

“Expert Cloud Consulting”

SOP | Jenkins Installation and CI/CD Setup on Ubuntu EC2

11 july 2025

Contributed by: Akshata

Approved by: Akshay (In Review)

Expert Cloud Consulting

Office #811, Gera Imperium Rise,

Hinjewadi Phase-II Rd, Pune, India – 411057

Jenkins Installation and CI/CD Setup on Ubuntu EC2



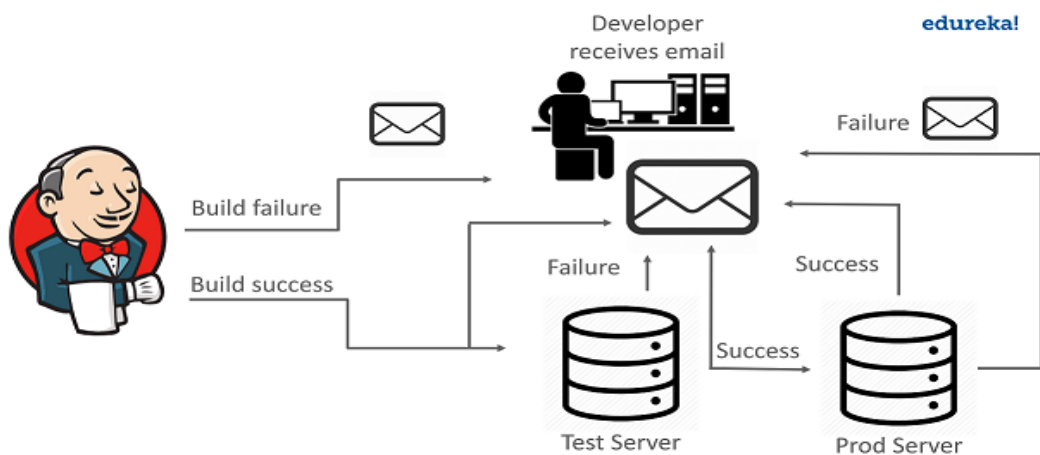
CICD Fundamentals

Topics:

- CI/CD principles and tools (Jenkins, GitLab CI).
- Setting up pipelines for build, test, deploy.

Assignments:

1. Build a CI/CD pipeline using Jenkins to:
 - Pull code from GitHub.
 - Run unit tests automatically.
 - Build and deploy a Docker container to a staging environment.
2. Integrate notifications:
 - Configure Jenkins to send Slack/email alerts on pipeline failures.



Objective

The objective of this document is to provide a step-by-step guide for installing and configuring Jenkins on an Ubuntu-based EC2 instance. It also covers integrating Jenkins with GitHub using webhooks for automated builds, deploying applications using Docker, and setting up email notifications for build status alerts. This setup enables continuous integration and continuous delivery (CI/CD) for seamless DevOps workflows.

Document Overview

This document provides a comprehensive guide for setting up Jenkins on an Ubuntu EC2 instance to implement a complete CI/CD pipeline. It includes the following key components:

System Preparation

Updating system packages and prerequisites for Jenkins installation.

Jenkins Installation

Adding official Jenkins repository and installing the latest stable version.

Service Configuration

Enabling and starting Jenkins, configuring firewall and systemd.

Docker Setup

Installing Docker and configuring user permissions for Jenkins and the current user.

Jenkins Dashboard Access

Accessing the Jenkins UI through a web browser, setting up the admin user.

Freestyle Project Creation

Creating a sample Jenkins project to pull code from GitHub and run custom Docker build steps.

GitHub Webhook Integration

Configuring GitHub to notify Jenkins on code changes via webhooks for automated builds.

Troubleshooting EC2 Issues



Solutions for common EC2 startup or SSH issues.

Email Notification Configuration

Sending build result notifications through Gmail using App Passwords.

Final Testing

Verifying auto-builds on code changes and email alerts on success or failure.

This guide ensures a practical understanding of Jenkins-based automation and helps set up a foundational DevOps environment using open-source tools.

Document References

The following resources were referred to during the creation and execution of this Terraform-based infrastructure setup

Date	Document	Filename / Url
9 July	Email Notification In Jenkins	https://www.edureka.co/blog/email-notification-in-jenkins/
10 July	Install Jenkins	https://www.jenkins.io/doc/book/installing/

Jenkins Installation

Update the system first

```
sudo apt update
```

Instructions are available on this page

<https://pkg.jenkins.io/debian-stable/>

Install Java

```
sudo apt update
```

```
sudo apt install fontconfig openjdk-21-jre
```

```
java -version
```



install Jenkins

```

sudo wget -O /etc/apt/keyrings/jenkins-keyring.asc \
  https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
echo "deb [signed-by=/etc/apt/keyrings/jenkins-keyring.asc]"
  https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update
sudo apt-get install jenkins

```

Start and Enable Jenkins

Check Jenkins Status

```
sudo systemctl status Jenkins
```

Install Docker and Configure Access

```
sudo apt install docker.io
```

```
sudo usermod -aG docker $USER
```

```
sudo usermod -aG docker Jenkins
```

Note: Accessing Jenkins dashboard if it is running through ec2 instance

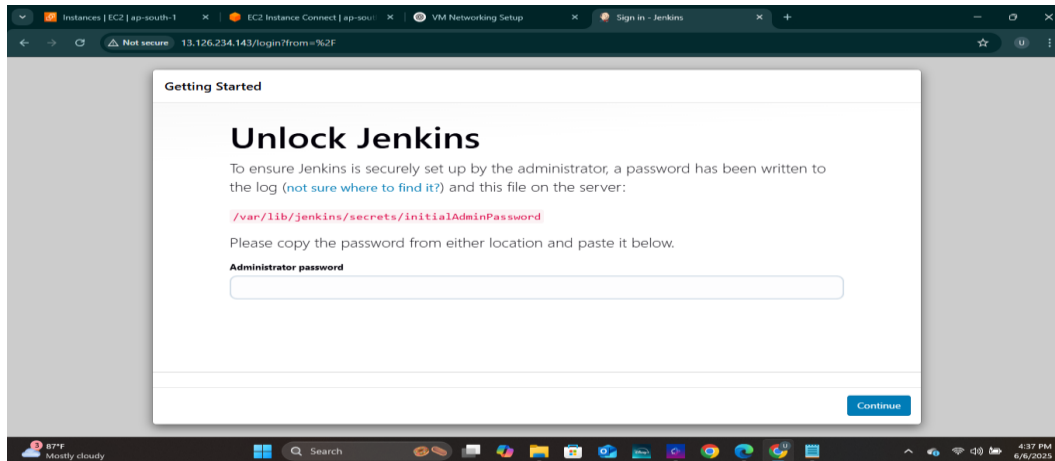
EC2 > Security Groups > sg-09bf47418eba670fc - launch-wizard-1 > Edit inbound rules

Security group rule ID	Type	Protocol	Port range	Source	Description - optional	
sgr-02224f4179f8b8a56	Custom TCP	TCP	8080	Cust...	0.0.0.0	Delete
sgr-0bd3002c472ec8845	SMTPS	TCP	465	Cust...	0.0.0.0	Delete
sgr-028c649c4e647c8c4	SSH	TCP	22	Cust...	0.0.0.0	Delete
sgr-0bdfb200a4593db82	Custom TCP	TCP	8000	Cust...	0.0.0.0	Delete
sgr-0b925cc3fbb2f3b67	HTTP	TCP	80	Cust...	0.0.0.0	Delete
sgr-08497036aec8e803e	HTTPS	TCP	443	Cust...	0.0.0.0	Delete



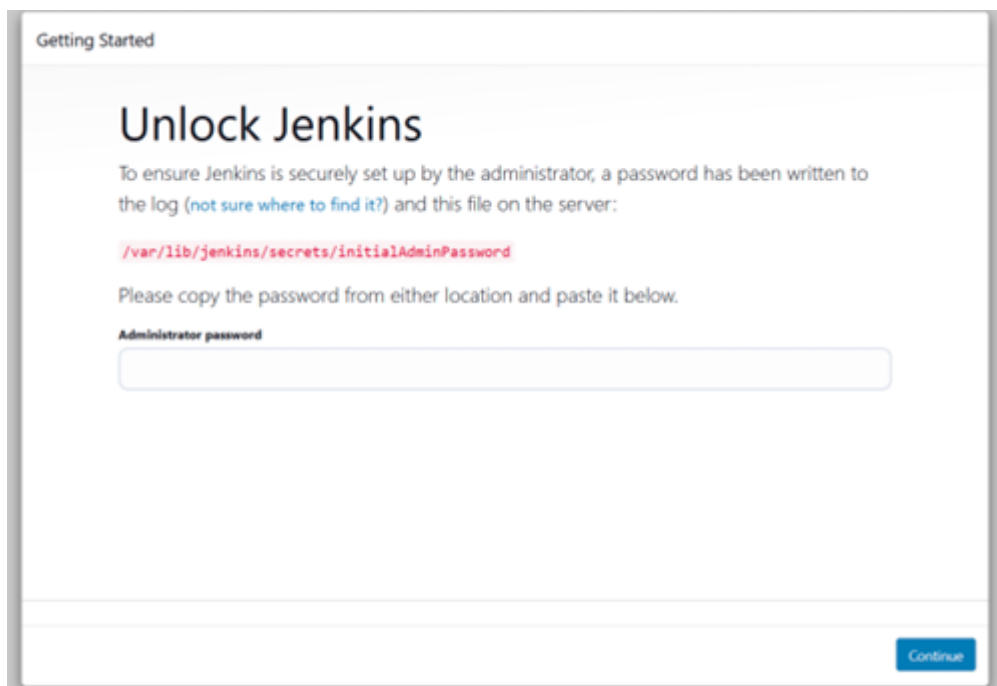
Add the above inbound rules to access successfully by default port 22 which is for ssh is enabled add the remaining type like HTTP, HTTPS also custom tcp 8080 too.

After the above changes dashboard is accessible at port 8080, for ec2 type public_ip4_address:8080

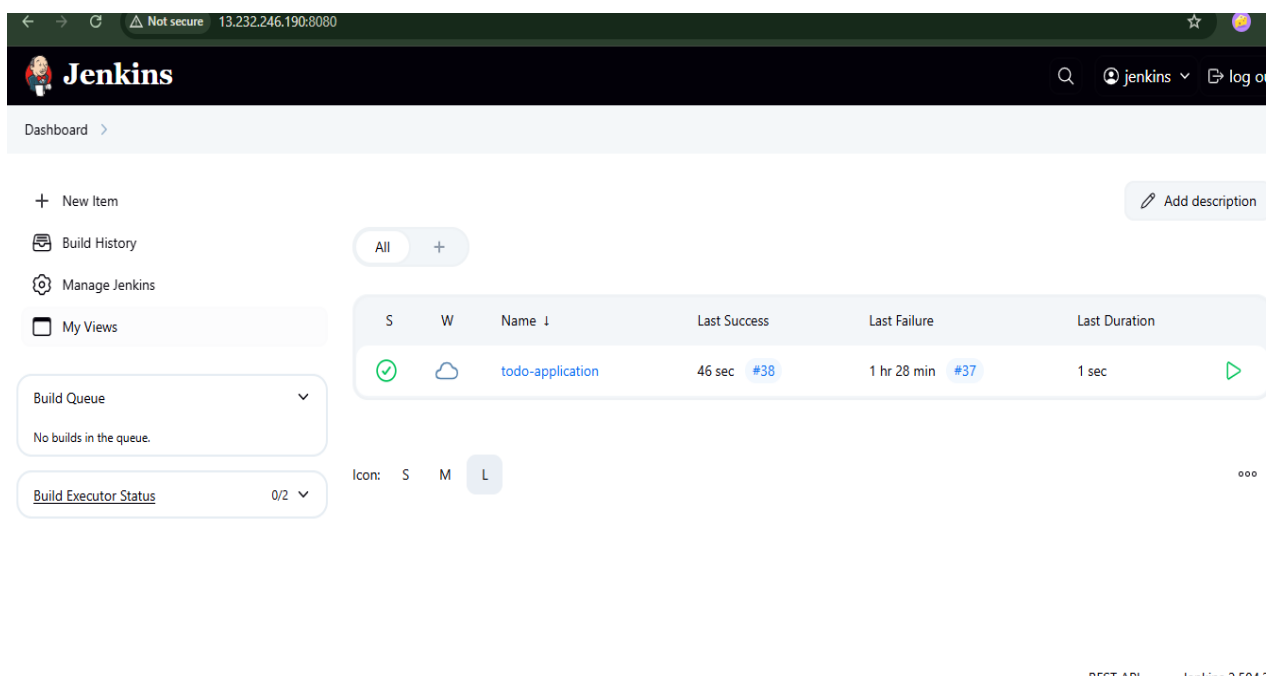
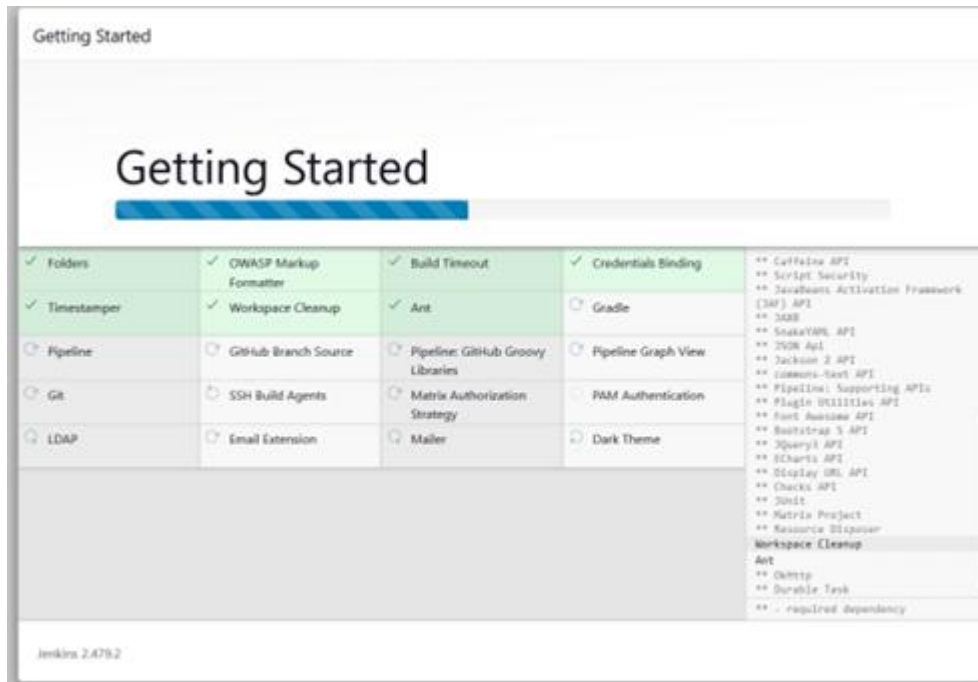


The above dashboard is visible, now to access admin password go to terminal and type
`sudo cat /var/lib/jenkins/secrets/initialAdminPassword`

Install suggested plugins, create admin user and now its ready



click on install suggested plugins



Go to jenkins dashboard click on create new item > freestyle project and then click ok


Dashboard > All > **New Item**


New Item


Enter an item name

todo-application

Select an item type

 **Freestyle project**
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

 **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.

OK

Source Code Management

Connect and manage your code repository to automatically pull the latest code for your builds.

☐ None

☒ Git ?

Repositories ?

Repository URL ?

https://github.com/akshataujawane/demo-app-apolicatin

Credentials ?

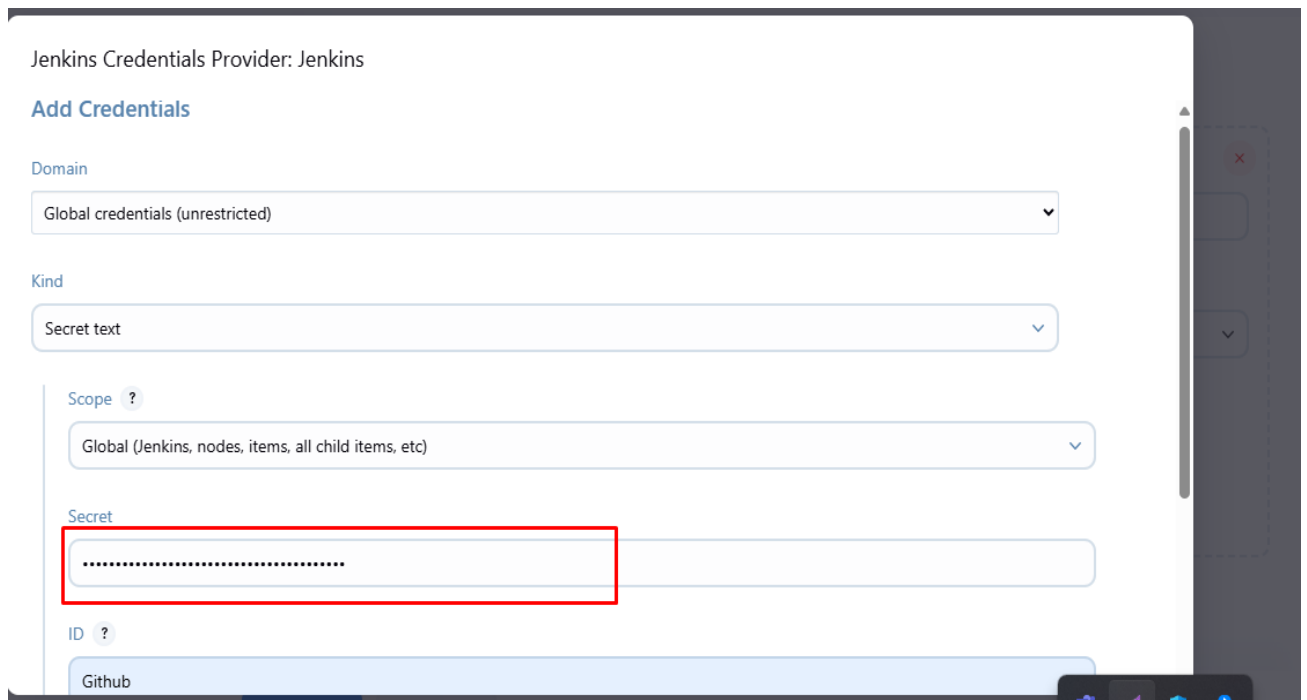
- none -

+ Add

Advanced ▾



after this we select Git and in credentials section click on add Jenkins Credential Provider



The screenshot shows the 'Jenkins Credentials Provider: Jenkins' configuration page. The 'Add Credentials' section is active. The 'Domain' dropdown is set to 'Global credentials (unrestricted)'. The 'Kind' dropdown is set to 'Secret text'. The 'Scope' dropdown is set to 'Global (Jenkins, nodes, items, all child items, etc)'. The 'Secret' field is highlighted with a red rectangle and contains a series of dots. The 'ID' field is set to 'Github'.

Add git branch



The screenshot shows the 'Branches to build' configuration page. The 'Branch Specifier (blank for 'any')' field is highlighted with a red rectangle and contains the text '*/develop'. The 'Add Branch' button is visible below the field.

after adding the credentials and branch click on build now and jenkins will clone the code from github repository to ec2 instance



go to configure then click on build environment and then click on build Scroll down you will see Build Steps option



Dashboard > todo-application > Configuration

Configure

- General
- Source Code Management
- Triggers
- Environment
- Build Steps**
- Post-build Actions

Build Steps

Automate your build process with ordered tasks like code compilation, testing, and deployment.

Execute shell ?

Command

[See the list of available environment variables](#)

```
docker build . -t todo-app
docker run -d -p 8000:8000 todo-app
```

Advanced ▾

Webhooks

When you create a webhook, you specify a URL and subscribe to events that occur on GitHub. When an event that your webhook is subscribed to occurs, GitHub will send an HTTP request with data about the event to the URL that you specified. If your server is set up to listen for webhook deliveries at that URL, it can take action when it receives one.

In short, whenever any changes are done to the repository and pushed then it will trigger the build the pipeline.

Dashboard > Manage Jenkins > Plugins

Plugins

Q git

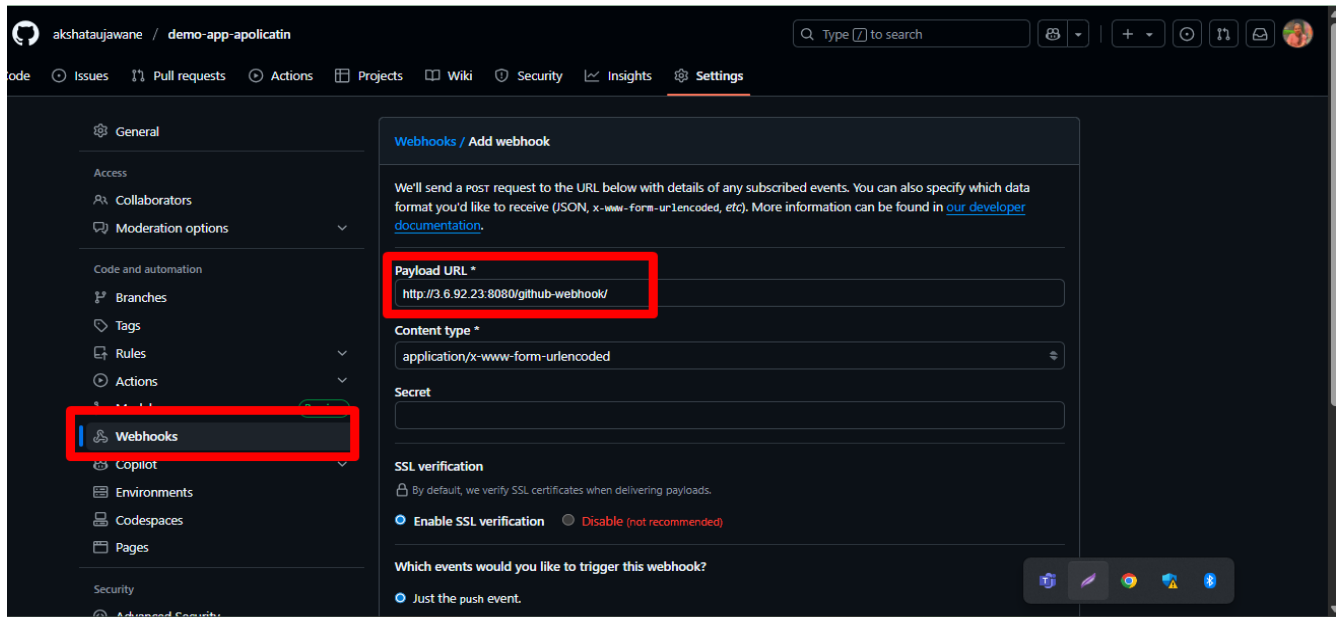
Name ↓	Enabled
Git client plugin 6.2.0 Utility plugin for Git support in Jenkins Report an issue with this plugin	<input checked="" type="checkbox"/>
Git plugin 5.7.0 This plugin integrates Git with Jenkins. Report an issue with this plugin	<input checked="" type="checkbox"/>
Git server Plugin 137.ve0060b_432302 Allows Jenkins to act as a Git server. Report an issue with this plugin	<input checked="" type="checkbox"/>
GitHub API Plugin 1.321-488.v9b_c0da_9533f8 This plugin provides GitHub API for other plugins.	<input checked="" type="checkbox"/>

Install github integration plugin

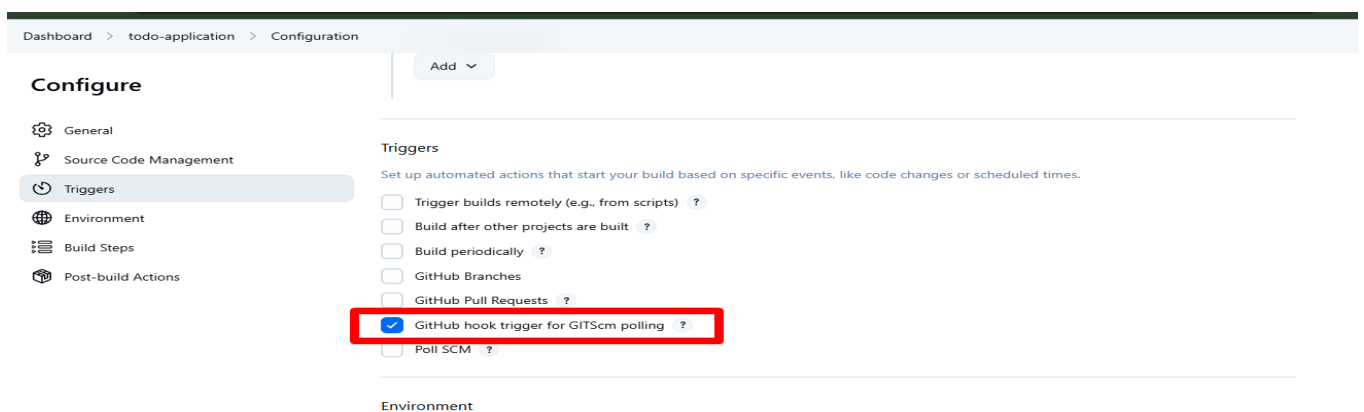
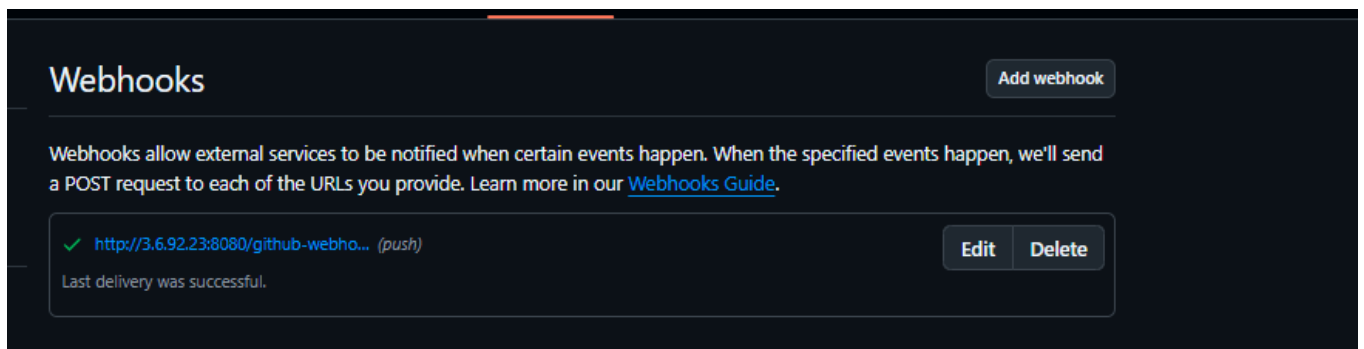


now go to your repository in github, click on settings

go to jenkins dashboard and copy the url add it as follows and changed the content type and click on add webhook



we can see here the webhook is created



now click on save



now we will make changes to code

```
{% endblock %}

{% block content %}
<div class="container">

  <!-- title row -->
  <div class="row">
    <div class="offset-md-2 col-lg-9">
      <div class="page-header">
        <h1>
          Welcome to ECC |
        </h1>
      </div>
    </div>
  </div>

  <!-- Add a todo row -->
  <div class="row">
    <div class="offset-md-2 col-lg-9">
      <form method="post" action="{% url 'todos:add' %}">
        {% csrf_token %}
        <div class="form-row">
          <div class="col-md-6">
```

here we will add some thing

```
{% block title %}
<title>Todo list</title>
{% endblock %}

{% block content %}
<div class="container">

  <!-- title row -->
  <div class="row">
    <div class="offset-md-2 col-lg-9">
      <div class="page-header">
        <h1>
          Welcome to ECC .....
        </h1>
      </div>
    </div>
  </div>

  <!-- Add a todo row -->
  <div class="row">
    <div class="offset-md-2 col-lg-9">
      <form method="post" action="{% url 'todos:add' %}">
        {% csrf_token %}
```



▶ Build Now

⚙️ Configure

🗑️ Delete Project

📄 GitHub Hook Log

✎ Rename

- Last build (#4), 18 hr ago
- Last stable build (#4), 18 hr ago
- Last successful build (#4), 18 hr ago
- Last failed build (#3), 18 hr ago
- Last unsuccessful build (#3), 18 hr ago
- Last completed build (#4), 18 hr ago

Builds

Today

✓ #5 7:29 AM

December 11, 2024

✓ #4 12:42 PM

✗ #3 12:38 PM

✗ #2 12:35 PM

✓ #1 7:05 AM

here we can see jenkins is running build automatically

▶ Build Now

⚙️ Configure

🗑️ Delete Project

📄 GitHub Hook Log

✎ Rename

- Last build (#4), 18 hr ago
- Last stable build (#4), 18 hr ago
- Last successful build (#4), 18 hr ago
- Last failed build (#3), 18 hr ago
- Last unsuccessful build (#3), 18 hr ago
- Last completed build (#4), 18 hr ago

Builds

Today

✓ #5 7:29 AM

December 11, 2024

✓ #4 12:42 PM

✗ #3 12:38 PM

✗ #2 12:35 PM

✓ #1 7:05 AM



the build failed now we will go through console to see what failed exactly by checking

Dashboard > todo-application > #38

Status

</> Changes

Console Output

Edit Build Information

Delete build '#38'

Timings

Git Build Data

← Previous Build

→ Next Build

✓ #38 (11 Jul 2025, 07:41:04)

Started by user [jenkins](#)

This run spent:

- 3 ms waiting;
- 1 sec build duration;
- 1 sec total from scheduled to completion.

Revision: b1e07243f0cf308857bed68849a8f947beafc8
Repository: <https://github.com/akshataujawane/demo-app-apolicatin>

- refs/remotes/origin/develop

</> No changes.

Add description

Keep this build forever

Started 2 hr 51 min ago
Took 1 sec

⚠ Not secure 13.232.246.190:8000/todos/

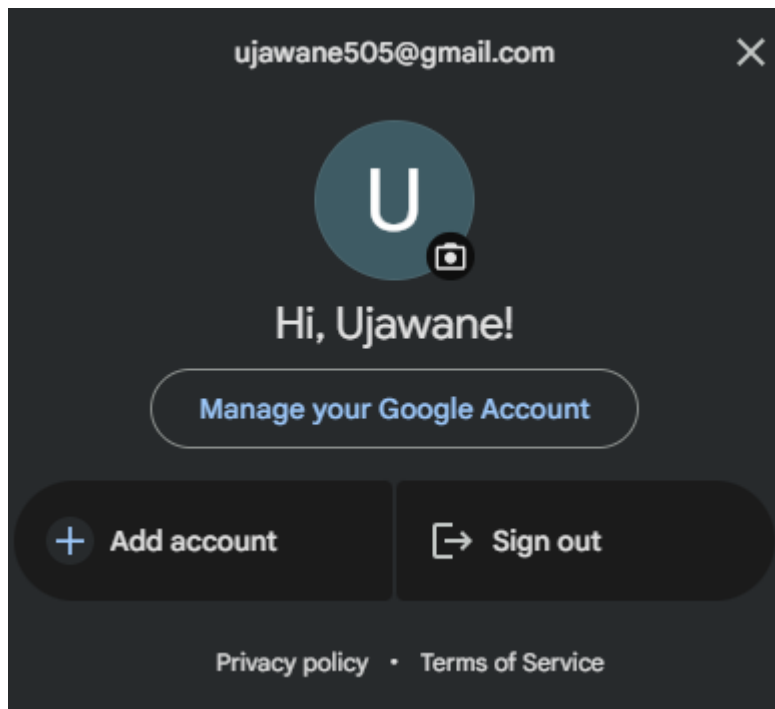
Welcome to ECC

☒ Docker file banao

☒ Hacktoberfest Updates

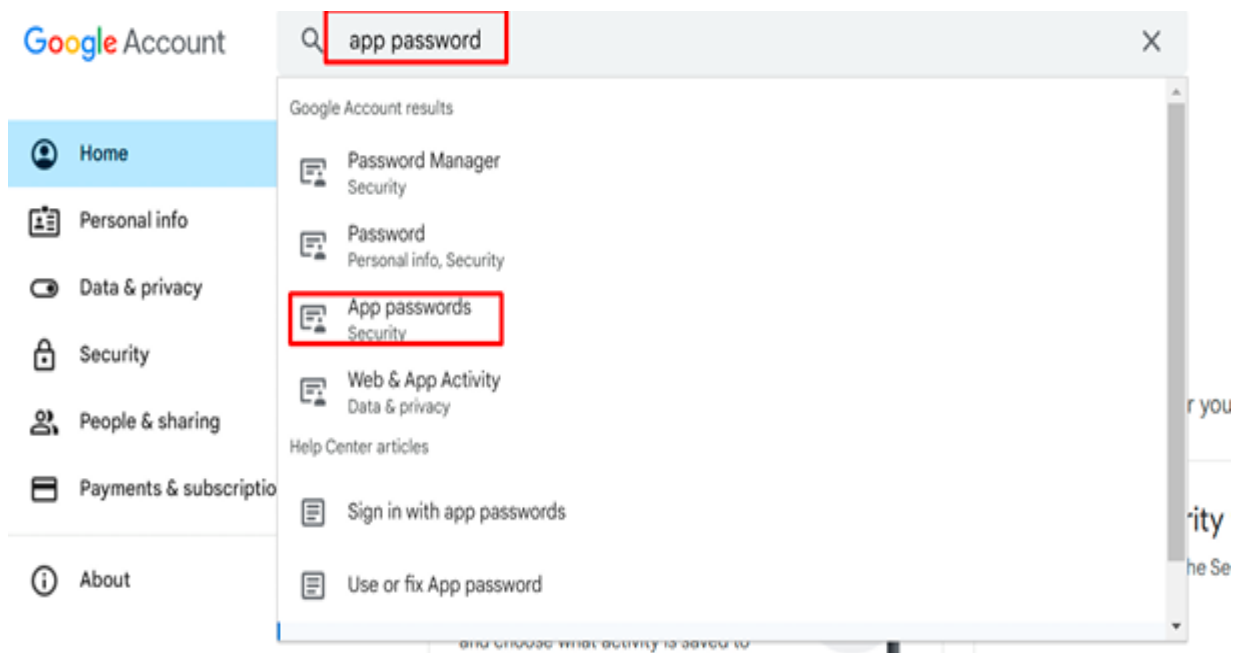
now to send email alerts, for that we will need to setup app password in google

Go to your google profile



click on manage your google

account search for app password and click on app password



enter your app name and click on create

← App passwords

App passwords help you sign into your Google Account on older apps and services that don't support modern security standards.

App passwords are less secure than using up-to-date apps and services that use modern security standards. Before you create an app password, you should check to see if your app needs this in order to sign in.

[Learn more](#)

Your app passwords

Mail

Created on Dec 3, last used on Dec 11

To create a new app specific password, type a name for it below...

App name
jenkins

Create

it will give you a password save it somewhere, we will require it. Go to your Jenkins dashboard click on Manage Jenkins

scroll down and go to Email Notification section

Manage Jenkins > System >

E-mail Notification

SMTP server

smtp.gmail.com

Default user e-mail suffix ?

Advanced ^ Edited

☒ Use SMTP Authentication ?

User Name

ujawane505@gmail.com

Password

Concealed

Change Password

Save

Apply



Jenkins Credentials Provider: Jenkins

Add Credentials

Domain

Global credentials (unrestricted)

Kind

Secret text

Scope ?

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

Secret

.....

ID ?

Github

Dashboard > Manage Jenkins > System >

Reply-To Address

ujawane505@gmail.com

Charset

UTF-8

☒ Test configuration by sending test e-mail

Test e-mail recipient

ujawane505@gmail.com

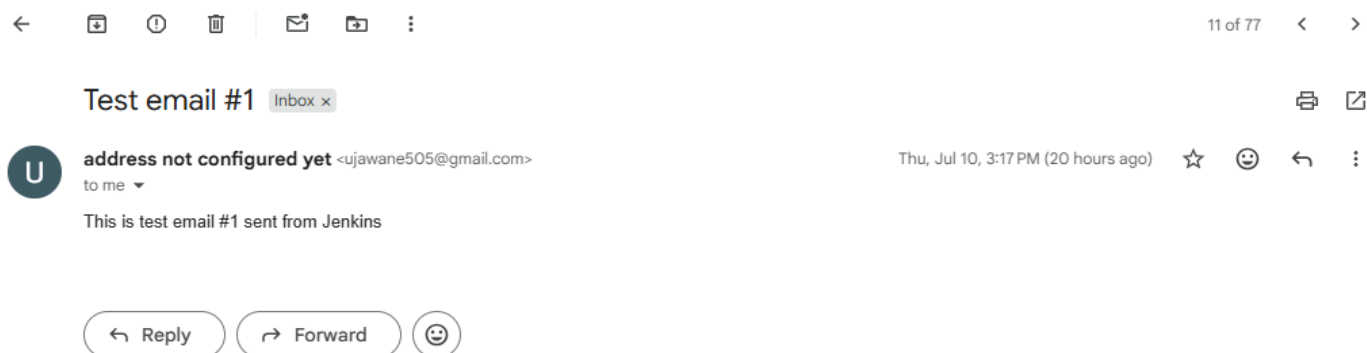
Email was successfully sent

Test configuration

Save Apply

REST API Jenkins 2.504.3

here we can see an email was sent



now go to extended email notifications section

Extended E-mail Notification

SMTP server

smtp.gmail.com

SMTP Port

465

Advanced ^ Edited

Credentials

E-mail-todo-application

+ Add

☒ Use SSL

☐ Use TLS

Save Apply

Jenkins Credentials Provider: Jenkins

Add Credentials

Domain

Global credentials (unrestricted)

Kind

Username with password

Scope ?

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

Username ?

email-todo app

☒ Treat username as secret ?



here the password is app password we created on google account

Jenkins Credentials Provider: Jenkins

Scope ?

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

Username ?

email-todo app

☒ Treat username as secret ?

Password ?

.....

ID ?

> Manage Jenkins > System >

Default Triggers ^

Default Triggers ?

☐ Aborted

☒ Always

☐ Before Build

☐ Failure - 1st

☐ Failure - 2nd

☒ Failure - Any

☐ Failure - Still

☐ Failure - X

☐ Failure -> Unstable (Test Failures)

☐ Fixed

☐ Not Built

☐ Script - After Build

☐ Script - Before Build

☐ Status Changed

☒ Success

☐ Test Improvement

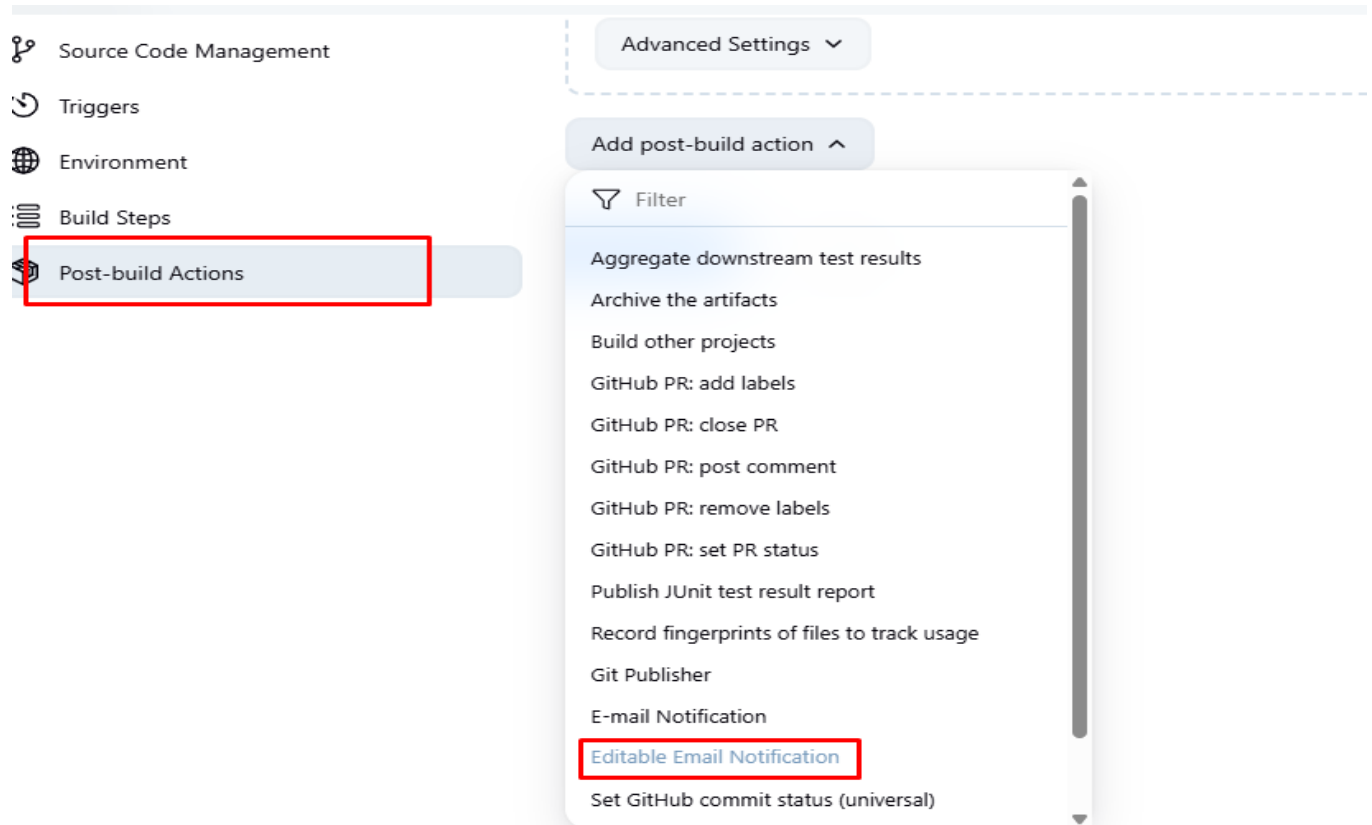
Save

Apply



save it

now go to configure from dashboard click on add post-build action and then click on editable email notification



Content Type ?

Project Content Type

Subject ?

\$PROJECT_DEFAULT_SUBJECT

Content ?

🔥 Your Jenkins pipeline has fail!
Please check your application
<http://3.6.92.23:8000/todos/>

Attachments ?

Can use wildcards like 'module/dist/**/*.*.zip'. See the [@includes of Ant fileset](#) for the exact format. The base directory is [the workspace](#).

When my pipeline fails, I receive a failure email

Dashboard > todo-application > #36 > Console Output

```

---> Using cache
---> 55f663c9307f
Step 6/7 : EXPOSE 8000
---> Using cache
---> f554f6c402fa
Step 7/7 : CMD ["python", "manage.py", "runserver", "0.0.0.0:8000"]
---> Using cache
---> d07ef3ed1f81
Successfully built d07ef3ed1f81
Successfully tagged todo-app:latest
+ docker run -d -p 8000:8000 todo-app
65c0c9c62e09b696118ec5460cb5a25d706b8c5cdda7942945bb66bb4c6eed41
docker: Error response from daemon: driver failed programming external connectivity on endpoint exciting_black
(4bab09f524470ec45ded574d24943476a61f335ca3d89e50b528005f094c882c): Bind for 0.0.0.0:8000 failed: port is already
allocated.
Build step 'Execute shell' marked build as failure
Email was triggered for: Failure - Any
Sending email for trigger: Failure - Any
Sending email to: ujawane505@gmail.com
Finished: FAILURE

```

1 of 77

todo-application - Build # 36 - Failure! Inbox x

U address not configured yet <ujawane505@gmail.com> 9:51 AM (0 minutes ago) ☆ 😊 ↶ ⋮
to me ▾

🔥 Your Jenkins pipeline has fail!
Please check your application
<http://3.6.92.23:8000/todos/>

↶ Reply ↷ Forward 😊



success email

Success ?

Send To

Recipient List ?

Add ▾

Advanced ^

Recipient List ?

Reply-To List ?

\$PROJECT_DEFAULT_REPLYTO

Subject ?

\$PROJECT_DEFAULT_SUBJECT

Content ?

🔥 Your Jenkins pipeline has Success!
Please check your application
<http://3.6.92.23:8000/todos/>

Attachments ?

Can use wildcards like 'module/dist/**/*.zip'. See the [@includes of Ant fileset](#) for the exact format. The base directory is [the workspace](#).

When my pipeline succeeds, I receive a success email

```

> todo-application > #35 > Console Output
Step 4// : COPY . .
--> Using cache
--> 6b7760f6f531
Step 5/7 : RUN python manage.py migrate
--> Using cache
--> 55f663c9307f
Step 6/7 : EXPOSE 8000
--> Using cache
--> f554f6c402fa
Step 7/7 : CMD ["python", "manage.py", "runserver", "0.0.0.0:8000"]
--> Using cache
--> d07ef3ed1f81
Successfully built d07ef3ed1f81
Successfully tagged todo-app:latest
+ docker run -d -p 8000:8000 todo-app
1ba57db91dd83e19788d16454184da2a1100ef8ff2a2ed0aa37b4458b6353732
Email was triggered for: Success
Sending email for trigger: Success
Sending email to: ujawane505@gmail.com
Finished: SUCCESS

```



