

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 token

Revoke 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

Review and document what you observed while setting up this deployment:

Diagram the pipeline:

What could you improve: