==**Objective : to upload index.html from local repo and output should be visible on apache web server .**==

==**Local repo: Also create a repo in Github account**==
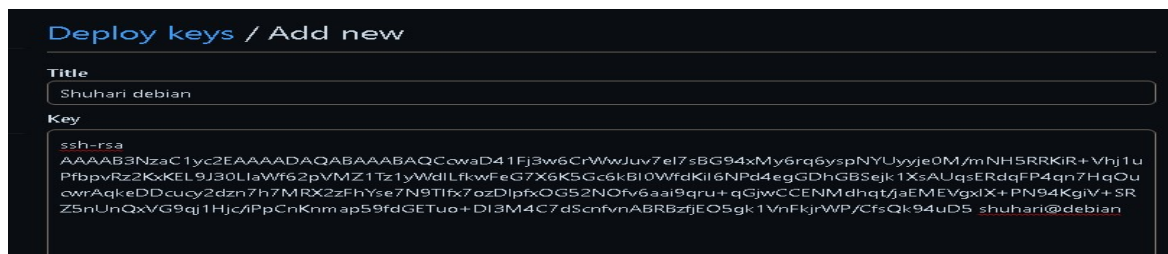
$sudo apt-get install git
$cd project
$git init
$git remote add origin "https/git"


```
HTTPS    SSH    GitHub CLI  (New)

You don't have any public SSH keys in your GitHub account.
You can add a new public key, or try cloning this repository via
HTTPS.

git@github.com:omg1410-gadre/demo27.git
```

$ssh-keygen
Add public key into github settin g-> deploy keys)


```
Deploy keys / Add new
Title
Shuhari debian
Key
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCowaD41Fj3w6CrWwJuv7el7sBG94xMy6rq6yspNYUyyje0M/mNH5RRKiR+Vhj1u
PfbpvRz2KxKEL9J30LIaWf62pVMZ1Tz1yWdILfkwFeG7X6K5Gc6kBI0WfdKil6NPd4egGDhGBSejk1XsAUqsERdqFP4qn7HqOu
cwrAqkeDDcucy2dzn7h7MRX2zFhYse7N9Tlfx7ozDlpfxOG52NOfv6aai9qru+qGjwCCENMdhqt/jaEMEVgxlX+PN94KgiV+SR
Z5nUnQxVG9qj1Hjc/iPpCnKnmap59fdGETuo+DI3M4C7dScnfvnABRBzfjEO5gk1VnFkjrWP/CfsQk94uD5 shuhari@debian
```

$git pull origin master
$git --config global user.name "…"
$git --config global user.email "…."
$git add index.html
$git commit -m "msg"
$git push origin master

Allow webhook
Setting :
Payload url : (……..url of jenkins…..)/github-webhook/
which event (just push)
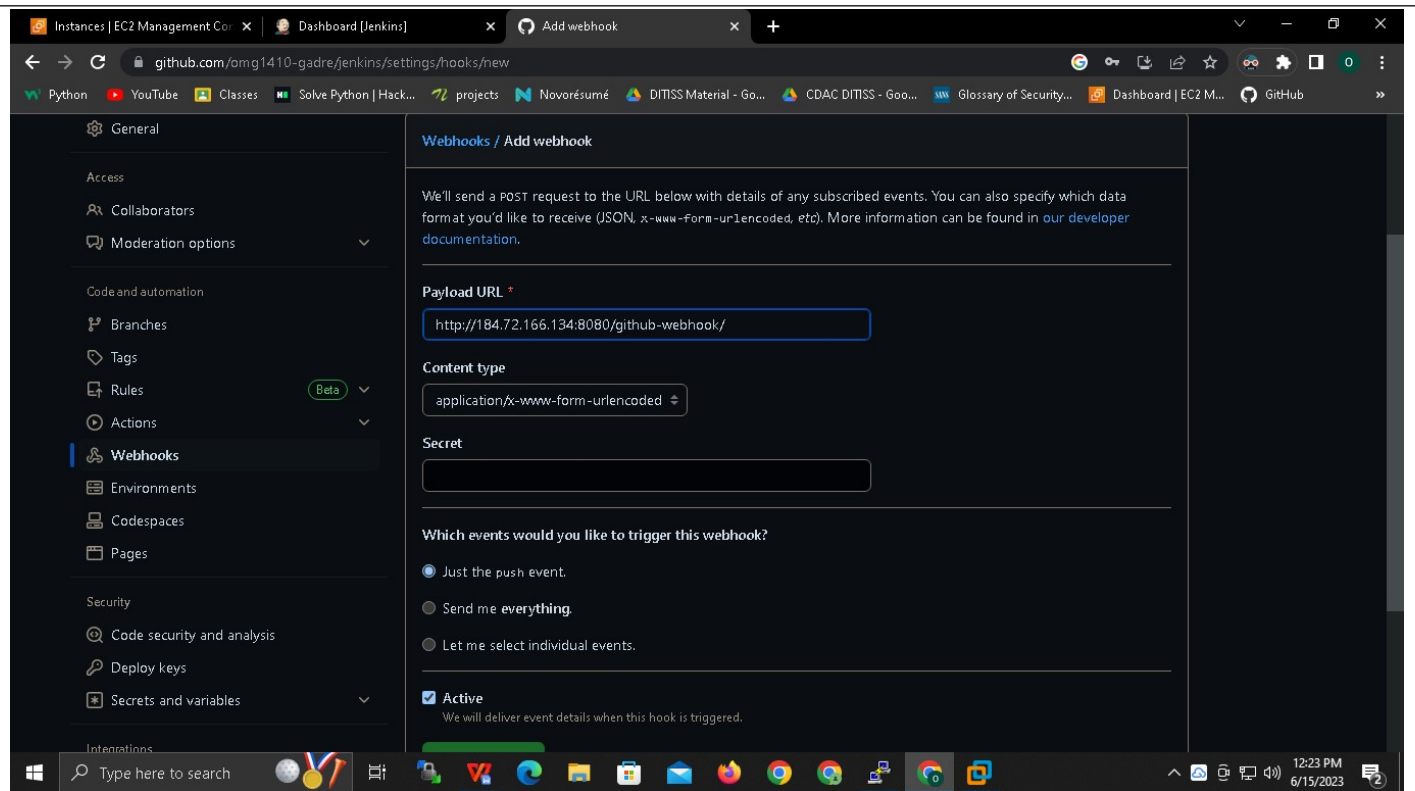
Jenkins

Create EC2 instance , allow http and https traffic as well .
Login in putty :


```
Using username "admin".
Authenticating with public key "jenkins"
Linux ip-172-31-91-90 5.10.0-23-cloud-amd64 #1 SMP Debian 5.10.179-1 (2023-05-12
) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
admin@ip-172-31-91-90:~$
```

$ apt-get update -y
$ apt-get upgrade -y
$ apt-get install gnupg -y *(it's a basic and simple tool for encrypting files)*
$ apt-get install default-jre -y (*used for compilig and launching java programm***)**
$ curl -fsSL https://pkg.jenkins.io/debian/jenkins.io-2023.key | sudo tee /usr/share/keyrings/jenkins-keyring.asc > /dev/null *(generating key with curl command)*
**$ echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] **
    **https://pkg.jenkins.io/debian binary/ | sudo tee **
    /etc/apt/sources.list.d/jenkins.list > /dev/null *(updating repo file)*
$ apt update -y
$ apt-get install jenkins -y
$ sudo usermod -a -G root jenkins *(adding jenkins users in root group I.e secondry)*
$ systemctl status jenkins

Port of jenkins is 8080

```
^C
admin@ip-172-31-91-90:~$ ss -ant
State      Recv-Q  Send-Q    Local Address:Port        Peer Address:Port      Process
LISTEN     0       128            0.0.0.0:22              0.0.0.0:*
ESTAB      0       64        172.31.91.90:22         202.71.157.78:57741
TIME-WAIT  0       0         172.31.91.90:53026       20.119.232.75:443
SYN-RECV   0       0         172.31.91.90:22        169.228.66.212:39691
LISTEN     0       50               *:8080                   *:*
LISTEN     0       128             [::]:22                 [::]:*
admin@ip-172-31-91-90:~$
```
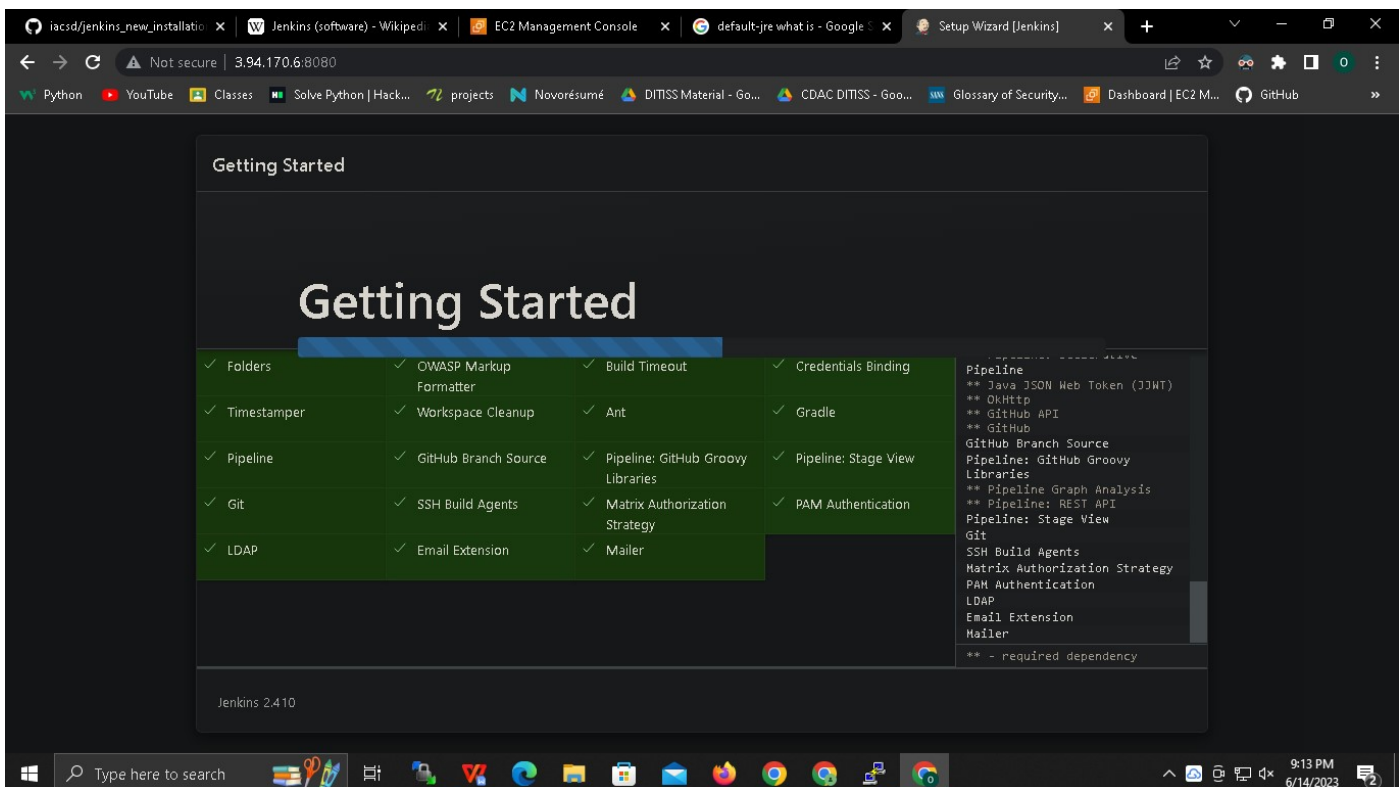
Go to aws -> Security -> edit inbound rules -> allow traffic at 8080

Security group rule ID	Type		Protocol Info	Port range Info	Source Info		Description - optional Info	
sgr-073af338da2607f16	HTTPS	▼	TCP	443	Custom ▼	Q		Delete
						0.0.0.0/0 ✕		
sgr-0eb549046fed0a7d6	HTTP	▼	TCP	80	Custom ▼	Q		Delete
						0.0.0.0/0 ✕		
sgr-04d1ef6b5f9fbc2ed	Custom TCP	▼	TCP	8080	Custom ▼	Q		Delete
						0.0.0.0/0 ✕		

Go to browser -> public ip : 8080
Copy password and paste it

```
admin@ip-172-31-91-90:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
a62630d952d9458f86b2e7b1e61d4bc1
admin@ip-172-31-91-90:~$
```

Getting Started

Getting Started

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	✓ Credentials Binding
✓ Timestamper	✓ Workspace Cleanup	✓ Ant	✓ Gradle
✓ Pipeline	✓ GitHub Branch Source	✓ Pipeline: GitHub Groovy Libraries	✓ Pipeline: Stage View
✓ Git	✓ SSH Build Agents	✓ Matrix Authorization Strategy	✓ PAM Authentication
✓ LDAP	✓ Email Extension	✓ Mailer	

```
Pipeline
** Java JSON Web Token (JJWT)
** OkHttp
** GitHub API
** GitHub
GitHub Branch Source
Pipeline: GitHub Groovy
Libraries
** Pipeline Graph Analysis
** Pipeline: REST API
Pipeline: Stage View
Git
SSH Build Agents
Matrix Authorization Strategy
PAM Authentication
LDAP
Email Extension
Mailer

** - required dependency
```

Jenkins 2.410

$sudo apt-get install git

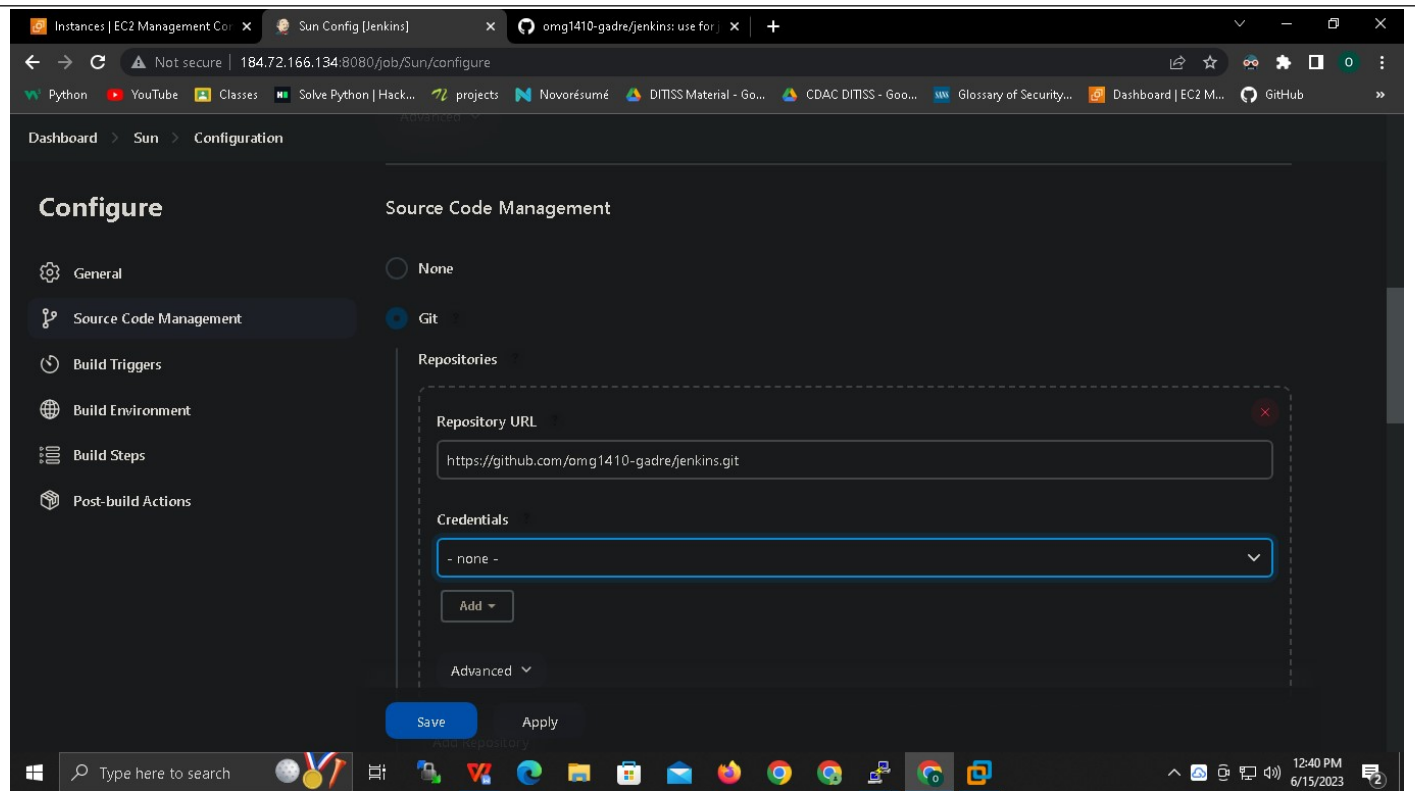==Define a project==

freestyle
Select github project add url -> copy url of github http
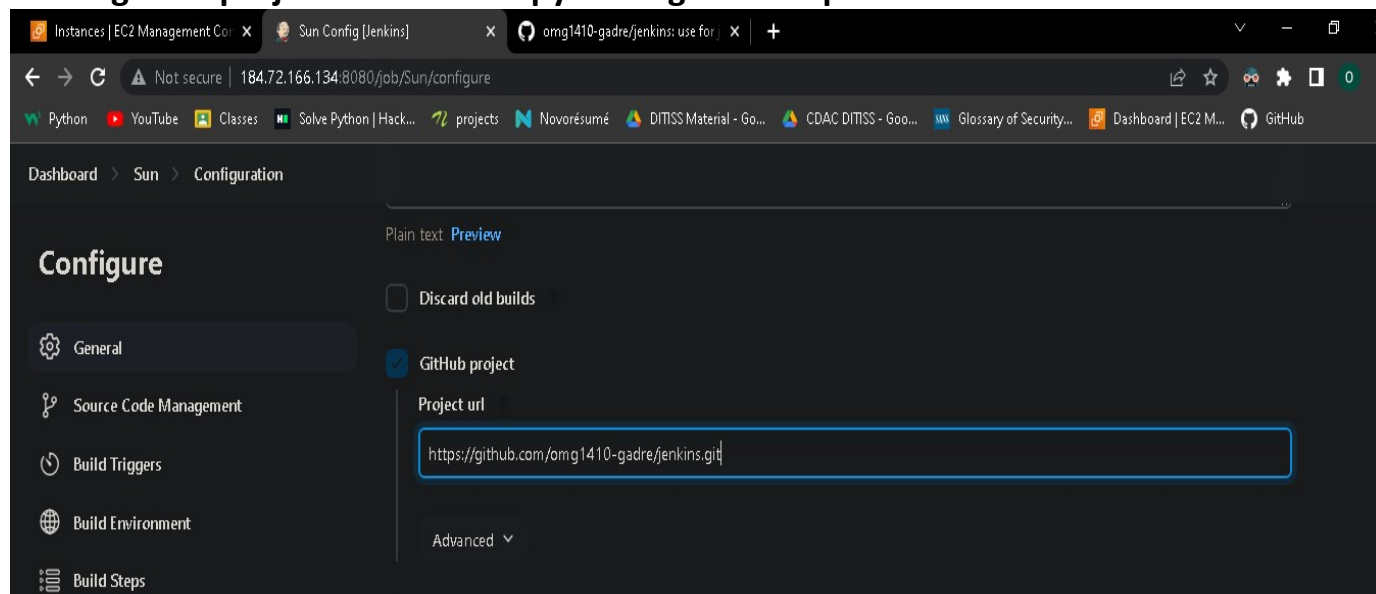
==**Source code mgmt**== **-> git -> repo url ->https link ->** *error will come because git is on jenkins*
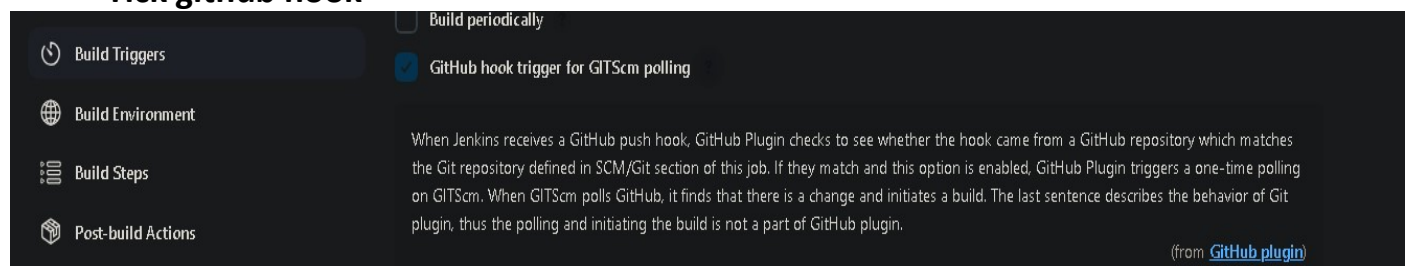
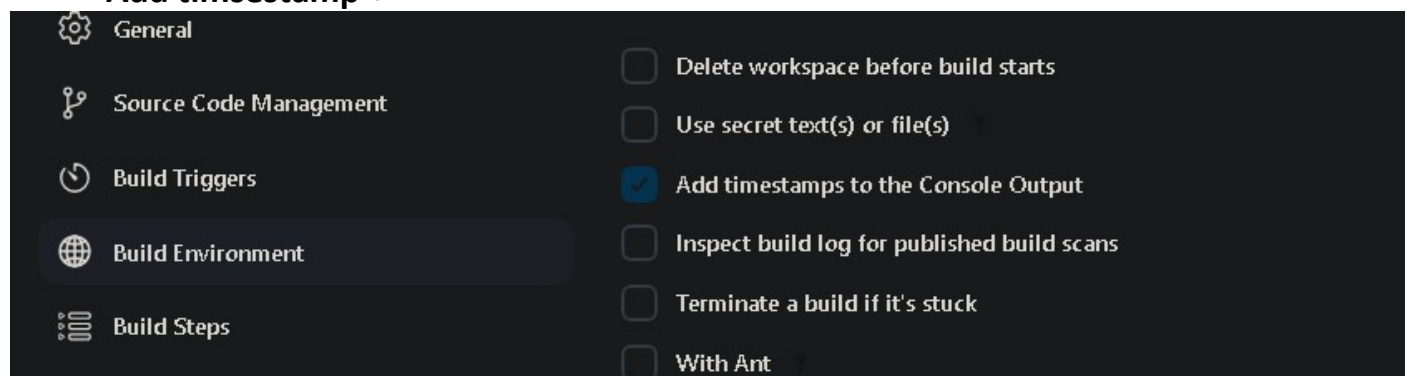   *We shall now install git in jenkins as well*
   No error

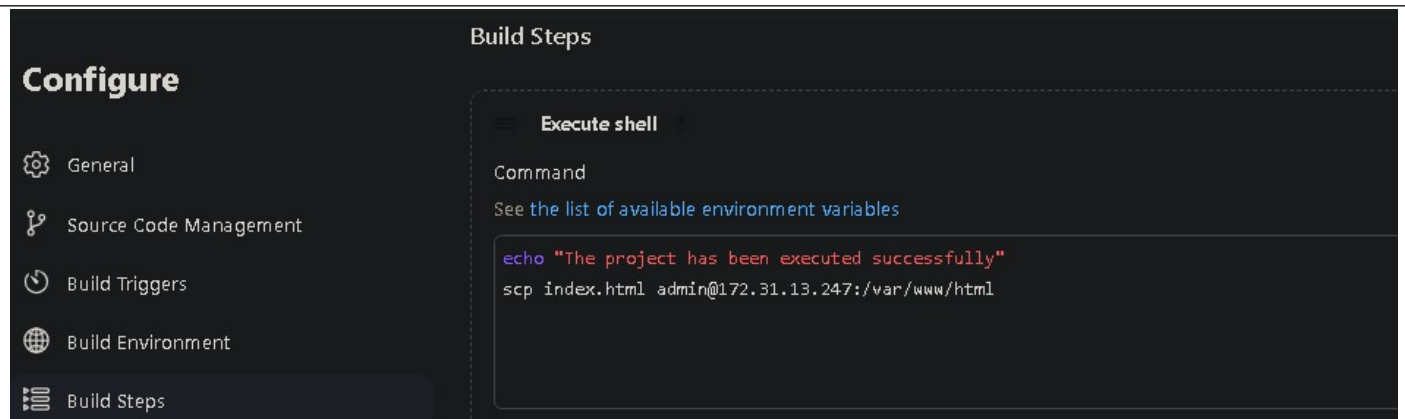## Build triggers :

Tick github hook

Build environment
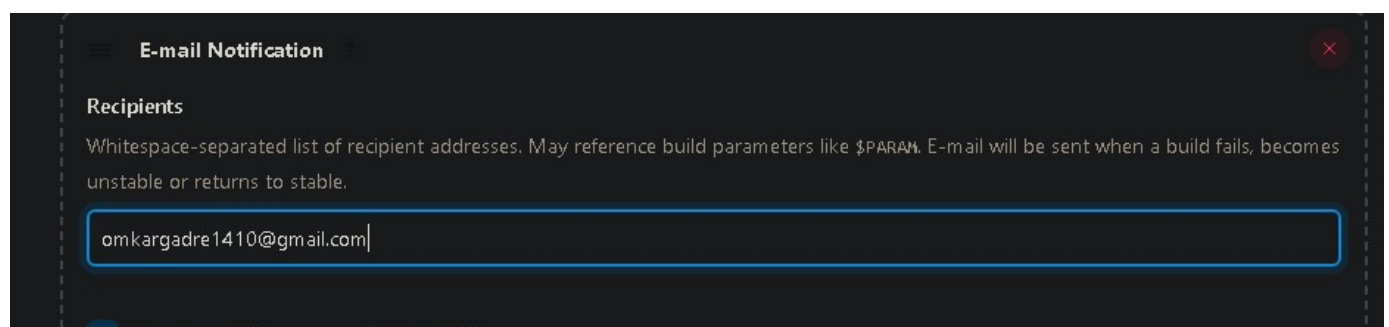
Add timsestamp +

Execute shells :

Echo "………………….."

scp admin@ip of apache:/var/www/html

Email notfn

Now Important steps start:

Our main objective is automation and its biggest hindarance is password hence we need to have paswordless ssh to allow scp command to work , always remember public key of client is stored with server not vice versa and always public key is distributed not private key.

Jenkins :
$sudo passwd jenkins *(changing passwd user jenkins)*
$su jenkins *(substituting user to jenkins as script is being run by that user)*
$ssh-keygen *(generating key)*
$cat ....pub key *(seeing key opening with nano is danegrous)*

Webserver
$cd /etc/ssh
$sudo nano sshd_config (uncomment pubkey authentication) it means alowing login who have pub key
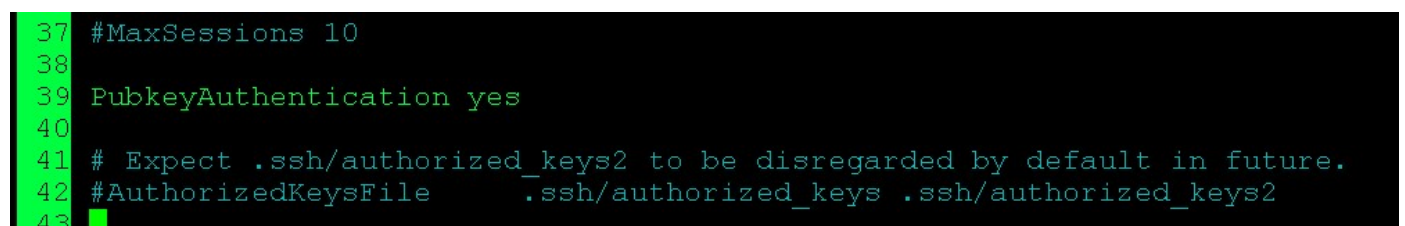
$ls -a (to show hidden files)
$cd .ssh
$sudo nano authorized_keys
        Add key of jenkins user here

```
  GNU nano 5.4                          authorized_keys

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDBXWD/WGynqmQVbPComFyxdGiF1763yPFzjLSO7nSFWDcw+17szC6IF8lGtecm+Z▶
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDg51CxLJlz0SuDk3hjiNFK8WWah/Csu1NH5dDHFdlY4OpsHf5rQmPgRw7S9R9glN▶
```

Also provide write permission to
$sudo chmod 7777 -R /var/www/html (-R recursive) 7777 user ,group,others and write

Manual Testing :
Jenkins :
$ scp demo.txt admin@ip of webserver:/var/www/html   (successful)

Now automation testing :

Go to local repo -> nano index.html -> git add index.html-> git commit -m "…"-> git push origin master.

```
shuhari@debian:~/project$ git push origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 412 bytes | 412.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To github.com:omg1410-gadre/project.git
   2567eea..9cdf714  master -> master
shuhari@debian:~/project$
```

Go in Jenkins dashboard check build :

Insert Public ip of webserver on browser

omg1410-gadre/project × | Instance details | EC2 Managemer × 44.197.187.119 × one #3 Console [Jenkins] × +

← → C ⚠ Not secure | 44.197.187.119

Python YouTube Classes Solve Python | Hack... projects Novorésumé DITISS Material - Go... CDAC DITISS - Goo... Glossary of

Automation is complete the developer ahs uploaded the apache page from local machine and jenkins have automaticallu upoaded it to webserver