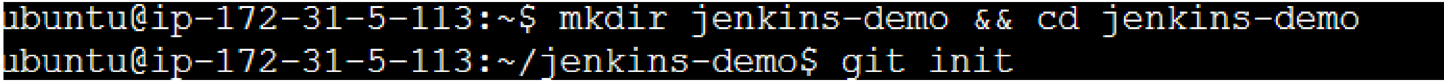
# Jenkins Task-2

## 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.

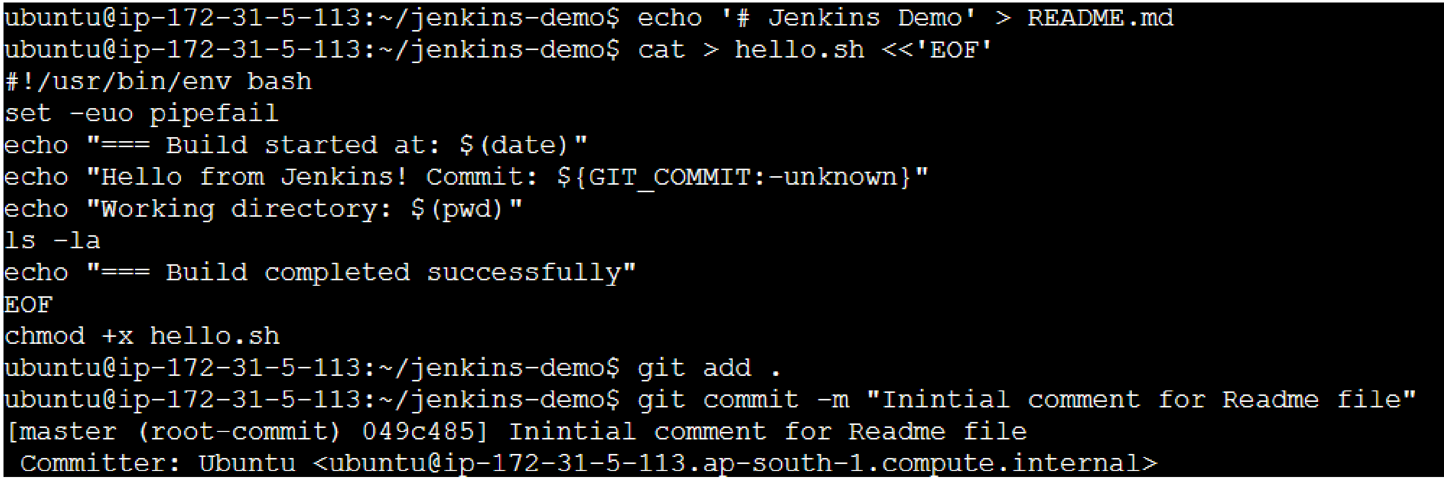
- ➔ Step 1: Pre-required set-up like:
  - -GitHub account
  - -EC2 with Jenkins installed
  - -Email

- ➔ Step 2: Make directory and initialize git.



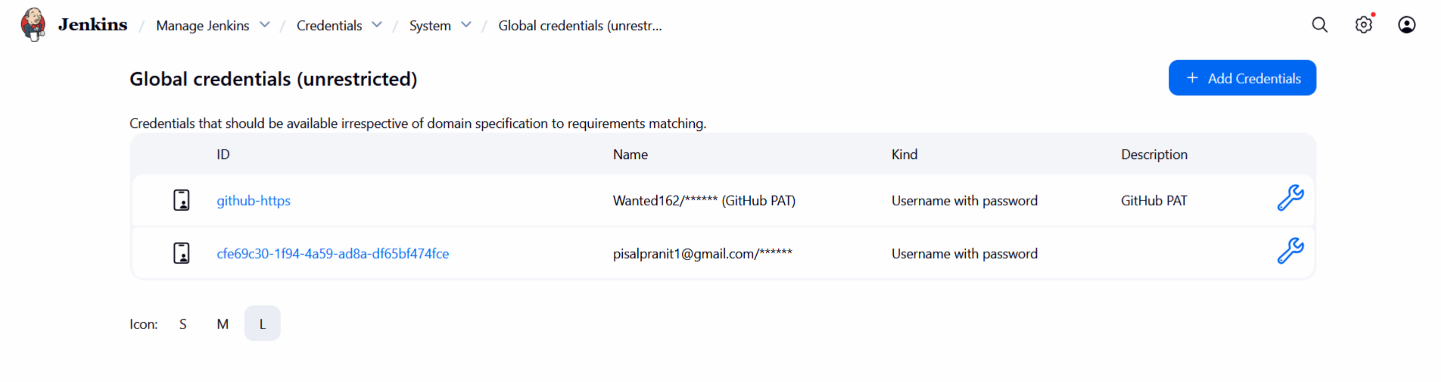
```
ubuntu@ip-172-31-5-113:~$ mkdir jenkins-demo && cd jenkins-demo
ubuntu@ip-172-31-5-113:~/jenkins-demo$ git init
```

- ➔ Step 3: Create a simple script and push it to GitHub



```
ubuntu@ip-172-31-5-113:~/jenkins-demo$ echo '# Jenkins Demo' > README.md
ubuntu@ip-172-31-5-113:~/jenkins-demo$ cat > hello.sh <<'EOF'
#!/usr/bin/env bash
set -euo pipefail
echo "=== Build started at: $(date)"
echo "Hello from Jenkins! Commit: ${GIT_COMMIT:-unknown}"
echo "Working directory: $(pwd)"
ls -la
echo "=== Build completed successfully"
EOF
chmod +x hello.sh
ubuntu@ip-172-31-5-113:~/jenkins-demo$ git add .
ubuntu@ip-172-31-5-113:~/jenkins-demo$ git commit -m "Inintial comment for Readme file"
[master (root-commit) 049c485] Inintial comment for Readme file
 Committer: Ubuntu <ubuntu@ip-172-31-5-113.ap-south-1.compute.internal>
```

- ➔ Step 4: In Jenkins make 2 Credentials. One for your GitHub and another for your Email.



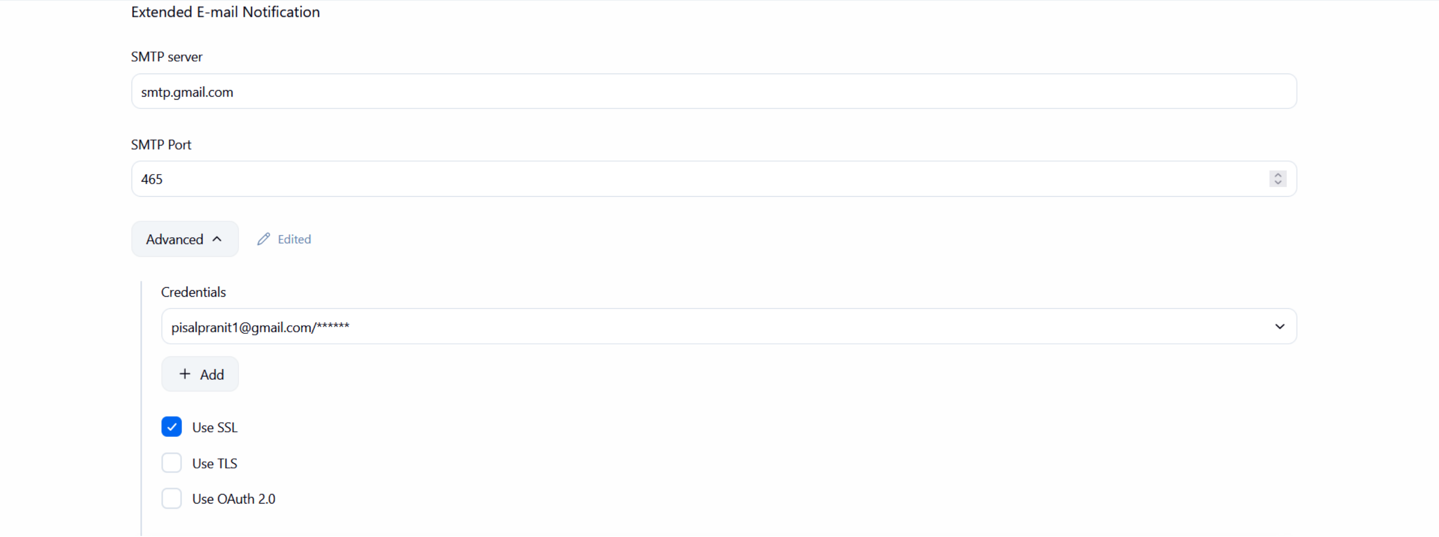
Jenkins / Manage Jenkins / Credentials / System / Global credentials (unrestr...

Global credentials (unrestricted)

Credentials that should be available irrespective of domain specification to requirements matching.

| | ID | Name | Kind | Description |
|---|---|---|---|---|
| | github-https | Wanted162/****** (GitHub PAT) | Username with password | GitHub PAT |
| | cfe69c30-1f94-4a59-ad8a-df65bf474fce | pisalpranit1@gmail.com/****** | Username with password | |

- Step 5: In Extended Email Notification fill the information as shown in picture to make sure that whenever the build happens it would notify you through the Mail



Extended E-mail Notification

SMTP server

smtp.gmail.com

SMTP Port

465

Credentials

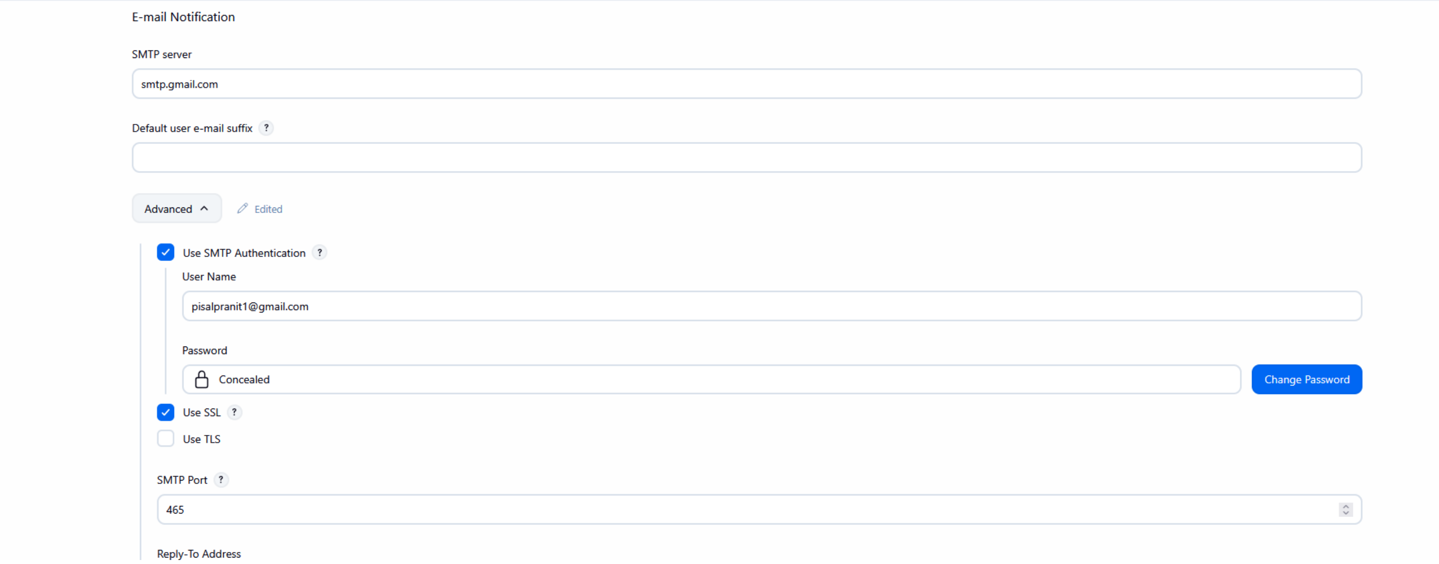
pisalpranit1@gmail.com/******

☑ Use SSL

□ Use TLS

□ Use OAuth 2.0

- Same goes for Email Notification



E-mail Notification

SMTP server

smtp.gmail.com

Default user e-mail suffix ?

☑ Use SMTP Authentication ?

User Name

pisalpranit1@gmail.com

Password

Concealed

☑ Use SSL ?

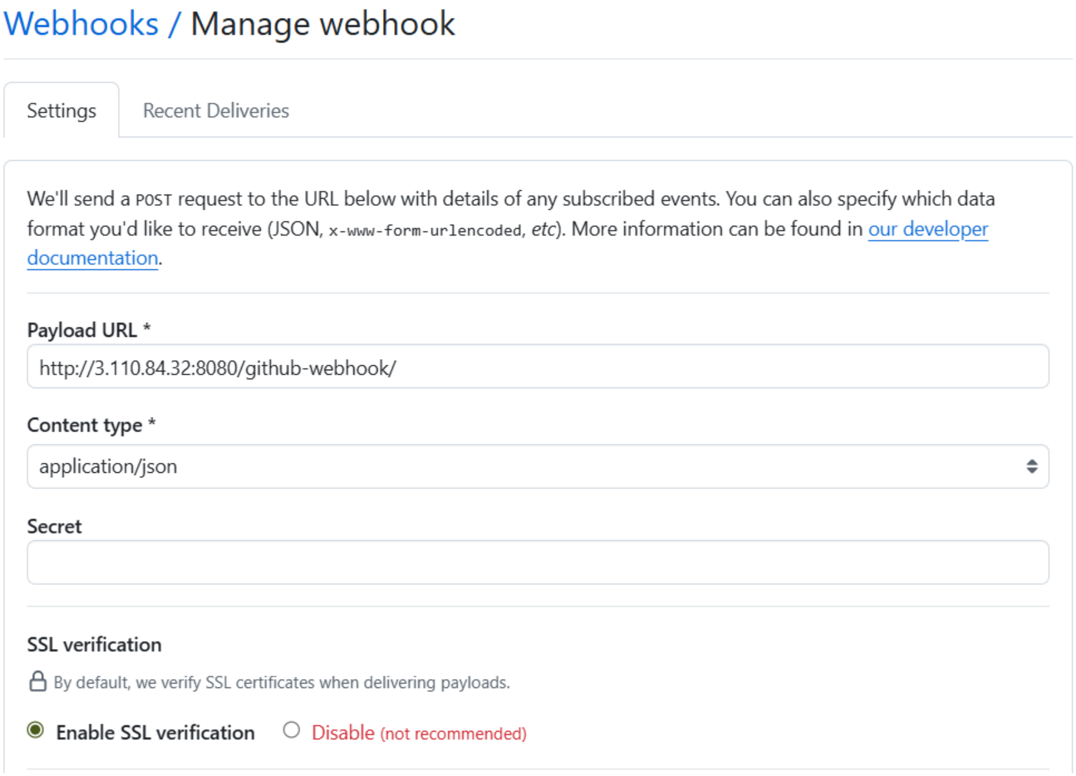
□ Use TLS

SMTP Port ?

465

Reply-To Address

- Step 6:  Turn on the web-hook of the GitHub to make sure the changes made in GitHub are triggered by Jenkins which will then notify it to mail



Webhooks / Manage webhook

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

Payload URL *

http://3.110.84.32:8080/github-webhook/

Content type *

application/json

Secret

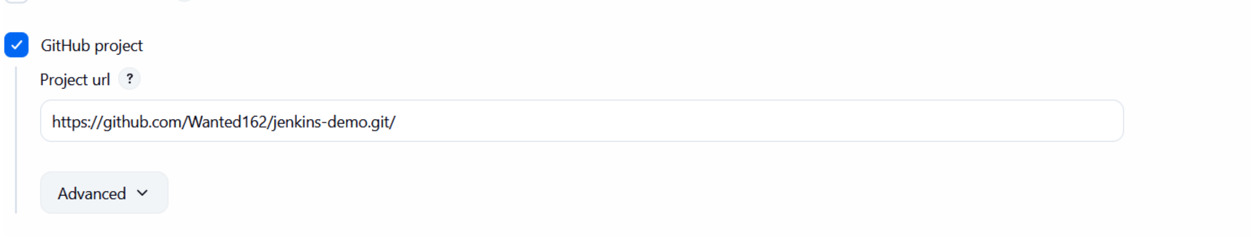
SSL verification

By default, we verify SSL certificates when delivering payloads.

● Enable SSL verification ○ Disable (not recommended)

➔ Step 7: Now in Jenkins create a freestyle project where we have to configure things like:

In General - > mention your Github repo link.

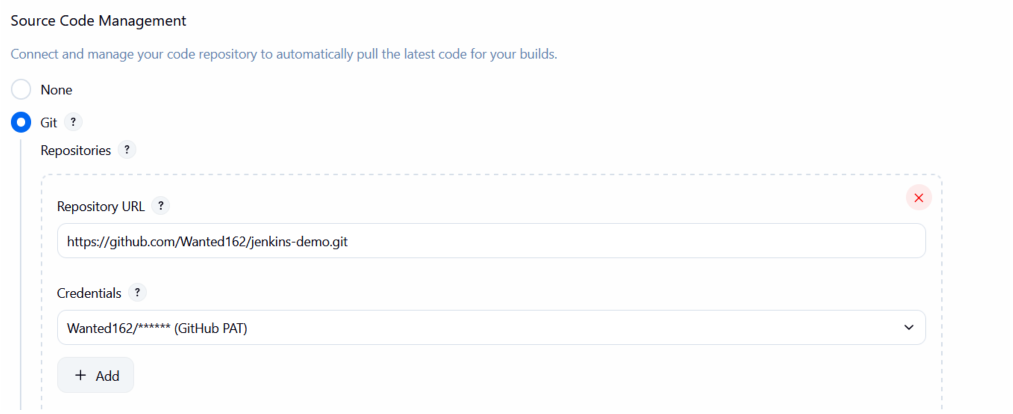


☑ GitHub project

Project url ?

https://github.com/Wanted162/jenkins-demo.git/

Under source code choose git and fill up as below



Source Code Management

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

○ None

● Git ?

Repositories ?

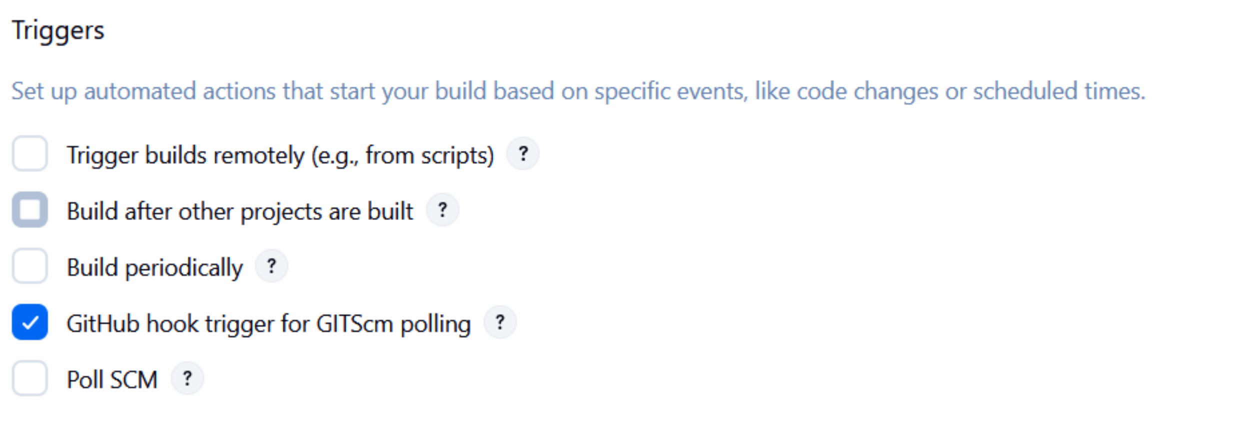
Repository URL ?

https://github.com/Wanted162/jenkins-demo.git

Credentials ?

Wanted162/****** (GitHub PAT)

Choose GitHub hook Trigger

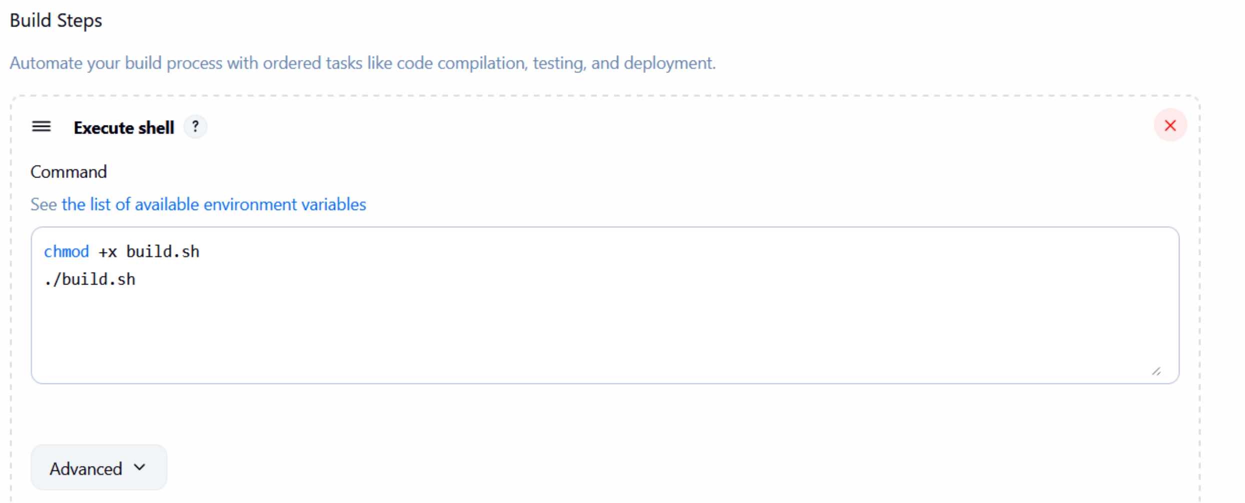


Triggers

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

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

Under Build steps choose Execute shell and paste that code so it runs the build.sh file from your git.



Build Steps

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

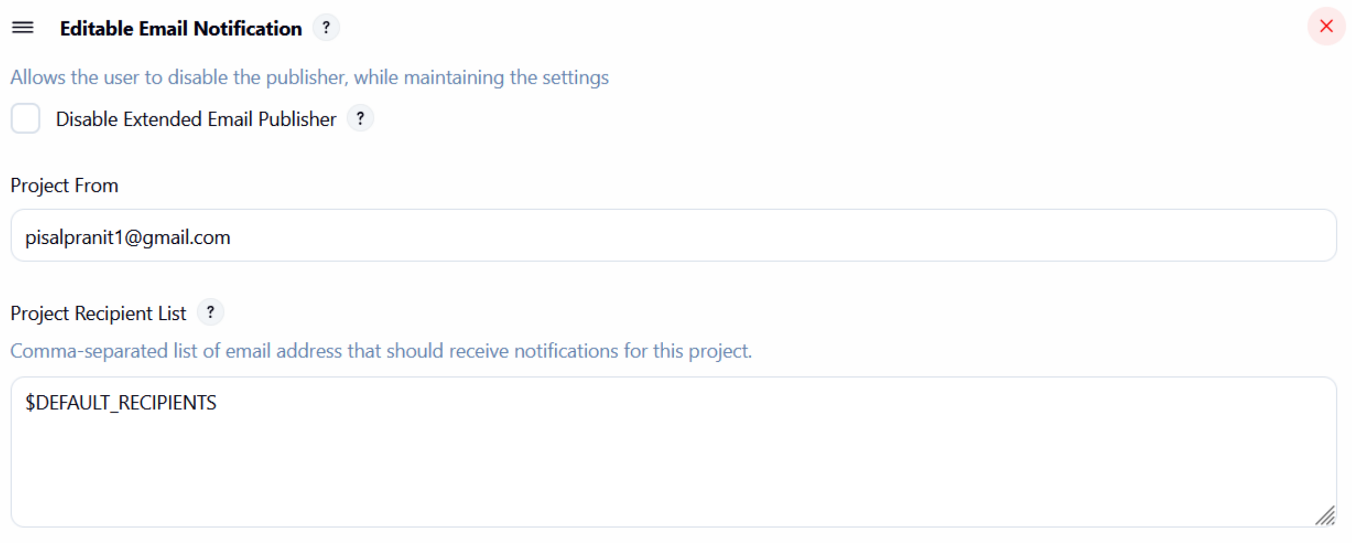
**Execute shell** ?

Command

See the list of available environment variables

```
chmod +x build.sh
./build.sh
```

Now under Post Build actions choose Editable Email Notification where you should fill the info as follows:



**Editable Email Notification** ?

Allows the user to disable the publisher, while maintaining the settings

☐ Disable Extended Email Publisher ?

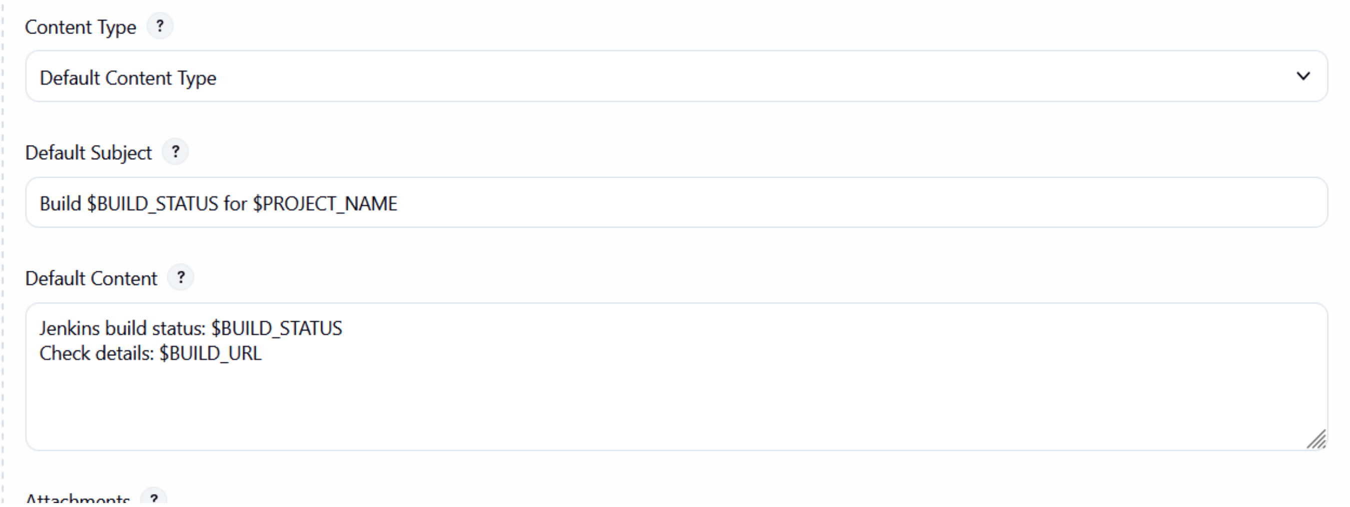
Project From

pisalpranit1@gmail.com

Project Recipient List ?

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

$DEFAULT_RECIPIENTS



Content Type ?

Default Content Type

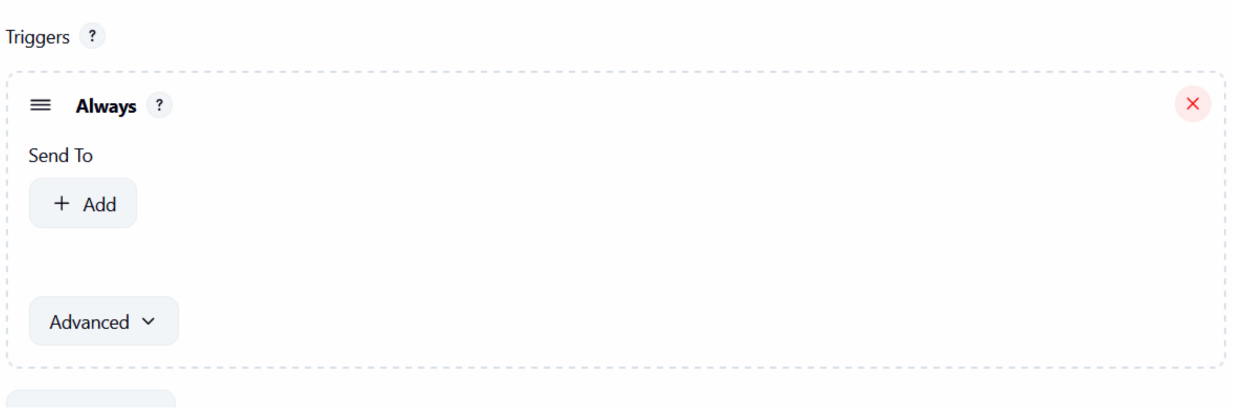
Default Subject ?

Build $BUILD_STATUS for $PROJECT_NAME

Default Content ?

```
Jenkins build status: $BUILD_STATUS
Check details: $BUILD_URL
```

Attachments ?

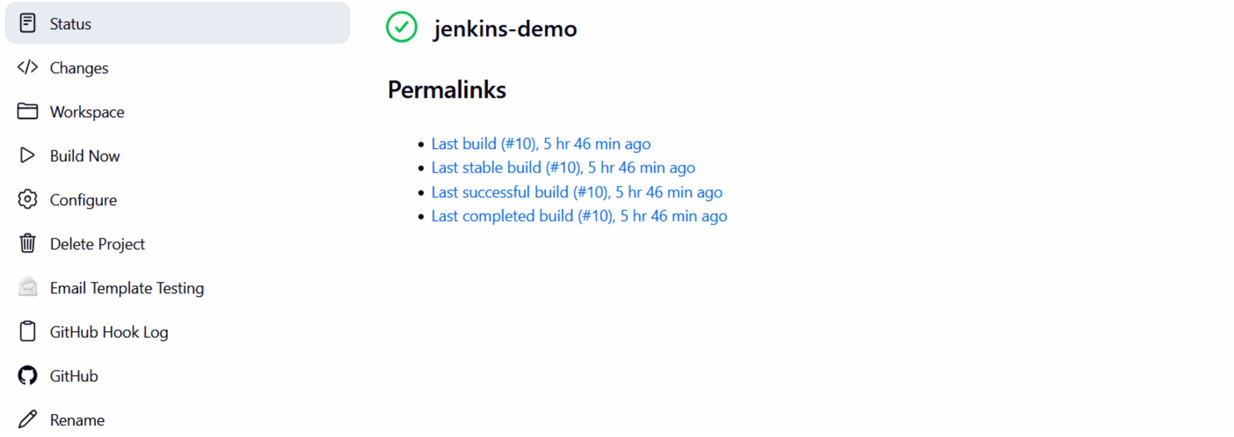


Triggers ?

Always ?

Send To

➔ Step 6: Finally save and build the Project.



jenkins-demo

Permalinks

- Last build (#10), 5 hr 46 min ago
- Last stable build (#10), 5 hr 46 min ago
- Last successful build (#10), 5 hr 46 min ago
- Last completed build (#10), 5 hr 46 min ago

➔ Now as soon as you make some changes in the Git files it send the triggers to Jenkins through Web-Hook where it will show you a build and notify you through Mail.

## jenkins-demo - Build # 10 - Successful

Inbox ×

**pisalpranit1@gmail.com**
to me

1:24 PM (5 hours ago)

Build result: Successful
Check console: http://3.110.84.32:8080/job/jenkins-demo/10/

Reply Forward