

Task Description:

- Create a simple script file and push it to repo. Create a project in Jenkins connected to your GitHub repository. When a commit is made to your repo, automatically build must get triggered from Jenkins and the output must be shared to me via email.

Tech Stack Used:

- GCP
- GitHub
- Jenkins
- G-mail

Steps :

Jenkins Server Creation:

- Go to the GCP platform (or AWS platform) and create a server with below specifications.

2vCPU with 8 GB Memory

20 GB Storage

Allow http and https in security

Allow port 8080 (for jenkins)

Linux OS

Details	Observability	OS Info	Screenshot
Basic information			
Name		jenkins	
Instance Id		5794870780496431583	
Description		None	
Type		Instance	
Status		Running	
Creation time		Jul 6, 2025, 6:46:33 PM UTC+05:30	
Location		us-central1-c	
Instance template		None	
In use by		None	
Physical host		None	
Maintenance status		—	
Reservations		Automatically choose	
Labels		None	
Tags		— 	
Deletion protection		Disabled	
Confidential VM service		Disabled	
Preserved state size		0 GB	

Machine configuration

Machine type	e2-standard-2 (2 vCPUs, 8 GB Memory)
CPU platform	Intel Broadwell
Minimum CPU platform	None
Architecture	x86_64
vCPUs to core ratio	—
Custom visible cores	—
All-core turbo-only mode	—

VM instances

Filter Enter property name or value

<input type="checkbox"/>	Status	Name	Zone	Recommendations	In use by	Internal IP	External IP	Connect
<input type="checkbox"/>		jenkins	us-central1-c			10.128.0.5 (nic0)	35.226.209.37 ↗ (nic0)	SSH

Jenkins Task

Jenkins Installation:

- Log into the sever and run the below bash script to install the Jenkins in automate way.

```
#!/bin/bash
# USE UBUNTU20.04 - INSTANCE: 2GB RAM + 2VCPU MIN - WILL ONLY WORK
sudo apt update -y
sudo apt install openjdk-17-jdk -y
sudo apt update -y
sudo apt install openjdk-8-jdk -y
sudo apt install maven -y
curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo
tee \
/usr/share/keyrings/jenkins-keyring.asc > /dev/null
echo deb [signed-by=/usr/share/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 -y
sudo apt-get install jenkins -y
###
```

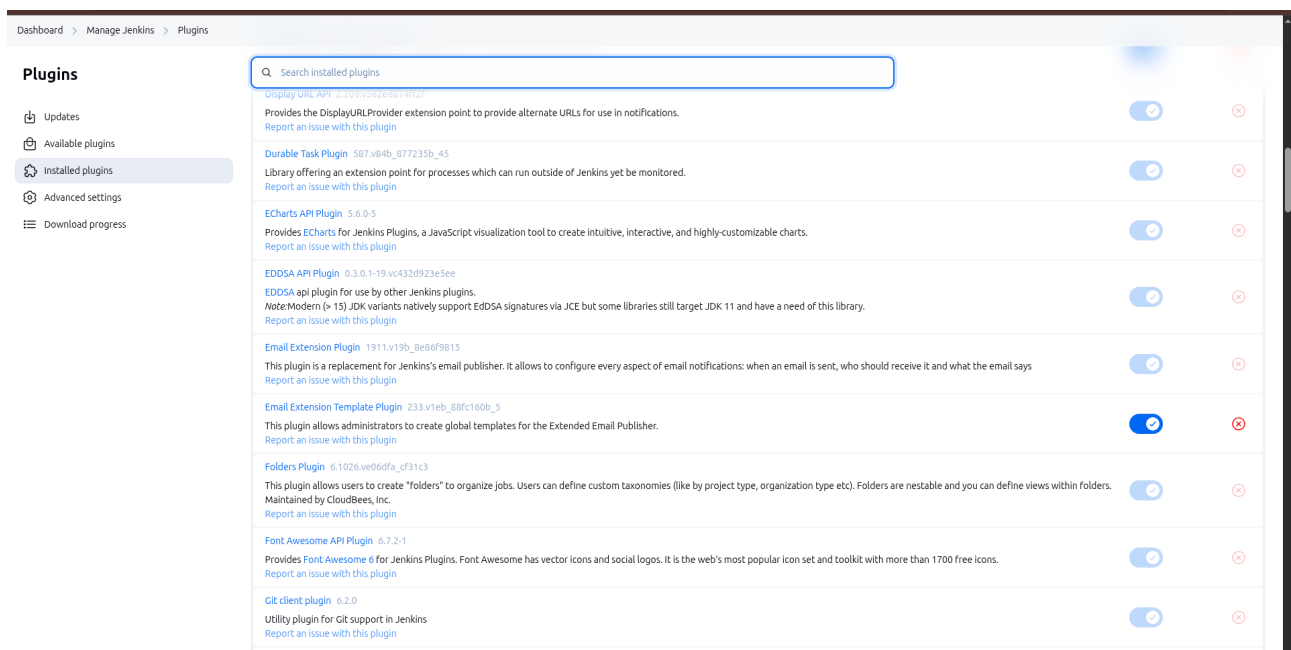
-Post installation try to connect the jenkins using the http://<IP of the server>:8080/

- Then try to set up the initial password and ID.

- Then follow the wizard to proceed set up the Jenkins.

- Once installation completed try to install the required plug-ins for Jenkins:

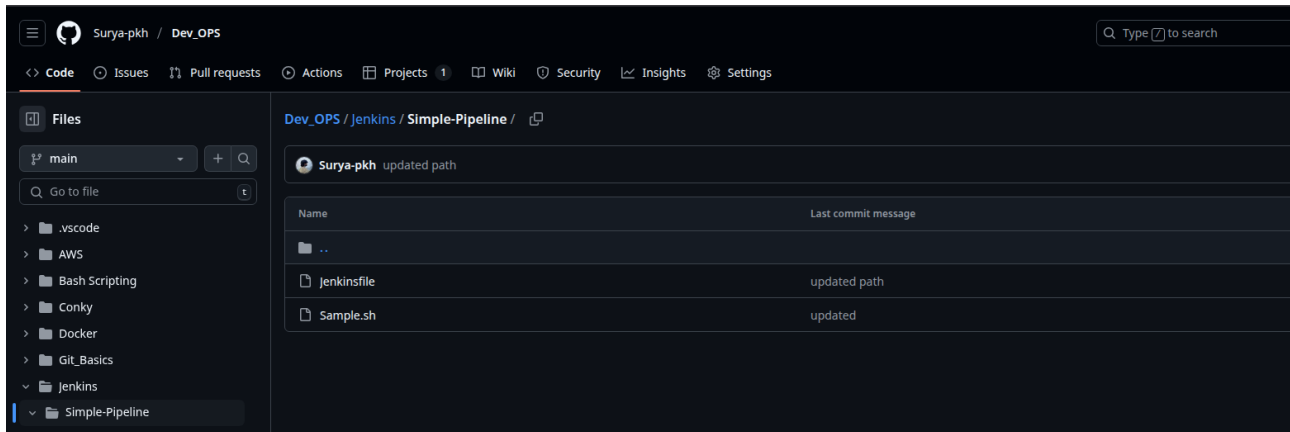
1. Build Timestamp
2. Pipeline Utility Steps
3. Email Notification template



Jenkins Task

GitHub Repo Setup:

- In Github create a repo.
- Add a SH file and a Jnekin file.



Sample.sh:

```
#!/bin/bash
echo "Hello from Jenkins Pipeline!" > output.txt
echo "Another Jenkins Build Test2"
```

Jenkinfile:

```
pipeline {
  agent any
  environment {
    RECIPIENT = 'gcpsurya27@gmail.com'
  }
  stages {
    stage('Checkout') {
      steps {
        checkout scm
      }
    }
    stage('Run Script') {
      steps {
        sh 'bash Jenkins/Simple-Pipeline/Sample.sh'
      }
    }
  }
  post {
    always {
      mail(
        to: 'gcpsurya27@gmail.com',
```

Jenkins Task

```
subject: "Pipeline Notification",  
body: "Pipeline completed with status: ${currentBuild.currentResult}"  
}  
}  
}
```

Creating Pipeline in Jenkins :

- Open jenkins and create **new item**.
- Give name and choose **Pipeline** option and click **ok** .







Jenkins

Dashboard > All > New Item

New Item

Enter an item name

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.
-  **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.
-  **Organization Folder**
Creates a set of multibranch project subfolders by scanning for repositories.

If you want to create a new item from other existing, you can use this option:

OK

- in the Configuration choose the **Trigger** – Github hook

Jenkins Task

Triggers

Set up automated actions that start your build based on specific events, like code changes or scheduled I

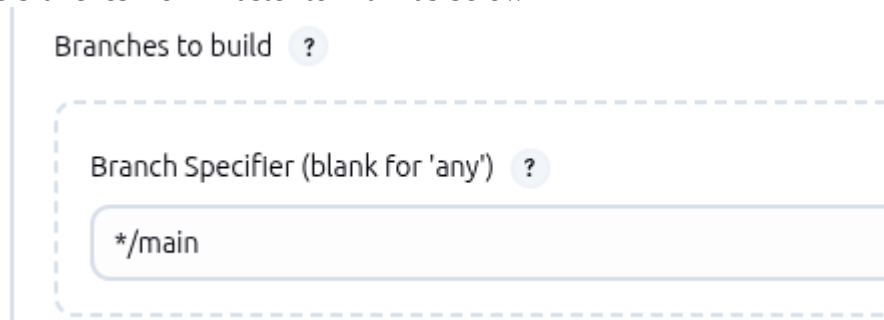
- ☐ Build after other projects are built ?
- ☐ Build periodically ?
- ☒ GitHub hook trigger for GITScm polling ?
- ☐ Poll SCM ?
- ☐ Trigger builds remotely (e.g., from scripts) ?

- In the pipeline groovy content choose the SCM as **Git** and add the **Repo URL** , also add the Git Credentials.



The screenshot shows the Jenkins Pipeline Configuration page. The 'Definition' section is set to 'Pipeline script from SCM'. The 'SCM' dropdown is set to 'Git'. Under 'Repositories', a new repository is added with the URL 'https://github.com/Surya-pkh/Dev_OPS.git' and the credential 'suryaprakash27032001@gmail.com/***** (Gitcred)'. The 'Advanced' options are expanded.

- Change the branches from Master to Main as below



The screenshot shows the 'Branches to build' configuration section. The 'Branch Specifier (blank for 'any')' field is set to '*/main'.

- Then at last copy the **Jenkinsfile** (groovy file) path and paste it the script path

Jenkins Task

Script Path ?

Jenkins/Simple-Pipeline/Jenkinsfile

☒ Lightweight checkout ?

[Pipeline Syntax](#)

Advanced

Advanced ▾

Save Apply

- Then **Save** the configuration.

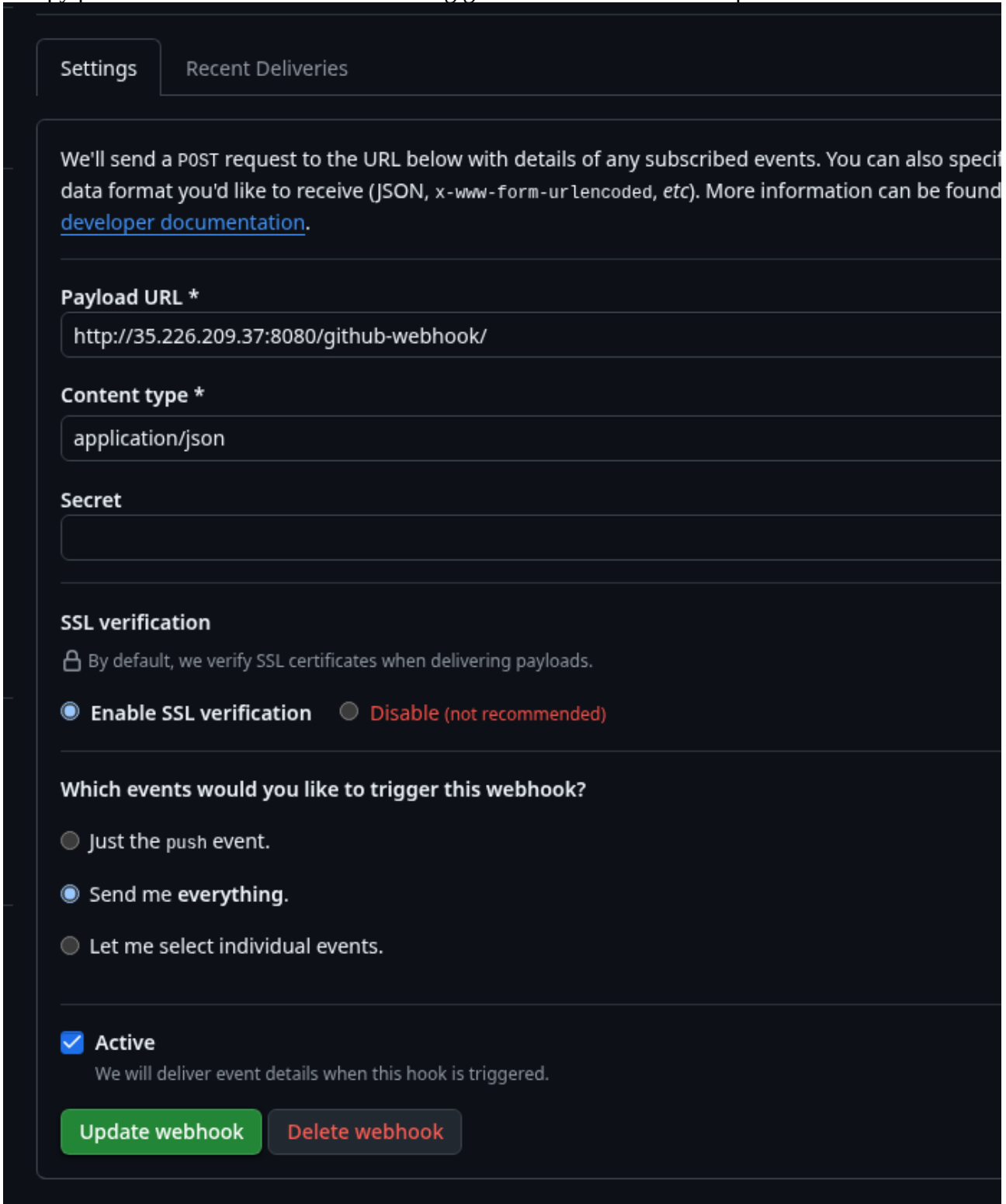
GitHub WebHook Configuration:

- go to the Github repo which has been created for this jenkins and move to the settings tab.

The screenshot shows the GitHub repository settings page for 'Surya-pkh / Dev_OPS'. The 'Settings' tab is selected in the top navigation bar. On the left, a sidebar lists various settings categories: General, Access, Collaborators, Moderation options, Code and automation, Branches, Tags, Rules, Actions, Models, Webhooks, Copilot, Environments, Codespaces, and Pages. The 'General' tab is currently active. The main content area displays the 'General' settings, including the 'Repository name' (Dev_OPS) with a 'Rename' button, and checkboxes for 'Template repository' and 'Require contributors to sign off on web-based commits'. The 'Default branch' is set to 'main'. A 'Social preview' section is partially visible at the bottom.

Jenkins Task

- Click on the **Webhooks**.
- Copy paste the **Jenkins** url with mentioning github webhook like in the picture



Settings Recent Deliveries

We'll send a POST request to the URL below with details of any subscribed events. You can also specify the data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in the [developer documentation](#).

Payload URL *

http://35.226.209.37:8080/github-webhook/

Content type *

application/json

Secret

SSL verification

By default, we verify SSL certificates when delivering payloads.

☒ Enable SSL verification ☐ Disable (not recommended)

Which events would you like to trigger this webhook?

☐ Just the push event.

☒ Send me everything.

☐ Let me select individual events.

☒ **Active**

We will deliver event details when this hook is triggered.

Update webhook Delete webhook

- Change the **Content Type** to json. And save it.

Email Notification Configuration on Jenkins:

- Open the Jenkins and go to the **Manage Jenkins**.

Dashboard > Manage Jenkins

+ New Item

Build History

Project Relationship

Check File Fingerprint

Manage Jenkins

My Views

Build Queue

Build Executor Status 0/2

Manage Jenkins

Building on the built-in node can be a security issue. You should set up distributed builds. See [the docu](#)

Java 17 end of life in Jenkins

You are running Jenkins on Java 17, support for which will end on or after Mar 31, 2026. Refer to [the d](#)

System Configuration

System

Configure global settings and paths.

Tools

Configure tools, their locations automatic installers.

Clouds

Add, remove, and configure cloud instances to provision agents on-demand.

Appearance

Configure the look and feel of J

Security

- Open the **System**
- go the **Email Notification** Setting

E-mail Notification

SMTP server

smtp.gmail.com

Default user e-mail suffix ?

Advanced Edited

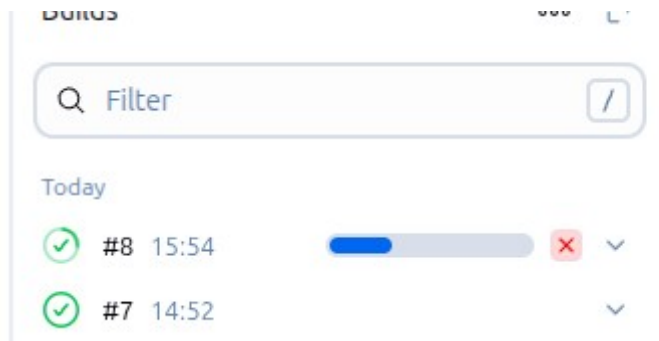
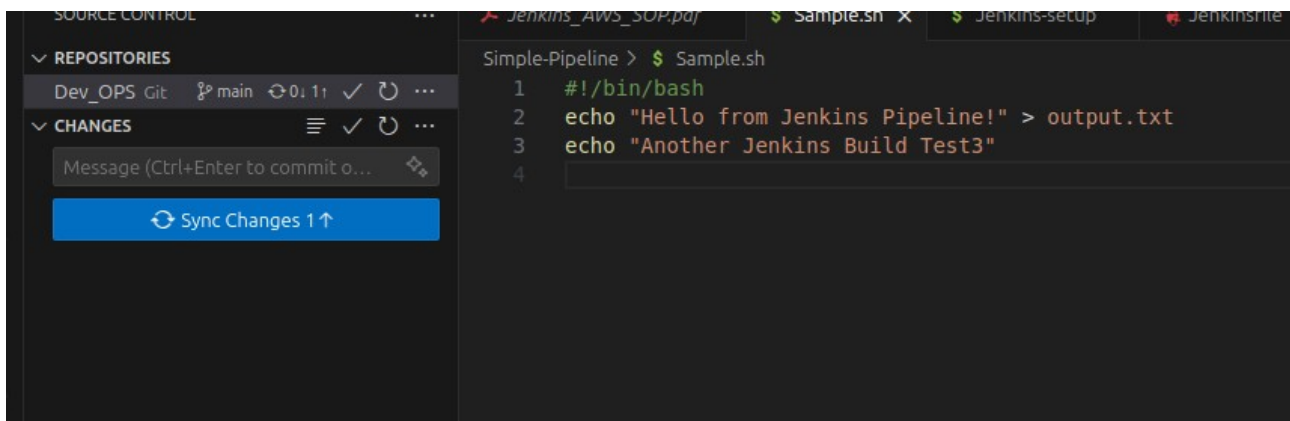
SMTP server: smtp.gmail.com

- Use SMTP Authentication: ✓

Jenkins Task

- Username: your Gmail address
- Password: **App password**
- SMTP Port: 587
- Use TLS: ✓
- Default Recipients: gcpsurya27@gmail.com
- Test configuration and **Save**

Now Trigger the by Committing changes on the Git Repo:



Stages

6 July 2025

