 40% better price performance over the comparable x86 processor. To upgrade to an arm64 instance type, choose the 'Capacity' settings in 'Additional configuration.'

Elastic Beanstalk > Environments

All environments

Filter results matching the display values

< 1 >

Environment name ▲	Health ▼	Application name ▼	Date created ▼	Last modified ▼	URL ▼	Running versions ▼	Platform ▼	Platform state ▼
Empty								
No environments to display.								

Create a new environment

- Select Web Server environment

AWS Graviton now supported

AWS Graviton, an arm64-based processor, can offer up to 40% better price performance over the comparable x86 processor. To upgrade to an arm64 instance type, choose the 'Capacity' settings in 'Additional configuration.'

Elastic Beanstalk > Create environment

Select environment tier

Amazon Elastic Beanstalk has two types of environment tiers to support different types of web applications. Web servers are standard applications that listen for and then process HTTP requests, typically over port 80. Workers are specialized applications that have a background processing task that listens for messages on an Amazon SQS queue. Worker applications post those messages to your application by using HTTP.

☒ Web server environment

Run a website, web application, or web API that serves HTTP requests.
[Learn more](#)

☐ Worker environment

Run a worker application that processes long-running workloads on demand or performs tasks on a schedule.
[Learn more](#)

Cancel

Select

- Enter the following configurations:
 - Application name: url-shortner
 - Environment name: Urlshortner-env
 - Platform: Python
 - Platform branch: 3.8
 - Platform version: 3.3.16
 - Application code: Upload your code, local file

Create a web server environment

Launch an environment with a sample application or your own code. By creating an environment, you allow Amazon Elastic Beanstalk to manage Amazon Web Services resources and permissions on your behalf. [Learn more](#)

Application information

Application name

Up to 100 Unicode characters, not including forward slash (/).

► Application tags (optional)

Environment information

Choose the name, subdomain, and description for your environment. These cannot be changed later.

Environment name

Domain

.us-east-1.elasticbeanstalk.

Check availability

Platform

☒ **Managed platform**
Platforms published and maintained by Amazon Elastic Beanstalk. [Learn more](#)

☐ **Custom platform**
Platforms created and owned by you.

Platform

Python

Platform branch

Python 3.8 running on 64bit Amazon Linux 2

Platform version

3.3.16 (Recommended)

Application code

☐ **Sample application**
Get started right away with sample code.

☐ **Existing version**
Application versions that you have uploaded for url-shortner.

-- Choose a version --

☒ **Upload your code**
Upload a source bundle from your computer or copy one from Amazon S3.

Version label


Unique name for this version of your application code.

url-shortner-source

Source code origin
Maximum size 512 MB

☒ **Local file**

☐ Public S3 URL

 **Choose file**

☐ No file uploaded

► Application code tags

Cancel

Configure more options

Create environment

- Once the environment finishes creating, select the URL AWS provides you on the page of your application health. See below:

Elastic Beanstalk ×

AWS Graviton now supported
AWS Graviton, an arm64-based processor, can offer up to 40% better price performance over the comparable x86 processor. To upgrade to an arm64 instance type, choose it in the 'Capacity' settings in 'Additional configuration.'

Elastic Beanstalk > Environments > Urlshortner4-env

Urlshortner4-env
Urlshortner4-env.eba-73wpsjuw.us-east-2.elasticbeanstalk.com (e-mqtkj@w7g)
Application name: url-shortner4

Refresh Actions

Health
Ok
Causes

Running version
url-shortner4-source
Upload and deploy

Platform
Python 3.8 running on 64bit Amazon Linux 2/3.3.16
Change

Recent events Show all

Time	Type	Details
2022-08-26 22:12:03 UTC-0400	INFO	Environment health has transitioned from Pending to Ok. Initialization completed 53 seconds ago and took 3 minutes.
2022-08-26 22:11:36 UTC-0400	INFO	Successfully launched environment: Urlshortner4-env
2022-08-26 22:11:36 UTC-0400	INFO	Application available at Urlshortner4-env.eba-73wpsjuw.us-east-2.elasticbeanstalk.com.
2022-08-26 22:11:05 UTC-0400	INFO	Instance deployment completed successfully.

Review and document what you observed while setting up this deployment

Diagram the pipeline

What could you improve

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