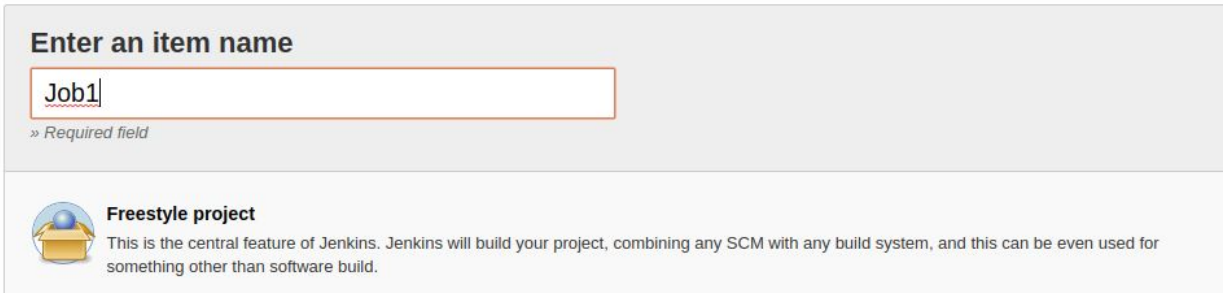
ASSESSMENT ON :

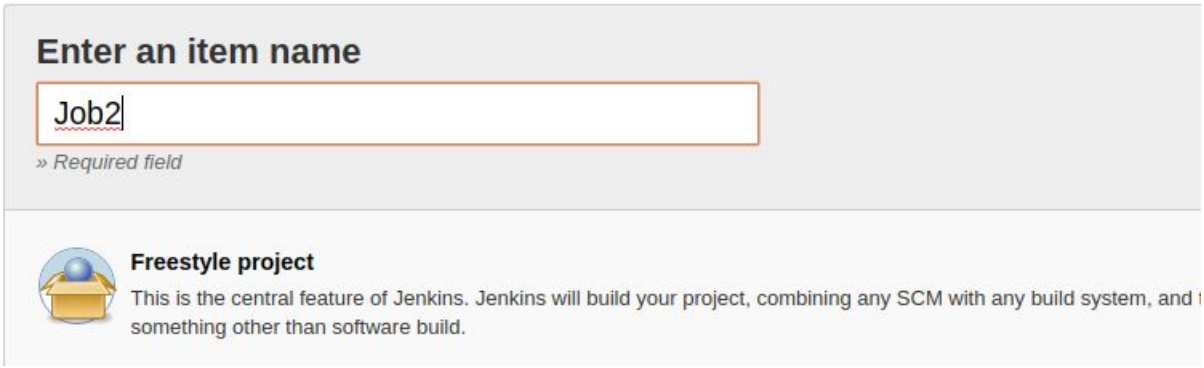
JENKINS(1)

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:

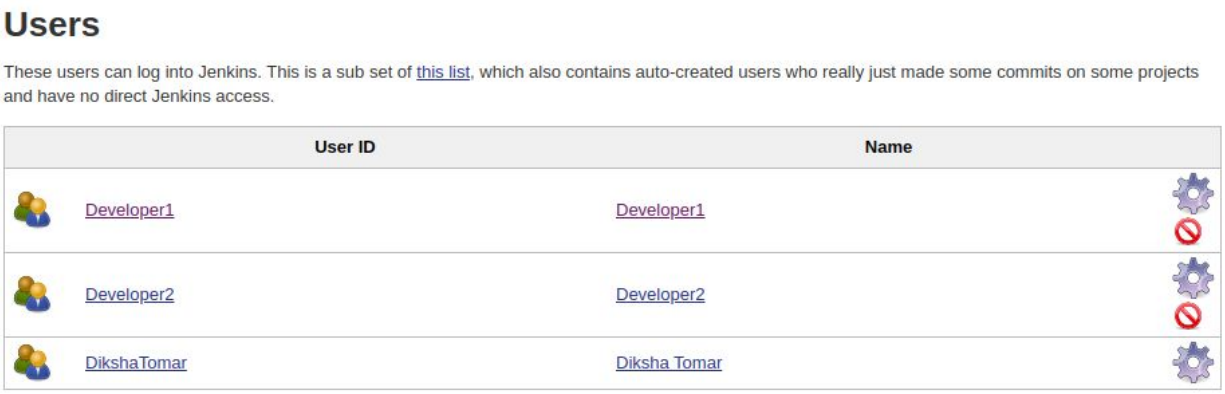
STEP 1: Create 2 jobs Job1 Job2



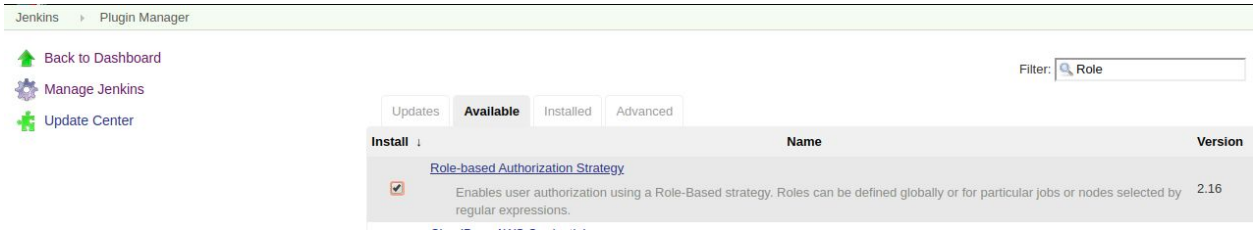


STEP 2: To create users go to Manage jenkins > Manage Users

STEP 3: Create two users Developer1 and Developer2



STEP 4: Navigate to Jenkins > Manage Jenkins > Manage Plugins > Available > Filter.
Type "Role-based Authorization Strategy" in the filter box and hit enter.
Select plugin and click the "Download now and install after restart".



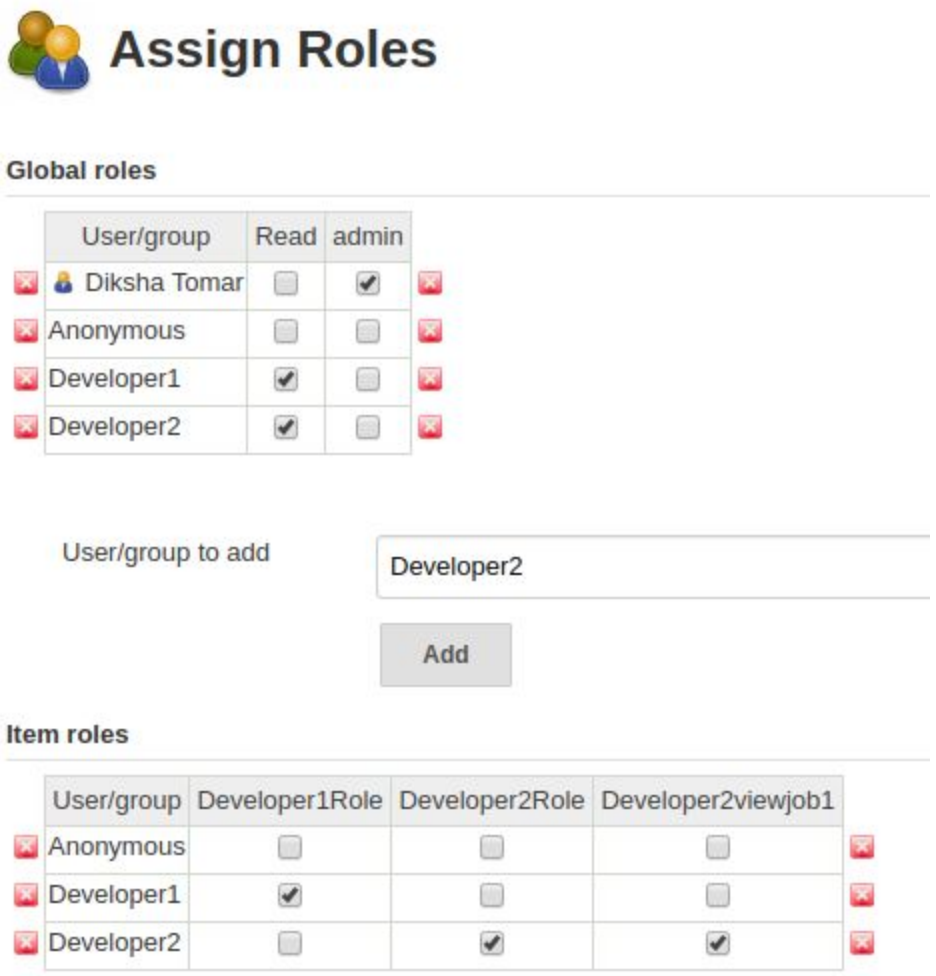
STEP 5: After plugin installation, navigate to "Jenkins > Configure Global Security". Tick Enable security and Role-Based Strategy then save settings.

STEP 6: Creating User Roles on Jenkins
Go to “Jenkins > Manage and Assign Roles > Manage Roles“.

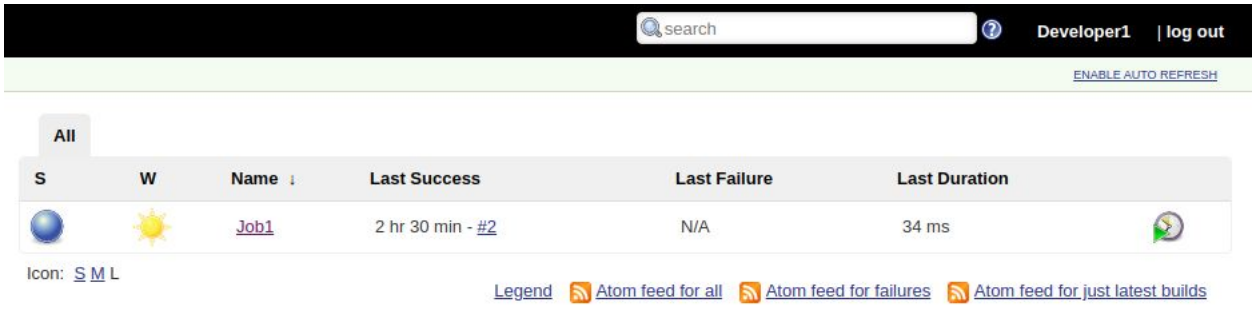
Item roles

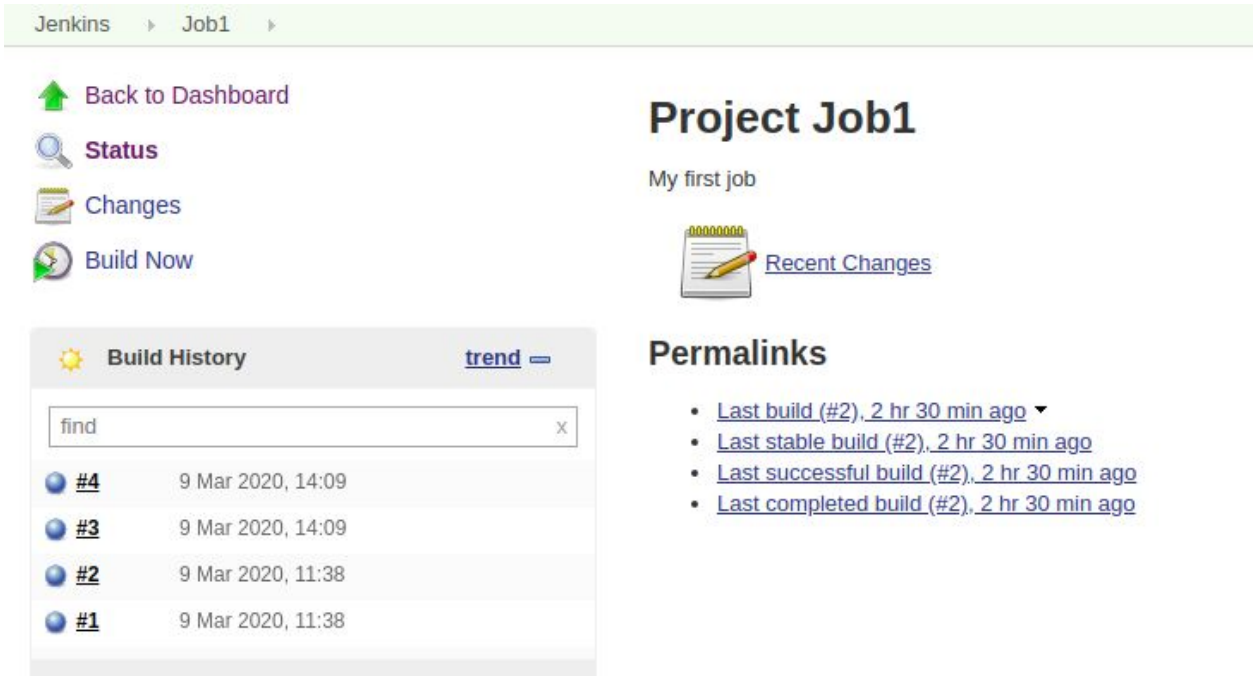
	Role	Pattern	Credentials					Job							
			Create	Delete	ManageDomains	Update	View	Build	Cancel	Configure	Create	Delete	Discover	Move	Read
	Developer1Role	"Job1"	☐	☐	☐	☐	☐	☑	☐	☐	☐	☐	☐	☐	☐
	Developer2Role	"Job2"	☐	☐	☐	☐	☐	☑	☐	☑	☐	☐	☐	☐	☐
	Developer2viewjob1	"Job1"	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☑

STEP 7: Assign roles to users
Go to “Jenkins > Manage and Assign Roles > Assign Roles“.

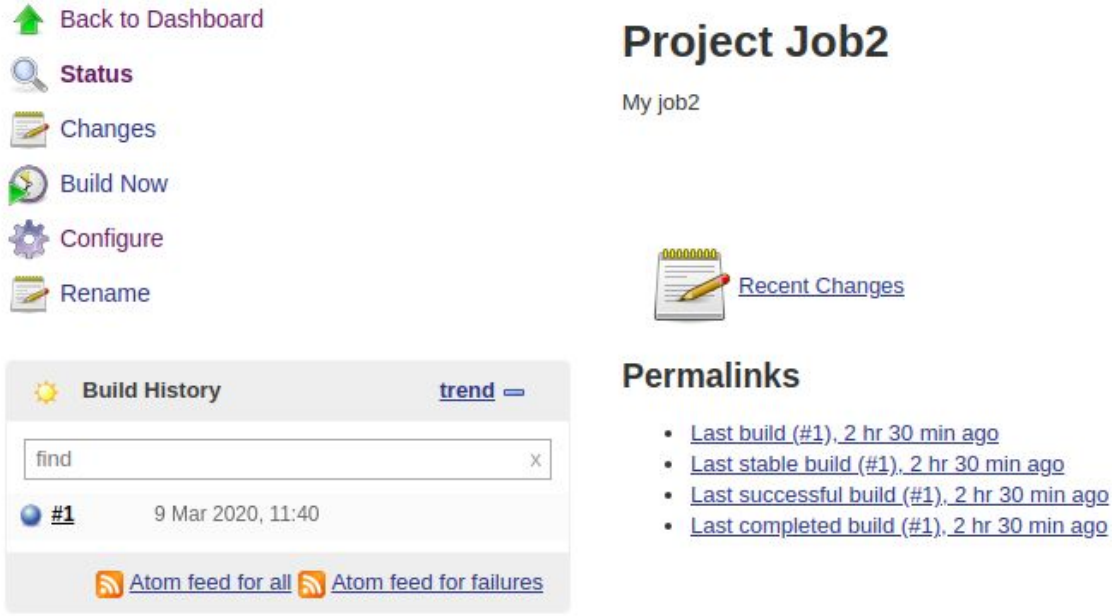


STEP 8: Login through Developer1 user and you can see Job1 with only access to build

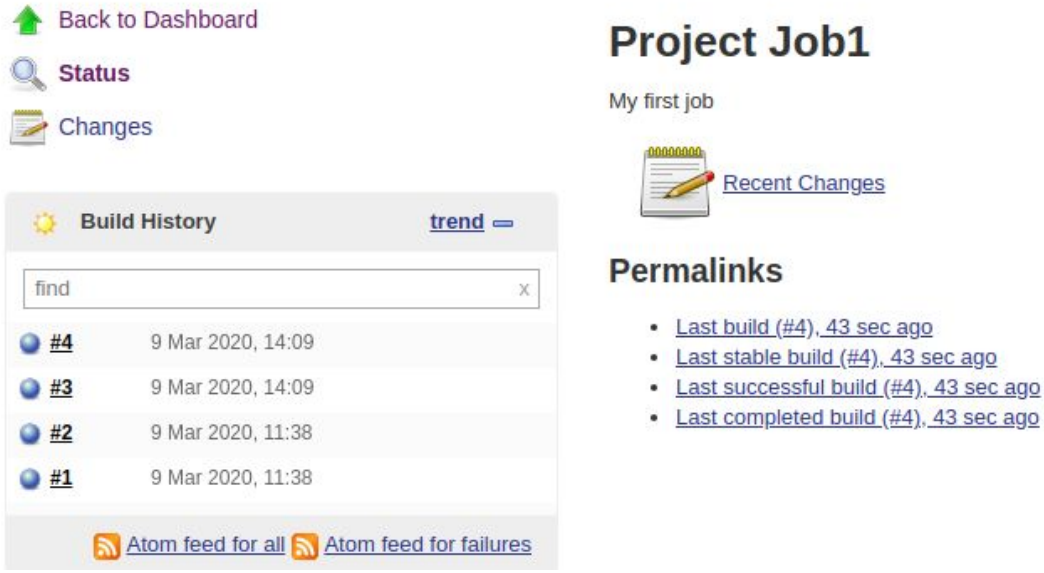




STEP 9: Login through Developer2 user and you can see Job2 with only access to build configure and build



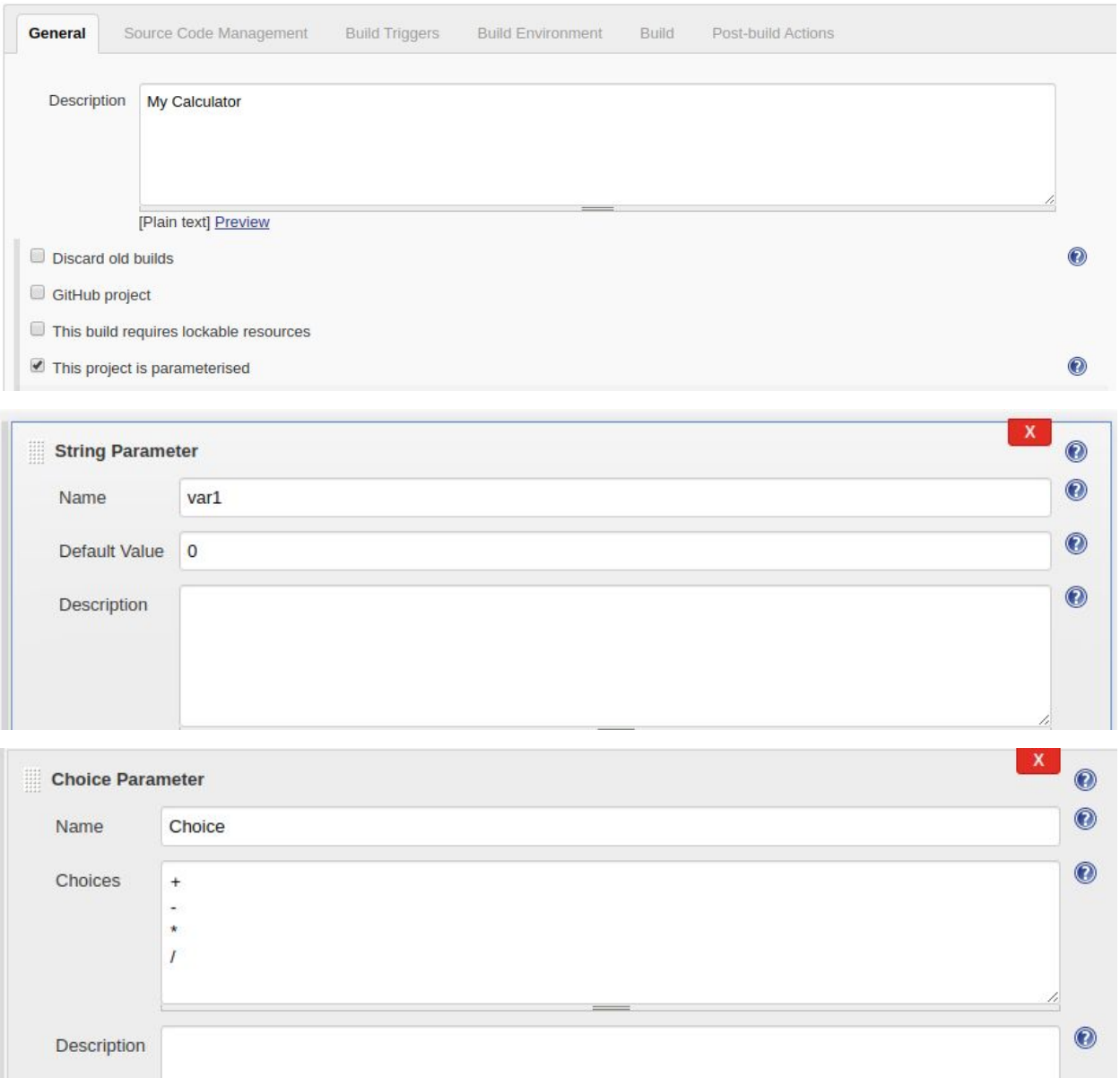
STEP 10: You can even observe that Job1 has only view access.



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:

STEP 1: Create a job and add parameters to it.



STEP 2: Build it using a script to implement calculator.

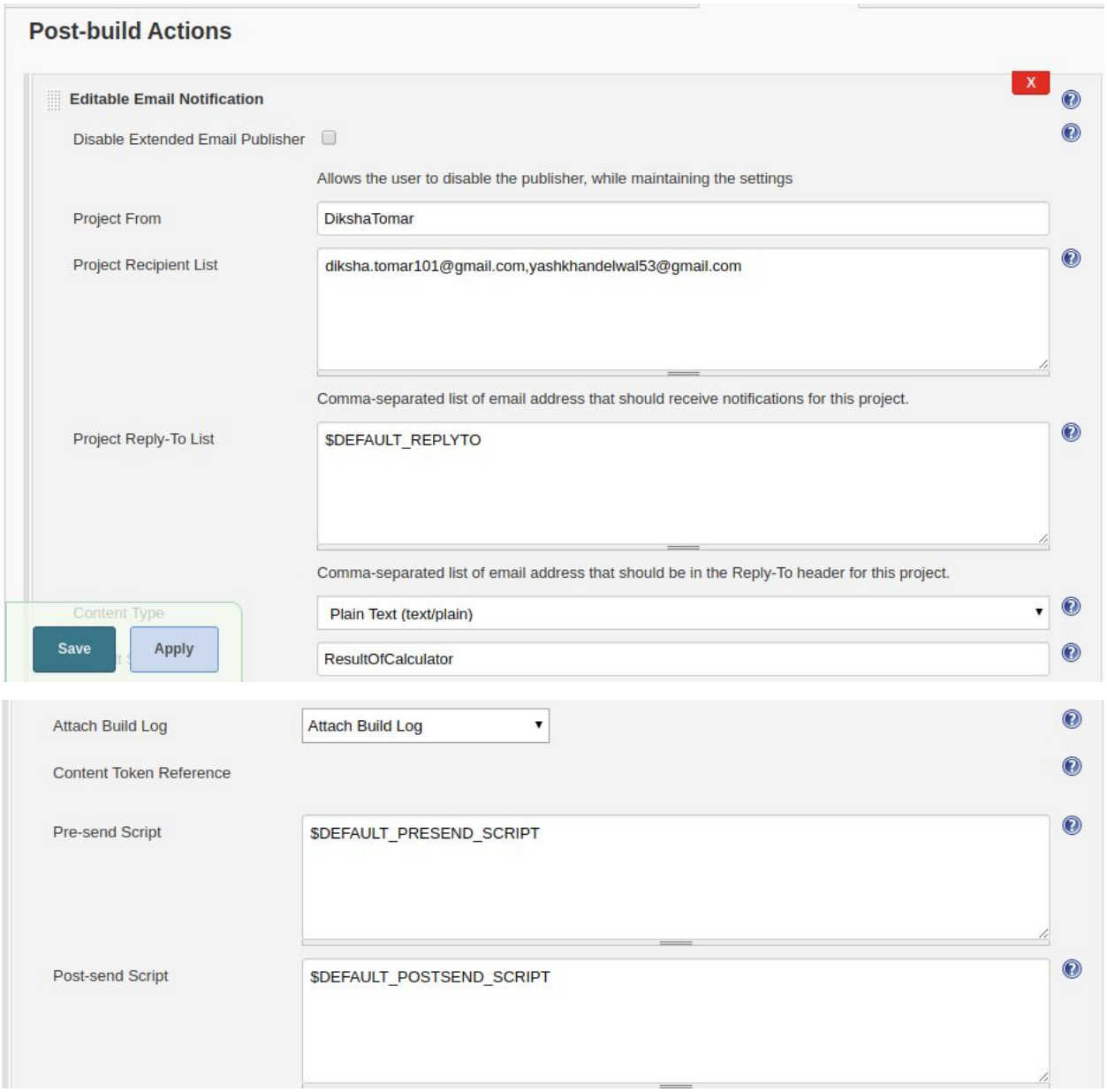
STEP 3: Manage jenkins > Configure Systems

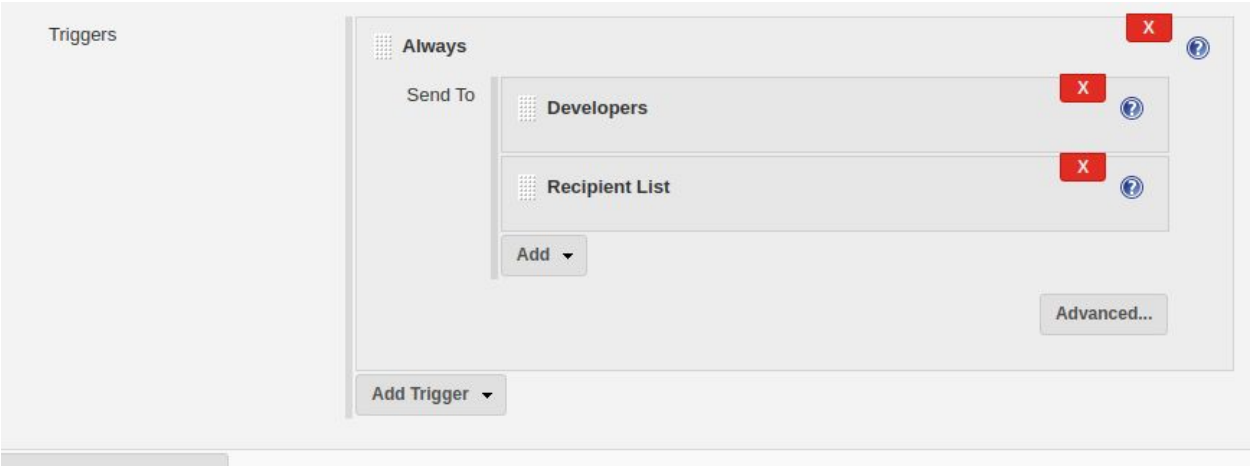
"E-mail Notification" section

1. Enter your SMTP server name to "SMTP server"
2. Click "Advanced"
3. Click "Use SMTP Authentication"
4. Enter required informations

STEP 4: Configure a project to send email at every build

1. Click “Add post-build action”
2. Click “Editable Email Notification”
3. Click “Advanced Settings…”
4. Click “Add Trigger”
5. Click “Always”
6. Save

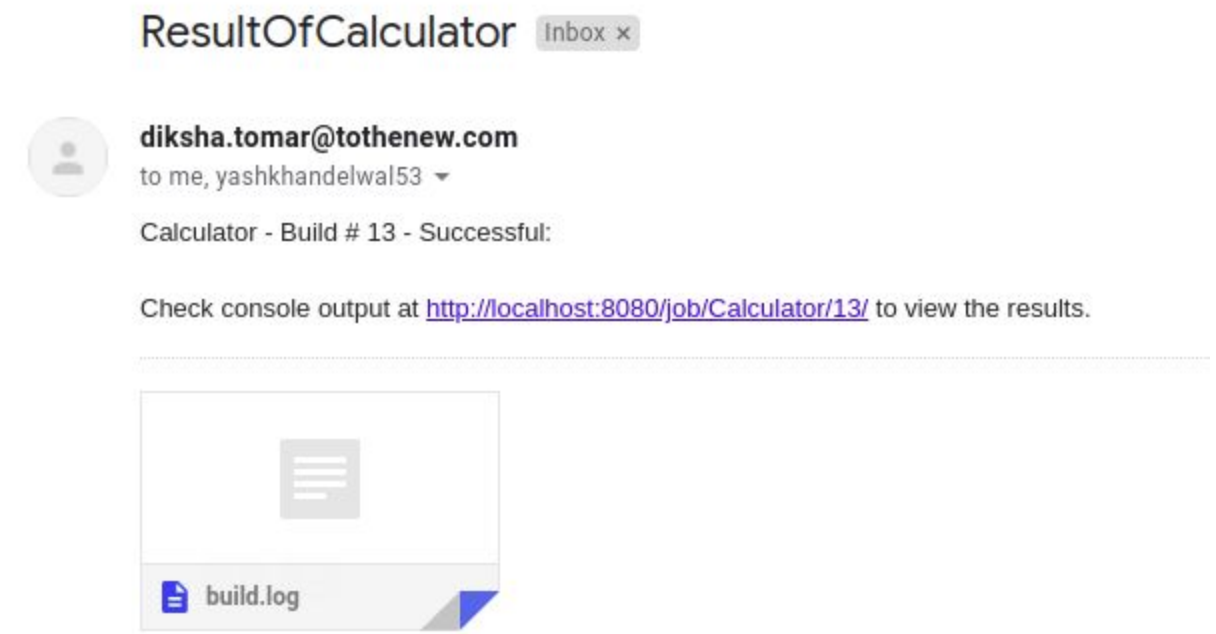

Post-build Actions
Editable Email Notification
X
Disable Extended Email Publisher
Allows the user to disable the publisher, while maintaining the settings
Project From
DikshaTomar
Project Recipient List
diksha.tomar101@gmail.com,yashkhandelwal53@gmail.com
Comma-separated list of email address that should receive notifications for this project.
Project Reply-To List
$DEFAULT_REPLYTO
Comma-separated list of email address that should be in the Reply-To header for this project.
Content Type
Plain Text (text/plain)
Save
Apply
ResultOfCalculator
Attach Build Log
Attach Build Log
Content Token Reference
Pre-send Script
$DEFAULT_PRESEND_SCRIPT
Post-send Script
$DEFAULT_POSTSEND_SCRIPT

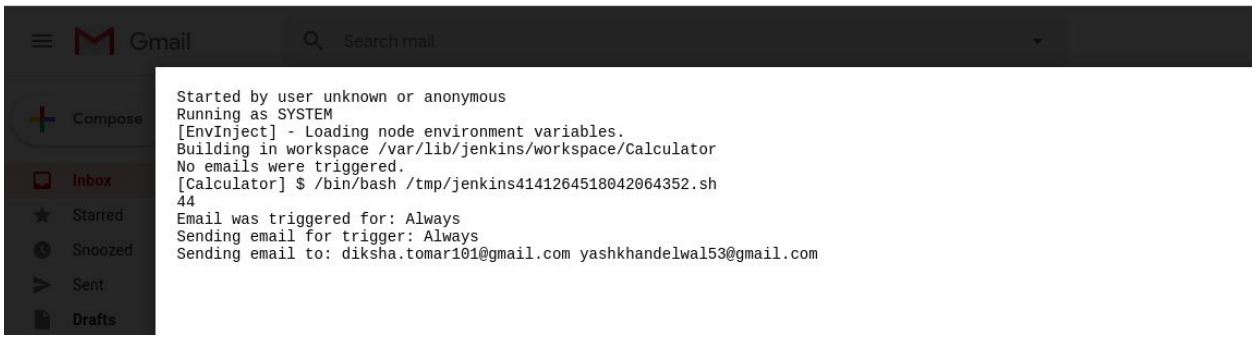


STEP 5: Build the job and check your mail for the result of calculator.

Console Output

```
Started by user unknown or anonymous
Running as SYSTEM
[EnvInject] - Loading node environment variables.
Building in workspace /var/lib/jenkins/workspace/Calculator
No emails were triggered.
[Calculator] $ /bin/bash /tmp/jenkins4141264518042064352.sh
44
Email was triggered for: Always
Sending email for trigger: Always
Sending email to: diksha.tomar101@gmail.com yashkhandelwal53@gmail.com
Finished: SUCCESS
```

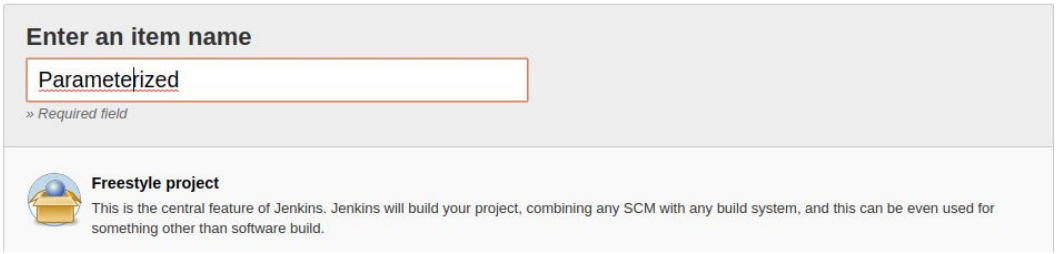




3. Create jenkins parameterized job which on selecting the different Env. will display different web pages by nginx.

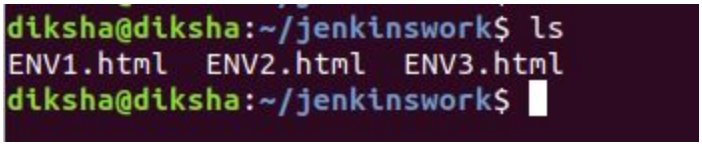
ANS:

STEP 1: Create a job with parameters.

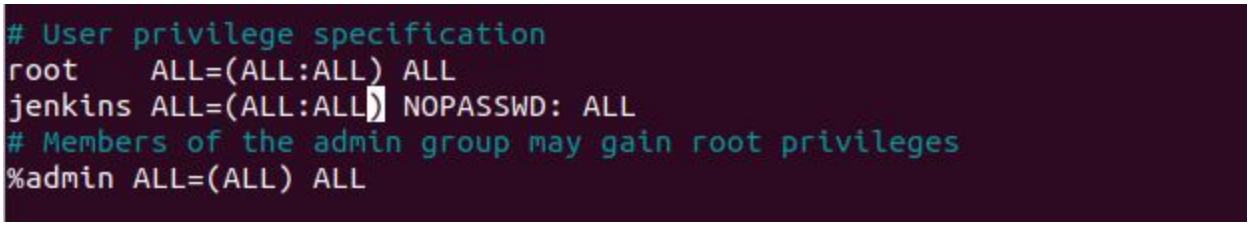


STEP 2: Build the job using script.

STEP 3: Make a jenkinswork directory and make 3 env.html files



STEP 4: In the sudoers file add jenkins with all the permission.



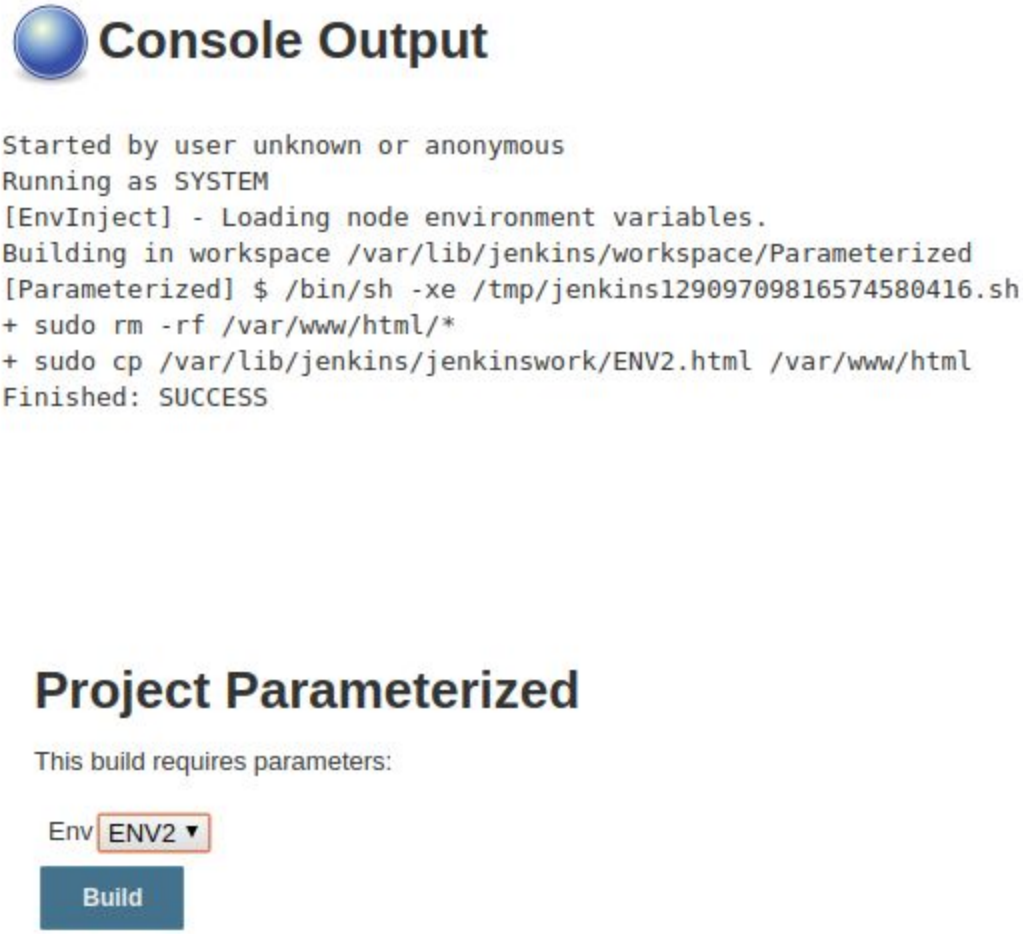
STEP 5: In /etc/nginx/sites-enabled add the env files and then restart nginx,Restart jenkins.

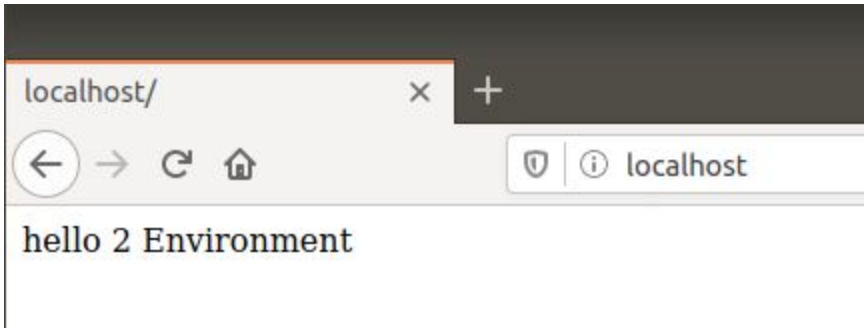
```
        # Add index.php to the list if you are using PHP
        index index.nginx-debian.html ENV1.html ENV2.html ENV3.html;

        server_name _;

        location / {
                # First attempt to serve request as file, then
                # as directory, then fall back to displaying a 404.
                try_files $uri $uri/ =404;
        }
"default" 91L, 2425C written                                          47,0-1
```

STEP 6: Build the job.

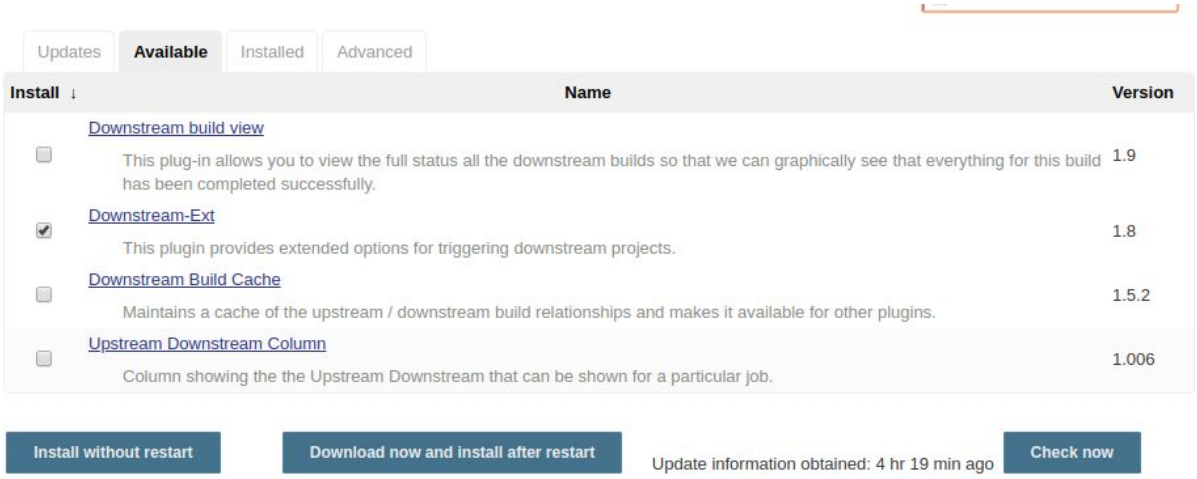




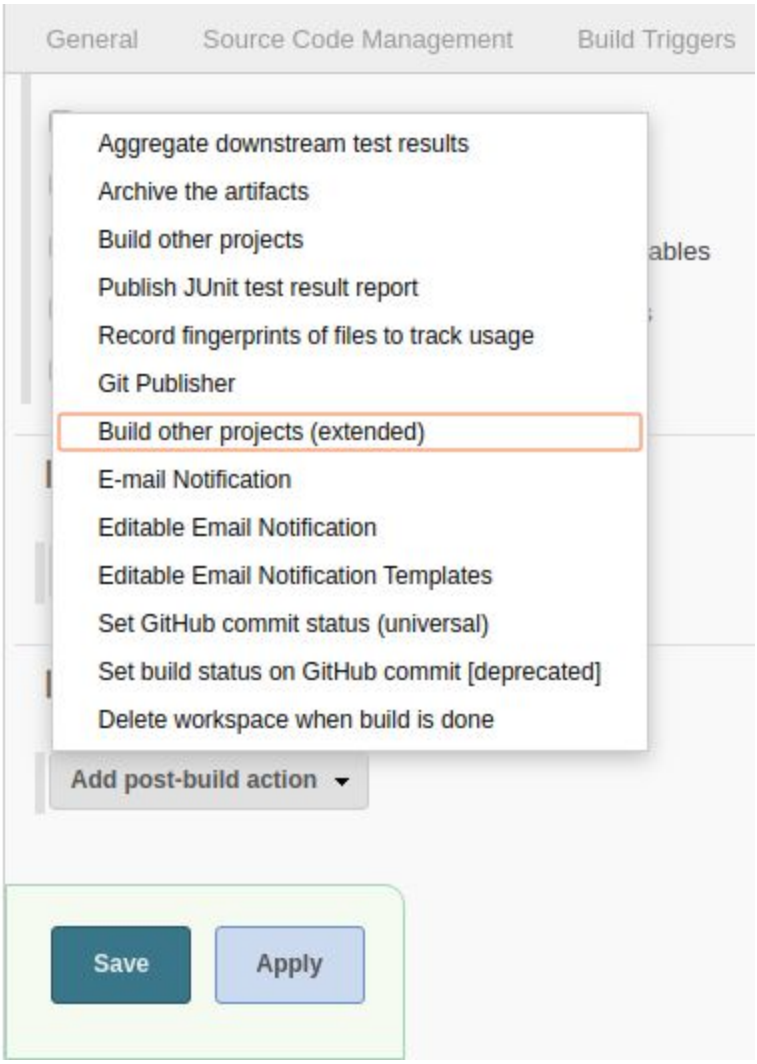
4. Create a job which on its failure will trigger another job.

ANS:

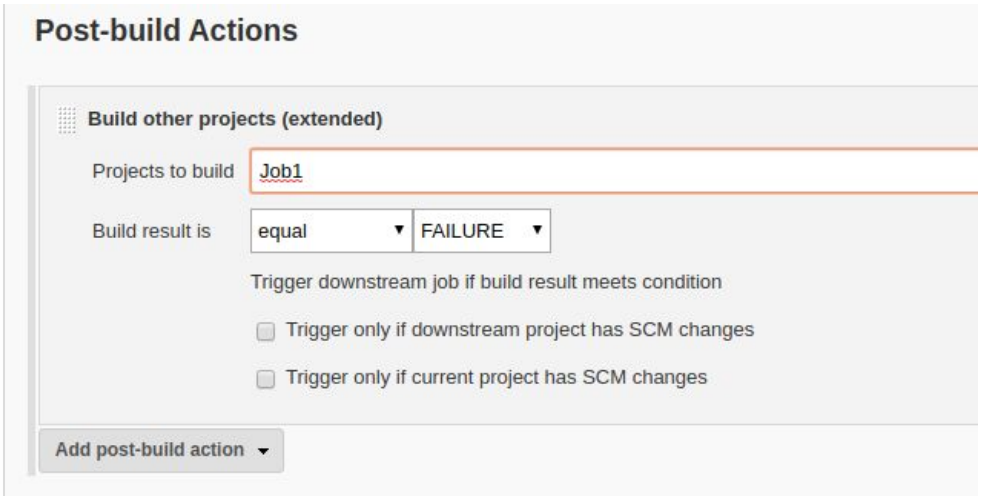
STEP 1: Install “Downstream-Ext” plugin.



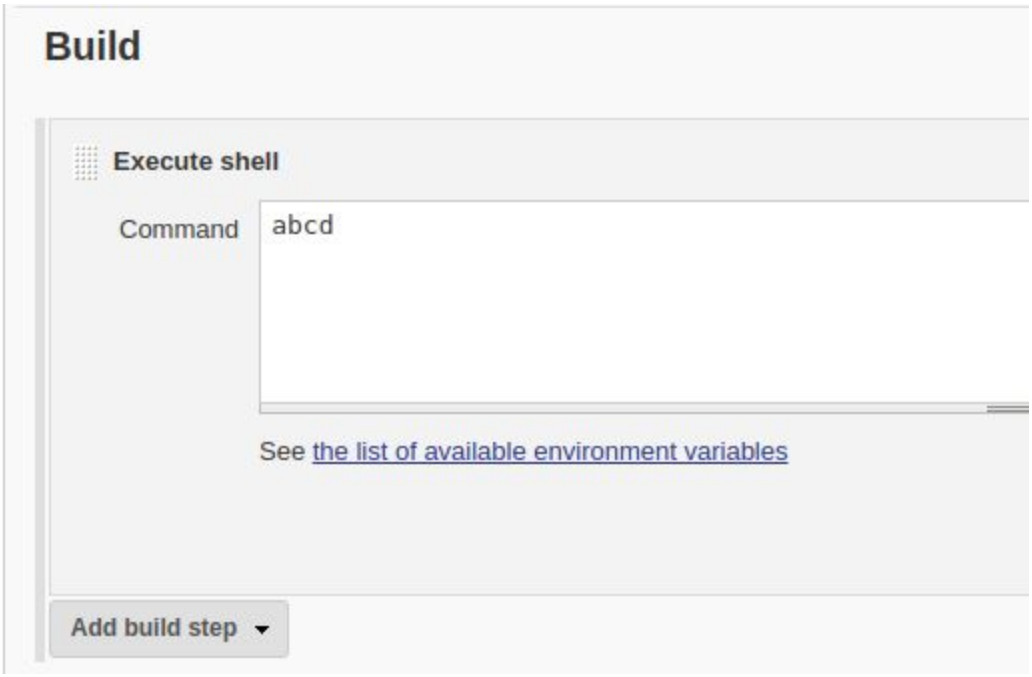
STEP 2: Go to Post build Action > Select: Build other projects(extended)



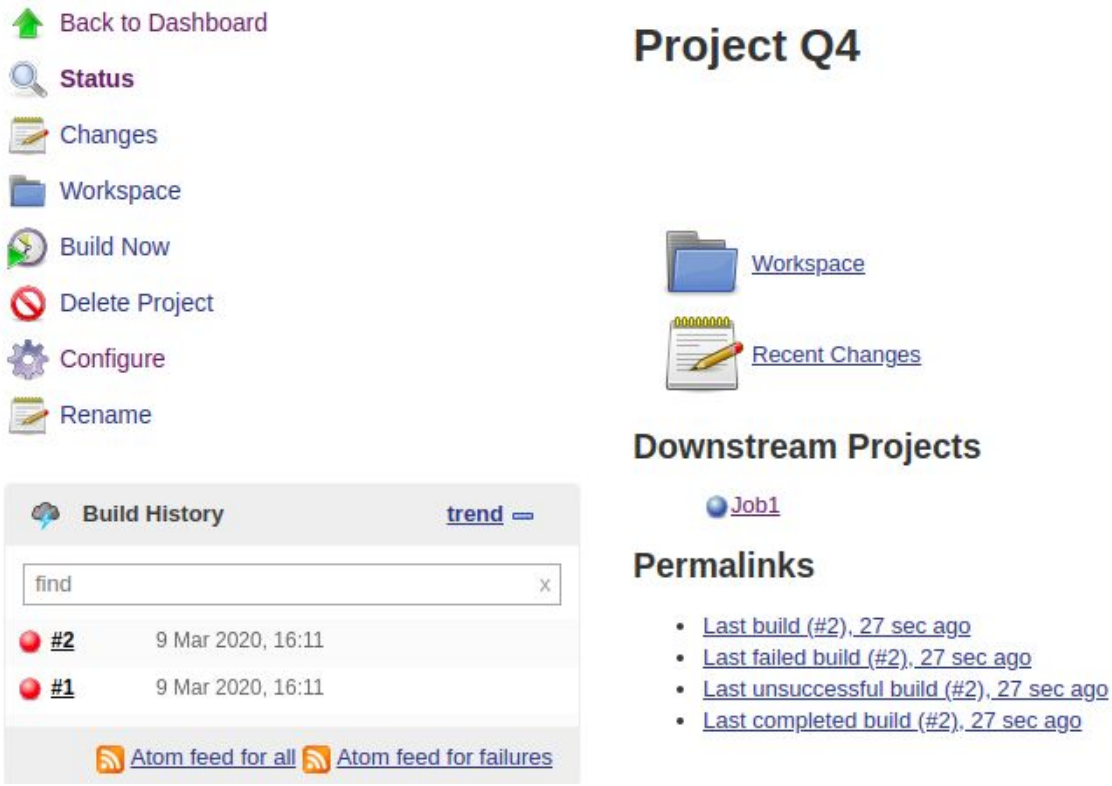
STEP 3: Specify the Job that will be triggered on failure

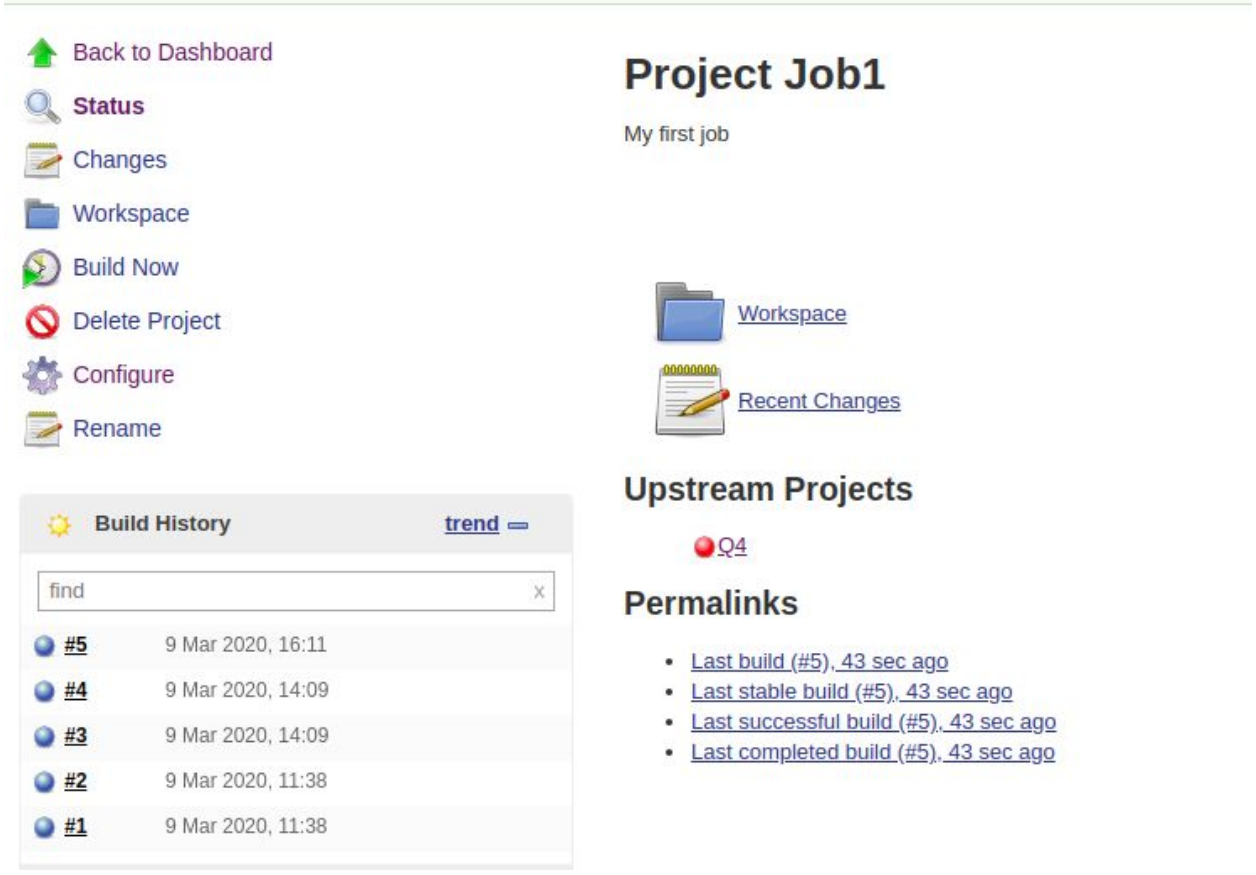


STEP 4: For error :-



STEP 5: Build the job and observe the changes

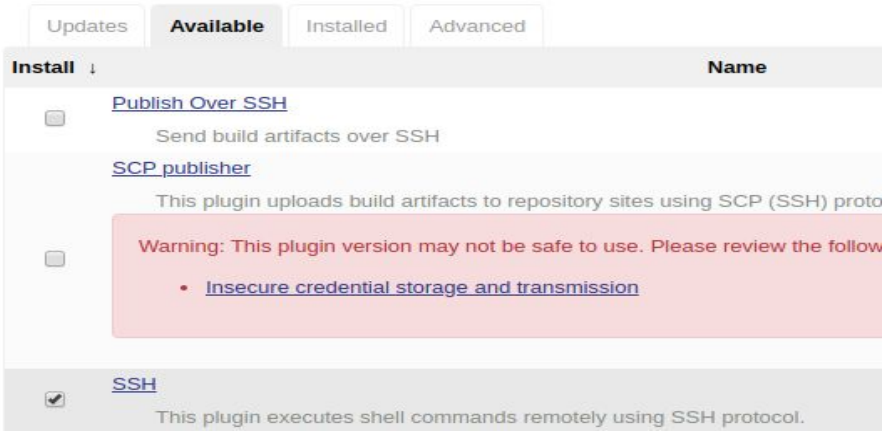




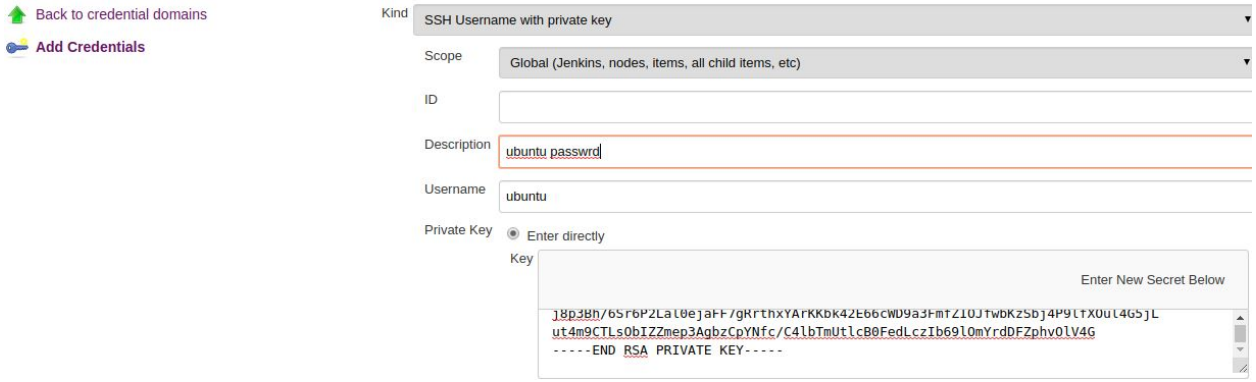
5. Create a job which can set a cronjob on another server. This server contains a script on its home directory and the script will print the two string parameters which will be given by the jenkins job.

ANS:

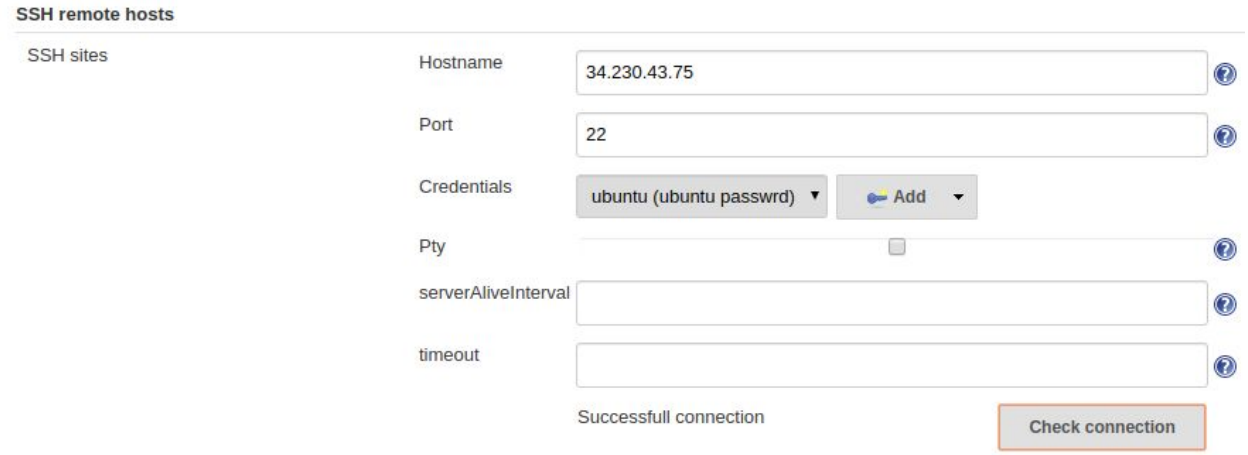
STEP 1: Install SSH plugin.



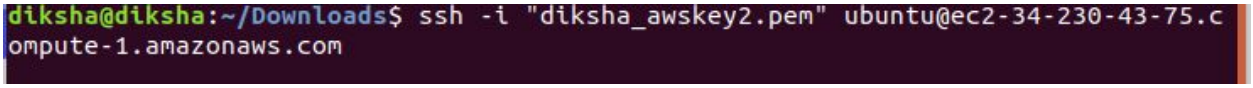
STEP 2: Credentials > Global credentials(unrestricted) >Add Credentials >Kind: SSH username with private key

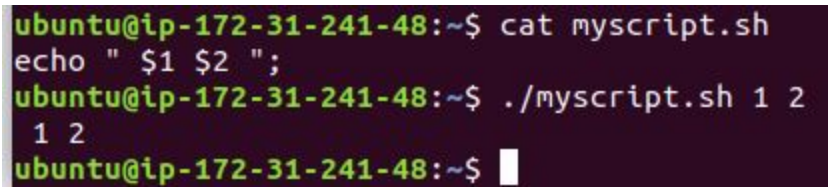


STEP 3: Manage jenkins > Configure System >SSH Hosts

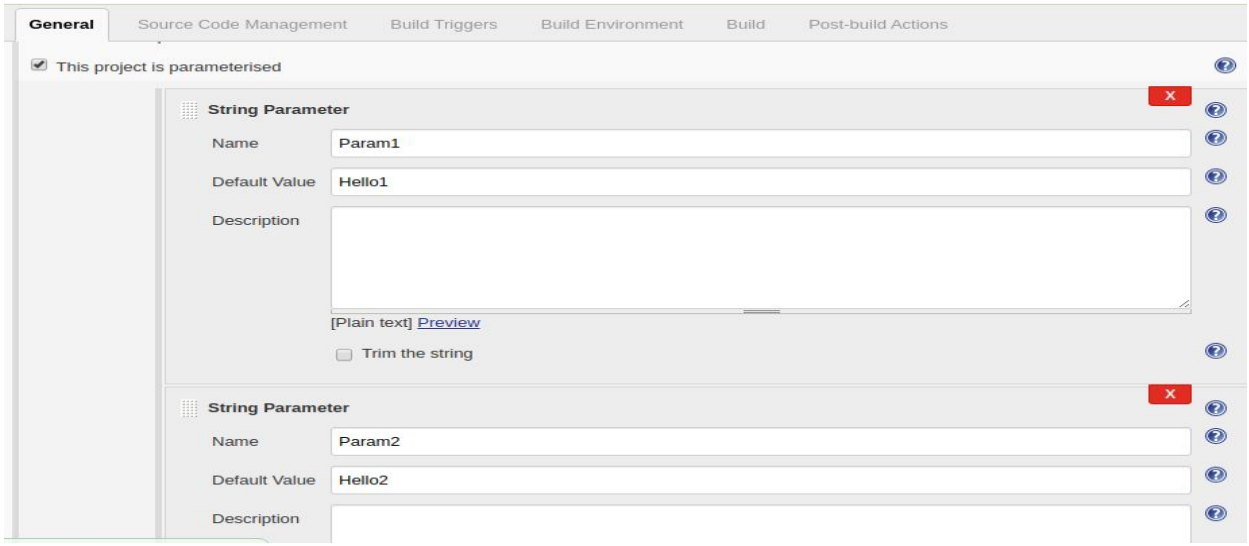


STEP 4: SSH into your instance and create a script

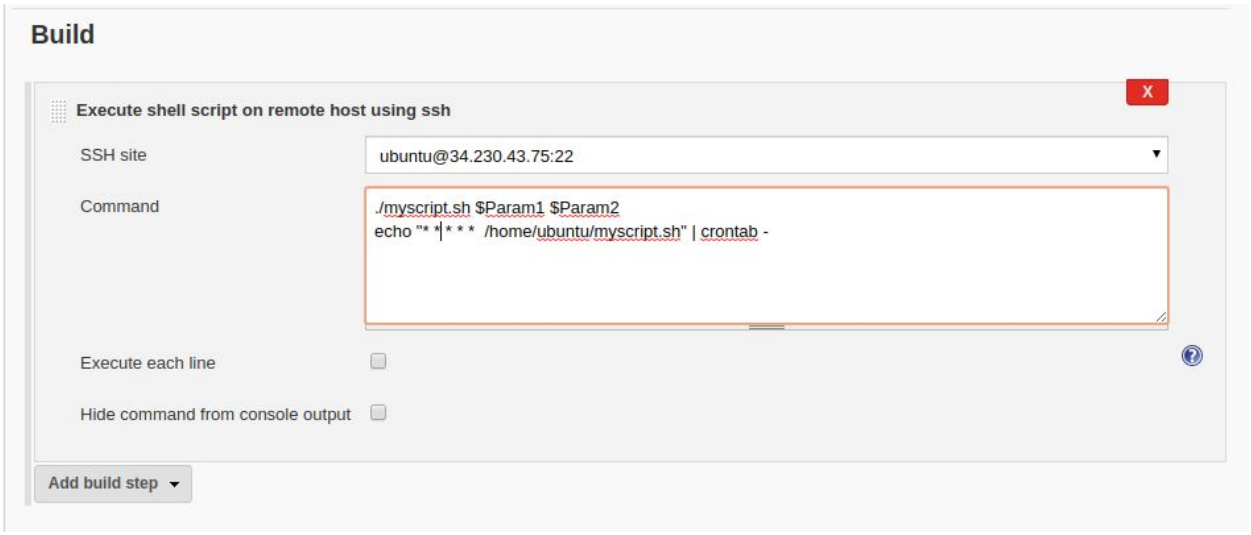




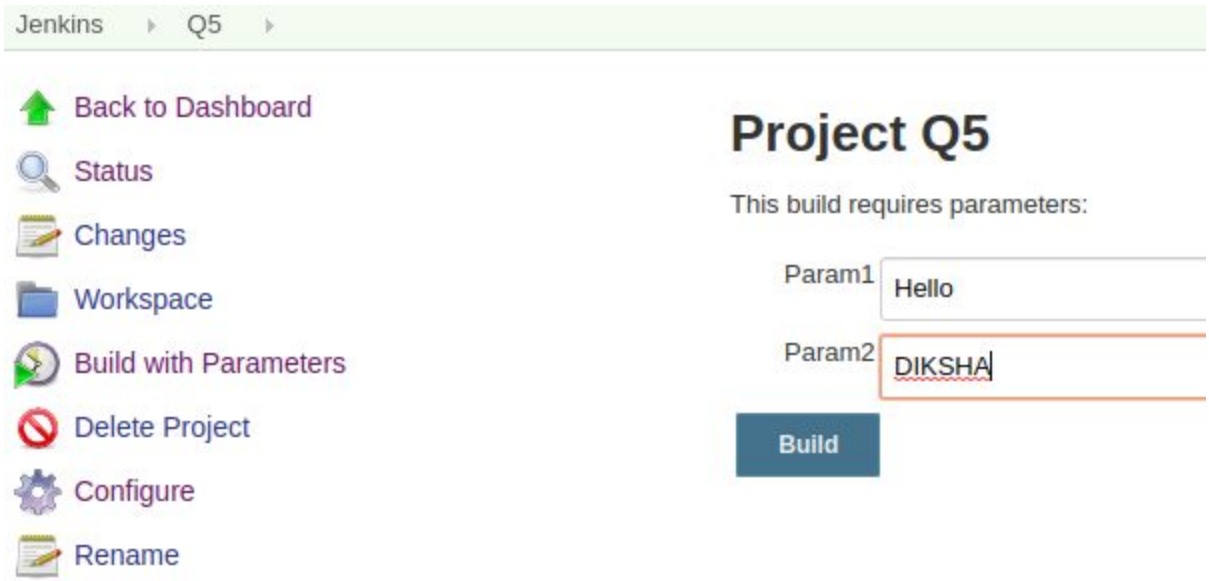
STEP 5: Create a new job and add parameters to it.

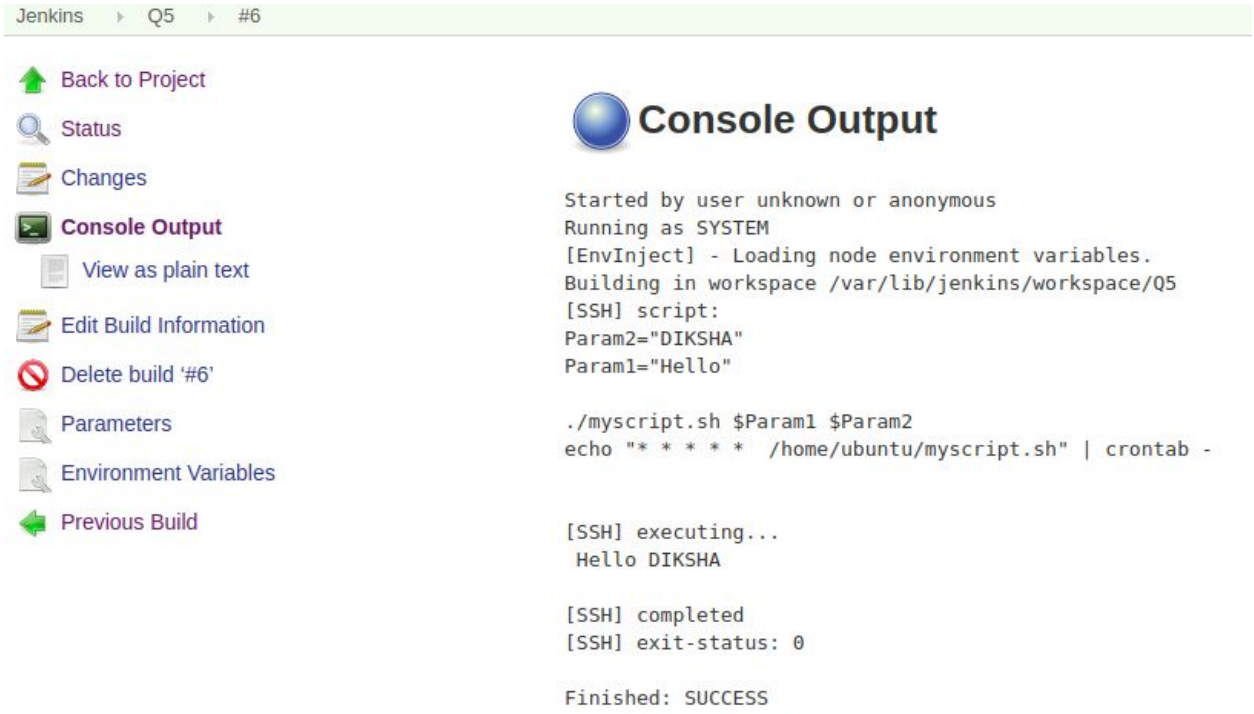


STEP 6: Build it using shell script on remote host using ssh



STEP 7: Build the job and give parameters value

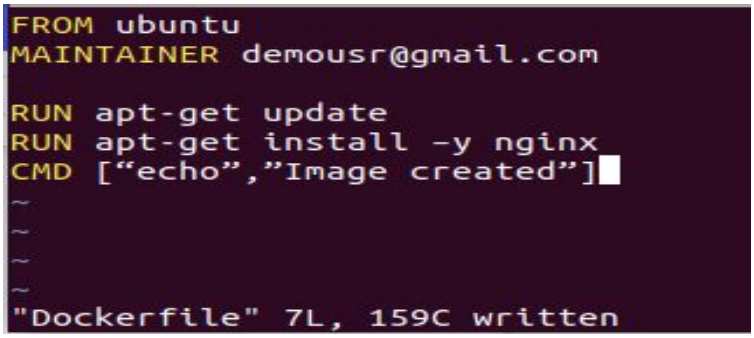


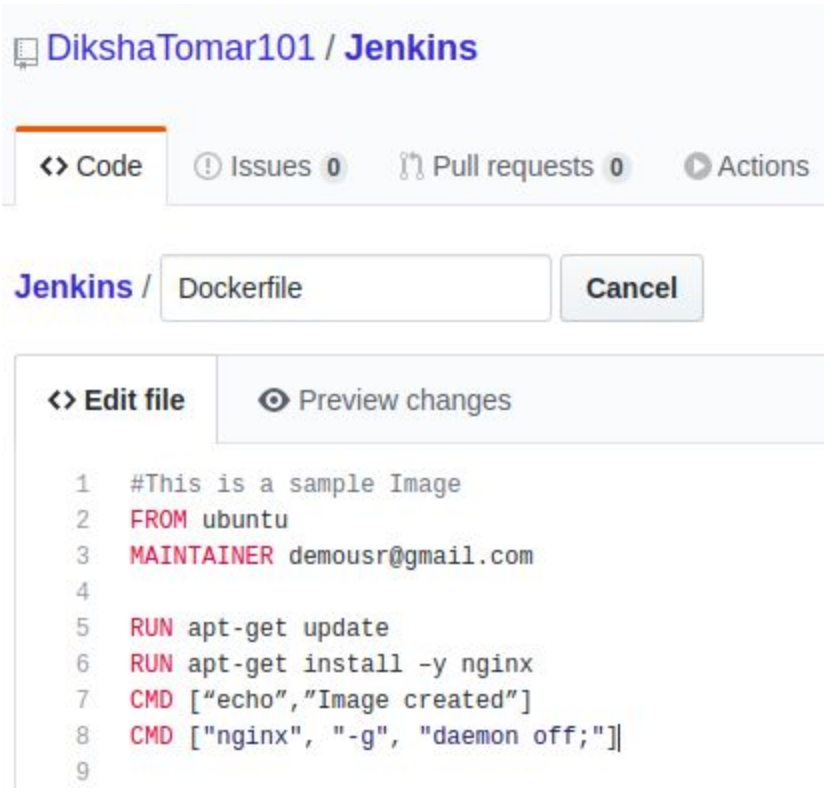


6. Create a job in which: Pull Dockerfile from GitHub, build it and push to Dockerhub. The docker image should have the tag: git commit id.

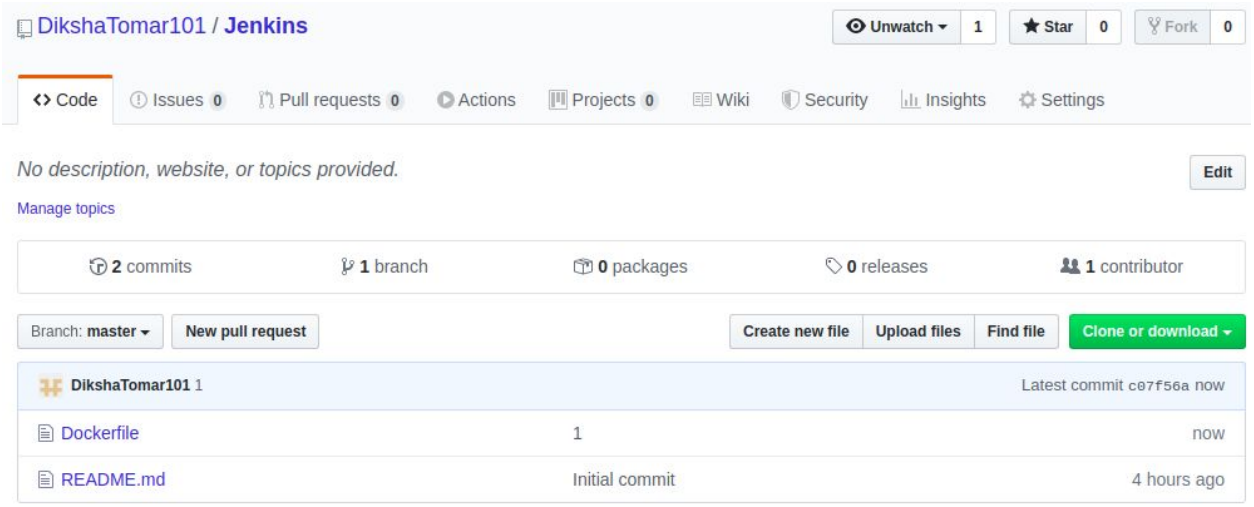
ANS:

STEP 1: Create a dockerfile in your local.

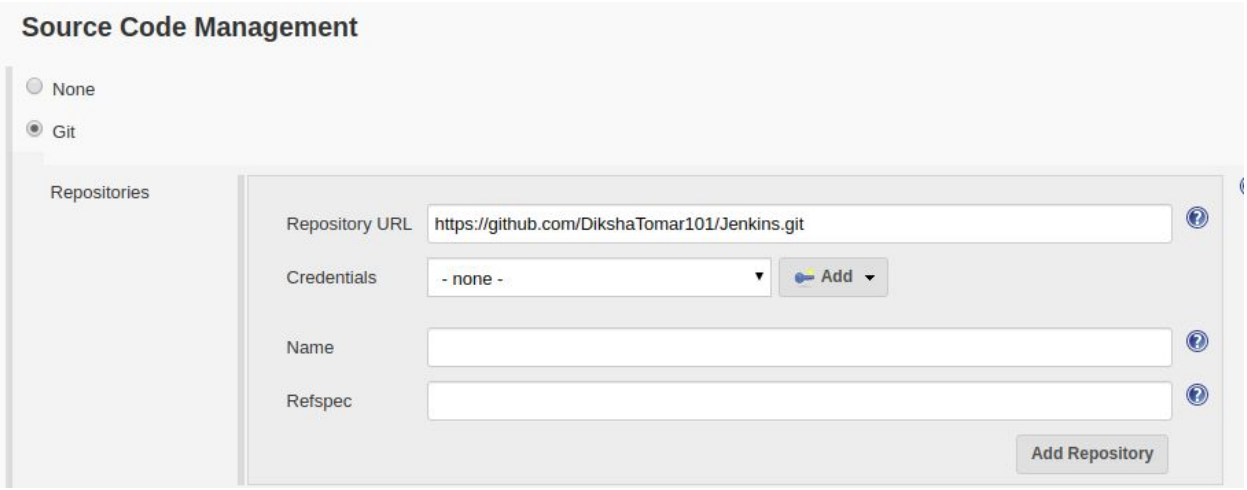




STEP 2: Push the dockerfile to your remote repository “Jenkins” in github.



STEP 3: IN SCM provide a git url.

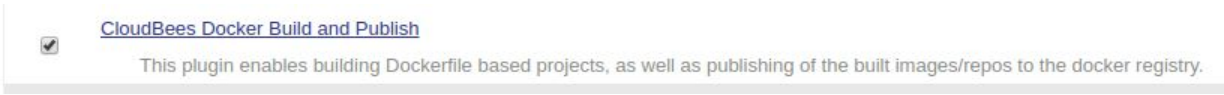


Deploy

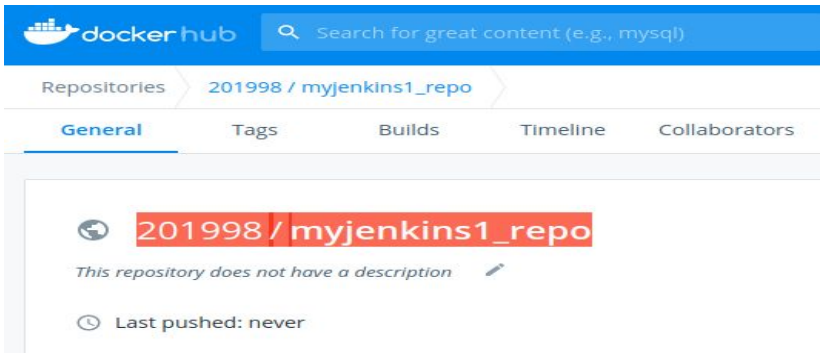
Now we're ready to create the job which will do the work for us. Click on New Item on the left side menu, and enter the item name as desired job name. Then select the *Freestyle project.*

STEP 4: Install plugin "Cloudbees Docker build and Publish"

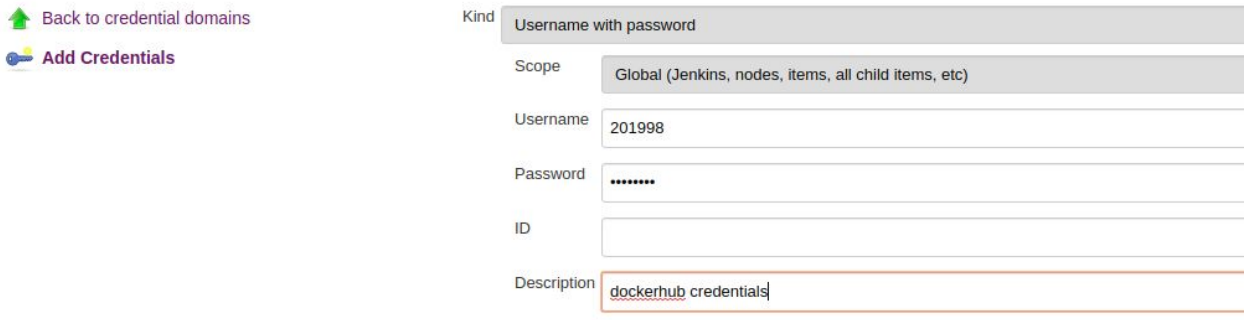
CloudBees Docker Build and Publish plugin: It enables us to build images from the Dockerfile present on the server and publishing them on the DockerHub.



STEP 5: In your Dockerhub account create a new repository.



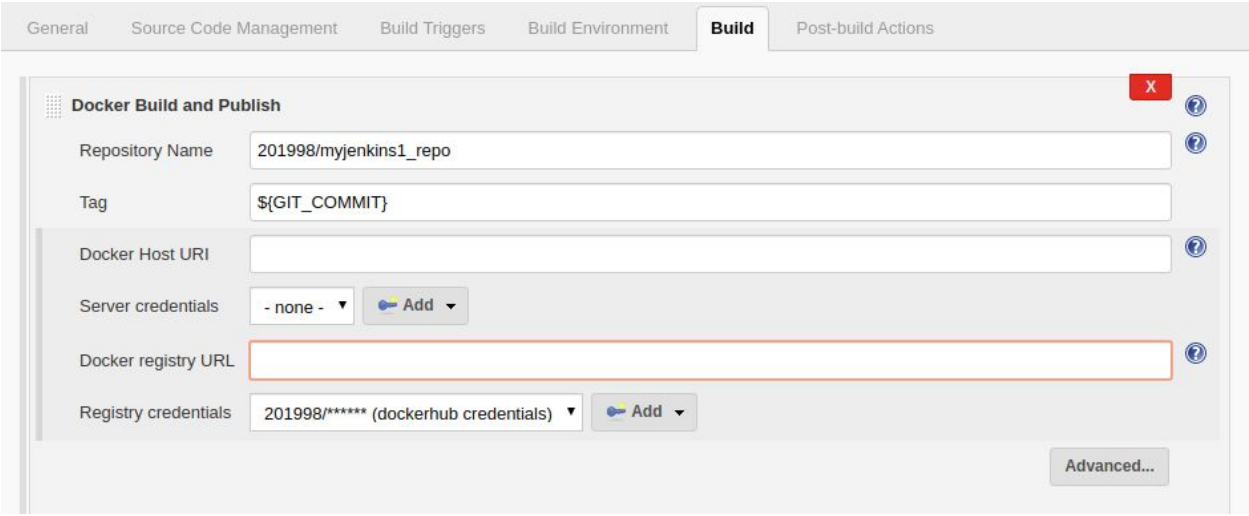
STEP 6: Add credentials of your Dockerhub account



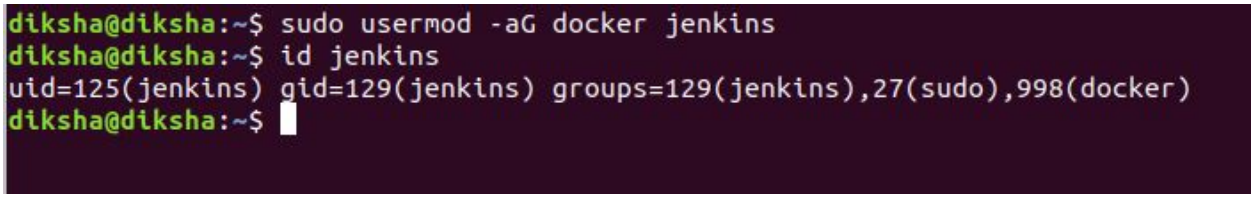
STEP 7: Creating and configuring Jenkins job to build images from Dockerfile:

Create a new Jenkins job (say "*Build Docker Image*") which will use *CloudBees Docker Build and Publish plugin* to build images from Dockerfile and push it on DockerHub.

Configure this job as follows under the build section:

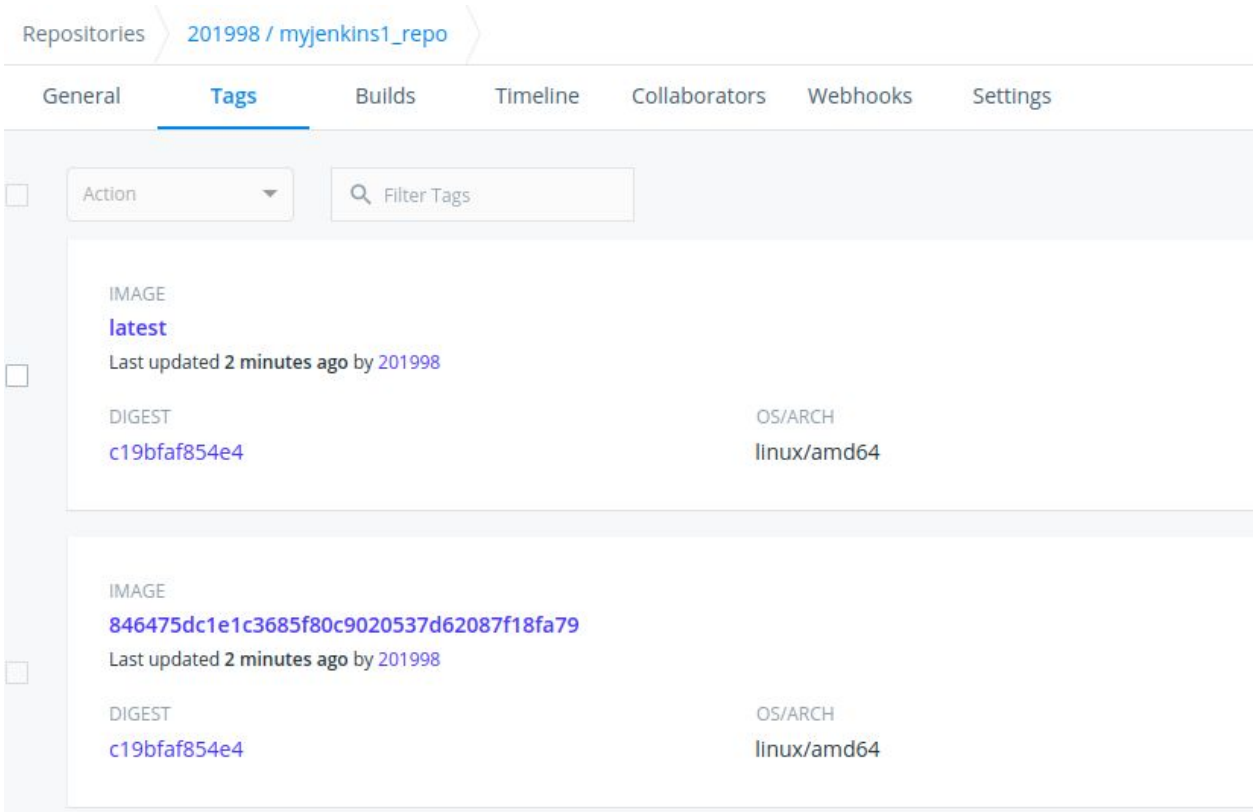


*Docker Host URI: Our Docker host is on the same machine where Jenkins is installed, so we are using default Docker 's default URI by keeping this field empty.



STEP 8: Now build the job

STEP 9: Check your Dockerhub for the commit tag



7. Host a static website on s3.

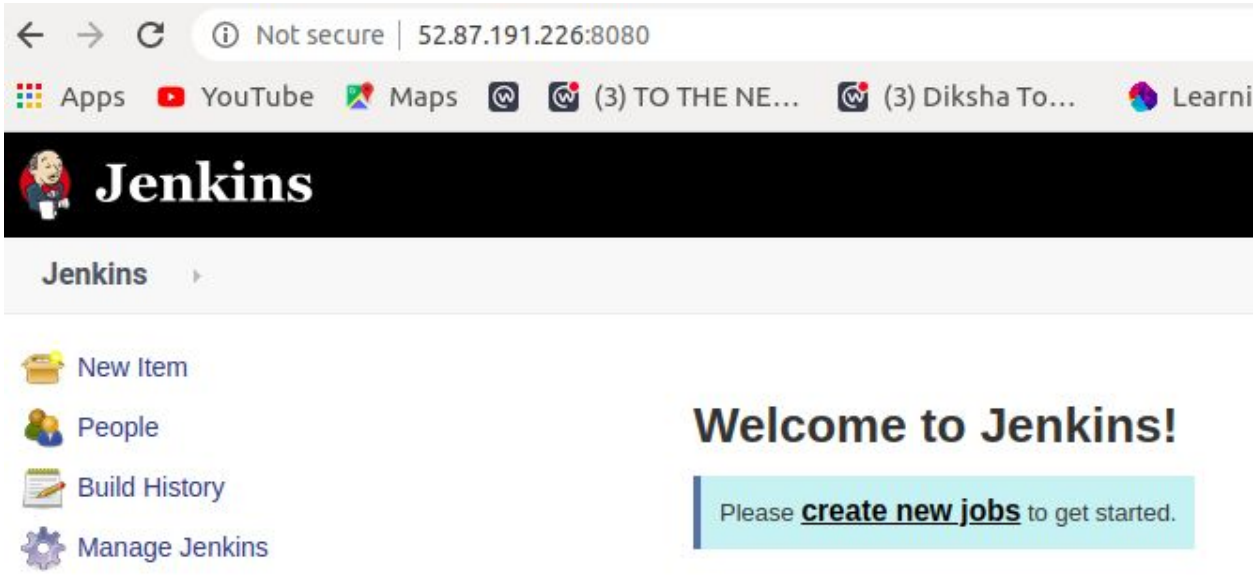
Its static content should be in git repo. When a person commits any change in the repo, the job should automatically reflect the changes in the s3 website.

STEP 1: Install jenkins on ec2 server.

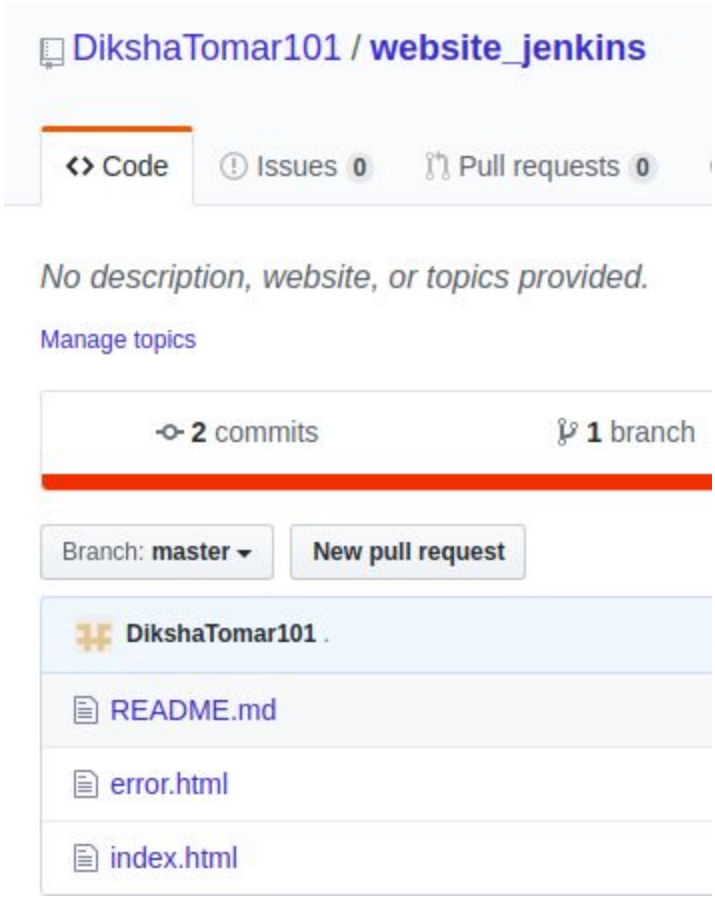
```
ubuntu@ip-172-31-181-104:~$ sudo apt install jenkins
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  daemon
The following NEW packages will be installed:
  daemon jenkins
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 65.3 MB of archives.
After this operation, 66.6 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

STEP 2:

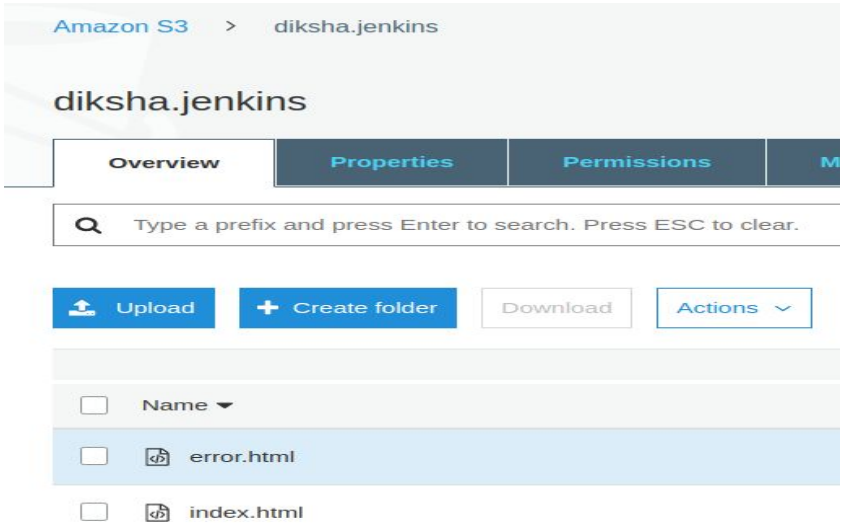
```
ubuntu@ip-172-31-225-166:~$ systemctl status jenkins
● jenkins.service - LSB: Start Jenkins at boot time
   Loaded: loaded (/etc/init.d/jenkins; generated)
   Active: active (exited) since Fri 2020-03-27 08:59:08 UTC; 10min ago
     Docs: man:systemd-sysv-generator(8)
    Tasks: 0 (limit: 1152)
   CGroup: /system.slice/jenkins.service
```



STEP 3 : Create a new repository

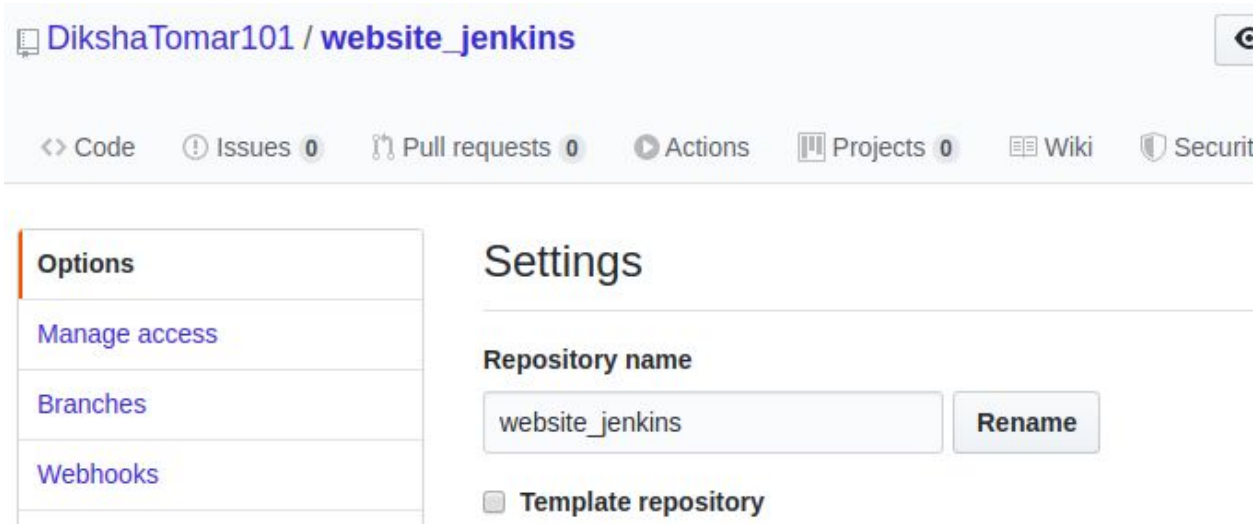


STEP 4: In S3 make your bucket and make a static website

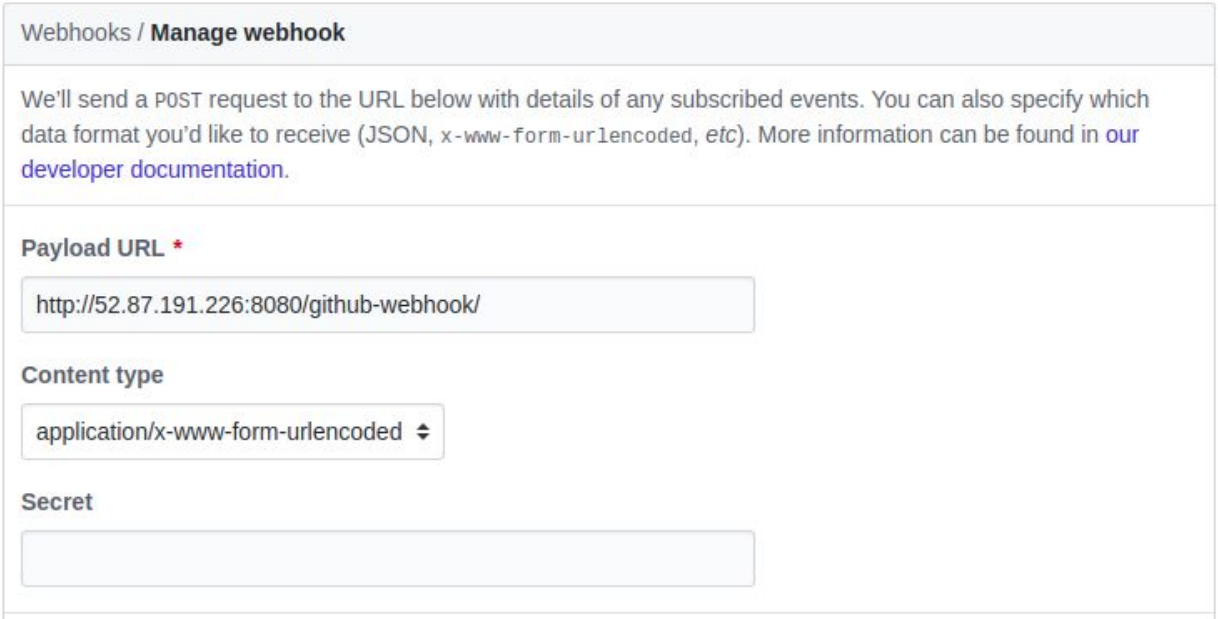


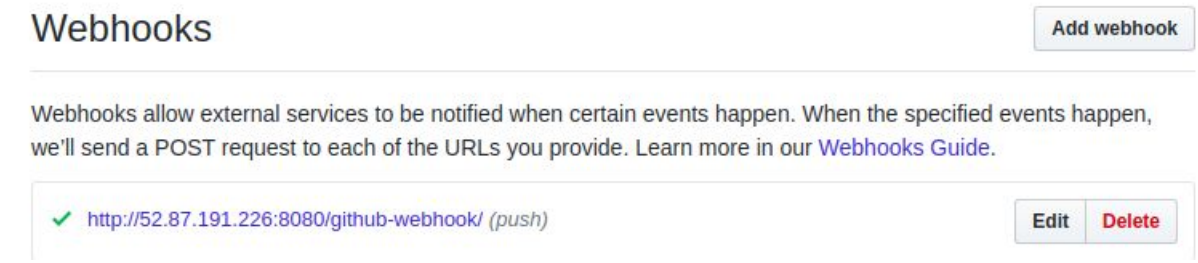
STEP 5: Check your website url if its working or not

STEP 6: Make a repository in git

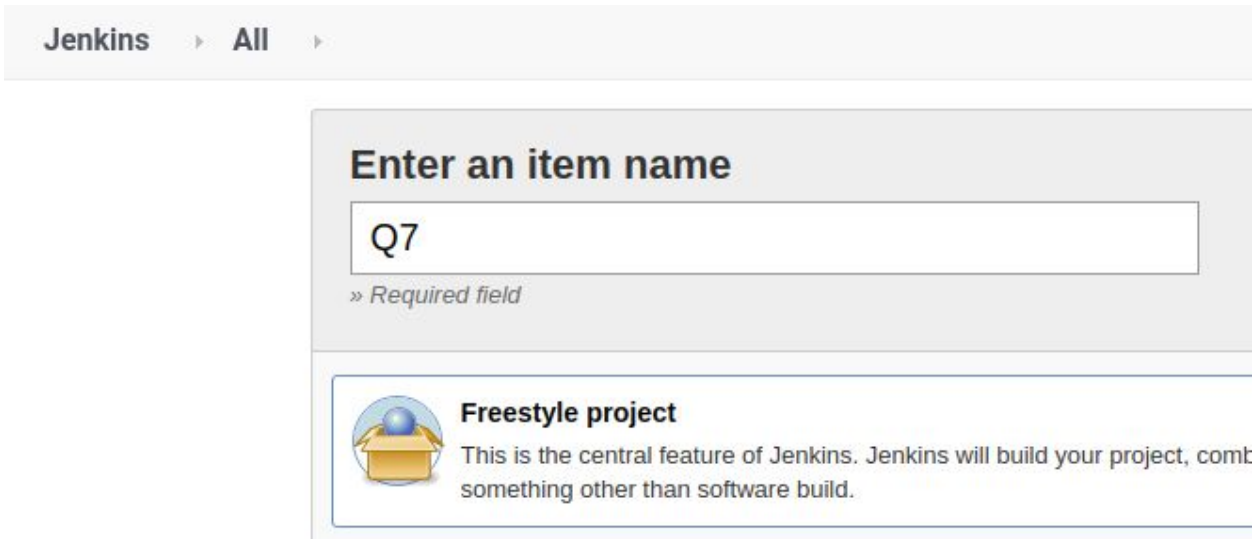


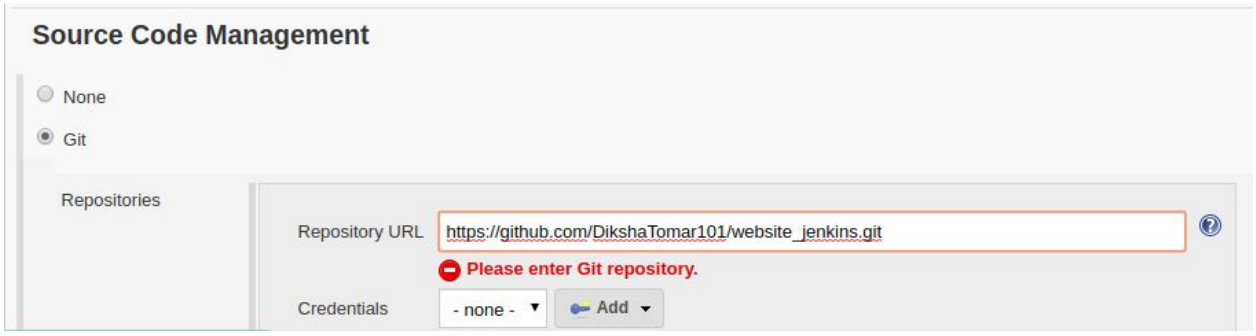
STEP 7: Apply webhooks to it





STEP 8: Make a new job





Build Triggers

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

Build

Execute shell

Command

```
#!/bin/bash
aws s3 cp * s3://diksha.jenkins/
```

See the list of available environment variables

Override Hook URL

- [x] Specify another hook URL for GitHub configuration

http://52.87.191.226:8080/github-webhook/

Shared secret

- none - ▾ Add ▾

```
ubuntu@ip-172-31-225-166:~$ cd website_jenkins/
ubuntu@ip-172-31-225-166:~/website_jenkins$ echo "Change made to index file" >in
dex.html
ubuntu@ip-172-31-225-166:~/website_jenkins$ git add .
ubuntu@ip-172-31-225-166:~/website_jenkins$ git commit -m "New commit"
[master 4423937] New commit
 Committer: Ubuntu <ubuntu@ip-172-31-225-166.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(+), 6 deletions(-)
```

```
ubuntu@ip-172-31-225-166:~/website_jenkins$ git push origin master
Username for 'https://github.com': DikshaTomar101
Password for 'https://DikshaTomar101@github.com':
Counting objects: 3, done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 288 bytes | 288.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/DikshaTomar101/website_jenkins.git
   4331df1..4423937  master -> master
```

Change made to index file