

Jenkins Task-2

Task Description:

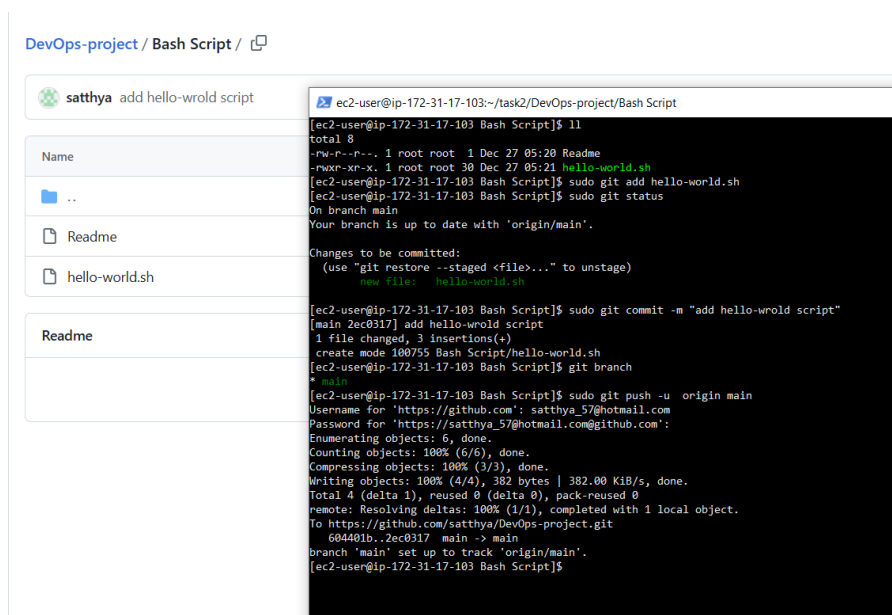
Create a simple script file and push it to repo. Create a project in Jenkins connected to your GitHub repository. When a commit is made to your repo, automatically build must get triggered from Jenkins and the output must be shared to me via email.

Techstacks needs to be used :


- AWS EC2
- Github
- Jenkins

Step1 : Create simple script and upload to Git Hub

- VI hello-world.sh
- chmod +x hello-world.sh
- git add hello-world.sh
- git commit -m "added hello-world script"
- git push -u origin main



The screenshot shows a terminal window with the following commands and output:

```
DevOps-project / Bash Script / 
sathya add hello-wrold script

Name
..
Readme
hello-world.sh

Readme

[ec2-user@ip-172-31-17-103:~/task2/DevOps-project/Bash Script]$ ll
total 8
-rw-r--r-- 1 root root 1 Dec 27 05:20 Readme
-rwxr-xr-x 1 root root 30 Dec 27 05:21 hello-world.sh
[ec2-user@ip-172-31-17-103 Bash Script]$ sudo git add hello-world.sh
[ec2-user@ip-172-31-17-103 Bash Script]$ sudo git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   hello-world.sh

[ec2-user@ip-172-31-17-103 Bash Script]$ sudo git commit -m "add hello-wrold script"
[main 2ec8317] add hello-wrold script
1 file changed, 3 insertions(+)
create mode 100755 Bash Script/hello-world.sh
[ec2-user@ip-172-31-17-103 Bash Script]$ git branch
* main
[ec2-user@ip-172-31-17-103 Bash Script]$ sudo git push -u origin main
Username for 'https://github.com': sathya_57@hotmail.com
Password for 'https://sathya_57@hotmail.com@github.com':
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 382 bytes | 382.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/sathya/DevOps-project.git
684481b..2ec8317 main -> main
branch 'main' set up to track 'origin/main'.
[ec2-user@ip-172-31-17-103 Bash Script]$
```

Step2 : Configure Jenkins Project

- Create Freestyle Project

New Item

Enter an item name

Guvi-Task2

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.

- Link to GitHub
 - Source Code Management > Git

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/satthya/DevOps-project.git

Credentials ?

- none -

+ Add

Advanced ▾

- Add Build Trigger
 - Build Triggers > Poll SCM

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 ?

Schedule ?

H/5 * * * *

Would last have run at Friday, December 27, 2024 at 8:29:32 AM Coordinated Universal Time; would next run at Friday, December 27, 2024 at 8:34:32 AM Coordinated Universal Time.

- ☐ Ignore post-commit hooks ?

- Define Build step
 - Add build step > Execute shell

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](#)

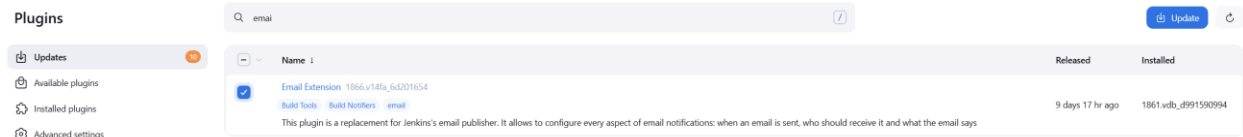
```
bash hello-world.sh
```

Advanced ▾

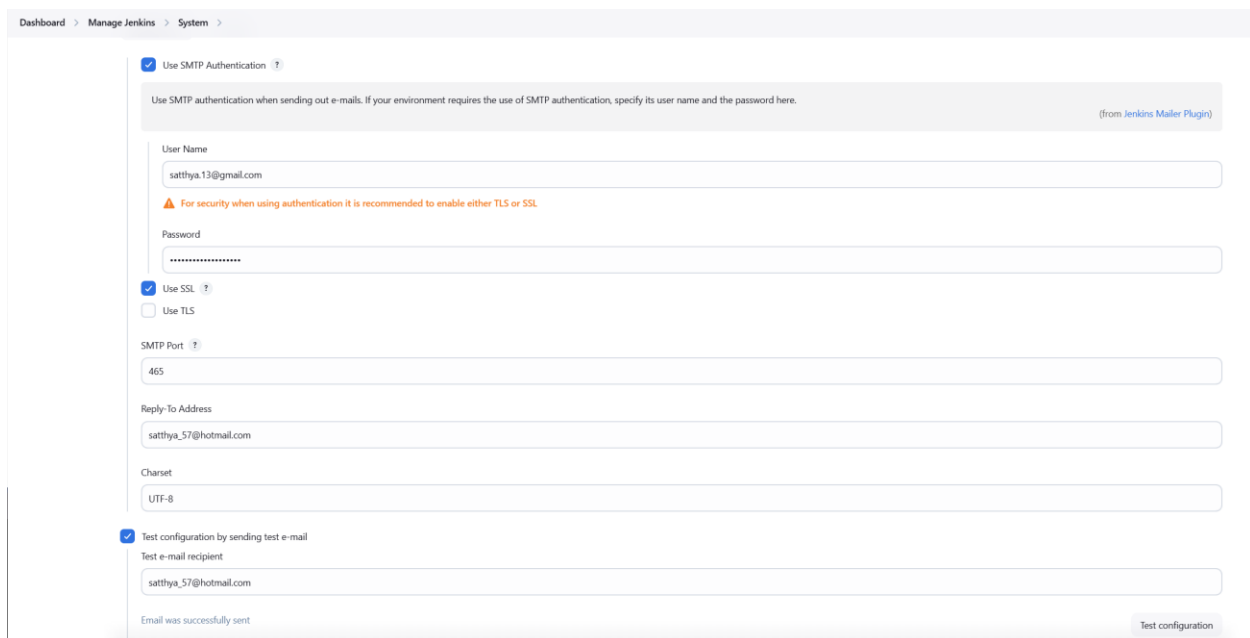
Add build step ▾

Step5 : Send Email Notifications

- Manage Jenkins > Plugin
- Install Email Extension Plugin



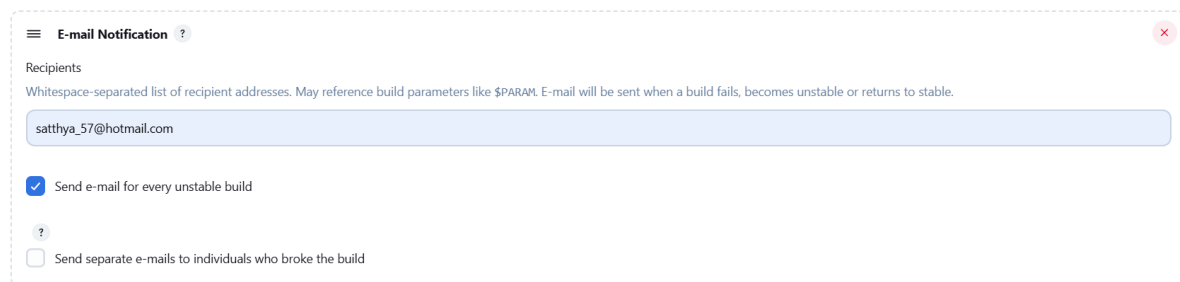
- Manage Jenkins > System
- Configure Email notification



- Jenkins project > Post-build Actions
- Select Email Notification

Post-build Actions

Define what happens after a build completes, like sending notifications, archiving artifacts, or triggering other jobs.



Step3 : Output Result

The screenshot shows the Jenkins web interface. At the top, there's a navigation bar with 'Dashboard' and 'Guvi-Task2'. The main content area is titled 'Guvi-Task2' with a green checkmark icon. Below this, there's a 'Status' section with a list of actions: Changes, Workspace, Build Now, Configure, Delete Project, Git Polling Log, and Rename. To the right of the 'Status' section, there's a 'Permalinks' section with a list of links: Last build (#7), Last stable build (#7), Last successful build (#7), Last failed build (#2), Last unsuccessful build (#2), and Last completed build (#7). Below the 'Status' section, there's a 'Builds' section with a filter input and a list of builds. The list shows two builds: #8 at 12:24 PM (failed) and #7 at 11:49 AM (successful).

The screenshot shows an Outlook email interface. The email is titled 'Build failed in Jenkins: Guvi-Task2 #8'. The sender is 'address not configured yet<satthya.13@gmail.com>' and the recipient is 'To: You'. The email content starts with 'This message was identified as junk.' followed by a link to the Jenkins build page: <http://18.141.178.222:8080/job/Guvi-Task2/8/display/redirect?page=changes>. The email then lists the changes: [satthya] update hello-world.sh v5. Below this, there's a detailed log of the build process, including the SCM change, the build workspace, the recommended git tool, and the build steps. The log ends with the message 'Build step 'Execute shell' marked build as failure'.