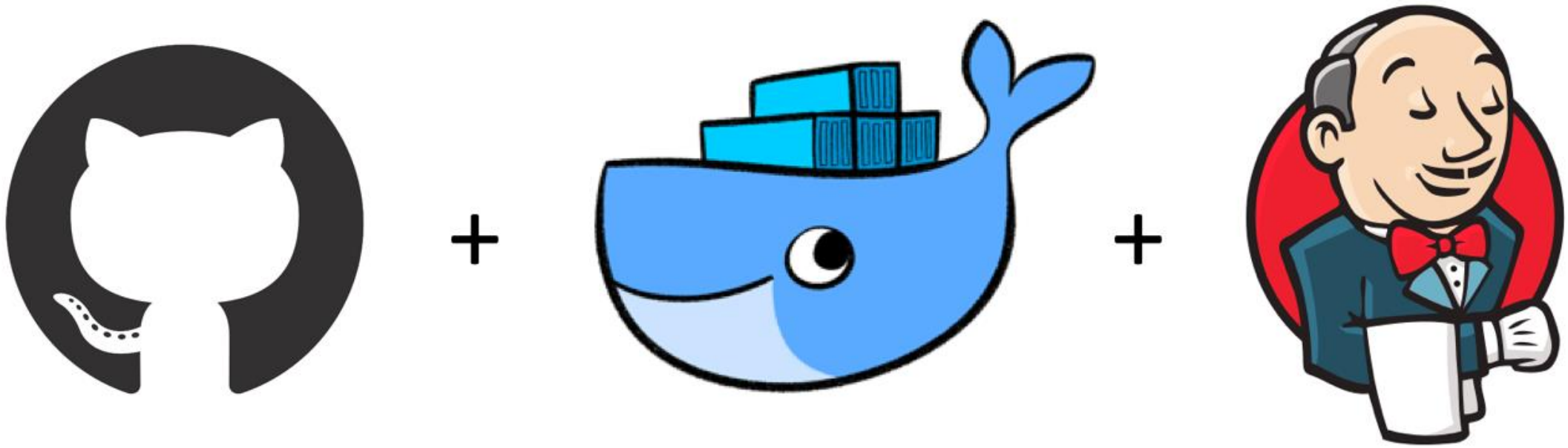
STEP BY STEP GUIDE



1. Push code → git
2. Trigger build →
3. Spin up docker environment → docker
4. Run tests →

1. Create a Local Repository

>>Create a local repo in your git

>>Initialize it : git init

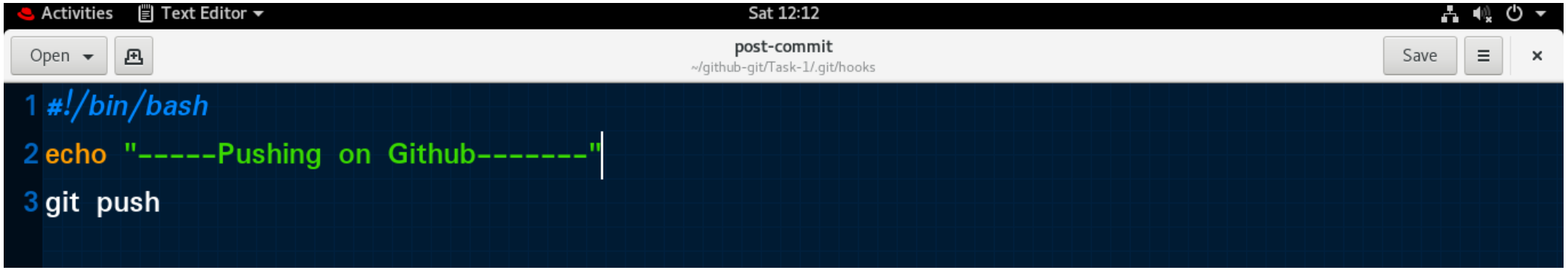
>>Create a new file : gedit file1

>>Add it in tracking system : git add .

>>Add post-commit hook to automate pushing onto github :

cd .git → cd hooks → gedit post-commit

and write the following code:

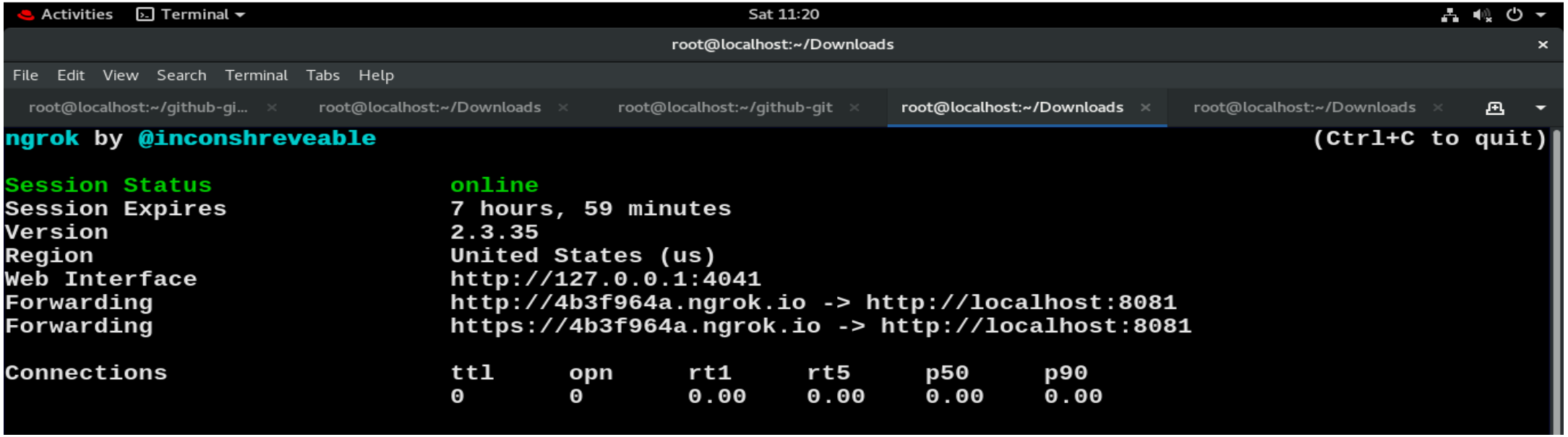


2. Create a Remote Repo in Github

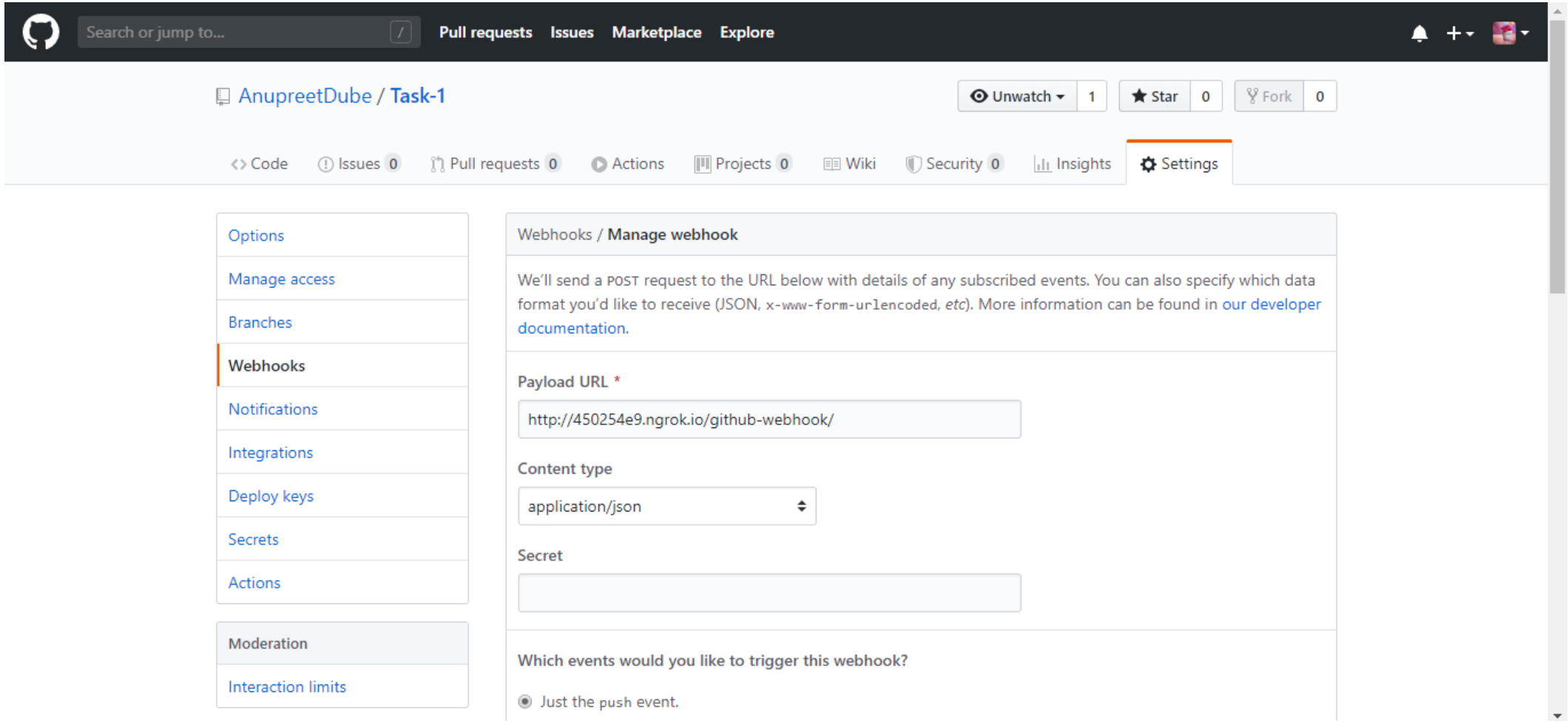
>>don’t initialize with readme

>>copy the command for adding remote path and paste this command in local repo terminal

>> Run ./ngrok command in local repo’s prompt to enable tunnling of Jenkins : ./ngrok http 8080 → copy the url as below



>>In remote repo settings → Webhooks → Create webhooks → paste the url path copied and add /github-webhook/ after the path and configure as shown:



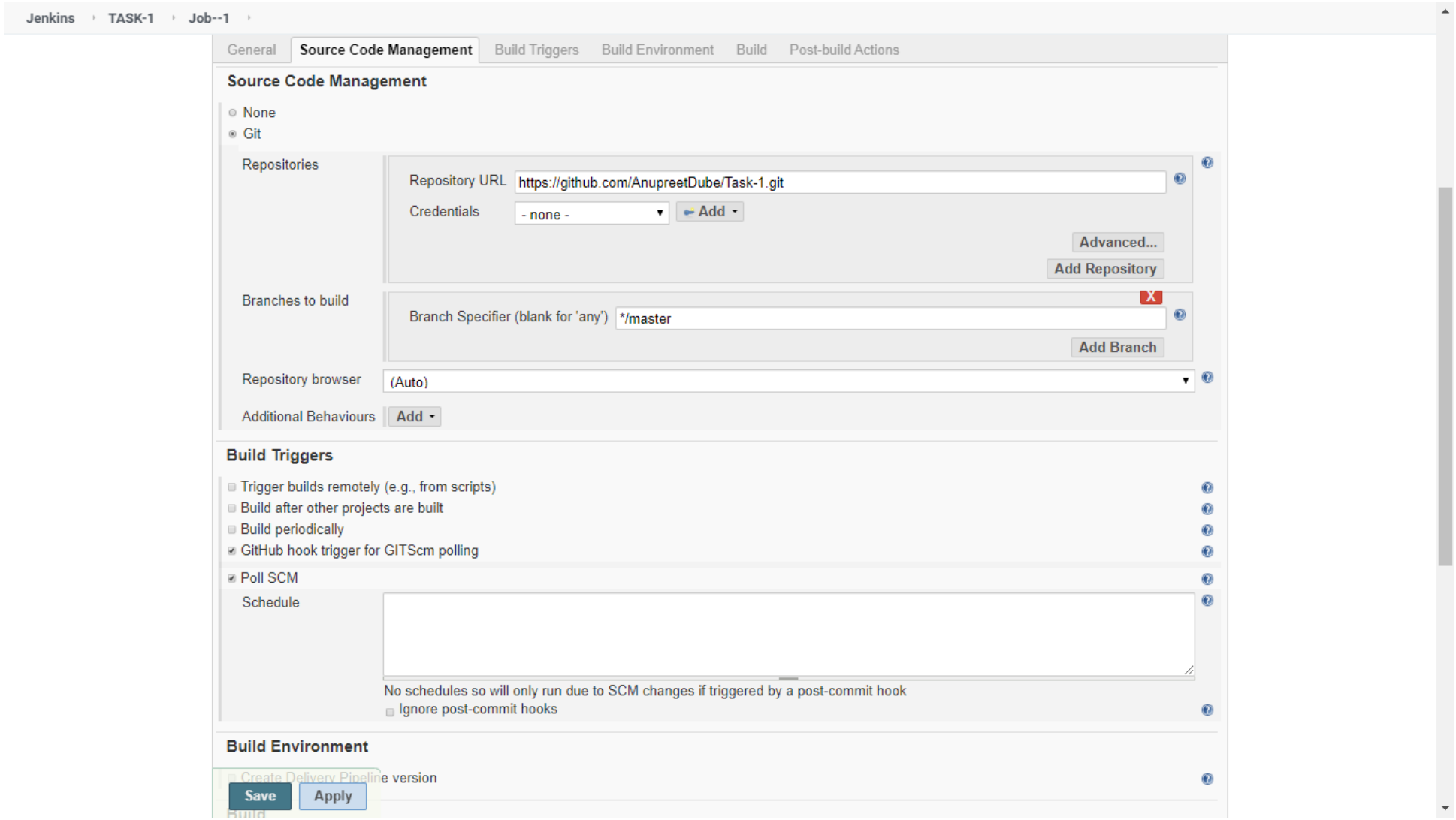
3. Create three Jenkins jobs

Job1 : To Download any new code pushed onto the master branch of the Remote Repo -->> Copy this code into a local folder inside RHEL_8 -->> Launch a webServer configured container via Docker : PRODUCTION ENVIRONMENT, and link that folder to this container's /var/www/httpd file to deploy the real time changes on the Client website

Job2 : To Download any new code pushed onto the feature branch (dev1) of the Remote Repo -->> Copy this code into a local folder inside RHEL_8 -->> Launch a webServer configured container via docker : TESTING ENVIRONMENT, and link that folder to this container's /var/www/httpd file to deploy the real time changes on the Test website

Job3 : To be run manually by the Quality Assurance team when the Feature branch code is verified -->> Merge the feature branch to the master branch -->> Run Job1 to deploy the changes on the Client Website

JOB-1


Jenkins
TASK-1
Job--1
General
Source Code Management
Build Triggers
Build Environment
Build
Post-build Actions
Source Code Management
None
Git
Repositories
Repository URL
https://github.com/AnupreetDube/Task-1.git
Credentials
- none -
Add
Advanced...
Add Repository
Branches to build
Branch Specifier (blank for 'any')
*/master
Add Branch
Repository browser
(Auto)
Additional Behaviours
Add
Build Triggers
Trigger builds remotely (e.g., from scripts)
Build after other projects are built
Build periodically
GitHub hook trigger for GITScm polling
Poll SCM
Schedule
No schedules so will only run due to SCM changes if triggered by a post-commit hook
Ignore post-commit hooks
Build Environment
Create Delivery Pipeline version
Save
Apply

Jenkins › TASK-1 › Job--1 ›

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

Build

Execute shell

Command

```
sudo cp -vrf * /root/github-git/linking

if sudo docker ps | grep Production
then
echo 'production active'
else
sudo systemctl start docker
sudo docker run -dti -p 8081:80 -v /root/github-git/linking/:/usr/local/apache2/htdocs --name Production httpd
fi
```

See the list of available environment variables

Advanced...

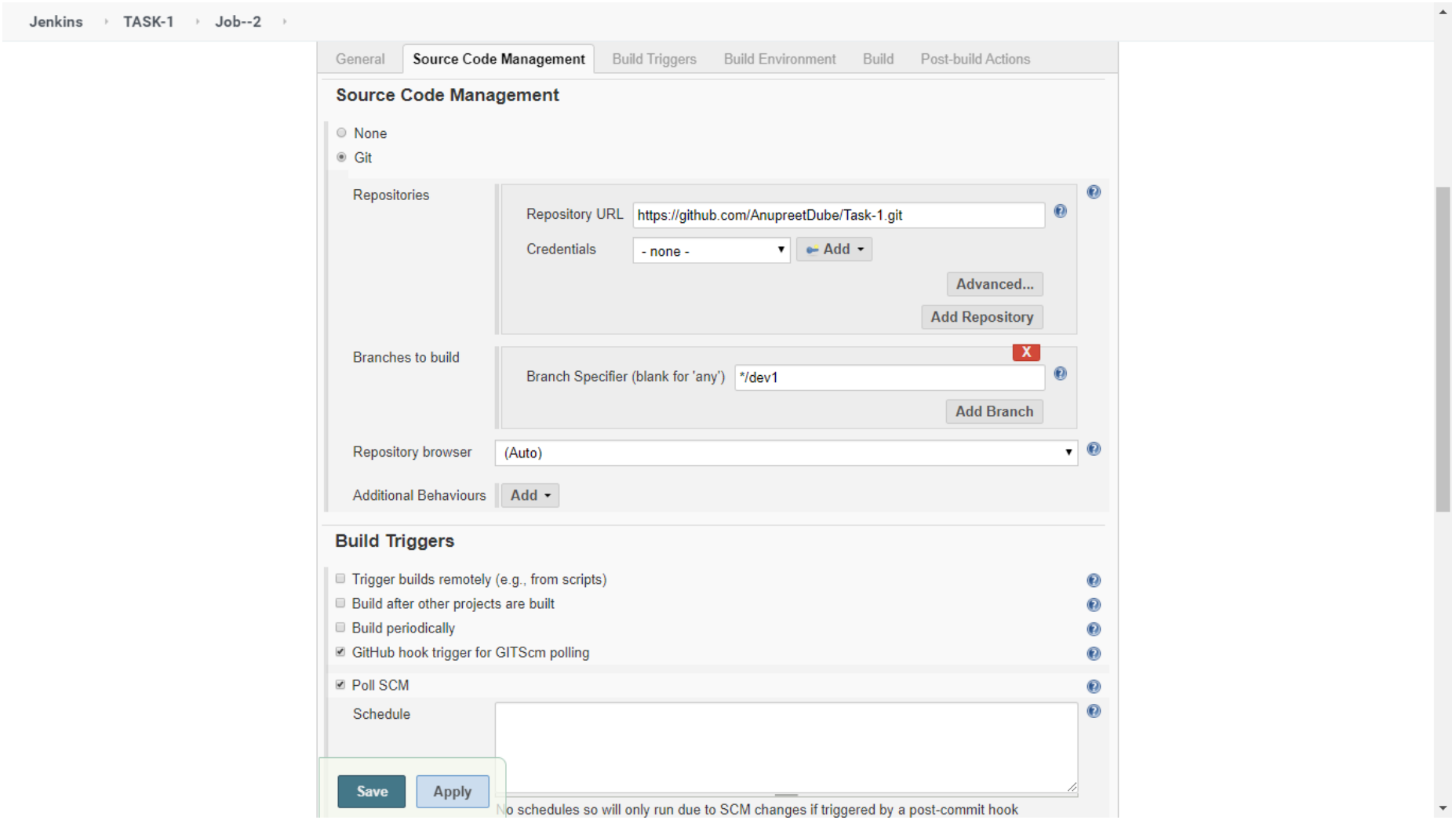
Add build step

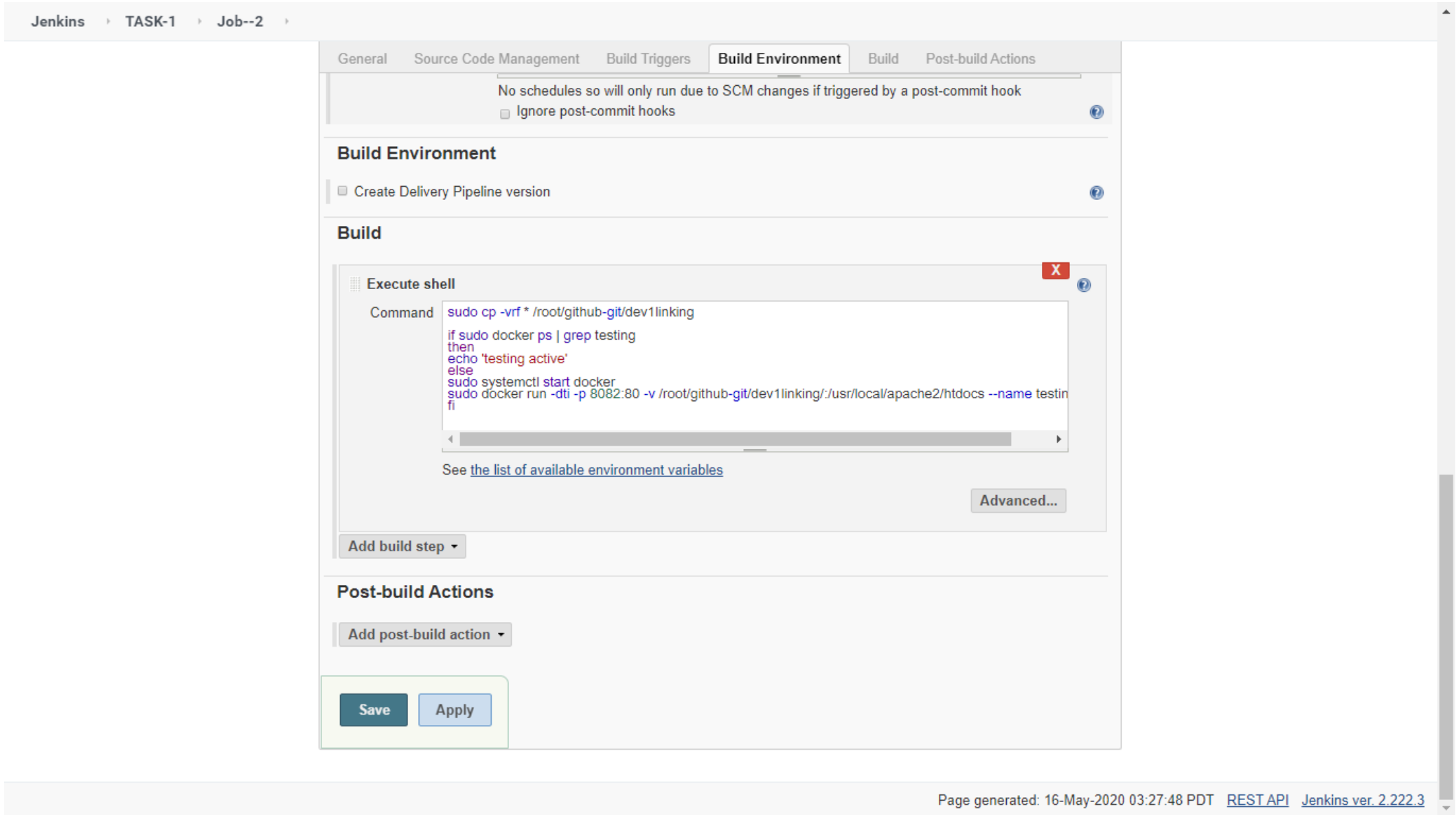
Post-build Actions

Add post-build action

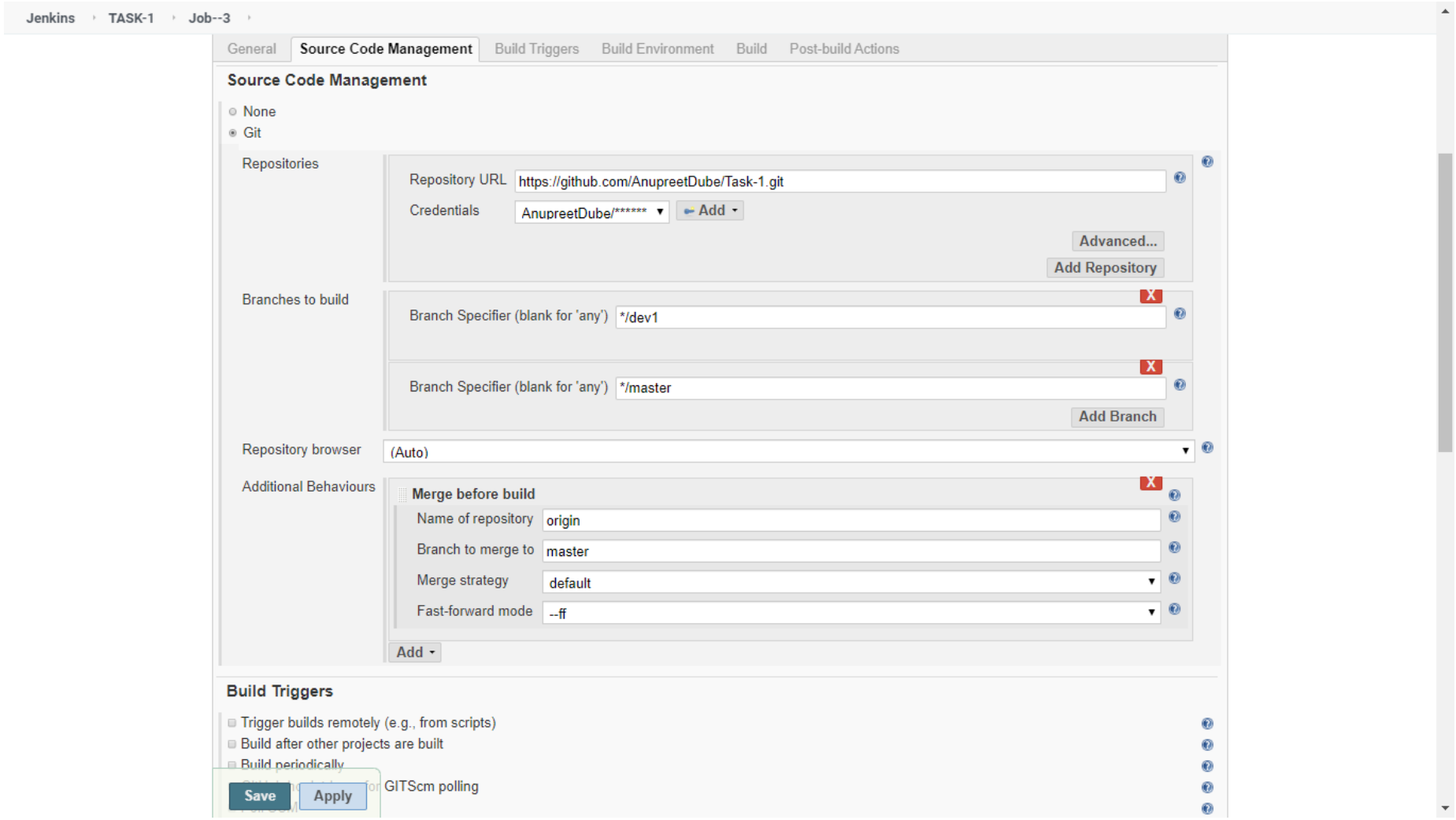
Save | Apply

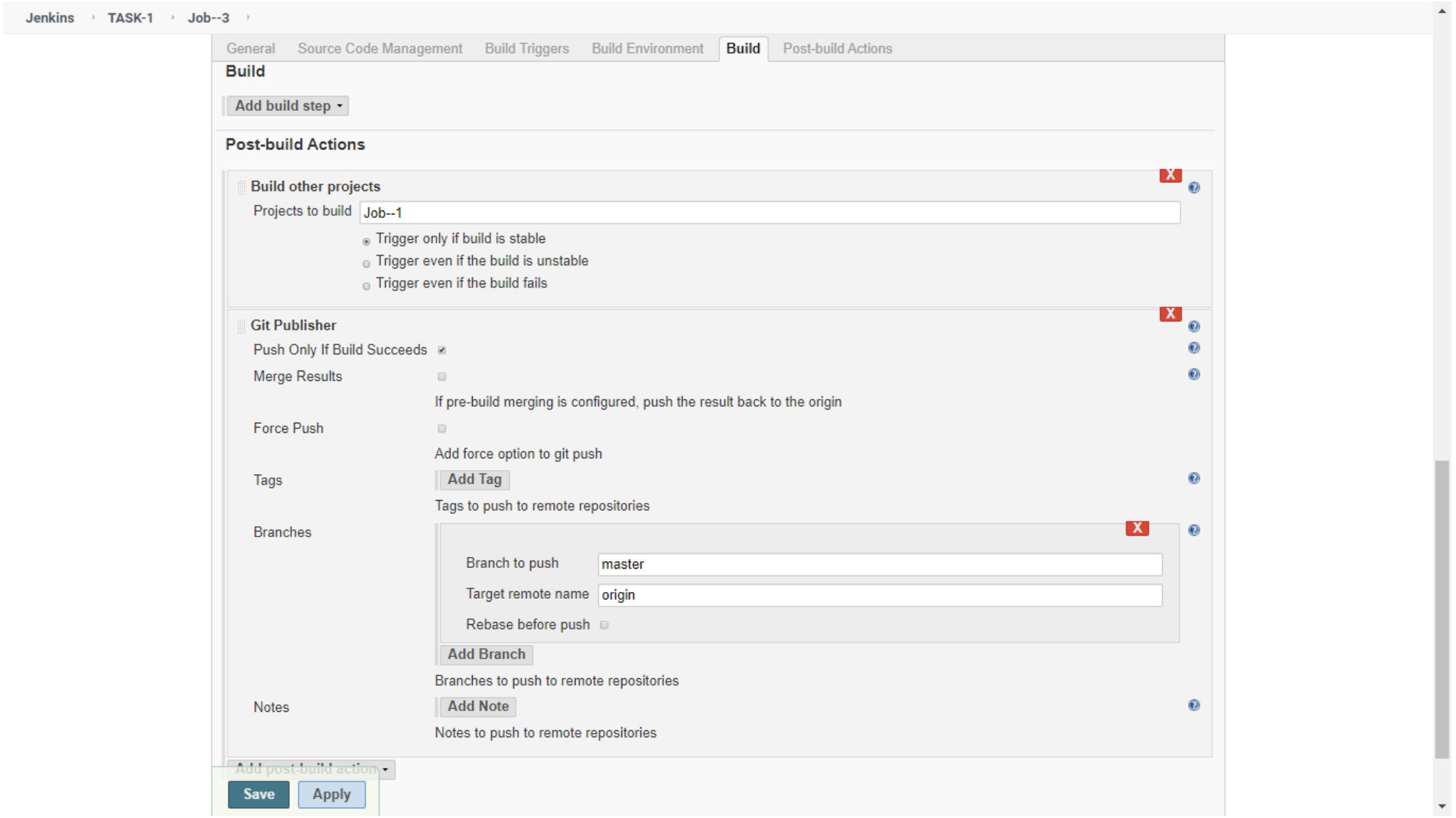
JOB-2


Jenkins
TASK-1
Job--2
General
Source Code Management
Build Triggers
Build Environment
Build
Post-build Actions
Source Code Management
None
Git
Repositories
Repository URL
https://github.com/AnupreetDube/Task-1.git
Credentials
- none -
Add
Advanced...
Add Repository
Branches to build
Branch Specifier (blank for 'any')
*/dev1
Add Branch
Repository browser
(Auto)
Additional Behaviours
Add
Build Triggers
Trigger builds remotely (e.g., from scripts)
Build after other projects are built
Build periodically
GitHub hook trigger for GITScm polling
Poll SCM
Schedule
Save
Apply
o schedules so will only run due to SCM changes if triggered by a post-commit hook


Jenkins › TASK-1 › Job--2 ›
General
Source Code Management
Build Triggers
Build Environment
Build
Post-build Actions
No schedules so will only run due to SCM changes if triggered by a post-commit hook
Ignore post-commit hooks
Build Environment
Create Delivery Pipeline version
Build
Execute shell
Command
sudo cp -vrf * /root/github-git/dev1linking
if sudo docker ps | grep testing
then
echo 'testing active'
else
sudo systemctl start docker
sudo docker run -dti -p 8082:80 -v /root/github-git/dev1linking/:/usr/local/apache2/htdocs --name testin
fi
See the list of available environment variables
Advanced...
Add build step
Post-build Actions
Add post-build action
Save
Apply
Page generated: 16-May-2020 03:27:48 PDT
REST API
Jenkins ver. 2.222.3

JOB-3


Jenkins › TASK-1 › Job--3 ›
General
Source Code Management
Build Triggers
Build Environment
Build
Post-build Actions
Source Code Management
None
Git
Repositories
Repository URL
https://github.com/AnupreetDube/Task-1.git
Credentials
AnupreetDube/******
Add
Advanced...
Add Repository
Branches to build
Branch Specifier (blank for 'any')
*/dev1
Branch Specifier (blank for 'any')
*/master
Add Branch
Repository browser
(Auto)
Additional Behaviours
Merge before build
Name of repository
origin
Branch to merge to
master
Merge strategy
default
Fast-forward mode
--ff
Add
Build Triggers
Trigger builds remotely (e.g., from scripts)
Build after other projects are built
Build periodically
GITScm polling
Save
Apply


Jenkins
TASK-1
Job--3
General
Source Code Management
Build Triggers
Build Environment
Build
Post-build Actions
Build
Add build step
Post-build Actions
Build other projects
Projects to build
Job--1
Trigger only if build is stable
Trigger even if the build is unstable
Trigger even if the build fails
Git Publisher
Push Only If Build Succeeds
Merge Results
If pre-build merging is configured, push the result back to the origin
Force Push
Add force option to git push
Tags
Add Tag
Tags to push to remote repositories
Branches
Branch to push
master
Target remote name
origin
Rebase before push
Add Branch
Branches to push to remote repositories
Notes
Add Note
Notes to push to remote repositories
Add post-build action
Save
Apply

4. Committing and testing

>> Commit the code in local repo : git commit . –m ‘1st commit’

>>Automated pushing will start

>> the code will be pushed to github

>>if the code is pushed via master branch then job1 will run

>>if the code is pushed via dev1 branch then job2 will run

>>enable tunnelling on the docker containers launched for production environment (via job1) and testing environment (via job2)

>> open the url obtained via tunnling of either containers

>>Run job3 manually if you are satisfied with the Testing website →

Dev1 branch will get merged to master branch → Job1 will run to deplo the changes

Tunneling URL obtained via ngrok

Index of /

Production Environment's Deployment : Client website

- d1.html
- d2
- m1.html
- m2.html

Index of /

Testing Environment's Deployment : Testing website

- d1.html
- d2
- m1.html