Jenkins Task-2

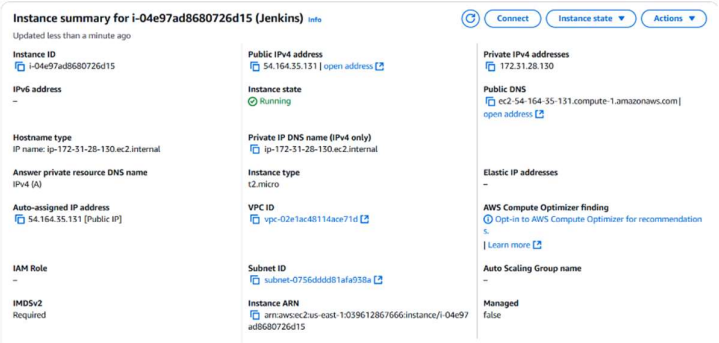
TASKS

Work Flow:

- Create an EC2 instance with the help of AWS Management Console with linux OS of required configuration and ensure that where the instance type should be “t2.medium” and also configure an security group and edit an inbound traffic to “all traffic”.
- Now, Connect an EC2 instance with an help of Windows Terminal or Gitbash or Vbox.
- To connect an EC2 instance the command is:
    - ssh -i “**key_file**” ec2-user@“**Public_IP_address**”

Key_file --- Key file of the instance with the extension .pem

Public_IP_address --- Public IP address of the instance.



1. **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.**

<u>Step 1: Install Java and Jenkins in EC2-instance</u>

Install An Java Package :

- ✓ Before installing an Jenkins, we have to install an java package because where Jenkins are developed using java programming language, so we need an java package to run an jenkins. In this case java package installation is mandatory.

- ✓ To list the java package from an repository, The command is:
    - **yum list | grep java**

- ✓ Once you got an right package, now to install that java package, the command is:
  - **sudo yum install java-21-amazon-corretto.x86_64**

```
[ec2-user@ip-172-31-28-130 ~]$ sudo yum install java-21-amazon-corretto.x86_64
Amazon Linux 2023 Kernel Livepatch repository                                                    119 kB/s |  16 kB     00:00
Dependencies resolved.
================================================================================================================================
 Package                                Architecture            Version                         Repository                  Size
================================================================================================================================
Installing:
 java-21-amazon-corretto                x86_64                  1:21.0.7+6-1.amzn2023.1         amazonlinux                 214 k
Installing dependencies:
 alsa-lib                               x86_64                  1.2.7.2-1.amzn2023.0.2          amazonlinux                 504 k
 cairo                                  x86_64                  1.18.0-4.amzn2023.0.1           amazonlinux                 718 k
 dejavu-sans-fonts                      noarch                  2.37-16.amzn2023.0.2            amazonlinux                 1.3 M
 dejavu-sans-mono-fonts                 noarch                  2.37-16.amzn2023.0.2            amazonlinux                 467 k
 dejavu-serif-fonts                     noarch                  2.37-16.amzn2023.0.2            amazonlinux                 1.0 M
 fontconfig                             x86_64                  2.13.94-2.amzn2023.0.2          amazonlinux                 273 k
 fonts-filesystem                       noarch                  1:2.0.5-12.amzn2023.0.2         amazonlinux                 9.5 k
 freetype                               x86_64                  2.13.2-5.amzn2023.0.1           amazonlinux                 423 k
 giflib                                 x86_64                  5.2.1-9.amzn2023.0.1            amazonlinux                  49 k
 google-noto-fonts-common               noarch                  20201206-2.amzn2023.0.2         amazonlinux                  15 k
 google-noto-sans-vf-fonts              noarch                  20201206-2.amzn2023.0.2         amazonlinux                 492 k
 graphite2                              x86_64                  1.3.14-7.amzn2023.0.2           amazonlinux                  97 k
 harfbuzz                               x86_64                  7.0.0-2.amzn2023.0.2            amazonlinux                 873 k
 java-21-amazon-corretto-headless       x86_64                  1:21.0.7+6-1.amzn2023.1         amazonlinux                  96 M
 javapackages-filesystem                noarch                  6.0.0-7.amzn2023.0.6            amazonlinux                  12 k
 langpacks-core-font-en                 noarch                  3.0-21.amzn2023.0.4             amazonlinux                  10 k
 libICE                                 x86_64                  1.1.1-3.amzn2023.0.1            amazonlinux                  76 k
 libSM                                  x86_64                  1.2.4-3.amzn2023.0.1            amazonlinux                  45 k
 libX11                                 x86_64                  1.8.10-2.amzn2023.0.1           amazonlinux                 659 k
 libX11-common                          noarch                  1.8.10-2.amzn2023.0.1           amazonlinux                 147 k
 libXau                                 x86_64                  1.0.11-6.amzn2023.0.1           amazonlinux                  33 k
 libXext                                x86_64                  1.3.6-1.amzn2023.0.1            amazonlinux                  42 k
 libXi                                  x86_64                  1.8.2-1.amzn2023.0.1            amazonlinux                  42 k
 libXinerama                            x86_64                  1.1.5-6.amzn2023.0.1            amazonlinux                  16 k
 libXrandr                              x86_64                  1.5.4-3.amzn2023.0.1            amazonlinux                  29 k
 libXrender                             x86_64                  0.9.11-6.amzn2023.0.1           amazonlinux                  29 k
 libXt                                  x86_64                  1.3.0-3.amzn2023.0.1            amazonlinux                 183 k
 libXtst                                x86_64                  1.2.5-1.amzn2023.0.1            amazonlinux                  22 k
 libbrotli                              x86_64                  1.0.9-4.amzn2023.0.2            amazonlinux                 315 k
 libjpeg-turbo                          x86_64                  2.1.4-2.amzn2023.0.5            amazonlinux                 190 k
 libpng                                 x86_64                  2:1.6.37-10.amzn2023.0.6        amazonlinux                 128 k
 libxcb                                 x86_64                  1.17.0-1.amzn2023.0.1           amazonlinux                 235 k
```

Install An Jenkins :

- ✓ To install an Jenkins in linux machine go to an official website by using below link.
- ✓ Link : https://www.jenkins.io/doc/book/installing/linux/#red-hat-centos
- ✓ Now you can see the instructions given in the official page to install an Jenkins, follow all the steps to install.
- ✓ And also where the command as given below to install an Jenkins from an official Page, run all the commands in your linux machine one by one.
  - sudo wget -O /etc/yum.repos.d/jenkins.repo \ https://pkg.jenkins.io/redhat-stable/jenkins.repo

```
[ec2-user@ip-172-31-28-130 ~]$ sudo wget -O /etc/yum.repos.d/jenkins.repo \
    https://pkg.jenkins.io/redhat-stable/jenkins.repo
--2025-06-02 16:28:10--  https://pkg.jenkins.io/redhat-stable/jenkins.repo
Resolving pkg.jenkins.io (pkg.jenkins.io)... 146.75.34.133, 2a04:4e42:78::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)|146.75.34.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 85
Saving to: '/etc/yum.repos.d/jenkins.repo'

/etc/yum.repos.d/jenkins.repo         100%[===================================================================>]      85  --.-KB/s    in 0s

2025-06-02 16:28:10 (2.52 MB/s) - '/etc/yum.repos.d/jenkins.repo' saved [85/85]
```

  - sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
  - sudo yum upgrade

```
[ec2-user@ip-172-31-28-130 ~]$ sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
[ec2-user@ip-172-31-28-130 ~]$ sudo yum upgrade
Jenkins-stable                                                                                   529 kB/s |  31 kB     00:00
Dependencies resolved.
Nothing to do.
Complete!
```

  - sudo yum install fontconfig

```
[ec2-user@ip-172-31-28-130 ~]$ sudo yum install fontconfig
Last metadata expiration check: 0:00:35 ago on Mon Jun  2 16:28:46 2025.
Package fontconfig-2.13.94-2.amzn2023.0.2.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
```

  - sudo yum install Jenkins

```
[ec2-user@ip-172-31-28-130 ~]$ sudo yum install jenkins
Last metadata expiration check: 0:00:52 ago on Mon Jun  2 16:28:46 2025.
Dependencies resolved.
================================================================================================================================================
 Package                         Architecture                    Version                                 Repository                          Size
================================================================================================================================================
Installing:
 jenkins                         noarch                          2.504.2-1.1                             jenkins                             90 M

Transaction Summary
================================================================================================================================================
Install  1 Package
```

- sudo systemctl daemon-reload

✓ To start and enable an jenkins service, The command is:

- **sudo systemctl start jenkins**
- **sudo systemctl enable jenkins**

✓ To check the status of the jenkins service, The command is:

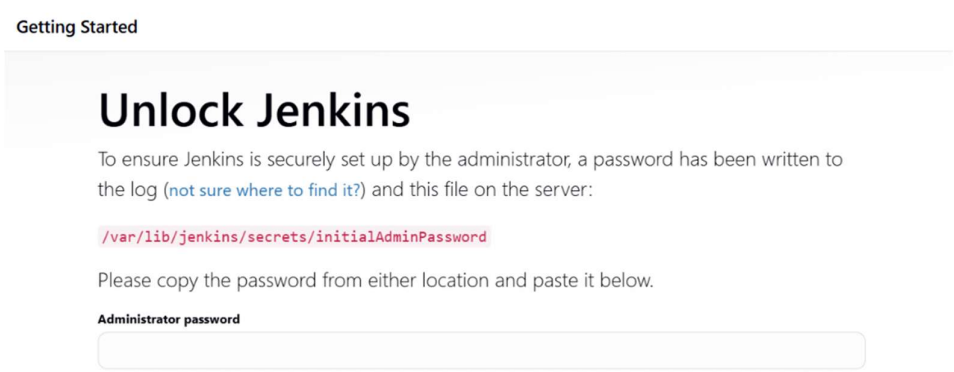
- **sudo systemctl status Jenkins**

```
[ec2-user@ip-172-31-28-130 ~]$ sudo systemctl enable jenkins
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /usr/lib/systemd/system/jenkins.service.
[ec2-user@ip-172-31-28-130 ~]$ sudo systemctl start jenkins
[ec2-user@ip-172-31-28-130 ~]$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
     Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; preset: disabled)
     Active: active (running) since Mon 2025-06-02 16:30:53 UTC; 8s ago
   Main PID: 26499 (java)
      Tasks: 45 (limit: 1111)
     Memory: 389.1M
        CPU: 15.887s
     CGroup: /system.slice/jenkins.service
             └─26499 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080
```

Step 2: Access Jenkins through an web browser.

✓ To accessing an Jenkins through an web browser copy / paste the public IP along with the localhost(:8080), The format is given below:

- http://54.226.235.178:8080



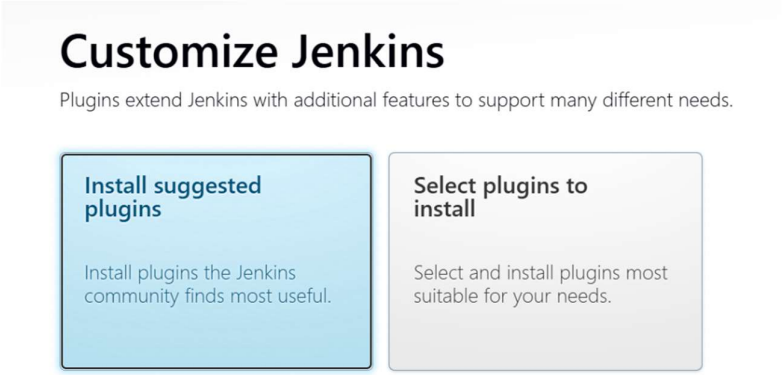
✓ To get an initial admin password run an below command it will give you an password for an initial login, The command is:

- **sudo cat /var/lib/jenkins/secrets/initialAdminPassword**

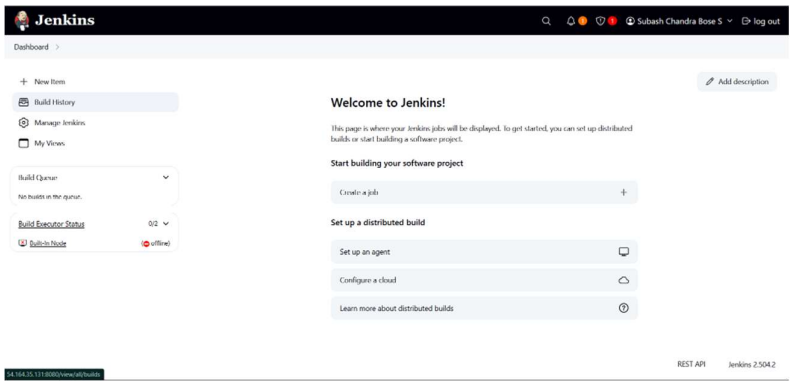
```
[ec2-user@ip-172-31-28-130 ~]$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
546b30aa1de04c7ab7d27763e4578152
[ec2-user@ip-172-31-28-130 ~]$
```

✓ Now copy and paste the password into the Jenkins UI to unlock.

- ✓ Now it will ask you to select an customize plugins, you can choose “Install suggested plugins”.



- ✓ And then it will move you to create an your first admin user account, Once created it will move you to the Jenkins dashboard.



Step 3: Create a Simple Script and Push to GitHub.

- ✓ To create an new scripting file and push to github, so first create an directory and move to the directory, the command is:
  - **mkdir jenkins-github-demo**
  - **cd jenkins-github-demo**
- ✓ Now create an bash scripting file and add an permission to it, The command is:
  - **touch build.sh**
  - **vi build.sh**

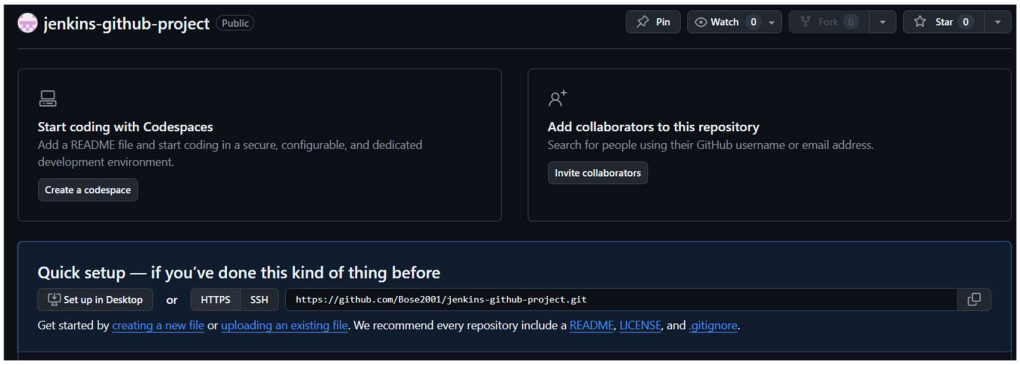
```
#!/bin/bash

echo 'Hello from jenkins build!'

~
```

- ✓ To add an permission for this bash scripting file, the command is:
  - **chmod +x build.sh**

```
[ec2-user@ip-172-31-7-145 ~]$ mkdir jenkins-github-demo
[ec2-user@ip-172-31-7-145 ~]$ cd jenkins-github-demo
[ec2-user@ip-172-31-7-145 jenkins-github-demo]$ touch build.sh
[ec2-user@ip-172-31-7-145 jenkins-github-demo]$ vi build.sh
[ec2-user@ip-172-31-7-145 jenkins-github-demo]$ ls
build.sh
[ec2-user@ip-172-31-7-145 jenkins-github-demo]$
```

Step 4: Initialize Git and Push to GitHub

- ✓ Before pushing an bash script in remote git repo we have to create one new repo to push the bash script to it.



- ✓ To initialize an git and add an bash script file to the git, the command is:
  - **git init**
  - **git add build.sh**

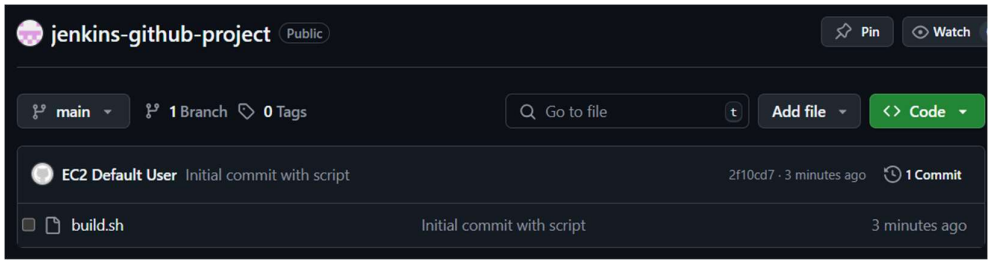
```
[ec2-user@ip-172-31-7-145 jenkins-github-demo]$ git add build.sh
[ec2-user@ip-172-31-7-145 jenkins-github-demo]$ git commit -m "Initial commit with script"
[master (root-commit) 2f10cd7] Initial commit with script
 Committer: EC2 Default User <ec2-user@ip-172-31-7-145.ap-south-1.compute.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 1 file changed, 5 insertions(+)
 create mode 100755 build.sh
[ec2-user@ip-172-31-7-145 jenkins-github-demo]$
```

- ✓ And commit the file to the git and give an branch for an file to store in an git, the command is:
  - **git commit -m "Initial commit with script"**
  - **git branch -M main**

```
[ec2-user@ip-172-31-7-145 jenkins-github-demo]$ git add build.sh
[ec2-user@ip-172-31-7-145 jenkins-github-demo]$ git commit -m "Initial commit with script"
[master (root-commit) 2f10cd7] Initial commit with script
 Committer: EC2 Default User <ec2-user@ip-172-31-7-145.ap-south-1.compute.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 1 file changed, 5 insertions(+)
 create mode 100755 build.sh
[ec2-user@ip-172-31-7-145 jenkins-github-demo]$ |
```

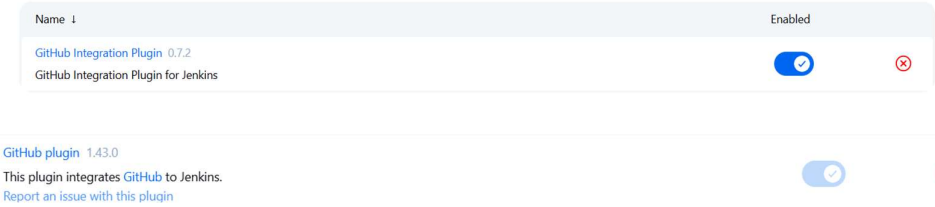
- ✓ Finally, to push an entire bash scripting file from local to remote git , the command is:
  - **git remote add origin https://github.com/Bose2001/jenkins-github-project.git**
  - **git push -u origin main**

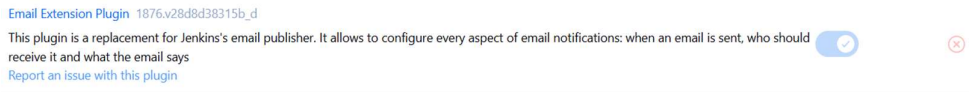
```
[ec2-user@ip-172-31-7-145 jenkins-github-demo]$ git remote add origin https://github.com/Bose2001/jenkins-github-project.git
[ec2-user@ip-172-31-7-145 jenkins-github-demo]$ git push -u origin main
Username for 'https://github.com': Bose2001
Password for 'https://Bose2001@github.com':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 301 bytes | 301.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Bose2001/jenkins-github-project.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
[ec2-user@ip-172-31-7-145 jenkins-github-demo]$ |
```



Step 5: Configure Jenkins for GitHub Project

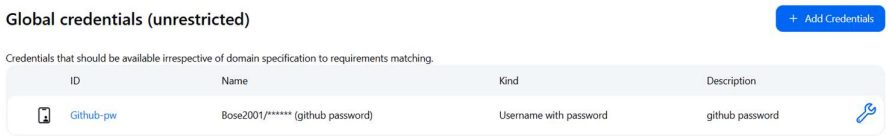
- ✓ To perform an task requirements, we have to install an required plugins to perform an required task, the plugins are:
  - **GitHub Integration**
  - **GitHub plugin**
  - **Email Extension Plugin**
  - **Pipeline** or **Freestyle** project type (if not present)





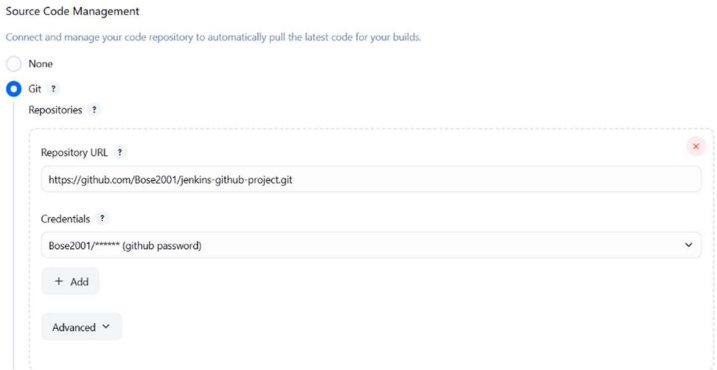
Step 5: Configure Jenkins Credentials

- ✓ To configure an Jenkins credentials and add an github credentials, the only we can able to access an github account.
- ✓ To add an credentials in Jenkins, follow the below path to configure.
  - **Manage Jenkins → Credentials → System → Global → Add Credentials**
- ✓ Now select an option called username with password and follow the below step.
  - **Username** = GitHub username
  - **Password** = GitHub personal access token (PAT)



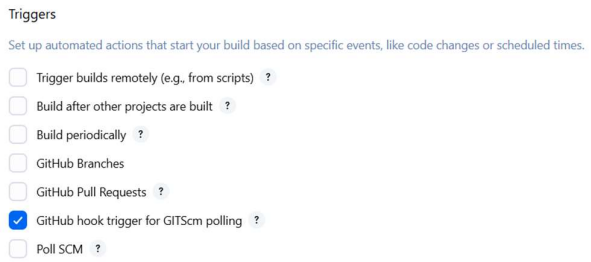
Step 6: Create Jenkins Job and Connect to GitHub

- ✓ **Create Freestyle Project**
  - Go to Jenkins Dashboard → New Item → Freestyle project
  - Name: github-auto-build
  - Click OK
- ✓ **Configure GitHub Source**
  - Under Source Code Management, select Git
  - Enter your repo URL: https://github.com/Bose2001/jenkins-github-project.git
  - Select the credentials you created



✓ **Add Build Trigger**

- Enable GitHub hook trigger for GITScm polling



Step 7: Set Up GitHub Webhook

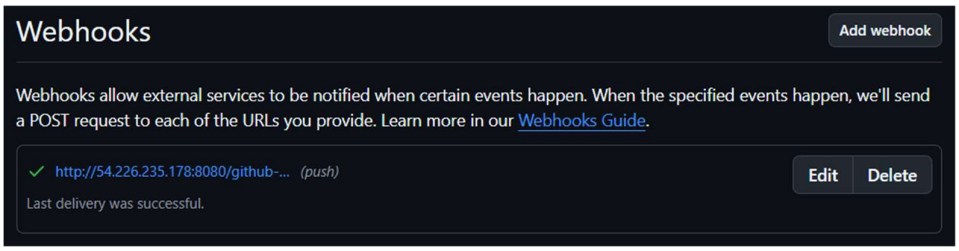
✓ Here webhook is used for an automation, which means, if we change anything in our bash scripting file, it will automatically trigger and build an job automatically.

✓ Now to enable an webhook in github go to settings and click add webhook

✓ Give an payload an URL as http://54.226.235.178:8080/github-webhook/

✓ Give an content type as application/json.

✓ Give an events as Just the push event and save it.



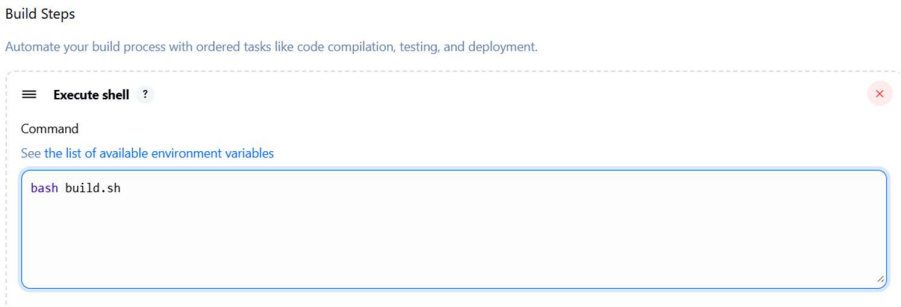
Step 8: Add Build Step

✓ This step tells Jenkins what to do after pulling the latest code from your GitHub repository.

✓ Where Jenkins can only able to pull an latest code from an github, so now we have to run that latest code.so here we are adding an build step.

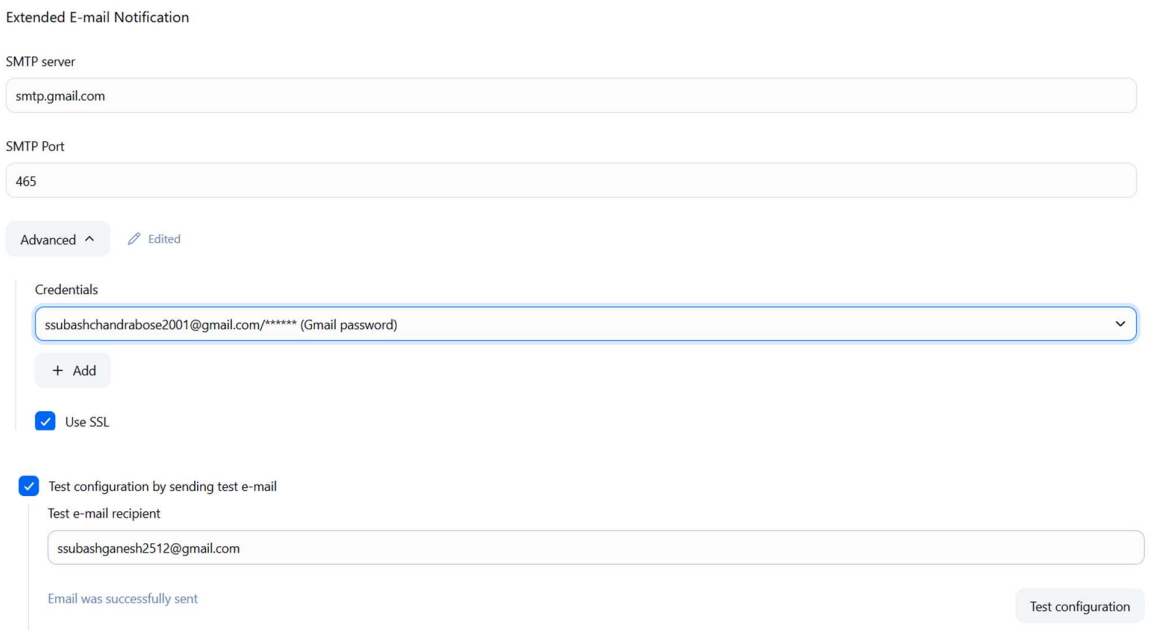
✓ Under Build, click Add build step → Execute shell.

- **bash build.sh**



Step 9: Configure Email Notification

- ✓ Now to enable an email notification from Jenkins, we have to configure SMTP in Jenkins
- ✓ **Configure SMTP in Jenkins**
  - Go to **Manage Jenkins → Configure System**:
  - SMTP server: smtp.gmail.com
  - Use SMTP Authentication: Yes
  - Username: your Gmail ID
  - Password: App Password (not your Gmail password)
  - Use SSL: Yes
  - SMTP Port: 465
  - Sender Email Address: your Gmail ID
- ✓ Click **Test configuration by sending test e-mail** to verify.



- ✓ **Add Email Notification to Job**

In your Jenkins job config:

- Post-build Actions →Add → Editable Email Notification
- Recipient list: your_email@example.com

Project Recipient List ?

Comma-separated list of email address that should receive notifications for this project.

ssubashganesh2512@gmail.com

Project Reply-To List ?

Comma-separated list of email address that should be in the Reply-To header for this project.

ssubashchandrabose2001@gmail.com

- Default Subject: Jenkins Build: $PROJECT_NAME - $BUILD_STATUS
- Default Content:

Greetings From Jenkins,

Project: $PROJECT_NAME
Build Status: $BUILD_STATUS
Console Output: $BUILD_URL/console

Content Type ?

Default Content Type

Default Subject ?

Jenkins Build: $PROJECT_NAME - $BUILD_STATUS

Default Content ?

Greetings From Jenkins,

Project: $PROJECT_NAME
Build Status: $BUILD_STATUS
Console Output: $BUILD_URL/console

✓ Now all add an trigger with an Always, it will help to trigger an email to your given email address.

Always ?

Send To

Developers ?

Recipient List ?

Add

Advanced

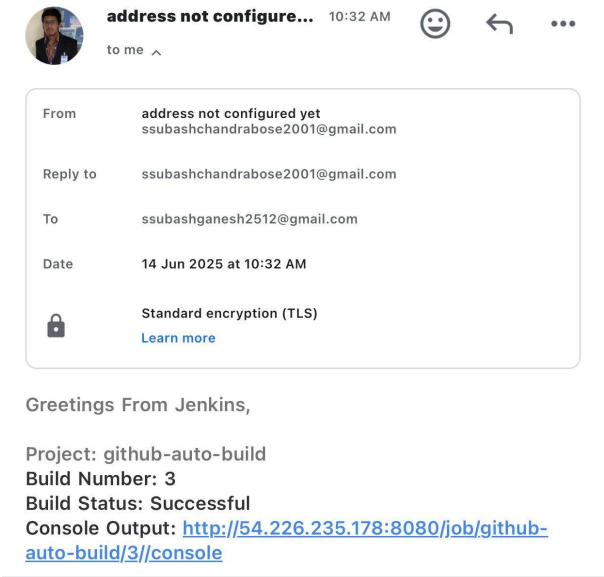
Step 10: Test the Setup

- ✓ Make a change in your GitHub repo (like edit build.sh).
- ✓ Now Commit and push.
- ✓ Where setup should:
  - Trigger Jenkins build automatically.
  - Send an email with build result.


```
[github-auto-build] $ /bin/sh -xe /tmp/jenkins10885803123468064696.sh
+ bash build.sh
Hello from jenkins build!
This is a test change for Jenkins auto build
This is a test change for Jenkins auto build 1
Email was triggered for: Always
Sending email for trigger: Always
Sending email to: ssubashganesh2512@gmail.com
Finished: SUCCESS
```

- ✓ Once the build gets success, where we will receive an email and also we can check with an console output.

✓ This is an email where received from Jenkins ,once the Jenkins is triggered to build an job for an latest bash script.



********** **TASK COMPLETED** **********