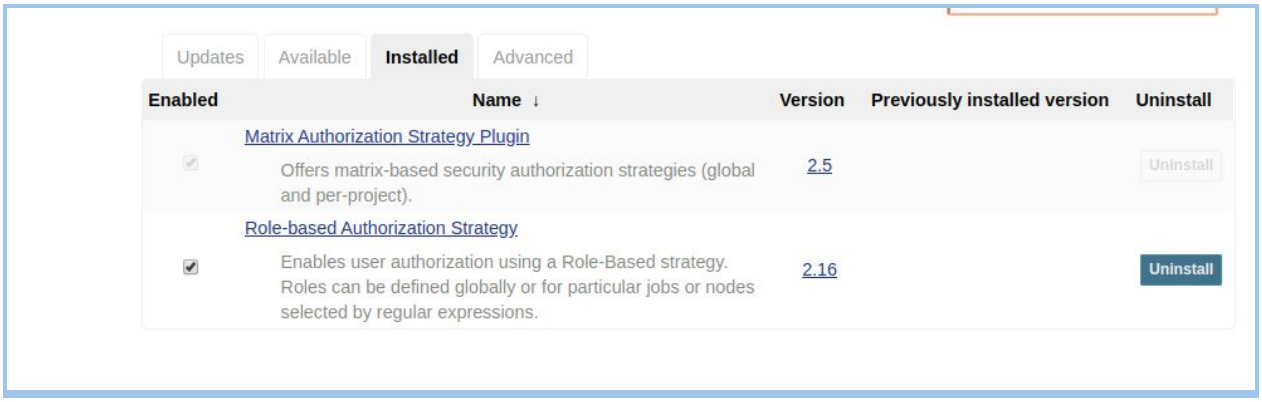
CI/CD : Jenkins

Trainee Name : Chhavi Sharma
Newers ID : 4023
College : UPES

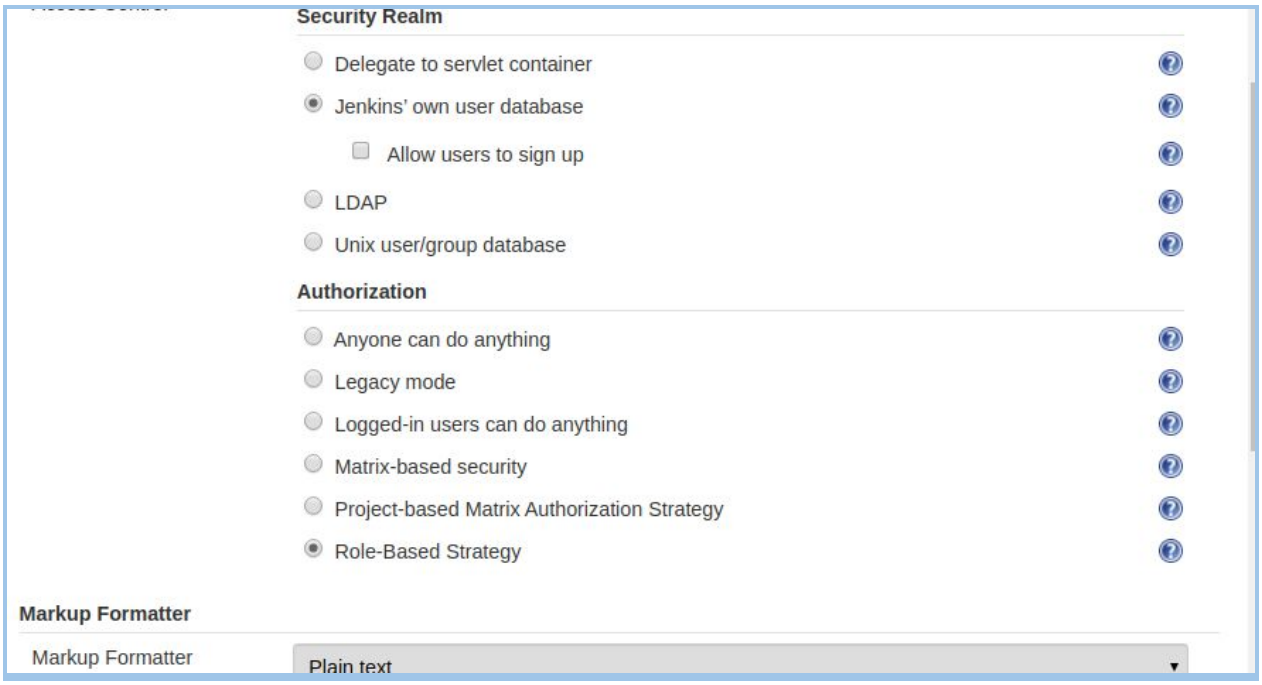
1. Create 2 users: developer1, developer2. The developer1 should be able to build job1 only and can't change the job configuration. The developer2 can configure and build the job2, also he is able to view job1 but can't build/configure it.

Ans.

Step 1: Install Role-based Authorization Strategy.



Step 2: In Configure Global security, select role based strategy under Authorization.

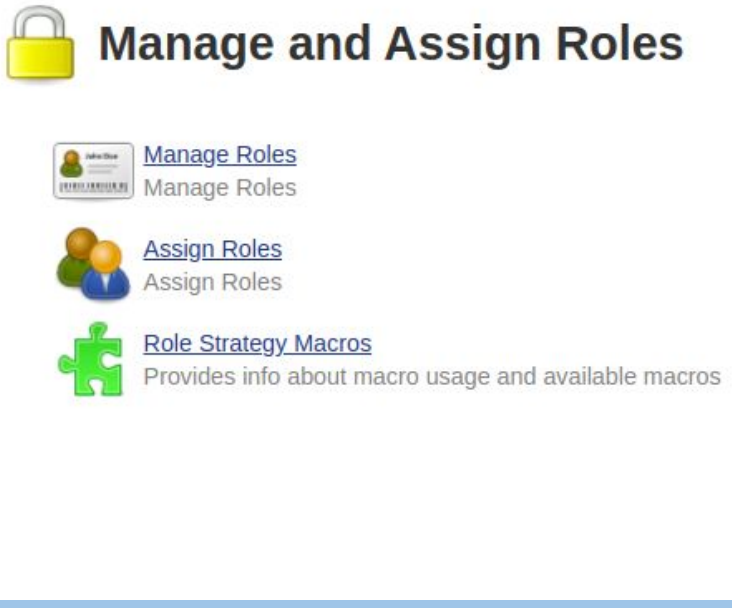


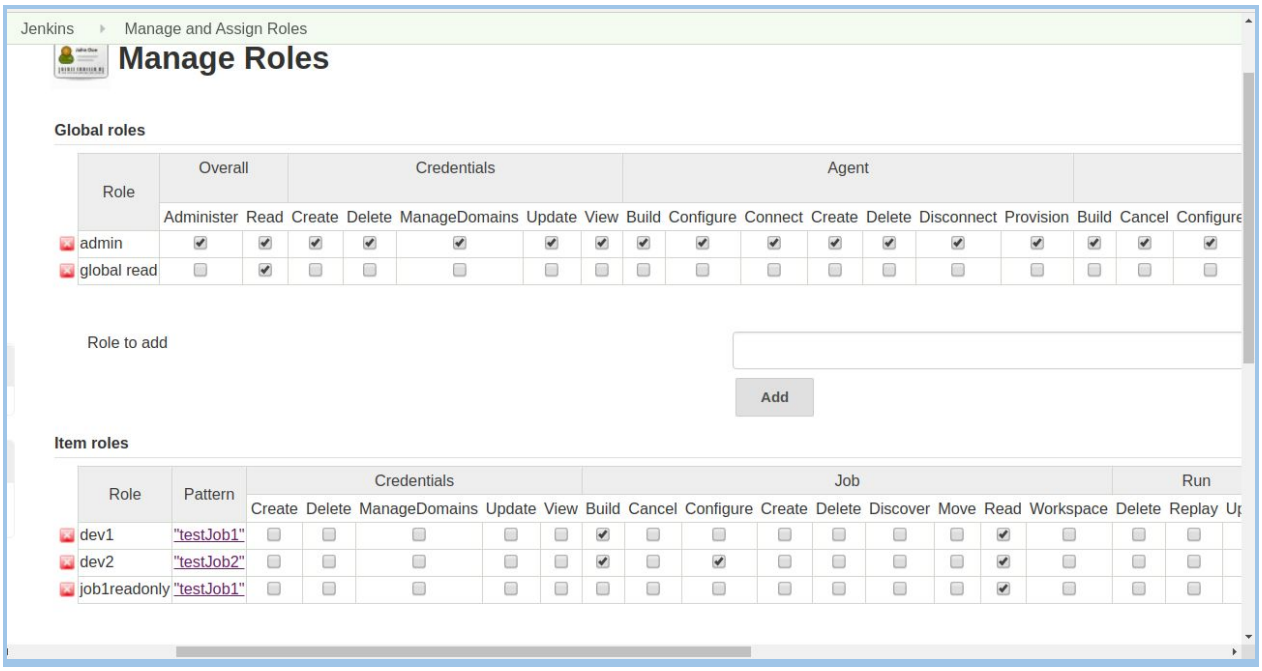
Step 3: Now create two jobs(testjob1, testjob2) and two users(developer1, developer2).

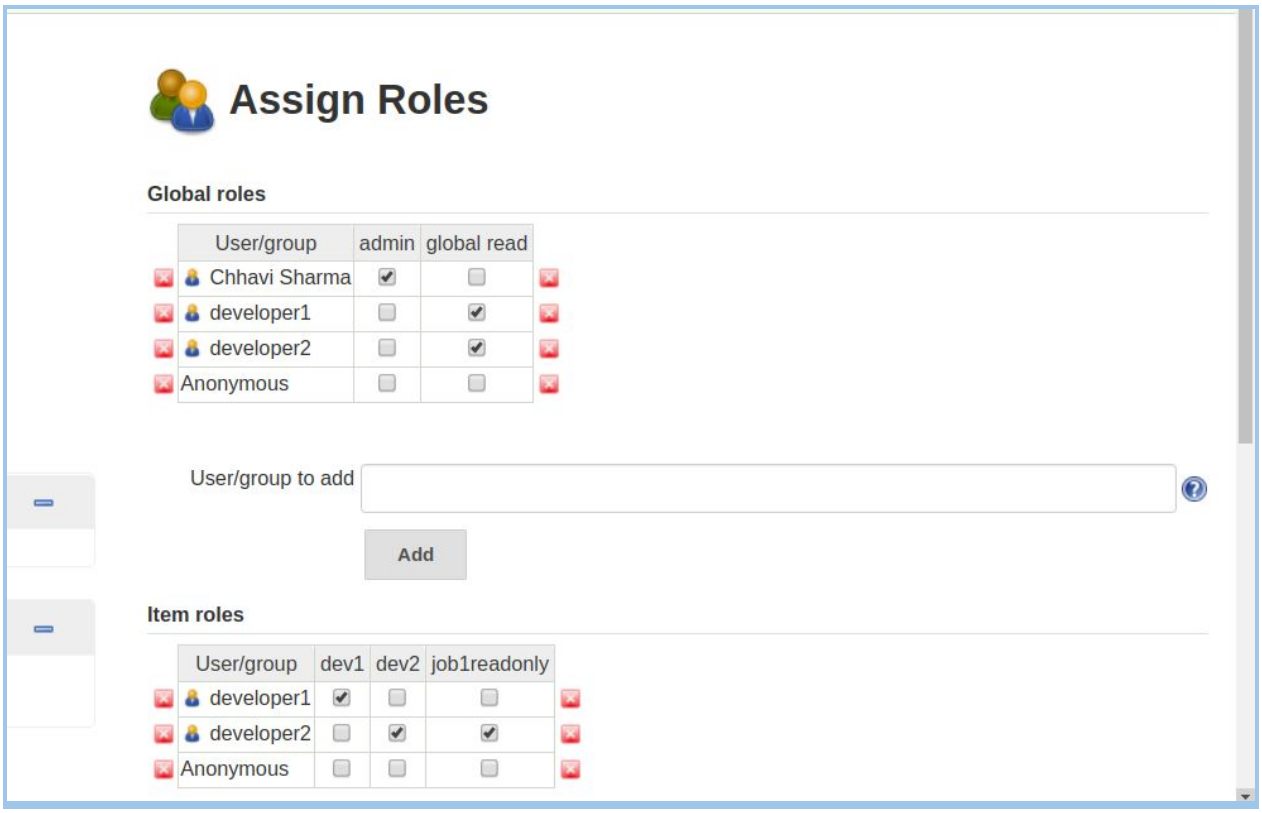
Users

These users can log into Jenkins. This is a sub set of this list, which also contains auto-created users who really just made some commits on some projects and have no direct Jenkins access.

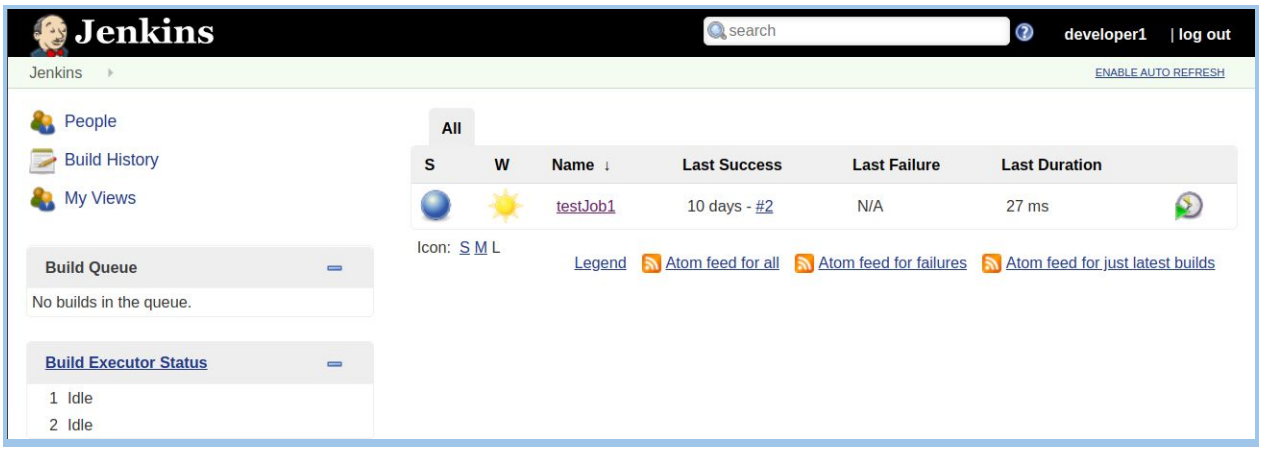
User ID	Name	
chhavi_97	Chhavi Sharma	
developer1	developer1	
developer2	developer2	

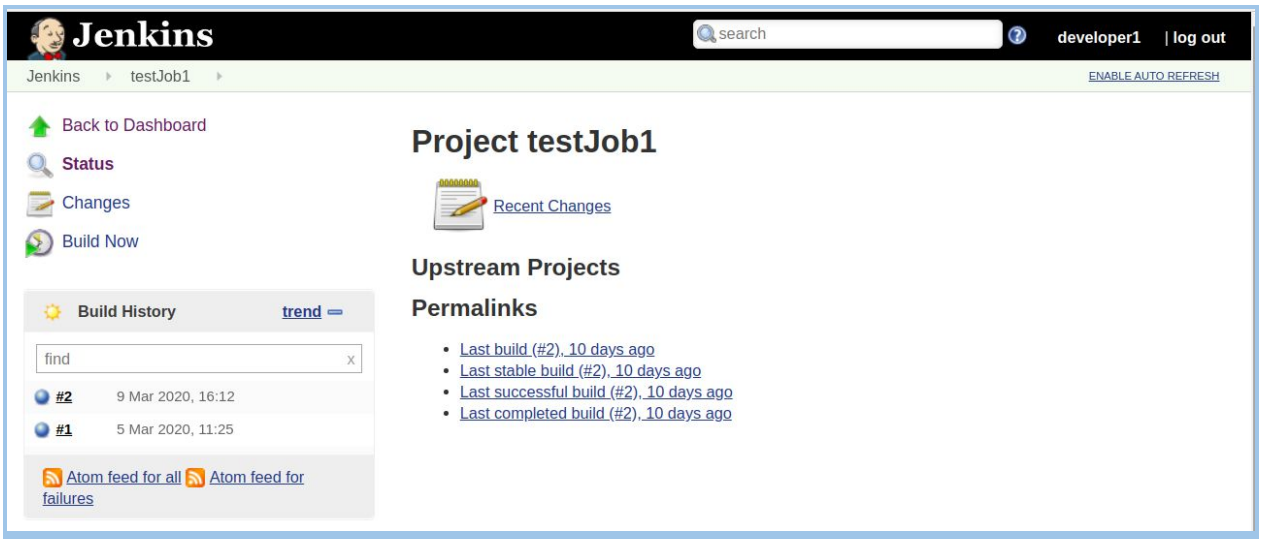




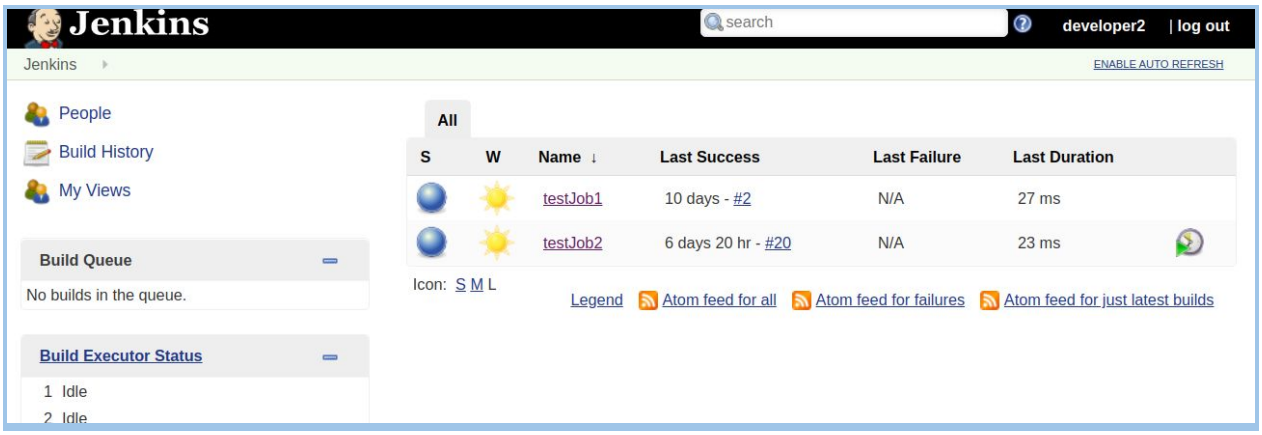


Developer1 console

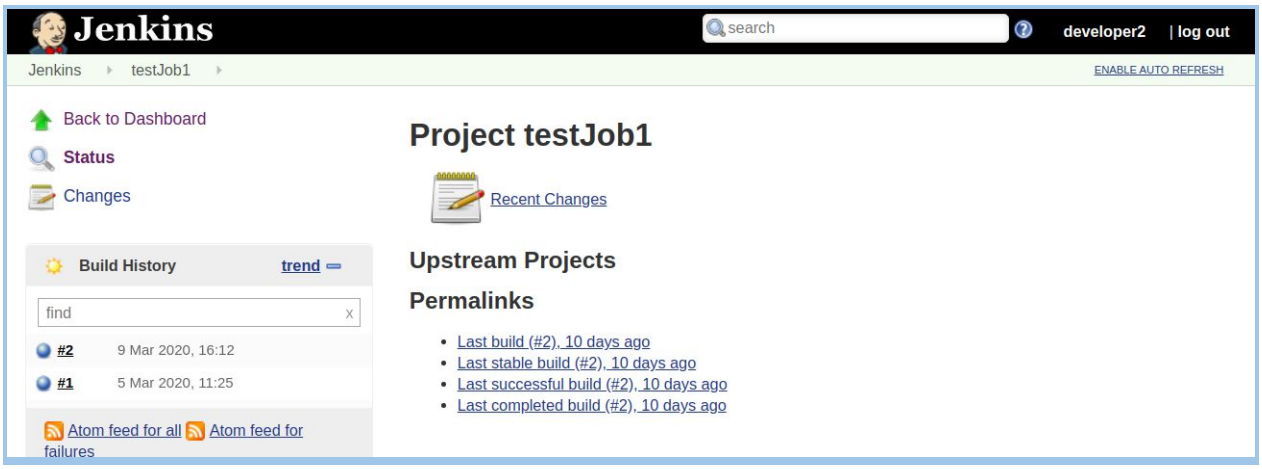




Developer2 console



He can build job 2 but not job 1.

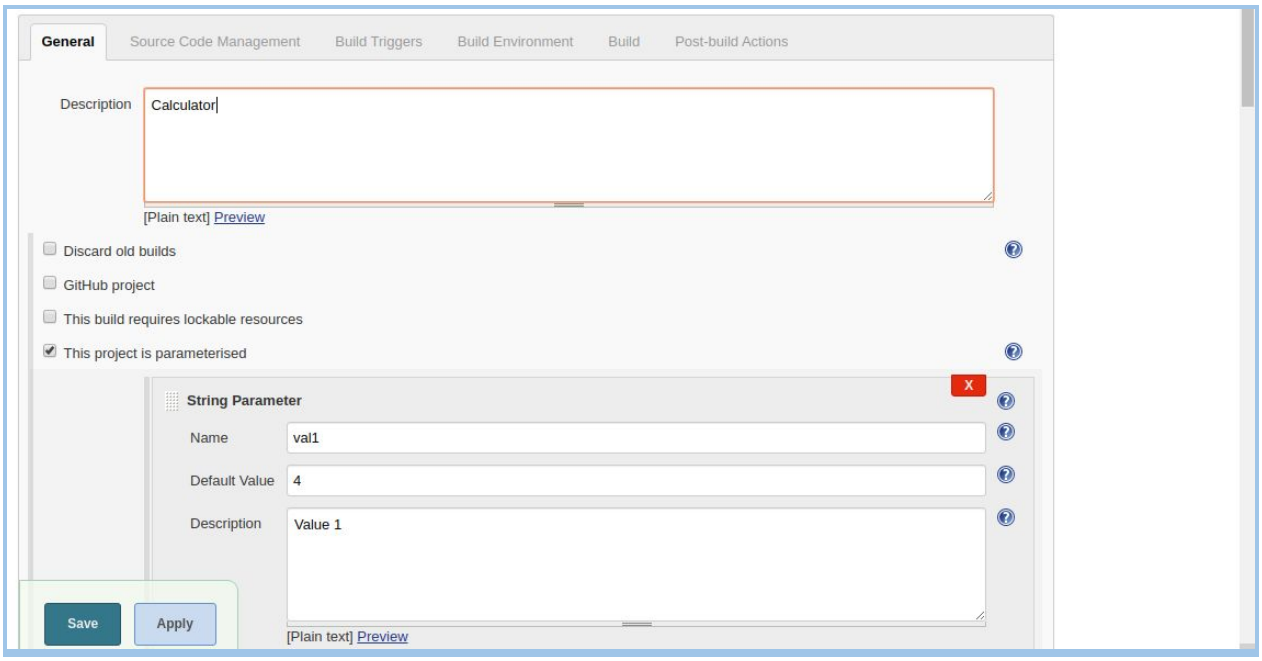


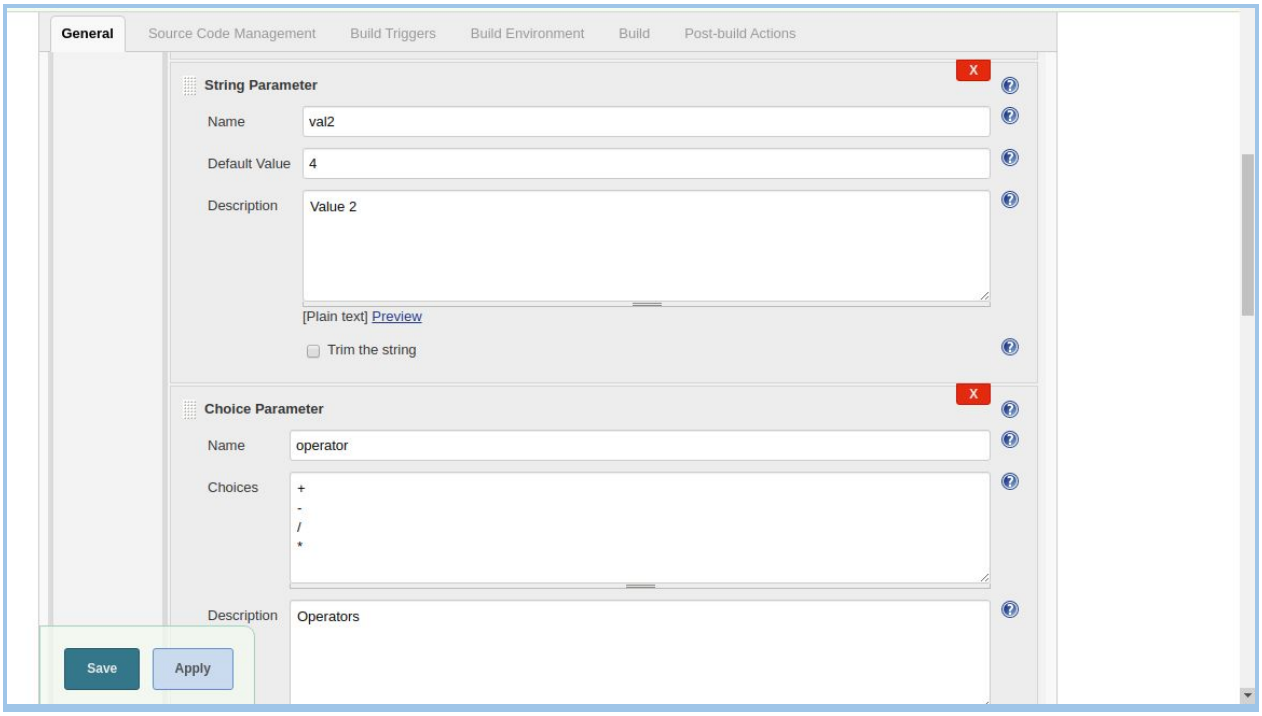
2. Create a Jenkins Job to create a calculator. It should give you a dropdown to ask Addition, Subtraction, Multiplication or Division and email the results.

Hint: Plugin Used (Environment Injector, Extended Email Notification)

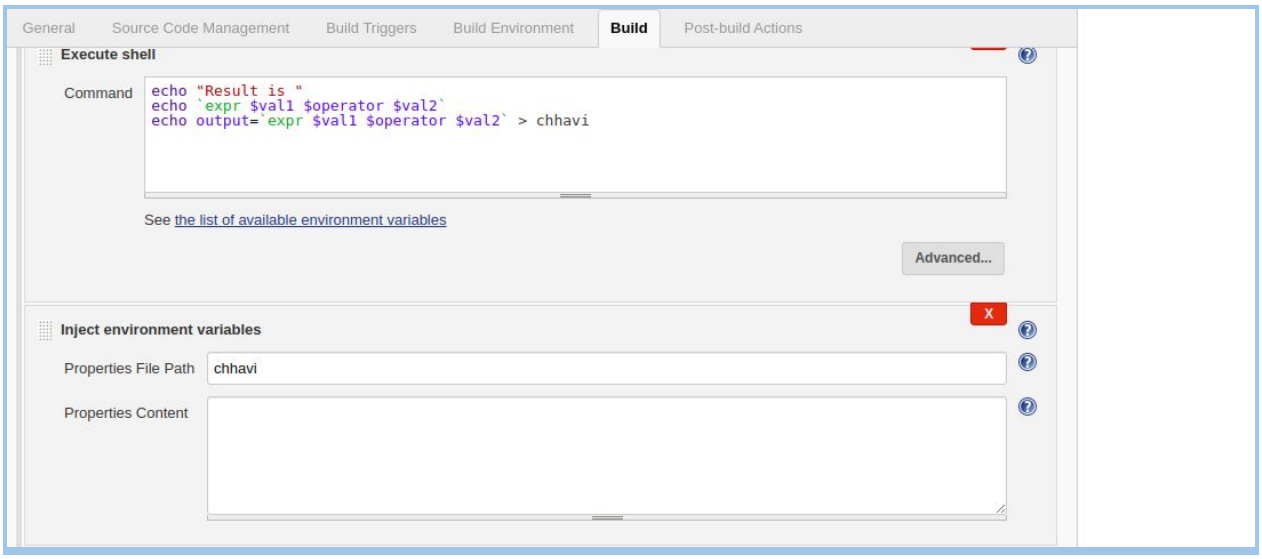
Ans.

Create a new freestyle job.Add two String parameters(to perform operations) and one choice parameter for choosing the type of operation.

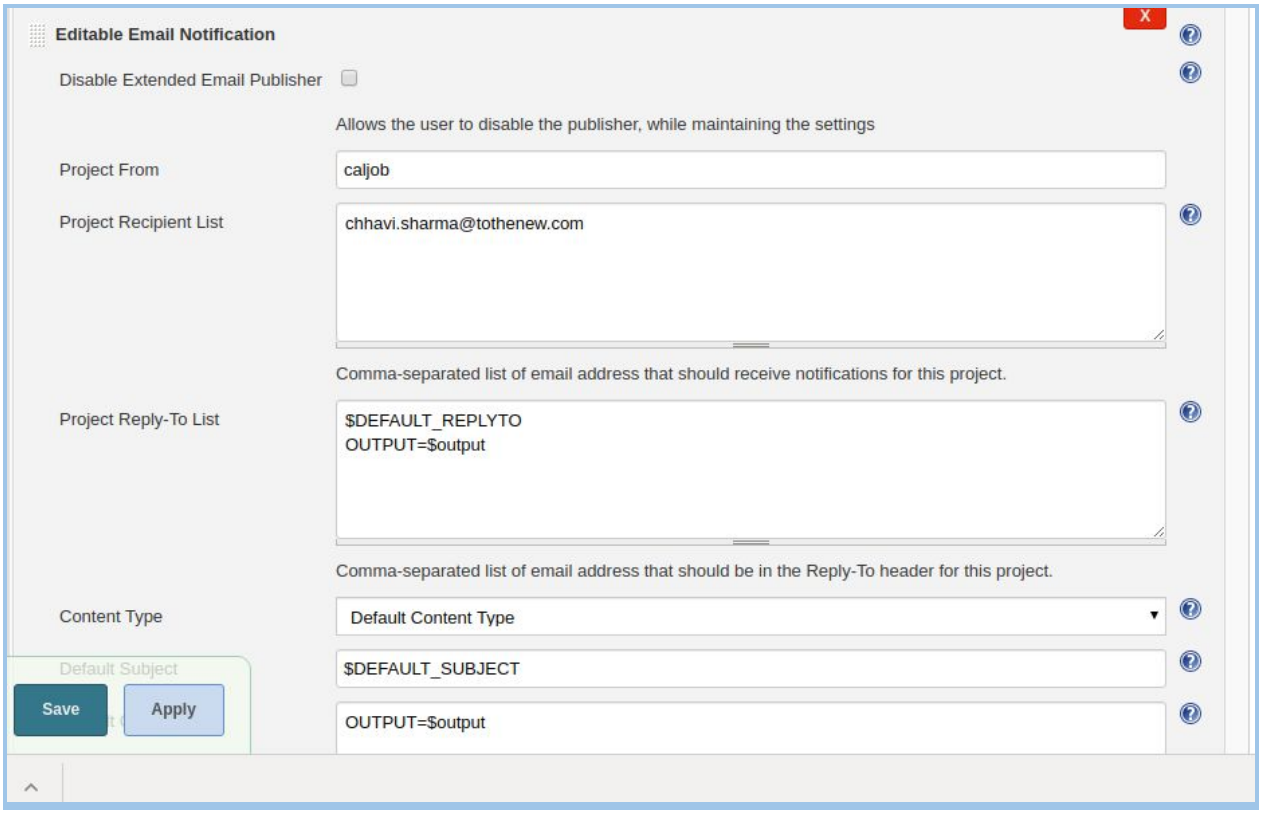




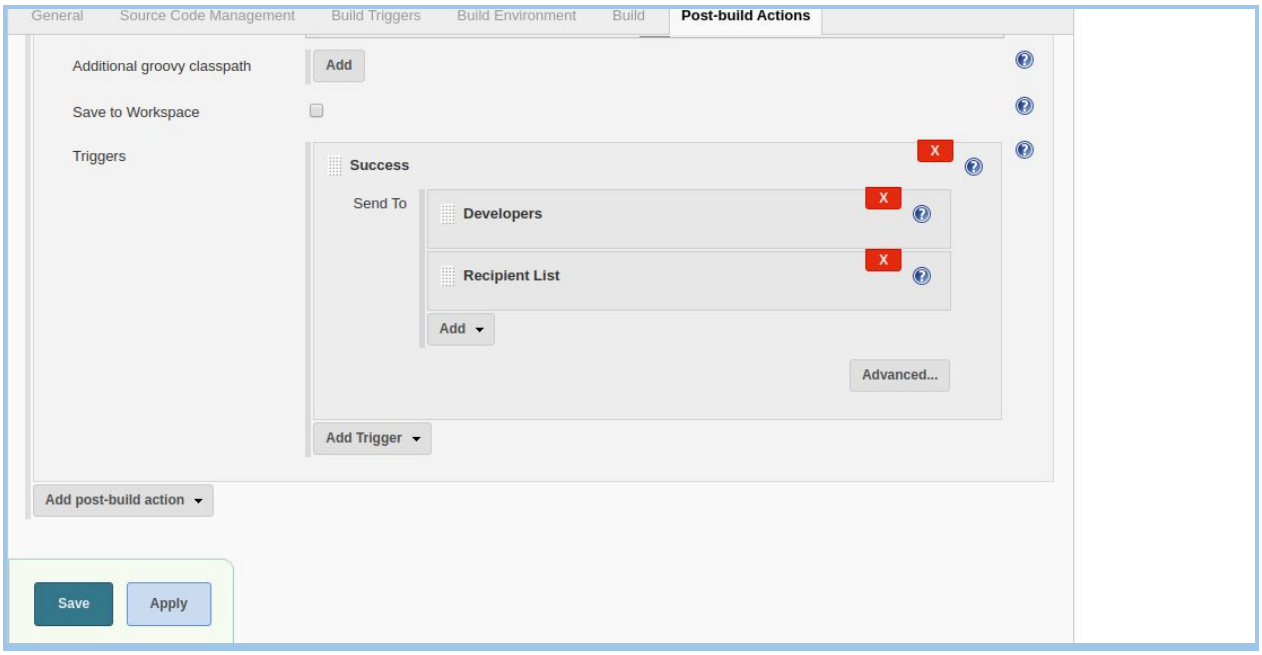
In Execute shell use the above defined parameters to compute the output.



Configure the project to send email at every build.
Click “Add post-build action”.
Click Editable Email Notification
Click Advanced Settings

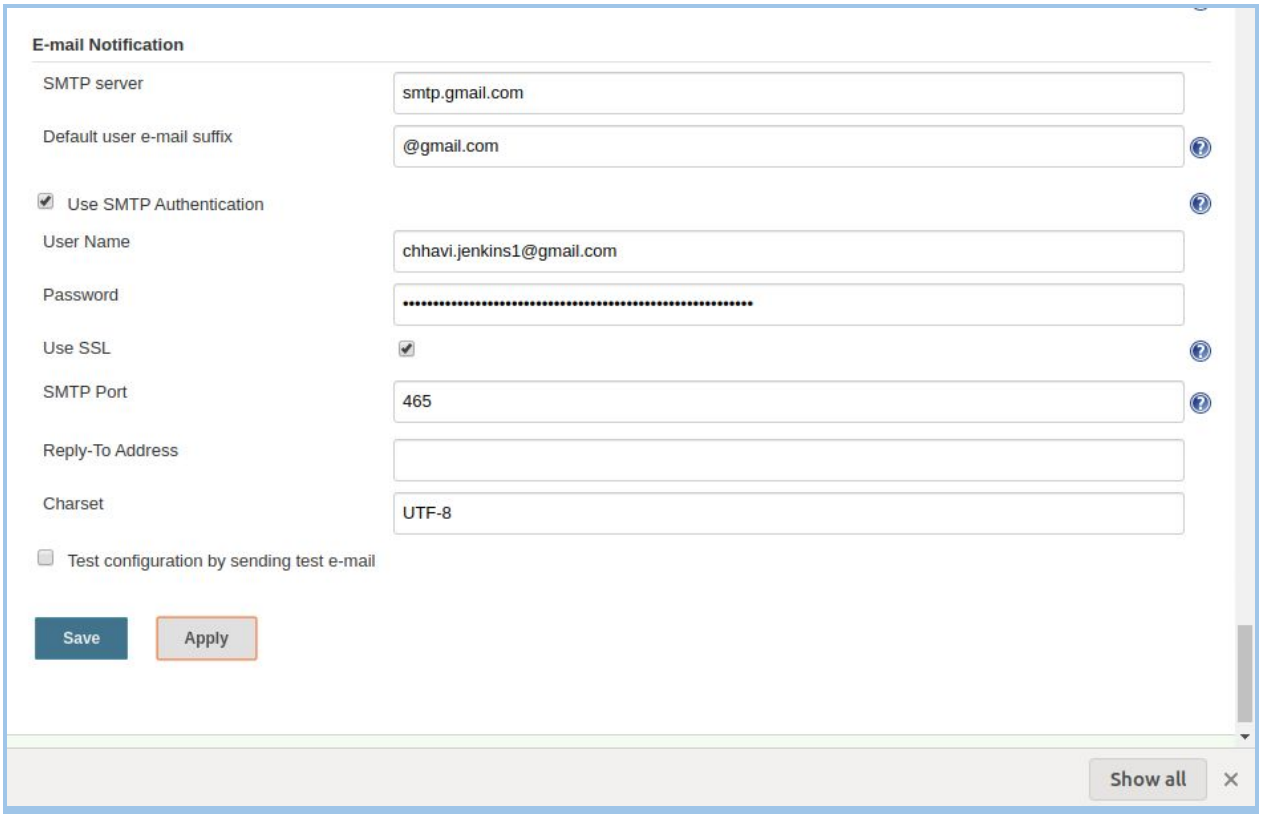


Click Add Trigger
Click “Always.”

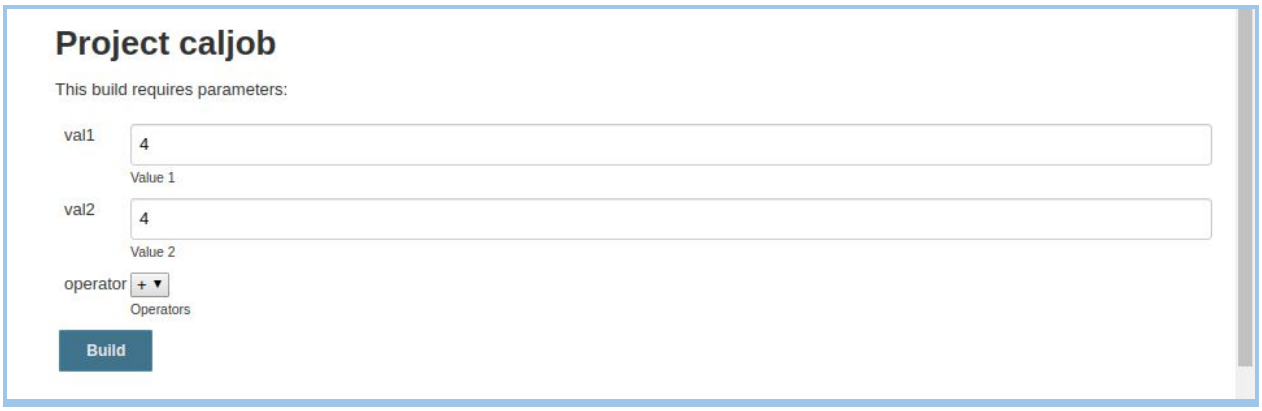


Goto Manage Jenkins -> Configure Systems

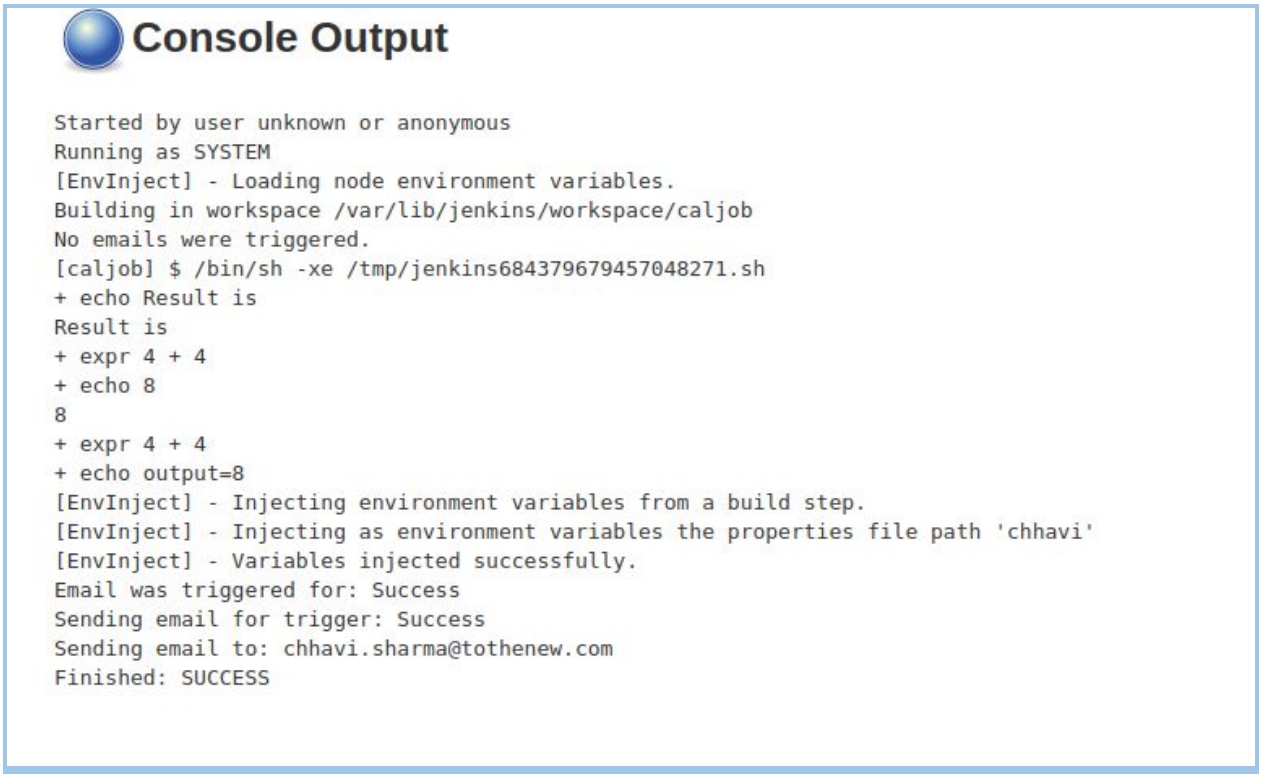
Email Notification section. Here enter the required information. Apply and Save



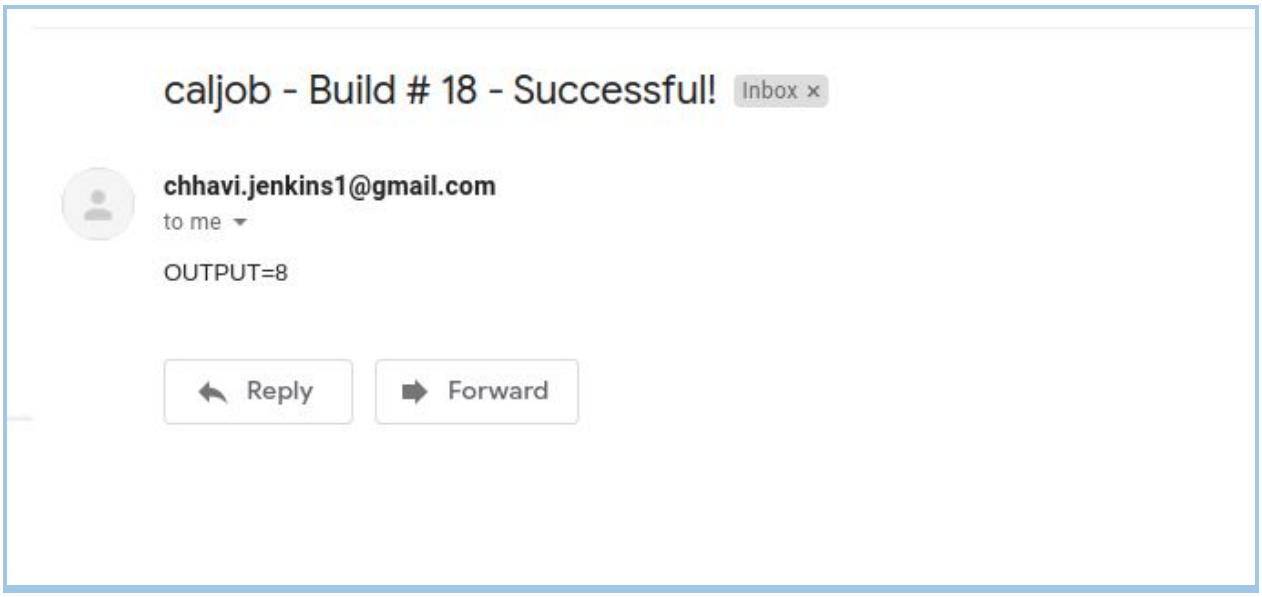
Next provide the parameters and build the job.



Check the console output for result.



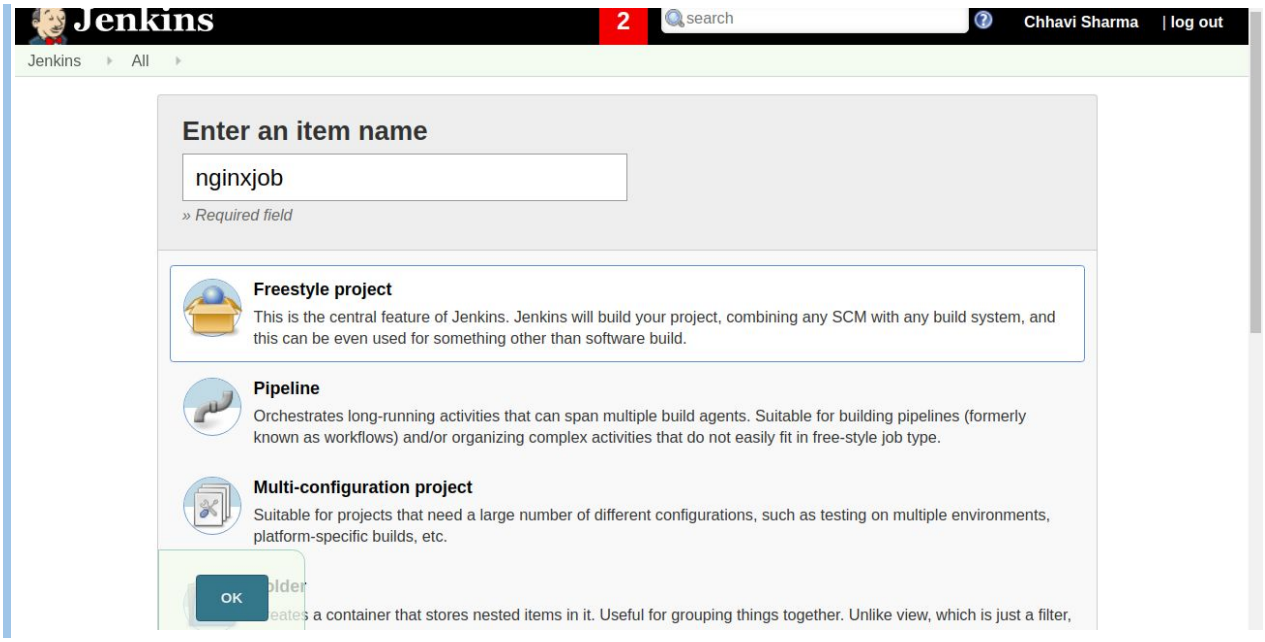
You must have also received the mail on the provided mail id.

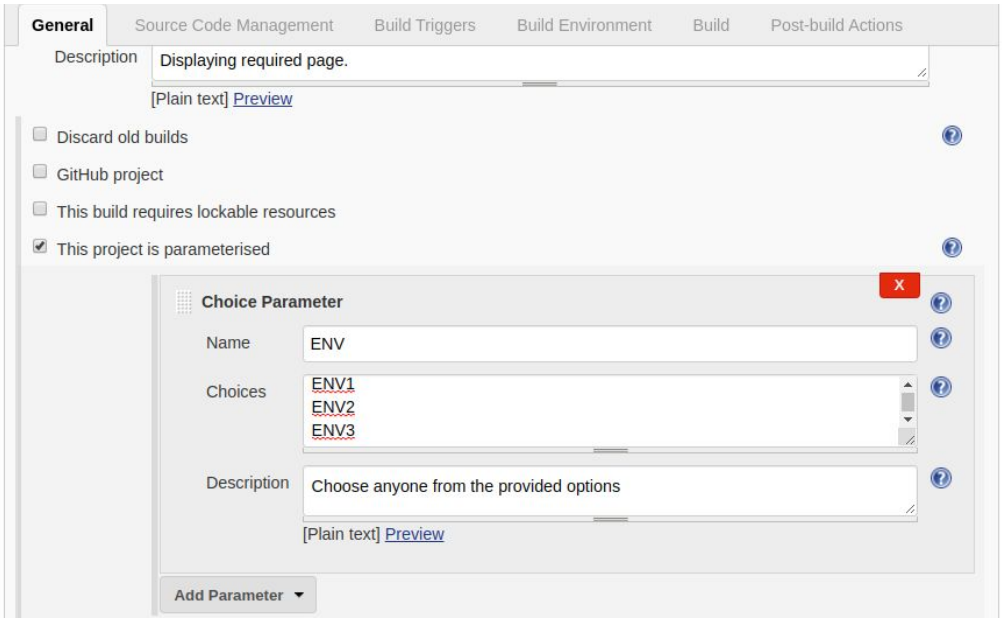


3. Create jenkins parameterized job which on selecting the different Env. will display different web pages by nginx.

Ans.

Step1:Create a new freestyle job. Choose **This project is parameterized** option.And then write the script as below.





Step 2:Now we create a directory ENV in /var/www/html and store our env indexes.

```
chhavi@chhavi:/var/www/html/ENV$ ls
ENV1.html  ENV2.html  ENV3.html
chhavi@chhavi:/var/www/html/ENV$ cat ENV1.html
YOU CHOSE ENV 1
chhavi@chhavi:/var/www/html/ENV$ cat ENV2.html
YOU CHOSE ENV 2
chhavi@chhavi:/var/www/html/ENV$ cat ENV3.html
YOU CHOSE ENV 3
chhavi@chhavi:/var/www/html/ENV$
```

Step 3: Now I created a file ENV.com which has our nginx server block.

```
chhavi@chhavi:/etc/nginx/sites-available$ cat nginxenv.com
server{
        listen 80;
        root /var/www/html;
        index ENV1.html ENV2.html ENV3.html;

}
chhavi@chhavi:/etc/nginx/sites-available$
```

Step 4: Edit /etc/hosts

```
chhavi@chhavi:/etc/nginx/sites-available$ cat /etc/hosts
127.0.0.1       localhost  nginxenv.com
127.0.1.1       chhavi

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
chhavi@chhavi:/etc/nginx/sites-available$
```

Step 5: We edit the sudoers file in /etc/ and give all permissions to jenkins with no passwd

```
# User privilege specification
root    ALL=(ALL:ALL) ALL
test    ALL=(ALL) NOPASSWD:ALL
jenkins ALL=(ALL:ALL) NOPASSWD:ALL
# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL
```

Step 6: Check the nginx file and then restart nginx and jenkins.

```
chhavi@chhavi:/etc$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
chhavi@chhavi:/etc$ sudo service nginx reload
```

```
chhavi@chhavi:/etc$ sudo service jenkins restart
chhavi@chhavi:/etc$
```

Step 7: Finally build the project by choosing different parameters and see the result in the browser.

YOU CHOSE ENV 2

Project nginxjob

This build requires parameters:

ENV ENV1 ▾

Choose anyone from the provided options

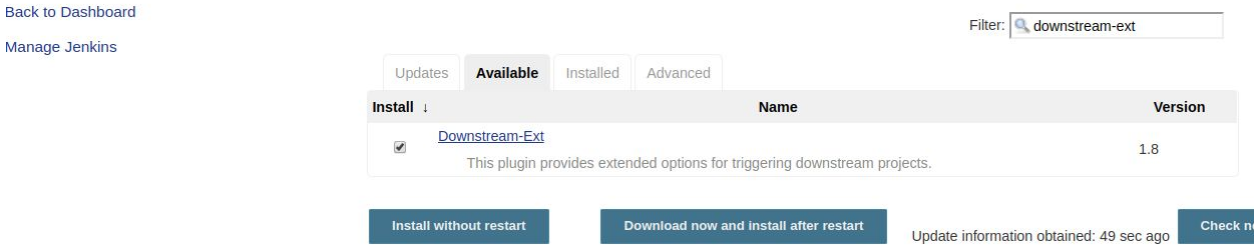
Build

YOU CHOSE ENV 1

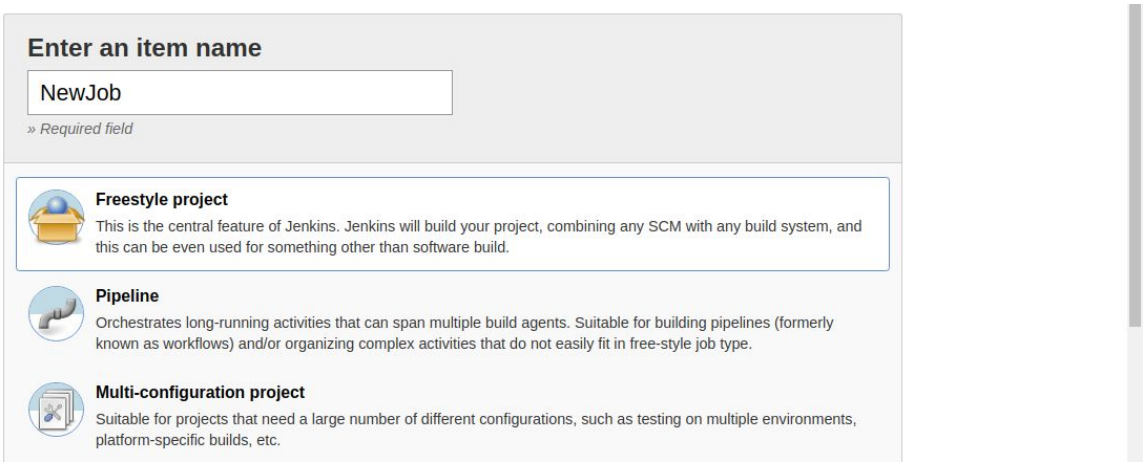
4. Create a job which on its failure will trigger another job.

Ans.

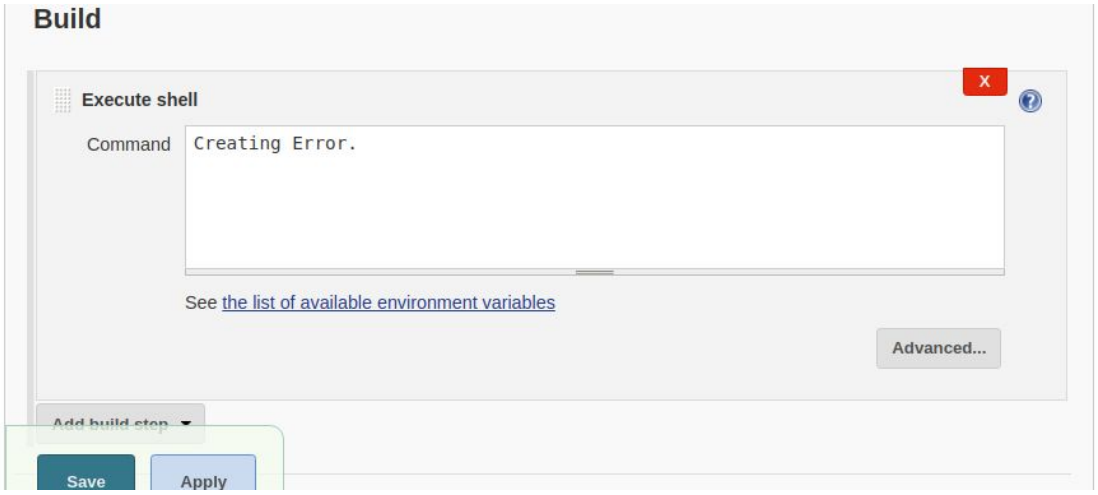
Install Downstream-Ext Plugin .



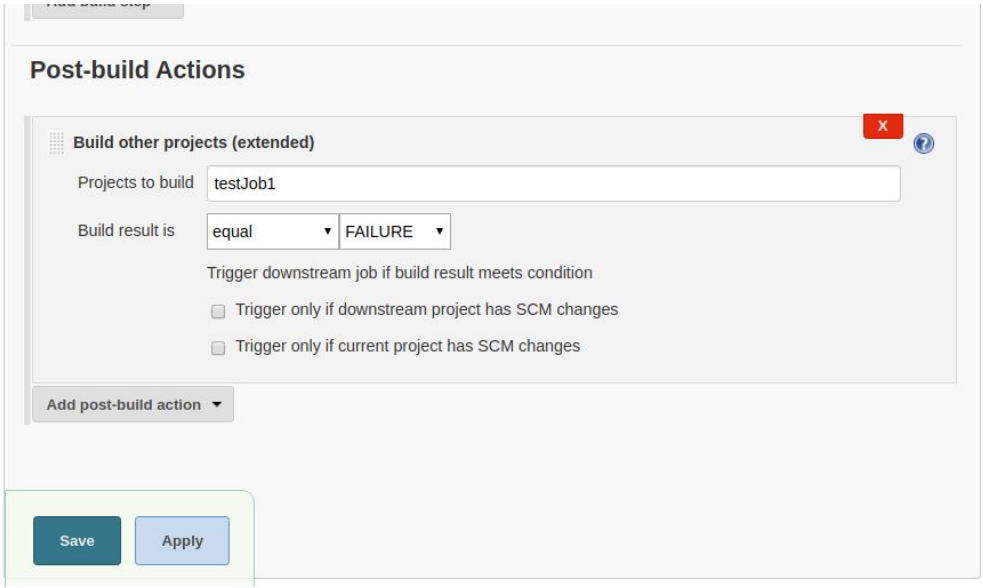
Now create a new job (upstream project).



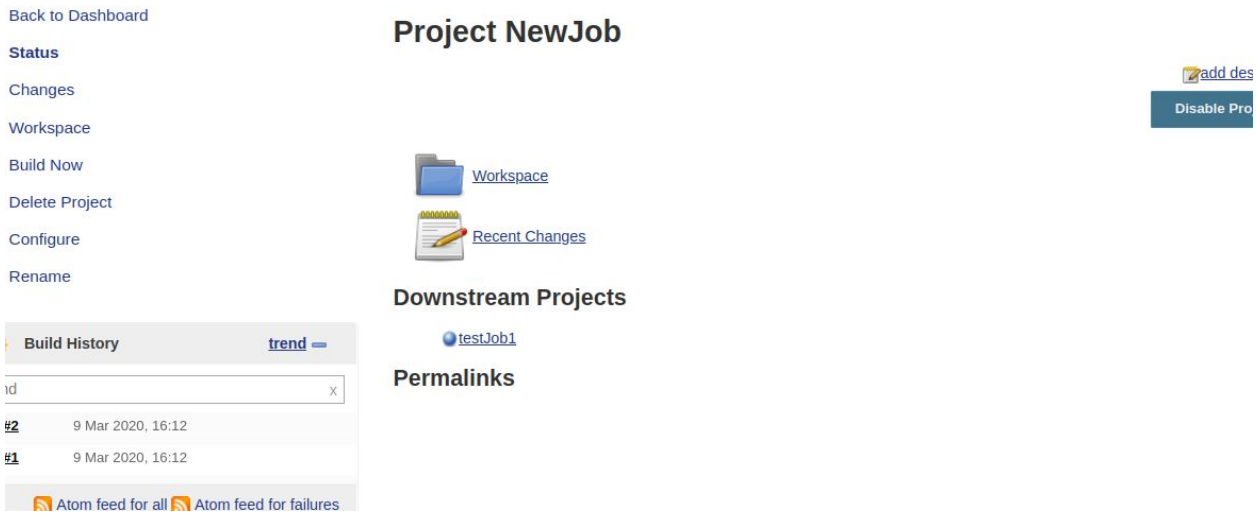
In the Build Action , Write a script with error.

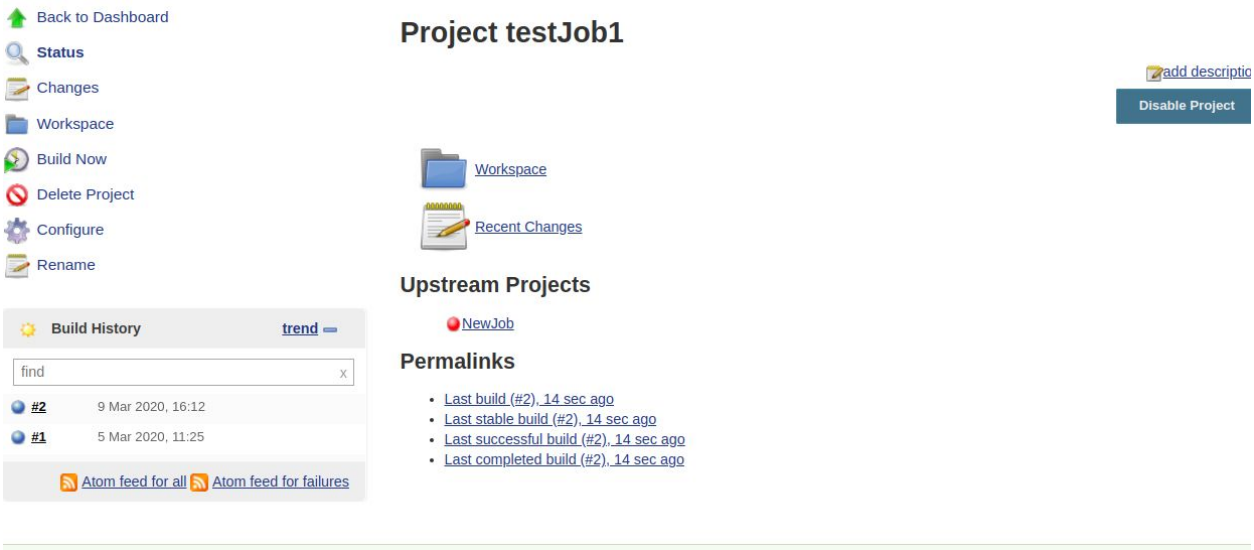


In the post build actions choose Build other projects.
In projects to build choose any other job that you want to build in case the current job fails.



Now build the new job. You can see the results.





5. Create a job which can set a cron job on another server. This server contains a script on its home directory and the script will print the two string parameters which will be given by the jenkins job.

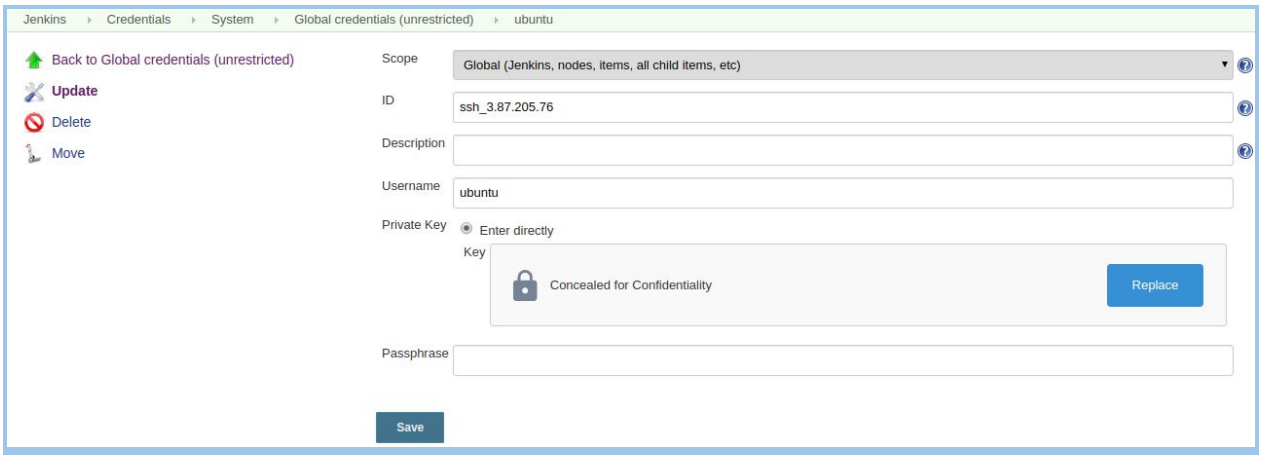
Ans.

Step 1: create a script on remote machine and make it executable.

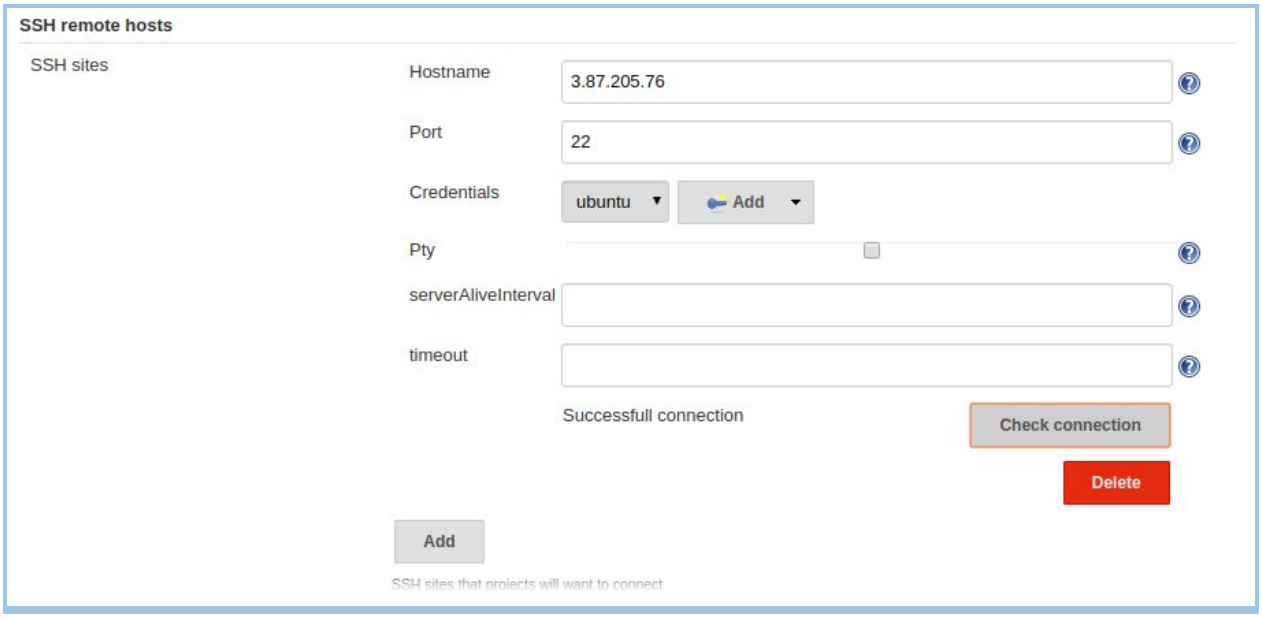
```
ubuntu@ip-172-31-179-174:~$ vim cronscript.sh
ubuntu@ip-172-31-179-174:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-179-174:~$ ls | grep cron
cronscript.sh
ubuntu@ip-172-31-179-174:~$ cat cronscript.sh
#!/bin/bash
echo $1
echo $2
ubuntu@ip-172-31-179-174:~$
```

```
ubuntu@ip-172-31-179-174:~$ chmod u+x  cronscript.sh
ubuntu@ip-172-31-179-174:~$ ll | grep cron
-rwxrw-r-- 1 ubuntu ubuntu   28 Mar 24 08:35 cronscript.sh*
ubuntu@ip-172-31-179-174:~$
```

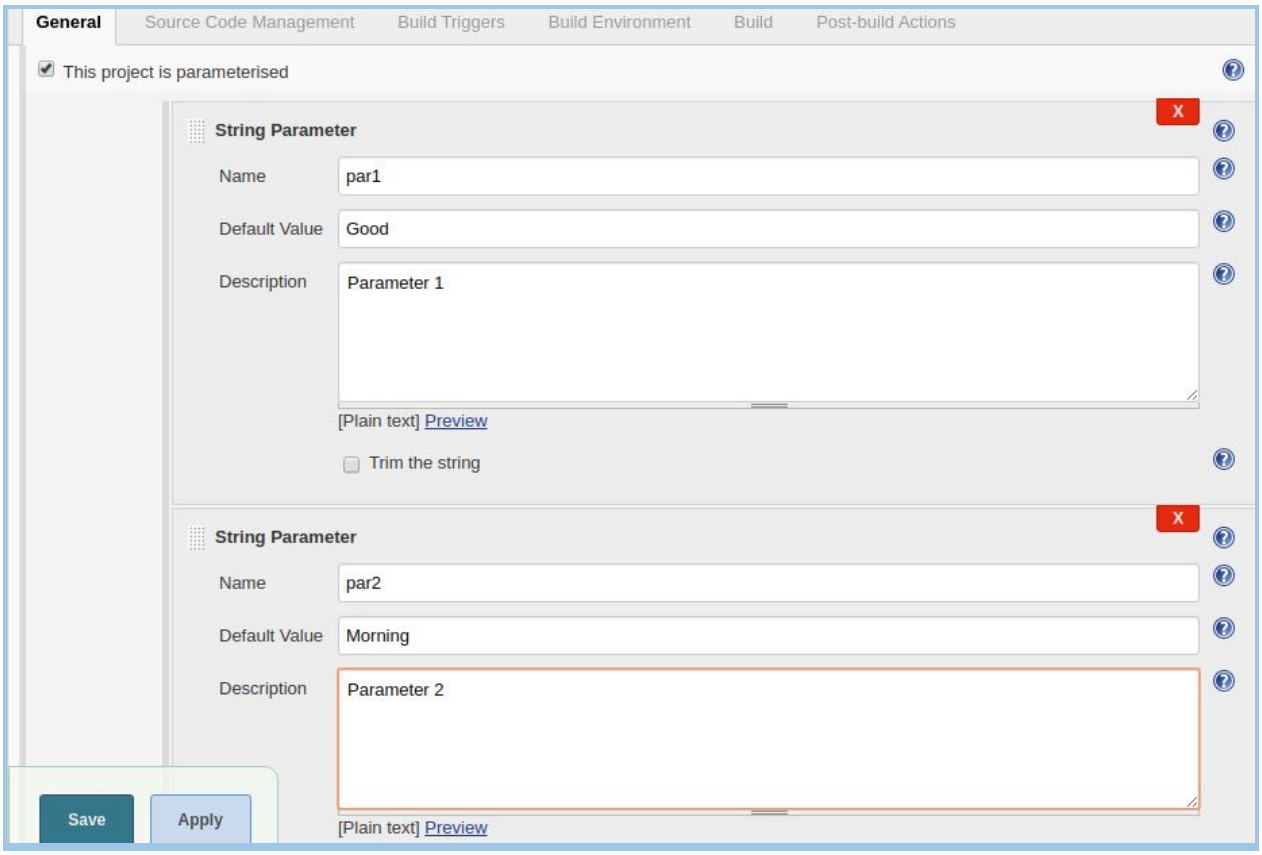
Step 2: Next create global credentials for your remote machine. Provide the necessary privat key.



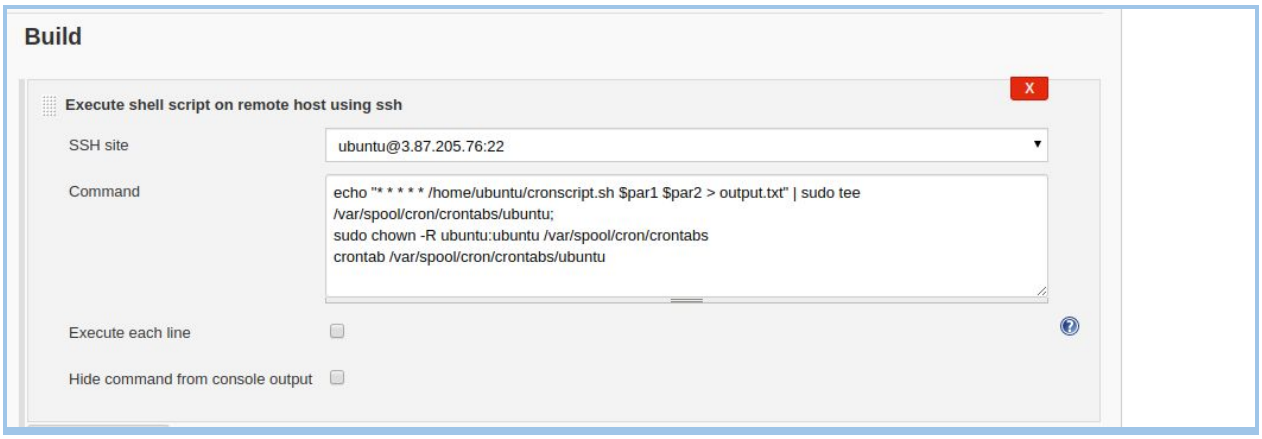
Step 3: Next fill in the required details for SSH remote hosts.Specify port 22 for successful ssh.Check connection.



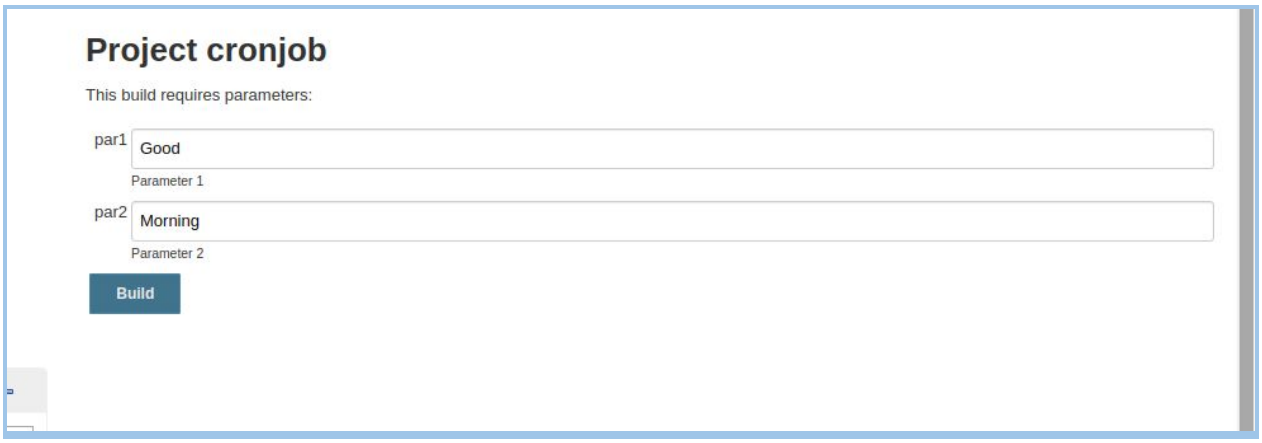
Step 4: Next build a freestyle job.Select “This job is parametrized option.”Add then select string parameters.



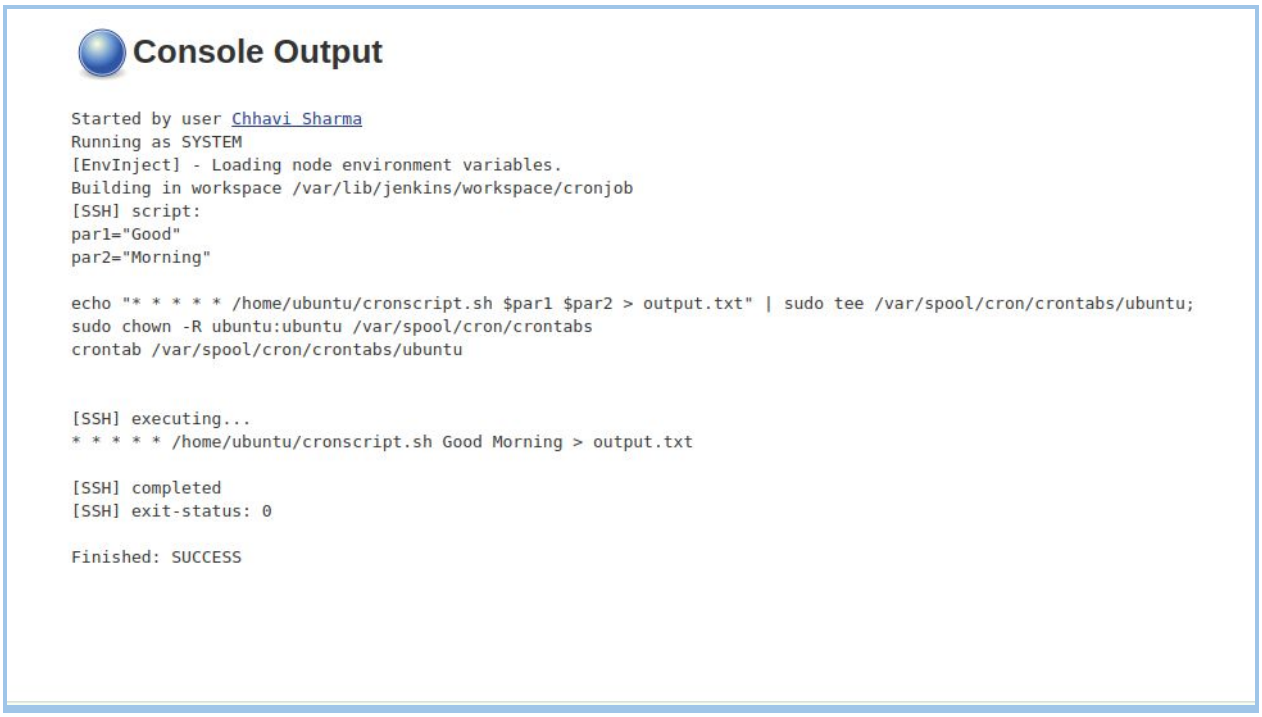
Step 5: In the Build section choose Execute shell script on remote host using ssh. And select the required SSH site (remote).Write an appropriate script as below.



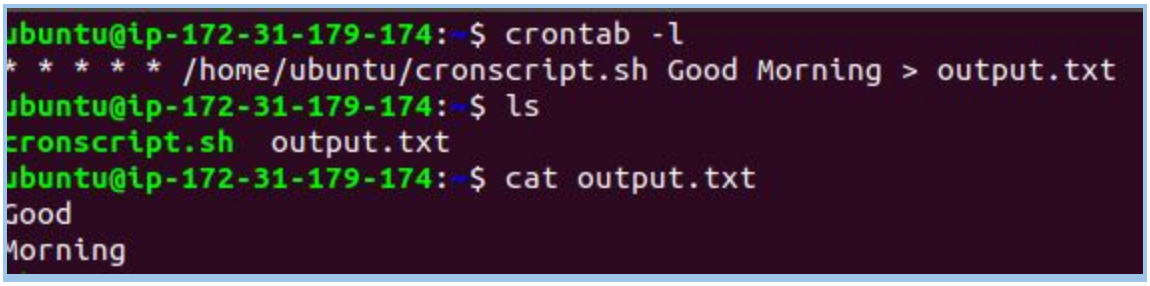
Step 6: Further build the job with the required parameters.



Step 7: Check the console output for any errors.



Step 8: Next ssh into remote to check if the job was successful.

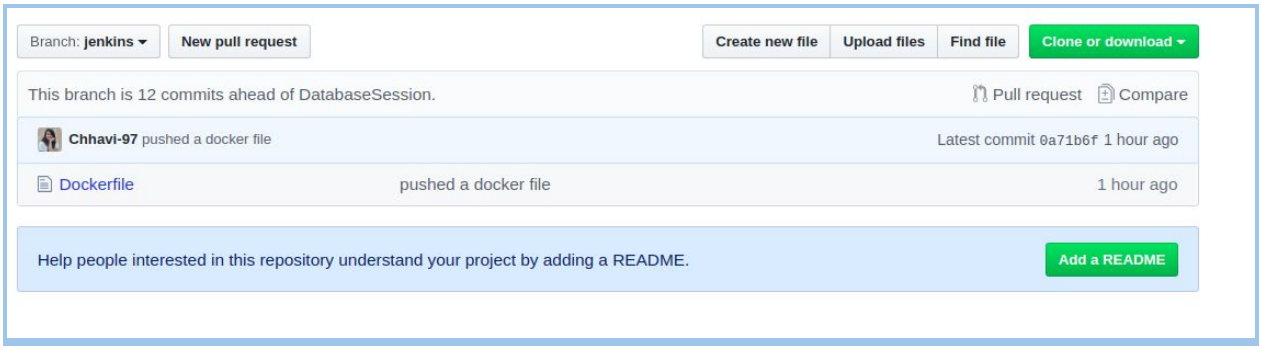


6. Create a job in which: Pull Dockerfile from GitHub, build it and push to Dockerhub. The docker image should have the tag: git commit id.

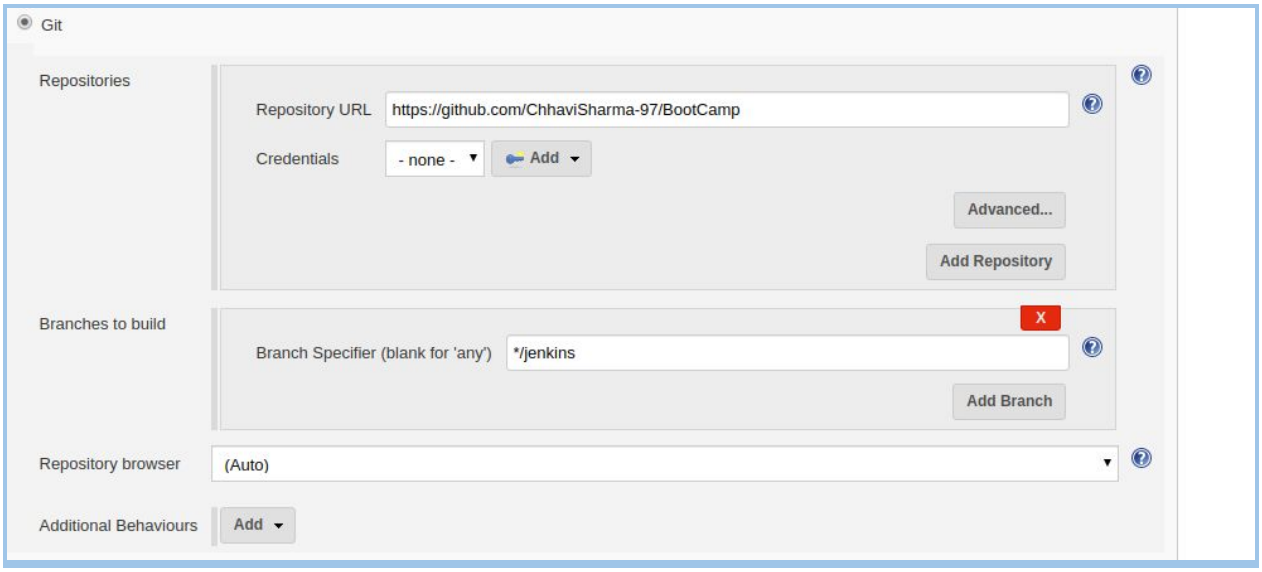
Ans.

VIA SHELL SCRIPT

Prerequisite: Push a dockerfile to your github repo.



Step 1: Create a freestyle job. Add Git URL and specify the path of Dockerfile.

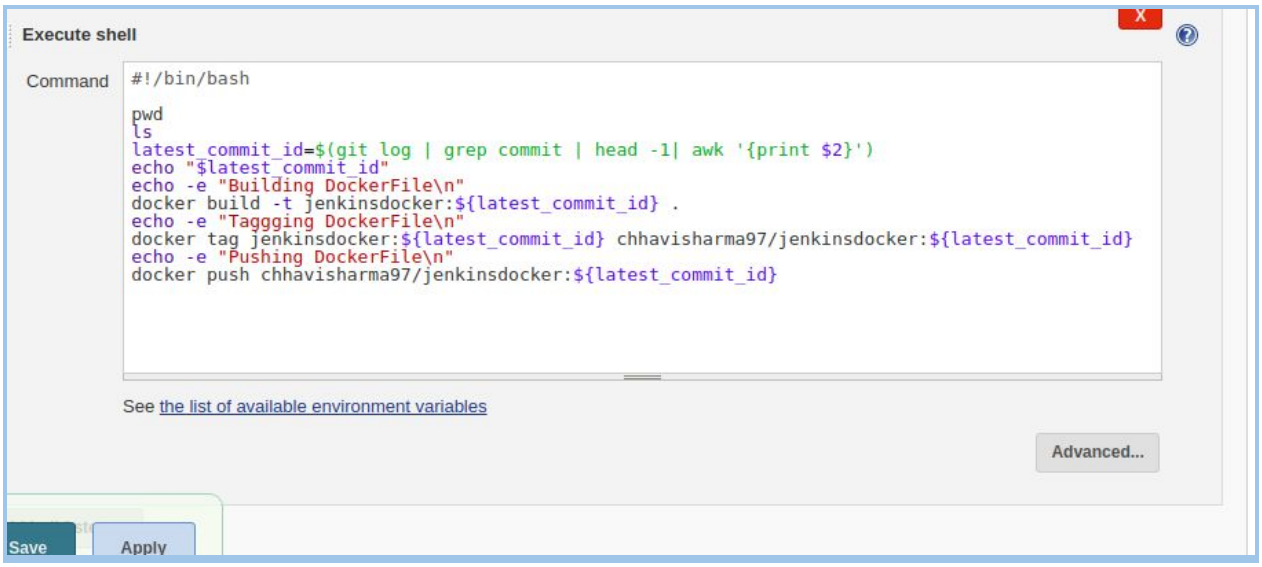


Step 2: Write a script to

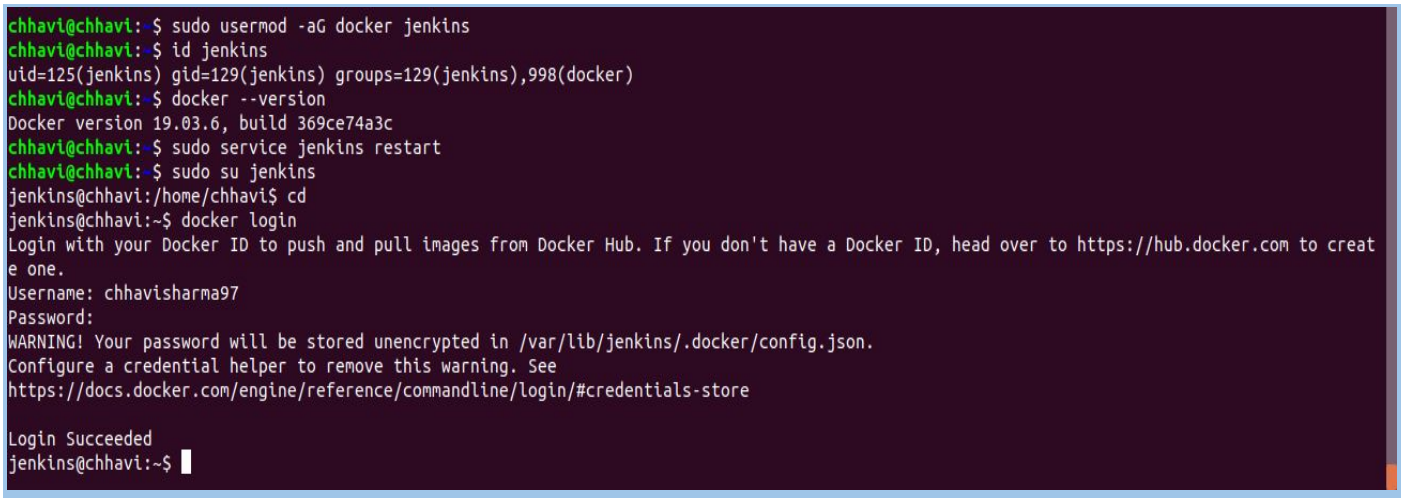
i.grep the latest commit id.

ii.Build a docker file and give a tag to it.

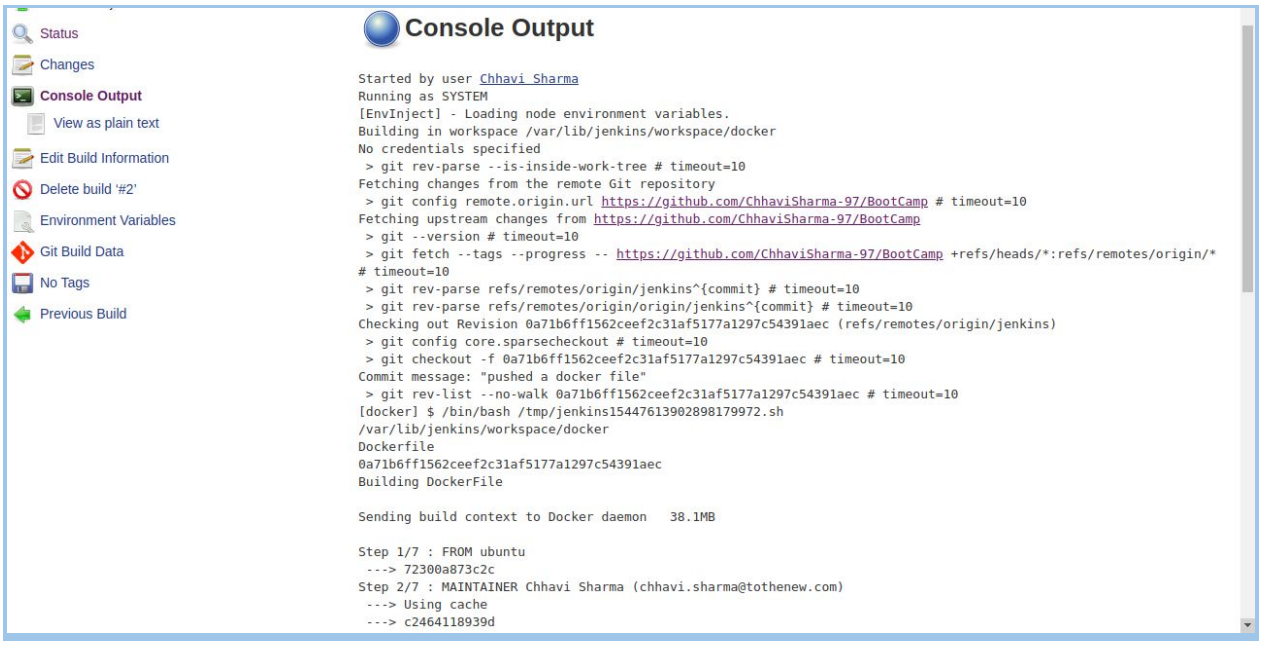
Iii. push the dockerfile to your dockerhub account repo.



Step 3: login into jenkins and then login docker hub.This is done so that the password gets saved in the config.json



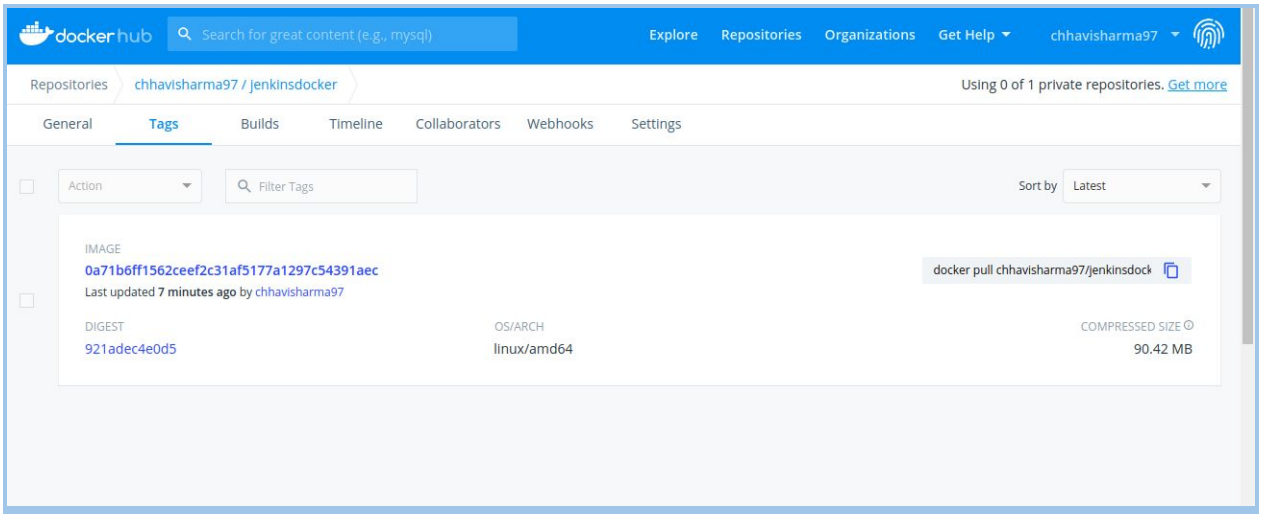
Step 4: Build the project.




```
Sending build context to Docker daemon   38.1MB

Step 1/7 : FROM ubuntu
 ---> 72300a873c2c
Step 2/7 : MAINTAINER Chhavi Sharma (chhavi.sharma@tothenew.com)
 ---> Using cache
 ---> c2464118939d
Step 3/7 : RUN apt-get update
 ---> Using cache
 ---> f42ff05b759f
Step 4/7 : RUN apt-get install vim wget curl net-tools -y
 ---> Using cache
 ---> b2d4cb60a72e
Step 5/7 : RUN apt-get install -y nginx
 ---> Using cache
 ---> 473b454232fe
Step 6/7 : EXPOSE 80
 ---> Using cache
 ---> b7a1894d98c2
Step 7/7 : CMD ["nginx", "-g", "daemon off;"]
 ---> Using cache
 ---> 5c52d7e81bb9
Successfully built 5c52d7e81bb9
Successfully tagged jenkinsdocker:0a71b6ff1562ceef2c31af5177a1297c54391aec
Taggging DockerFile
```

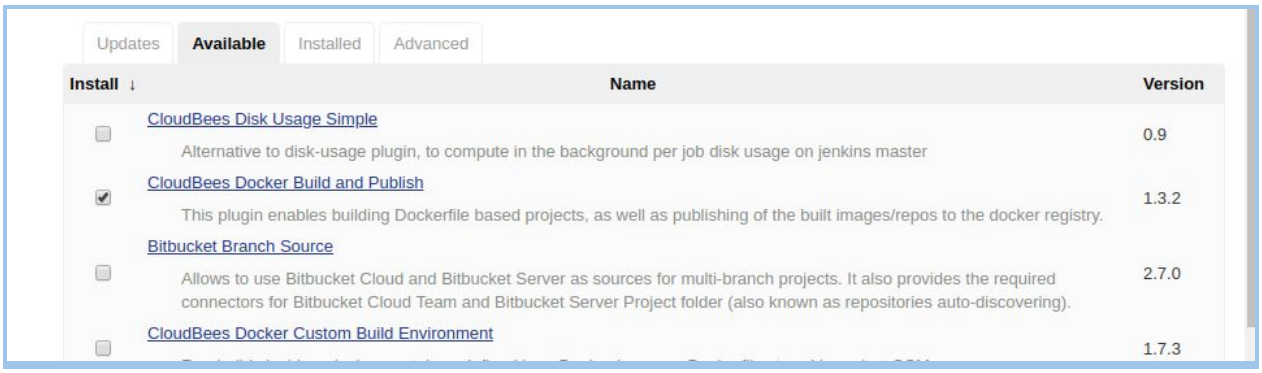
```
Pushing DockerFile

The push refers to repository [docker.io/chhavisharma97/jenkinsdocker]
4f94cd6ccb94: Preparing
40eac20a0958: Preparing
2744aab3a613: Preparing
1852b2300972: Preparing
03c9b9f537a4: Preparing
8c98131d2d1d: Preparing
cc4590d6a718: Preparing
8c98131d2d1d: Waiting
cc4590d6a718: Waiting
2744aab3a613: Mounted from chhavisharma97/commit_id
03c9b9f537a4: Mounted from chhavisharma97/commit_id
1852b2300972: Mounted from chhavisharma97/commit_id
40eac20a0958: Mounted from chhavisharma97/commit_id
4f94cd6ccb94: Mounted from chhavisharma97/commit_id
cc4590d6a718: Mounted from chhavisharma97/commit_id
8c98131d2d1d: Mounted from chhavisharma97/commit_id
0a71b6ff1562ceef2c31af5177a1297c54391aec: digest:
sha256:921adec4e0d59f6053b588b30cb3e291208132430c04c1c92895cce30f63244c size: 1788
Finished: SUCCESS
```

Step 5: You can see your updated repo in dockerhub console.

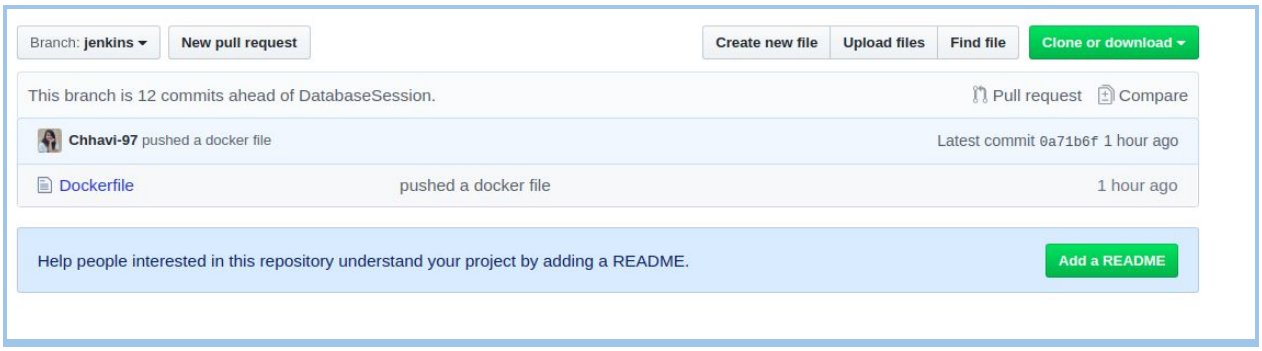


VIA CLOUDBEES

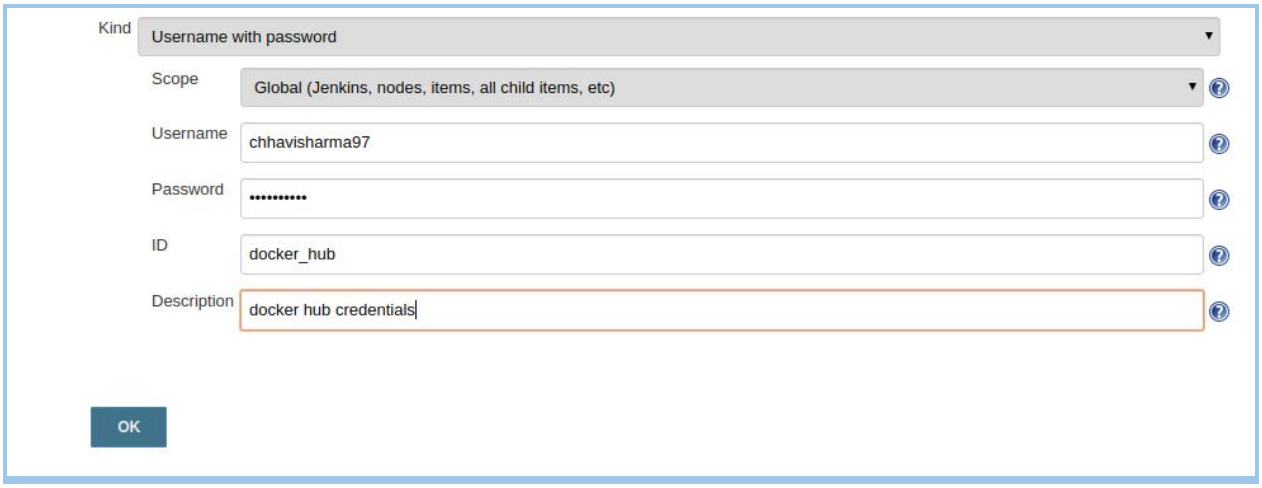
Step 1: Install cloudbees plugin.

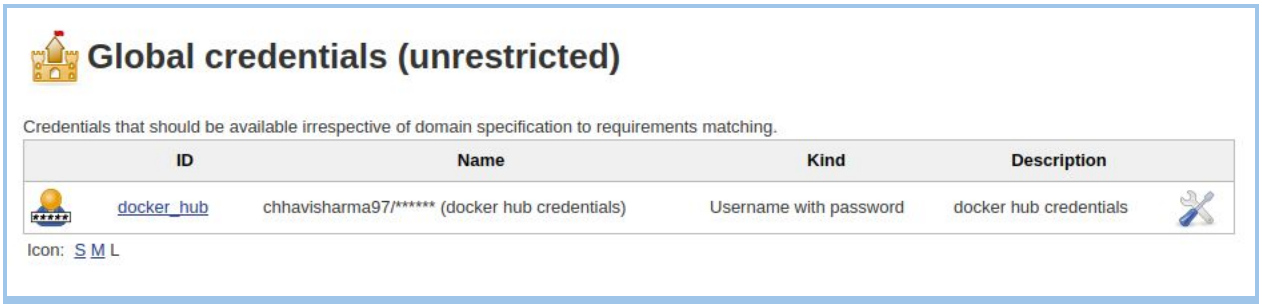


Step 2: Create and push a dockerfile to your github repo.

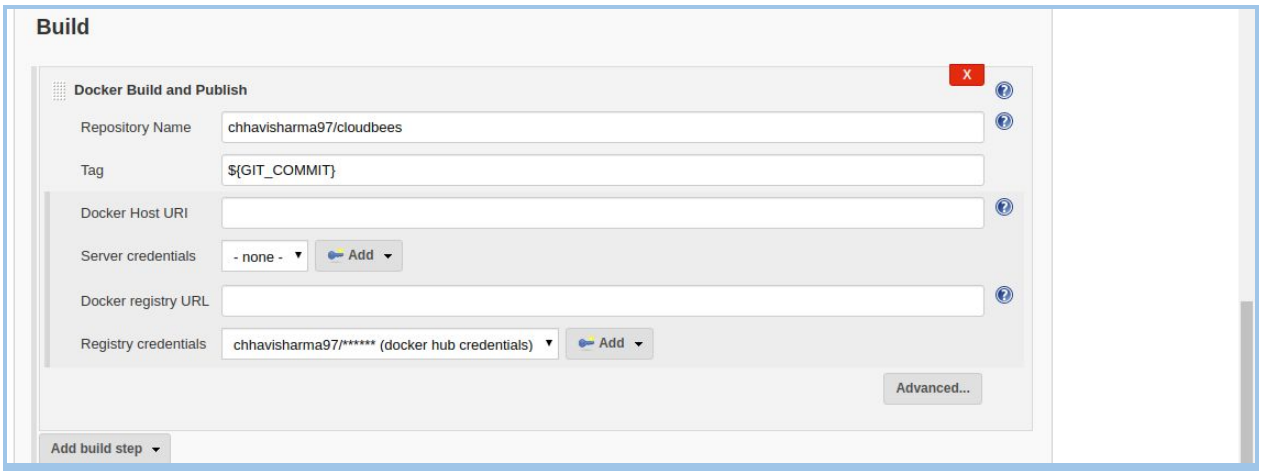


Step 3: Create Dockerhub credentials





Step 4: Add a build step.



Step 5: Add jenkins to docker group. Restart jenkins

```
hhavi@chhavi:~$ sudo usermod -aG docker jenkins
hhavi@chhavi:~$ id jenkins
id=125(jenkins) gid=129(jenkins) groups=129(jenkins),998(docker)
hhavi@chhavi:~$ docker --version
ocker version 19.03.6, build 369ce74a3c
hhavi@chhavi:~$ sudo service jenkins restart
```

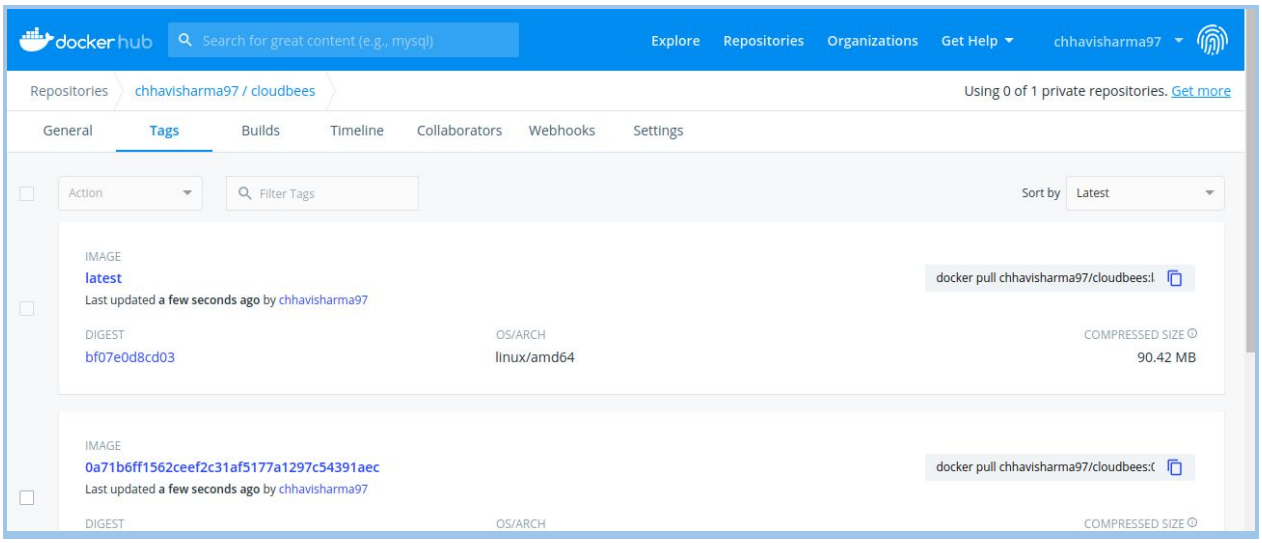
Step 6: Build the project.

Console Output

```
Started by user Chhavi Sharma
Running as SYSTEM
[EnvInject] - Loading node environment variables.
Building in workspace /var/lib/jenkins/workspace/dockerviacloudbees
No credentials specified
 > git rev-parse --is-inside-work-tree # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/ChhaviSharma-97/BootCamp # timeout=10
Fetching upstream changes from https://github.com/ChhaviSharma-97/BootCamp
 > git --version # timeout=10
 > git fetch --tags --progress -- https://github.com/ChhaviSharma-97/BootCamp +refs/heads/*:refs/remotes/origin/*
# timeout=10
 > git rev-parse refs/remotes/origin/jenkins^{commit} # timeout=10
 > git rev-parse refs/remotes/origin/origin/jenkins^{commit} # timeout=10
Checking out Revision 0a71b6ff1562ceef2c31af5177a1297c54391aec (refs/remotes/origin/jenkins)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 0a71b6ff1562ceef2c31af5177a1297c54391aec # timeout=10
Commit message: "pushed a docker file"
 > git rev-list --no-walk 0a71b6ff1562ceef2c31af5177a1297c54391aec # timeout=10
[dockerviacloudbees] $ docker build -t chhavisharma97/cloudbees:0a71b6ff1562ceef2c31af5177a1297c54391aec --
pull=true /var/lib/jenkins/workspace/dockerviacloudbees
Sending build context to Docker daemon   38.1MB

Step 1/7 : FROM ubuntu
latest: Pulling from library/ubuntu
5bed26d33875: Pulling fs layer
f11b29a9c730: Pulling fs layer
930bda195c84: Pulling fs layer
```

```
c8De1b8f4d60: Layer already exists
977183d4e999: Layer already exists
latest: digest: sha256:bf07e0d8cd0341b6c0ea754c68de11f3cced970ee96ddcc0c65b5840098b4d57 size: 1788
Finished: SUCCESS
```

Step 7: See the pushed image on docker hub.

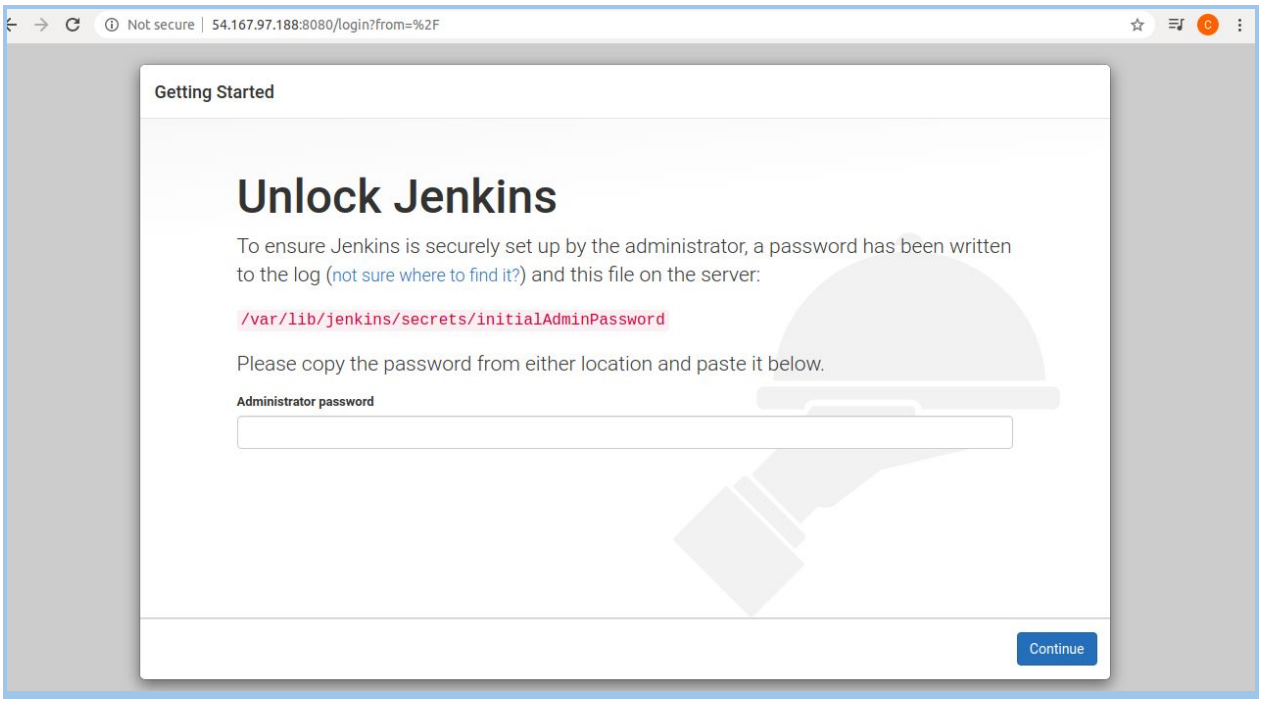


7. Host a static website on s3.Its static content should be in git repo. When a person commits any change in the repo, the job should automatically reflect the changes in the s3 website

Ans.

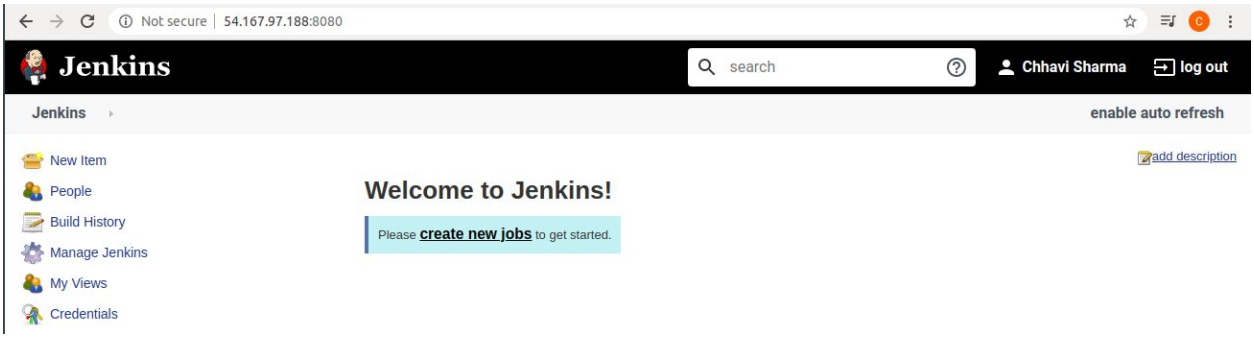
Step 1: Install jenkins on ec2 instance. Also install awscli and configure your aws.

```
ubuntu@ip-172-31-37-88:~$ sudo apt-get install jenkins -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  daemon
The following NEW packages will be installed:
  daemon jenkins
0 upgraded, 2 newly installed, 0 to remove and 70 not upgraded.
Need to get 65.3 MB of archives.
After this operation, 66.6 MB of additional disk space will be used.
```

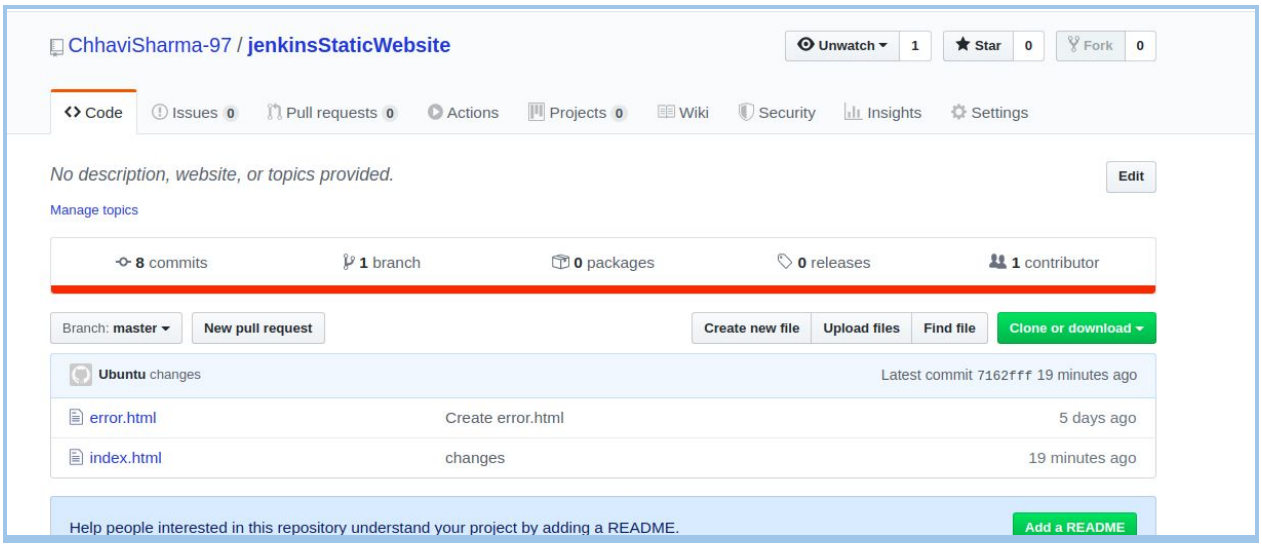


Step 2: Check if jenkins is running.

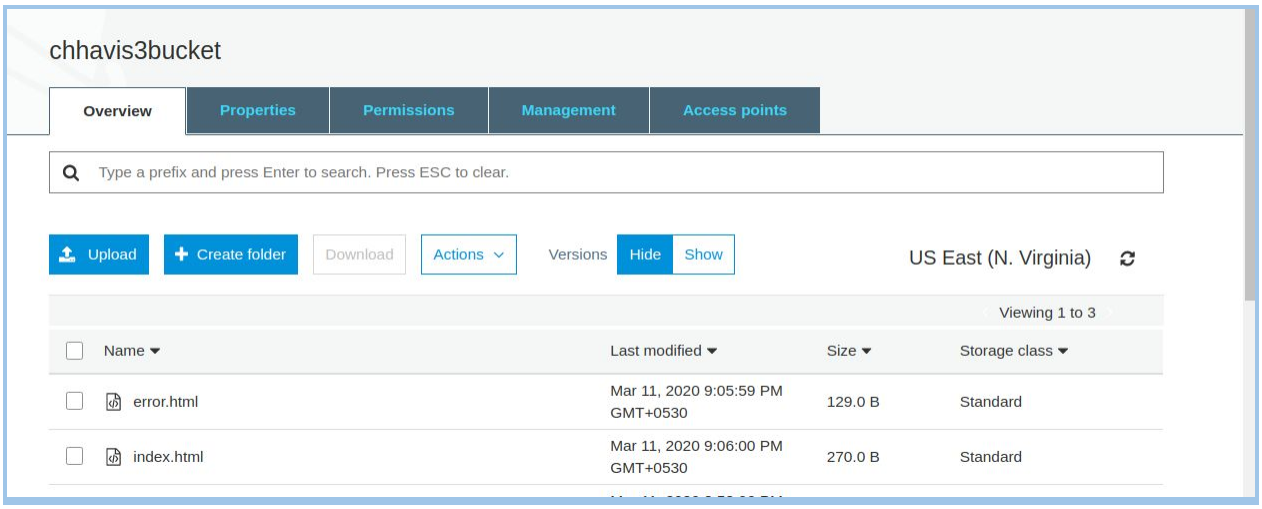
```
ubuntu@ip-172-31-37-88:~$ sudo systemctl status jenkins.service
● jenkins.service - LSB: Start Jenkins at boot time
   Loaded: loaded (/etc/init.d/jenkins; generated)
   Active: active (exited) since Mon 2020-04-06 12:18:03 UTC; 6min ago
     Docs: man:systemd-sysv-generator(8)
    Tasks: 0 (limit: 1152)
   CGroup: /system.slice/jenkins.service

Apr 06 12:18:02 ip-172-31-37-88 systemd[1]: Starting LSB: Start Jenkins at boot time...
Apr 06 12:18:02 ip-172-31-37-88 jenkins[7905]: Correct java version found
Apr 06 12:18:02 ip-172-31-37-88 jenkins[7905]:  * Starting Jenkins Automation Server jenkins
Apr 06 12:18:02 ip-172-31-37-88 su[7951]: Successful su for jenkins by root
Apr 06 12:18:02 ip-172-31-37-88 su[7951]: + ??? root:jenkins
Apr 06 12:18:02 ip-172-31-37-88 su[7951]: pam_unix(su:session): session opened for user jenkins by (uid=0)
Apr 06 12:18:02 ip-172-31-37-88 su[7951]: pam_unix(su:session): session closed for user jenkins
Apr 06 12:18:03 ip-172-31-37-88 jenkins[7905]:    ...done.
Apr 06 12:18:03 ip-172-31-37-88 systemd[1]: Started LSB: Start Jenkins at boot time.
ubuntu@ip-172-31-37-88:~$
```



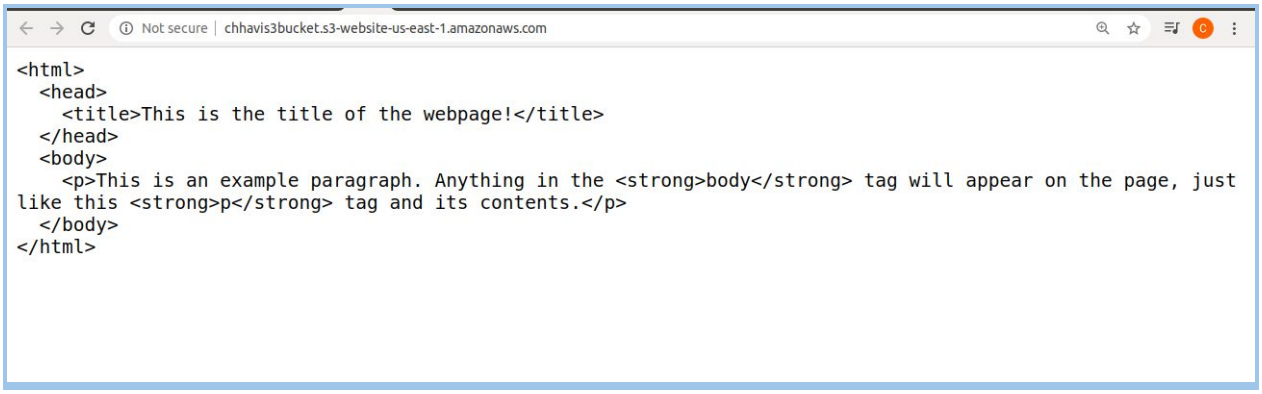
Step 3: Create a new repository



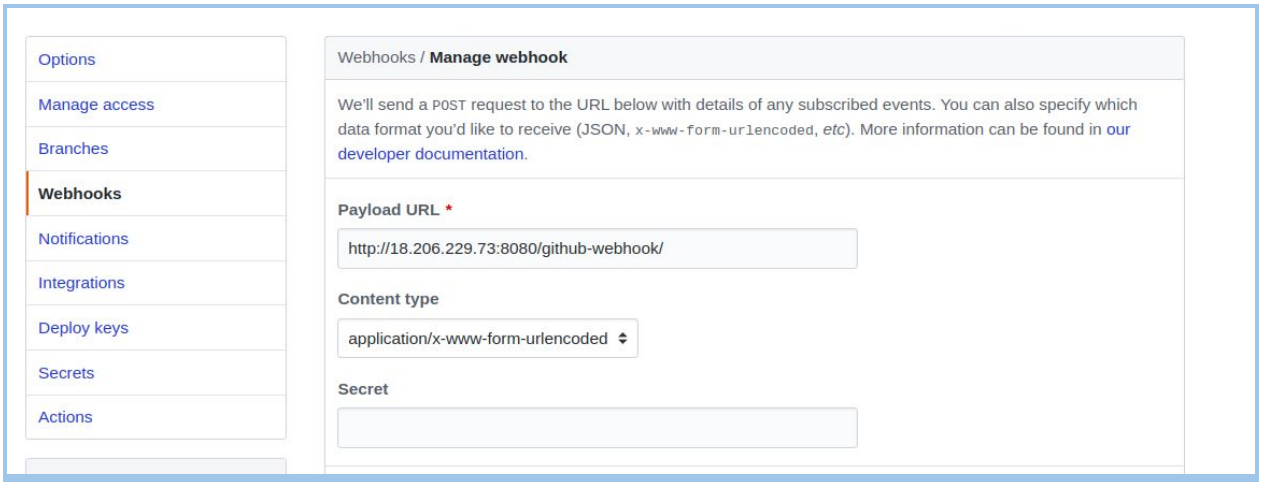
Step 4: Create a static website in s3 .

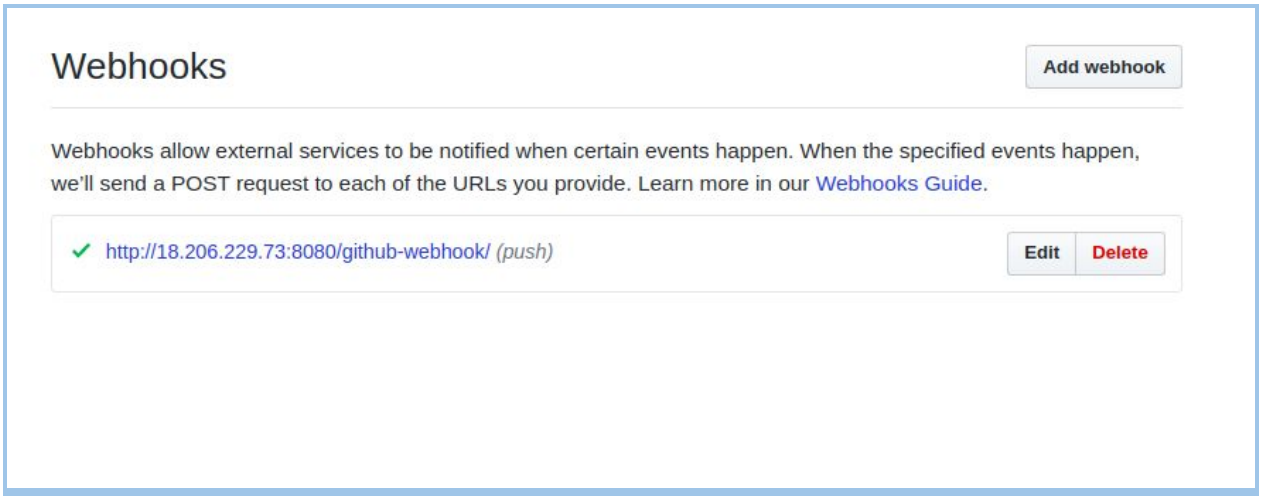


Check if the website is working .

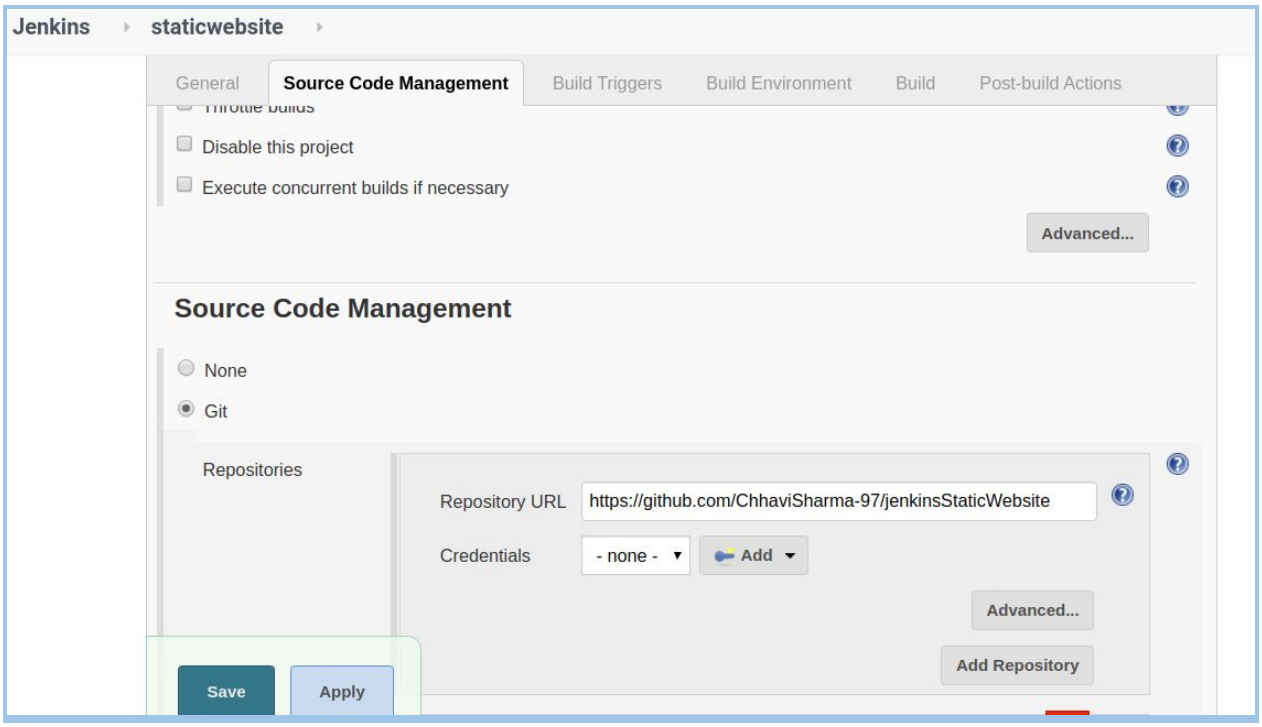


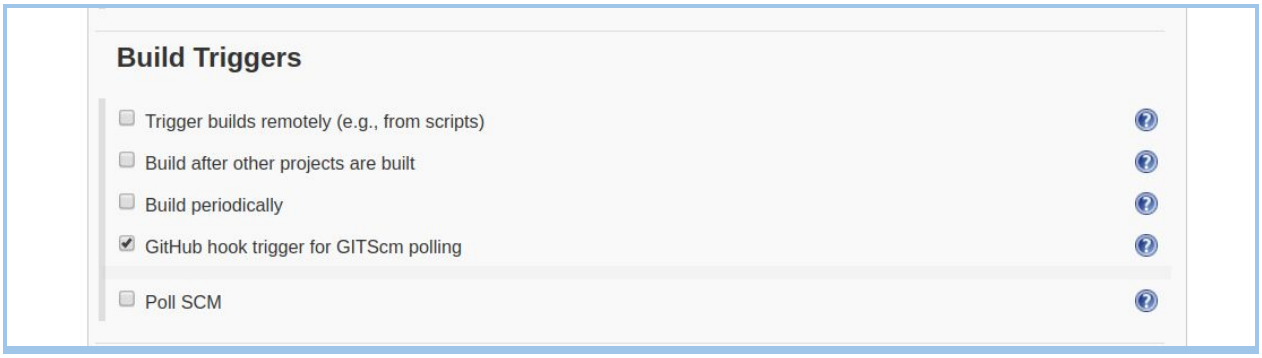
Step 5 : Apply webhooks to it. Go to Settings -> Webhooks -> Add Webhook



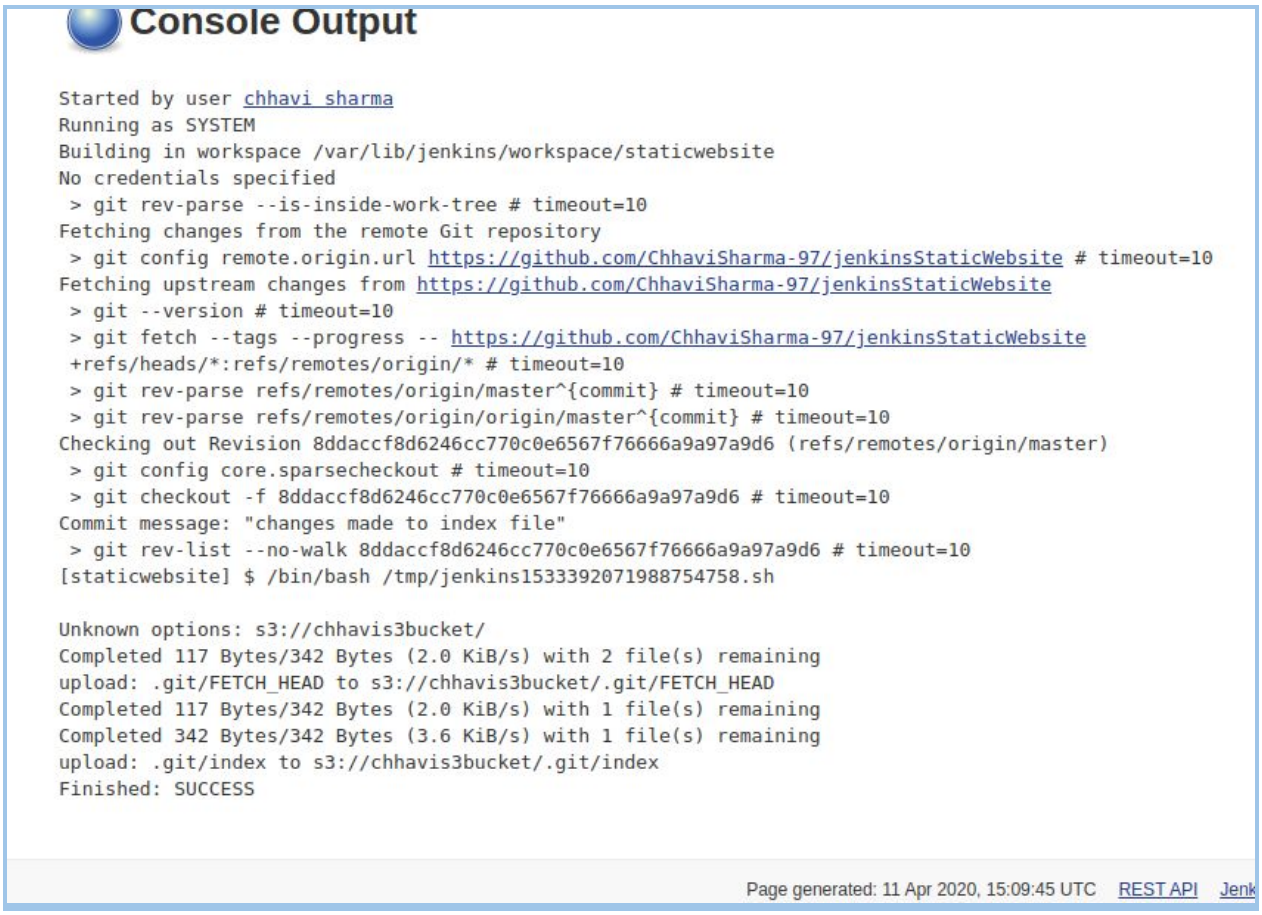


Step 6: Create a jenkins freestyle job with following configuration





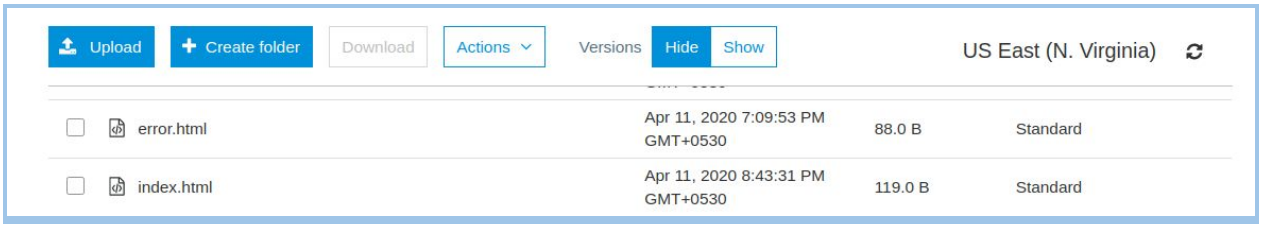
Step 7: Build the job.

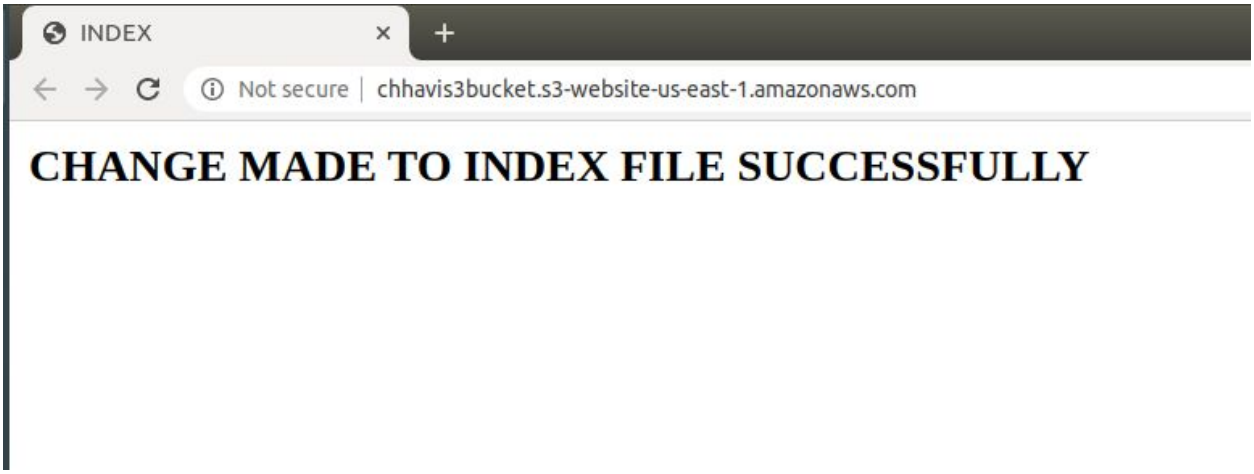


Step 8: Now push changes to your repo

```
ubuntu@ip-172-31-215-131:~/website$ git add .
ubuntu@ip-172-31-215-131:~/website$ git commit -m "changes"
[master 7162fff] changes
 Committer: Ubuntu <ubuntu@ip-172-31-215-131.ec2.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, 1 insertion(+), 1 deletion(-)
ubuntu@ip-172-31-215-131:~/website$ git push origin1 master
Username for 'https://github.com': ChhaviSharma-97
Password for 'https://ChhaviSharma-97@github.com':
Counting objects: 3, done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 369 bytes | 369.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/ChhaviSharma-97/jenkinsStaticWebsite.git
   8ddaccf..7162fff  master -> master
```

Step 9: See that the changes are reflected in your website.




INDEX
Not secure | chhavis3bucket.s3-website-us-east-1.amazonaws.com
CHANGE MADE TO INDEX FILE SUCCESSFULLY