

# **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

**Enter an item name**

» Required field

**Freestyle project**  
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.


-  [Back to Project](#)
-  [Status](#)
-  [Changes](#)
-  **Console Output**
  -  [View as plain text](#)
-  [Edit Build Information](#)
-  [Delete build '#1'](#)
-  [Next Build](#)

## Console Output

```
Started by user Diksha Tomar  
Running as SYSTEM  
Building in workspace /var/lib/jenkins/workspace/Job1  
[Job1] $ /bin/sh -xe /tmp/jenkins13715890696520300354.sh  
+ echo Hello job1  
Hello job1  
Finished: SUCCESS
```

**Enter an item name**

» Required field

**Freestyle project**  
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and something other than software build.

Jenkins > Job2 > #1

[Back to Project](#)  
[Status](#)  
[Changes](#)  
**Console Output**  
[View as plain text](#)  
[Edit Build Information](#)  
[Delete build '#1'](#)

## Console Output

Started by user [Diksha Tomar](#)  
Running as SYSTEM  
Building in workspace /var/lib/jenkins/workspace/Job2  
[Job2] \$ /bin/sh -xe /tmp/jenkins1841183973604626873.sh  
+ echo Hello Job2  
Hello Job2  
Finished: SUCCESS

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



### Manage Users

Create/delete/modify users that can log in to this Jenkins

**STEP 3:** Create two users Developer1 and Developer2









## Create User

Username:	<input type="text" value="Developer1"/>
Password:	<input type="password" value="....."/>
Confirm password:	<input type="password" value="....."/>
Full name:	<input type="text" value="Developer1"/>
E-mail address:	<input type="text" value="developer1@gmail.com"/>

Create User




## 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	
 <a href="#">Developer1</a>	<a href="#">Developer1</a>	 
 <a href="#">Developer2</a>	<a href="#">Developer2</a>	 
 <a href="#">DikshaTomar</a>	<a href="#">Diksha Tomar</a>	

**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”.

Jenkins > Plugin Manager

 Back to Dashboard  
 Manage Jenkins  
 Update Center

Filter:

Updates Available Installed Advanced

Install	Name	Version
<input checked="" type="checkbox"/>	<a href="#">Role-based Authorization Strategy</a> Enables user authorization using a Role-Based strategy. Roles can be defined globally or for particular jobs or nodes selected by regular expressions.	2.16

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



## Configure Global Security

☒ Enable security

## Authorization

- ☐ Anyone can do anything
- ☐ Legacy mode
- ☐ Logged-in users can do anything
- ☐ Matrix-based security
- ☐ Project-based Matrix Authorization Strategy
- ☒ Role-Based Strategy

### STEP 6: Creating User Roles on Jenkins

Go to “Jenkins > Manage and Assign Roles > Manage Roles”.



## Manage and Assign Roles



[Manage Roles](#)

Manage Roles

### Item roles

	Role	Pattern	Credentials					Job						
			Create	Delete	ManageDomains	Update	View	Build	Cancel	Configure	Create	Delete	Discover	Move
<input checked="" type="checkbox"/>	Developer1Role	<a href="#">"Job1"</a>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	Developer2Role	<a href="#">"Job2"</a>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	Developer2viewjob1	<a href="#">"Job1"</a>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

## STEP 7: Assign roles to users

Go to “Jenkins > Manage and Assign Roles > Assign Roles”.

### Assign Roles

#### Global roles

	User/group	Read	admin	
	Diksha Tomar	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
	Anonymous	<input type="checkbox"/>	<input type="checkbox"/>	
	Developer1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	Developer2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

User/group to add

Developer2

Add

#### Item roles

	User/group	Developer1Role	Developer2Role	Developer2viewjob1	
	Anonymous	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	Developer1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	Developer2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

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

Developer1 | log out [ENABLE AUTO REFRESH](#)

All

S	W	Name	Last Success	Last Failure	Last Duration
		<a href="#">Job1</a>	2 hr 30 min - <a href="#">#2</a>	N/A	34 ms

Icon: [S](#) [M](#) [L](#)

[Legend](#) [Atom feed for all](#) [Atom feed for failures](#) [Atom feed for just latest builds](#)

Jenkins
Job1

Back to Dashboard
 Status
 Changes
 Build Now

# Project Job1

My first job

[Recent Changes](#)

## Build History

[trend](#)

find X

	<a href="#">#4</a>	9 Mar 2020, 14:09
	<a href="#">#3</a>	9 Mar 2020, 14:09
	<a href="#">#2</a>	9 Mar 2020, 11:38
	<a href="#">#1</a>	9 Mar 2020, 11:38

## Permalinks

- [Last build \(#2\), 2 hr 30 min ago](#) ▼
- [Last stable build \(#2\), 2 hr 30 min ago](#)
- [Last successful build \(#2\), 2 hr 30 min ago](#)
- [Last completed build \(#2\), 2 hr 30 min ago](#)

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

Jenkins

Jenkins

People
Build History
My Views

### Build Queue

No builds in the queue.

All

S	W	Name	Last Success	Last Failure	Last Duration
		<a href="#">Job1</a>	18 sec - <a href="#">#4</a>	N/A	28 ms
		<a href="#">Job2</a>	2 hr 29 min - <a href="#">#1</a>	N/A	29 ms

Icon: [S](#) [M](#) [L](#)

[Legend](#)
[Atom feed for all](#)
[Atom feed for failures](#)
[Atom feed for just latest builds](#)

 [Back to Dashboard](#)

 [Status](#)

 [Changes](#)

 [Build Now](#)

 [Configure](#)

 [Rename](#)

 **Build History** [trend](#) 



 <b>#1</b>	9 Mar 2020, 11:40
---	-------------------

 [Atom feed for all](#)  [Atom feed for failures](#)

## Project Job2

My job2





[Recent Changes](#)

### Permalinks

- [Last build \(#1\), 2 hr 30 min ago](#)
- [Last stable build \(#1\), 2 hr 30 min ago](#)
- [Last successful build \(#1\), 2 hr 30 min ago](#)
- [Last completed build \(#1\), 2 hr 30 min ago](#)

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

 [Back to Dashboard](#)


 [Status](#)

 [Changes](#)

 **Build History** [trend](#) 



 <b>#4</b>	9 Mar 2020, 14:09
 <b>#3</b>	9 Mar 2020, 14:09
 <b>#2</b>	9 Mar 2020, 11:38
 <b>#1</b>	9 Mar 2020, 11:38

 [Atom feed for all](#)  [Atom feed for failures](#)

## Project Job1

My first job



[Recent Changes](#)

### Permalinks

- [Last build \(#4\), 43 sec ago](#)
- [Last stable build \(#4\), 43 sec ago](#)
- [Last successful build \(#4\), 43 sec ago](#)
- [Last completed build \(#4\), 43 sec ago](#)

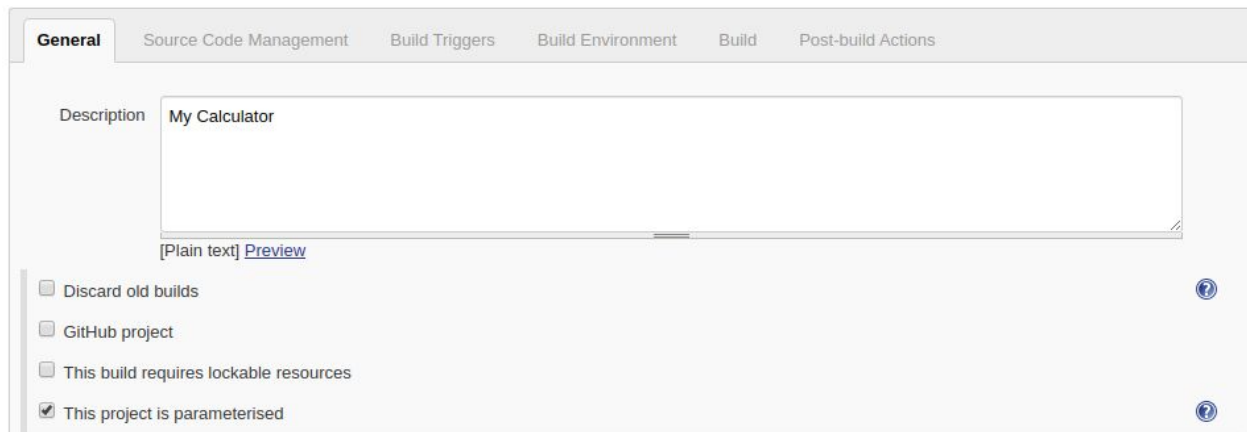


2. Create a Jenkins Job to create a calculator. It should give you a dropdown to ask Addition, Subtraction, Multiplication or Division and email the results.

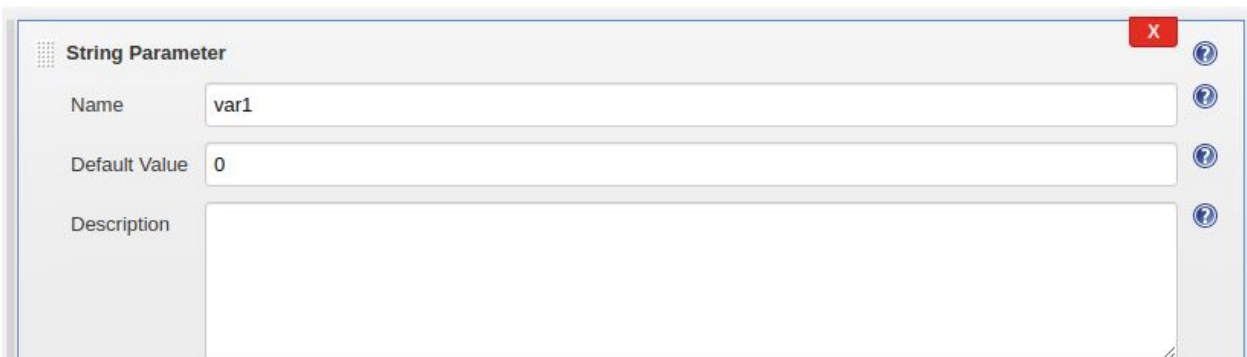
*Hint: Plugin Used (Environment Injector, Extended Email Notification)*

**ANS:**

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



This screenshot shows the 'General' tab of a Jenkins job configuration page. The job name is 'My Calculator'. The description field contains the text 'My Calculator'. Below the description, there are four checkboxes: 'Discard old builds', 'GitHub project', 'This build requires lockable resources', and 'This project is parameterised' (which is checked). A 'Preview' link is visible below the description field.



This screenshot shows the configuration for a 'String Parameter'. The 'Name' field is 'var1', the 'Default Value' is '0', and the 'Description' field is empty. There are help icons (question marks) and a close button (X) in the top right corner.




This screenshot shows the configuration for a 'Choice Parameter'. The 'Name' field is 'Choice', and the 'Choices' field contains a list of operators: '+', '-', '\*', and '/'. The 'Description' field is empty. There are help icons (question marks) and a close button (X) in the top right corner.



The image shows the 'String Parameter' configuration window in Jenkins. It has a title bar with a red 'X' and a help icon. The main area contains three fields: 'Name' with the value 'var2', 'Default Value' with the value '0', and a large 'Description' text area. Below the description area, there is a '[Plain text] Preview' link and a checkbox labeled 'Trim the string' which is currently unchecked. There are also help icons on the right side of the window.

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



The image shows the 'Build' configuration window in Jenkins. It has a title bar with a red 'X' and a help icon. The main area is titled 'Execute shell' and contains a 'Command' field with the following script:

```
#!/bin/bash
case $Choice in
+) echo $var1 + $var2 | bc;;
-) echo $var1 - $var2 | bc;;
*) echo $var1 \* $var2 | bc;;
/) echo $var1 / $var2 | bc;;
esac
```

Below the command field, there is a link that says 'See the list of available environment variables'. At the bottom right, there is an 'Advanced...' button. At the bottom left, there is an 'Add build step' button with a dropdown arrow.

**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

**Extended E-mail Notification**

SMTP server

smtp.gmail.com

Default user E-mail suffix

Project name

## Post-build Actions

### Editable Email Notification

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

Save

Apply

Plain Text (text/plain)

ResultOfCalculator

Attach Build Log

Attach Build Log

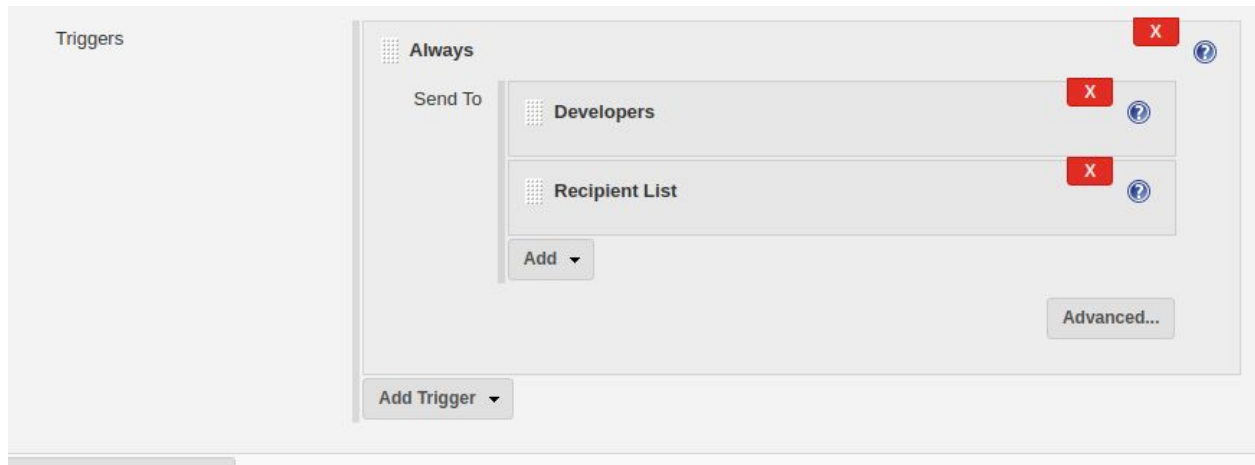
Content Token Reference

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
```

## ResultOfCalculator Inbox x

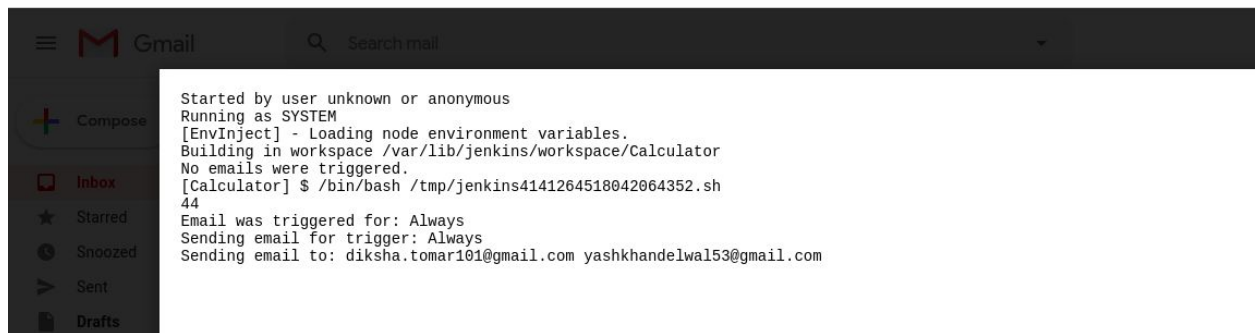
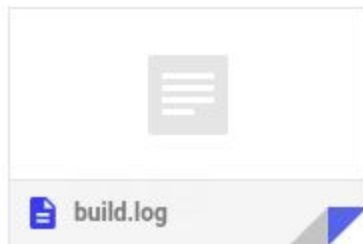


**diksha.tomar@tothenew.com**

to me, yashkhandelwal53 ▾

Calculator - Build # 13 - Successful:

Check console output at <http://localhost:8080/job/Calculator/13/> to view the results.




**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.

**Enter an item name**  
  
» Required field

**Freestyle project**  
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

☒ This project is parameterised

Choice Parameter

X

Name

Env

Choices

ENV1  
ENV2  
ENV3

Description

**STEP 2:** Build the job using script.

Build

Execute shell

Command

```
sudo rm -rf /var/www/html/*
sudo cp ~/jenkinswork/$Env.html /var/www/html
```

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

```
diksha@diksha:~/jenkinswork$ ls
ENV1.html ENV2.html ENV3.html
diksha@diksha:~/jenkinswork$
```

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

```
diksha@diksha:~/jenkinswork$ vi /etc/sudoers

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

**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.



## Console Output

```
Started by user unknown or anonymous
Running as SYSTEM
[EnvInject] - Loading node environment variables.
Building in workspace /var/lib/jenkins/workspace/Parameterized
[Parameterized] $ /bin/sh -xe /tmp/jenkins12909709816574580416.sh
+ sudo rm -rf /var/www/html/*
+ sudo cp /var/lib/jenkins/jenkinswork/ENV2.html /var/www/html
Finished: SUCCESS
```

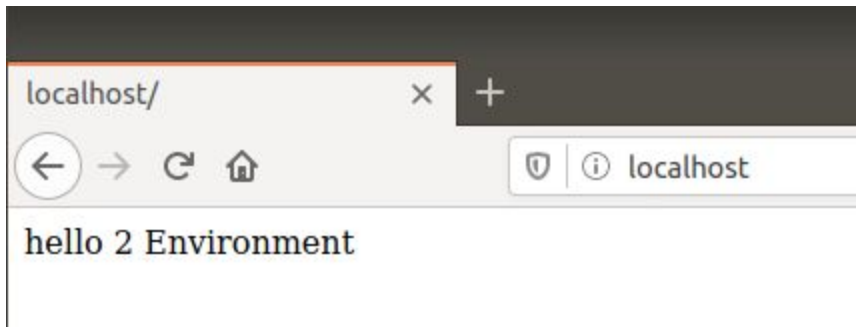
## Project Parameterized

This build requires parameters:

Env

**Build**

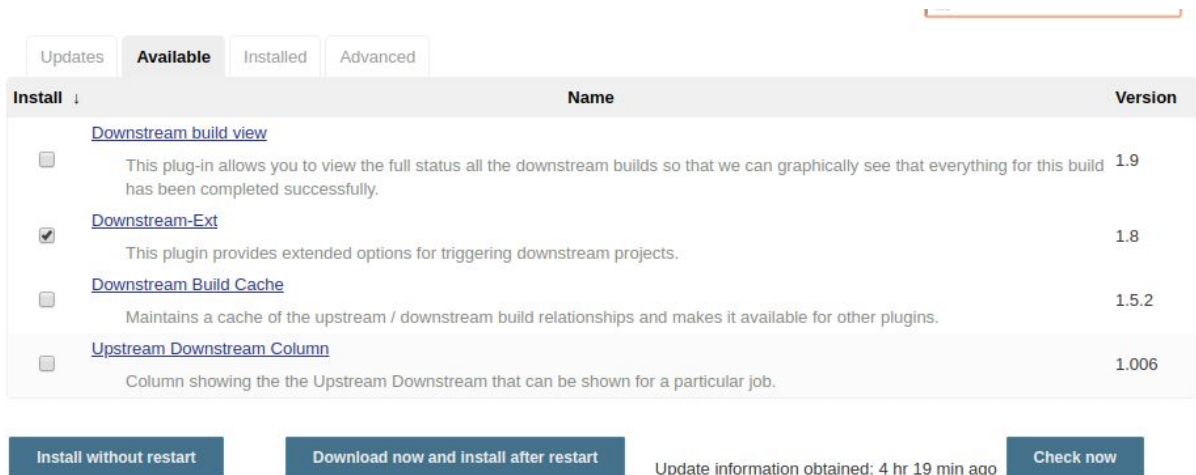




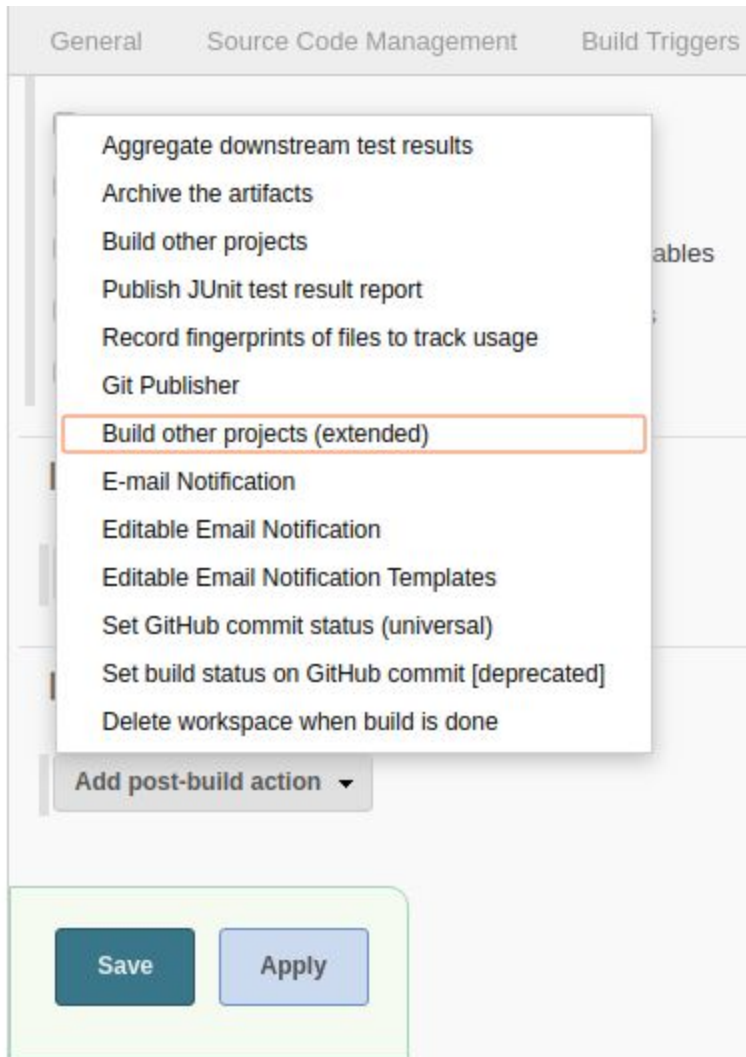
**4. Create a job which on its failure will trigger another job.**

**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

#### Post-build Actions

**Build other projects (extended)**

Projects to build

Build result is

Trigger downstream job if build result meets condition

☐ Trigger only if downstream project has SCM changes









☐ Trigger only if current project has SCM changes

Add post-build action ▼

**STEP 4:** For error :-



**STEP 5:** Build the job and observe the changes

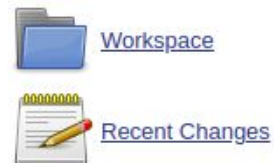
-  [Back to Dashboard](#)
-  [Status](#)
-  [Changes](#)
-  [Workspace](#)
-  [Build Now](#)
-  [Delete Project](#)
-  [Configure](#)
-  [Rename](#)

 **Build History** [trend](#)

 <b>#2</b>	9 Mar 2020, 16:11
 <b>#1</b>	9 Mar 2020, 16:11

 [Atom feed for all](#)  [Atom feed for failures](#)

## Project Q4



## Downstream Projects

 [Job1](#)

## Permalinks

- [Last build \(#2\), 27 sec ago](#)
- [Last failed build \(#2\), 27 sec ago](#)
- [Last unsuccessful build \(#2\), 27 sec ago](#)
- [Last completed build \(#2\), 27 sec ago](#)

 [Back to Dashboard](#)

 [Status](#)

 [Changes](#)

 [Workspace](#)

 [Build Now](#)

 [Delete Project](#)

 [Configure](#)

 [Rename](#)

## Project Job1

My first job

 [Workspace](#)






 [Recent Changes](#)

## Upstream Projects

 Q4

## Permalinks

- [Last build \(#5\), 43 sec ago](#)
- [Last stable build \(#5\), 43 sec ago](#)
- [Last successful build \(#5\), 43 sec ago](#)
- [Last completed build \(#5\), 43 sec ago](#)

Build History <span>trend —</span>	
<input type="text" value="find"/>	
 <a href="#">#5</a>	9 Mar 2020, 16:11
 <a href="#">#4</a>	9 Mar 2020, 14:09
 <a href="#">#3</a>	9 Mar 2020, 14:09
 <a href="#">#2</a>	9 Mar 2020, 11:38
 <a href="#">#1</a>	9 Mar 2020, 11:38

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.

Updates

Available

Installed

Advanced

Install

Name

☐

[Publish Over SSH](#)  
Send build artifacts over SSH  
[SCP publisher](#)  
This plugin uploads build artifacts to repository sites using SCP (SSH) proto  

Warning: This plugin version may not be safe to use. Please review the follow

- [Insecure credential storage and transmission](#)

☒

[SSH](#)  
This plugin executes shell commands remotely using SSH protocol.

[Back to credential domains](#)

[Add Credentials](#)



## Global credentials (unrestricted)

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

Jenkins > Credentials > System >

New Item

People

Build History

Manage Jenkins

Lockable Resources

Credentials

System

Add domain

New View

System

Domain

Global credentials (unrestricted)

Icon: [S](#) [M](#) [L](#)

[Back to credential domains](#)

[Add Credentials](#)

Kind: SSH Username with private key

Scope: Global (Jenkins, nodes, items, all child items, etc)

ID:

Description: ubuntu passwd

Username: ubuntu

Private Key: ☒ Enter directly

Key: 

Enter New Secret Below

j8p3Bh/6Sr6P2La10ejaFF/gKrtHxYArKKbK42E66cWD9a3FmTZ10JtwbKzSbJ4P9LTx0uL4G5jL  
ut4m9CTLs0bIZZmep3AgbzCpYNfc/C41bTmUtlcB0FedLczIb69lOmYrdDFZphv0LV4G  
-----END RSA PRIVATE KEY-----

### STEP 3: Manage jenkins > Configure System > SSH Hosts

SSH remote hosts

SSH sites

Hostname

Port

Credentials

Pty ☐

serverAliveInterval

timeout

Successfull connection

### STEP 4: SSH into your instance and create a script

```
diksha@diksha:~/Downloads$ ssh -i "diksha_awskey2.pem" ubuntu@ec2-34-230-43-75.compute-1.amazonaws.com
```

```
ubuntu@ip-172-31-241-48:~$ cat myscript.sh
echo " $1 $2 ";
ubuntu@ip-172-31-241-48:~$ ./myscript.sh 1 2
1 2
ubuntu@ip-172-31-241-48:~$
```

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

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

☒ This project is parameterised

String Parameter

Name

Default Value

Description

[Plain text] [Preview](#)

☐ Trim the string

String Parameter

Name

Default Value

Description

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

**Build**

Execute shell script on remote host using ssh

SSH site

ubuntu@34.230.43.75:22

Command

```
./myscript.sh $Param1 $Param2  
echo "*" | * * * /home/ubuntu/myscript.sh" | crontab -
```

Execute each line

☐


Hide command from console output


☐


Add build step ▾


## STEP 7: Build the job and give parameters value


Jenkins ▸ Q5 ▸


 Back to Dashboard


 Status


 Changes

 Workspace

 Build with Parameters

 Delete Project

 Configure

 Rename

## Project Q5

This build requires parameters:

Param1

Hello

Param2

DIKSHA

Build



Back to Project

Status

Changes

Console Output

View as plain text


Edit Build Information

Delete build '#6'

Parameters

Environment Variables

Previous Build



## Console Output

Started by user unknown or anonymous

Running as SYSTEM

[EnvInject] - Loading node environment variables.

Building in workspace /var/lib/jenkins/workspace/Q5

[SSH] script:

Param2="DIKSHA"

Param1="Hello"

./myscript.sh \$Param1 \$Param2

echo " \* \* \* \* \* /home/ubuntu/myscript.sh" | crontab -

[SSH] executing...

Hello DIKSHA

[SSH] completed

[SSH] exit-status: 0

Finished: SUCCESS

```
FROM ubuntu
MAINTAINER demousr@gmail.com

RUN apt-get update
RUN apt-get install -y nginx
CMD ["echo","Image created"]
```

~

~

~

~

"Dockerfile" 7L, 159C written



DikshaTomar101 / Jenkins

<> Code ⓘ Issues 0 🔗 Pull requests 0 ▶ Actions

Jenkins / Dockerfile Cancel

<> Edit file 👁 Preview changes

```

1  #This is a sample Image
2  FROM ubuntu
3  MAINTAINER demousr@gmail.com
4
5  RUN apt-get update
6  RUN apt-get install -y nginx
7  CMD ["echo","Image created"]
8  CMD ["nginx", "-g", "daemon off;"]
9

```

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

DikshaTomar101 / Jenkins

👁 Unwatch 1 ⭐ Star 0 🍴 Fork 0

<> Code ⓘ Issues 0 🔗 Pull requests 0 ▶ Actions 📁 Projects 0 📖 Wiki 🛡 Security 📊 Insights ⚙ Settings

No description, website, or topics provided. Edit

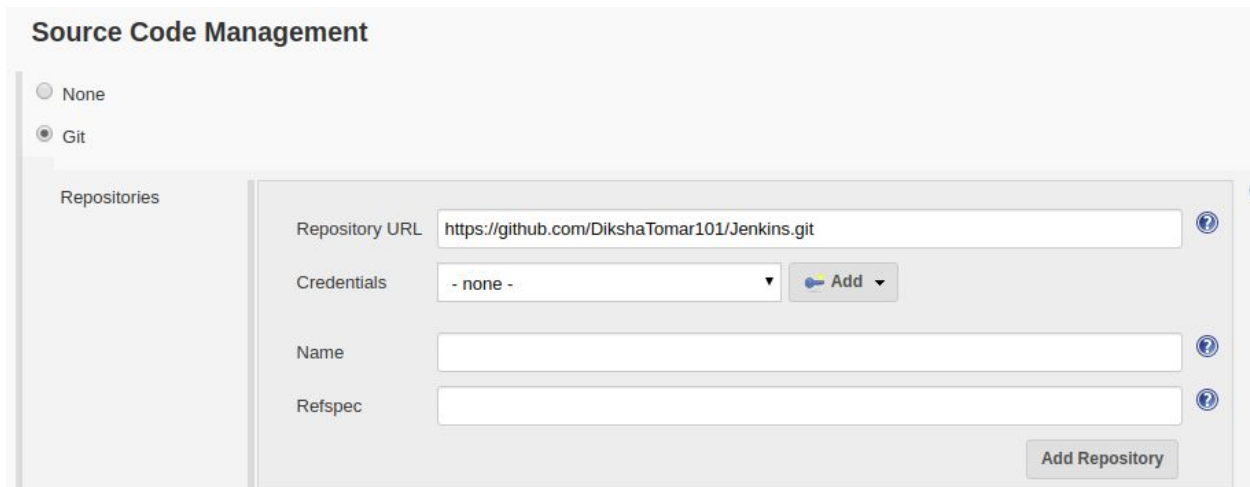
Manage topics

📁 2 commits 🌿 1 branch 📦 0 packages 📦 0 releases 👤 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

DikshaTomar101 1		Latest commit c07f56a now
Dockerfile	1	now
README.md	Initial commit	4 hours ago

**STEP 3:** IN SCM provide a git url.



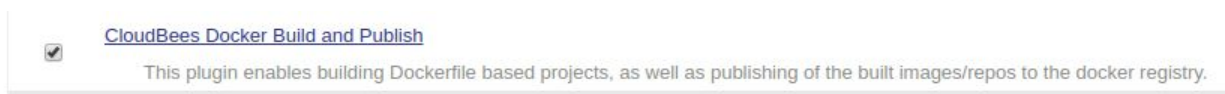
The image shows the 'Source Code Management' configuration page in Jenkins. On the left, there are radio buttons for 'None' and 'Git', with 'Git' selected. Below this is a 'Repositories' section. The main area contains four input fields: 'Repository URL' with the value 'https://github.com/DikshaTomar101/Jenkins.git', 'Credentials' with a dropdown menu showing '- none -' and an 'Add' button, 'Name' (empty), and 'Refspec' (empty). Each input field has a help icon (question mark) to its right. At the bottom right, there is an 'Add Repository' button.

## Deploy

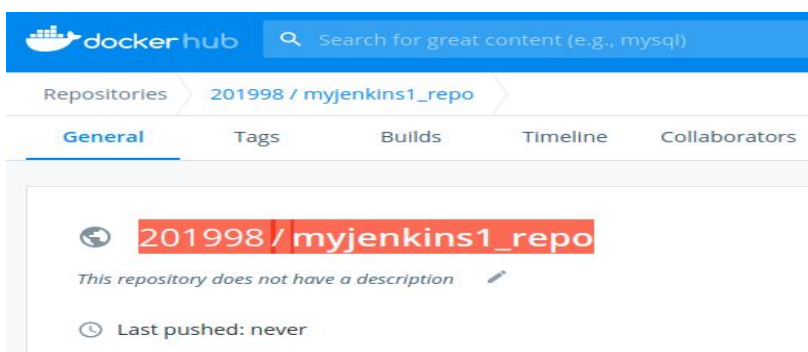
Now we're ready to create the job which will do the work for us. Click on New Item on the left side menu, and enter the item name as desired job name. Then select the *Freestyle project*.

**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

[Back to credential domains](#)

[Add Credentials](#)

Kind	Username with password
Scope	Global (Jenkins, nodes, items, all child items, etc)
Username	201998
Password	*****
ID	
Description	dockerhub credentials

## 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:

General	Source Code Management	Build Triggers	Build Environment	<b>Build</b>	Post-build Actions
---------	------------------------	----------------	-------------------	--------------	--------------------

Docker Build and Publish

X

?

Repository Name

201998/myjenkins1\_repo

?

Tag

\${GIT\_COMMIT}

?

Docker Host URI

?

Server credentials

- none -

Add

Docker registry URL

?

Registry credentials

201998/\*\*\*\*\* (dockerhub credentials)

Add

Advanced...

\*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.

```
diksha@diksha:~$ sudo usermod -aG docker jenkins
diksha@diksha:~$ id jenkins
uid=125(jenkins) gid=129(jenkins) groups=129(jenkins),27(sudo),998(docker)
diksha@diksha:~$
```

## STEP 8: Now build the job

Back to Project

Status

Changes

**Console Output**

View as plain text

Edit Build Information

Environment Variables

Git Build Data

No Tags

Docker Fingerprints

Previous Build

### Console Output

```
Started by user unknown or anonymous
Running as SYSTEM
[EnvInject] - Loading node environment variables.
Building in workspace /var/lib/jenkins/workspace/Q6
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/DikshaTomar101/Jenkins.git # timeout=10
Fetching upstream changes from https://github.com/DikshaTomar101/Jenkins.git
> git --version # timeout=10
> git fetch --tags --progress -- https://github.com/DikshaTomar101/Jenkins.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
> git rev-parse refs/remotes/origin/origin/master^{commit} # timeout=10
Checking out Revision 846475dc1e1c3685f80c9020537d62087f18fa79 (refs/remotes/origin/master)
```

## STEP 9: Check your Dockerhub for the commit tag

Repositories

201998 / myjenkins1\_repo

General

**Tags**

Builds

Timeline

Collaborators

Webhooks

Settings

Action

Filter Tags

IMAGE

**latest**

Last updated 2 minutes ago by 201998

DIGEST

**c19bfaf854e4**

OS/ARCH

**linux/amd64**

IMAGE

**846475dc1e1c3685f80c9020537d62087f18fa79**

Last updated 2 minutes ago by 201998

DIGEST

**c19bfaf854e4**

OS/ARCH

**linux/amd64**

## 7. Host a static website on s3.

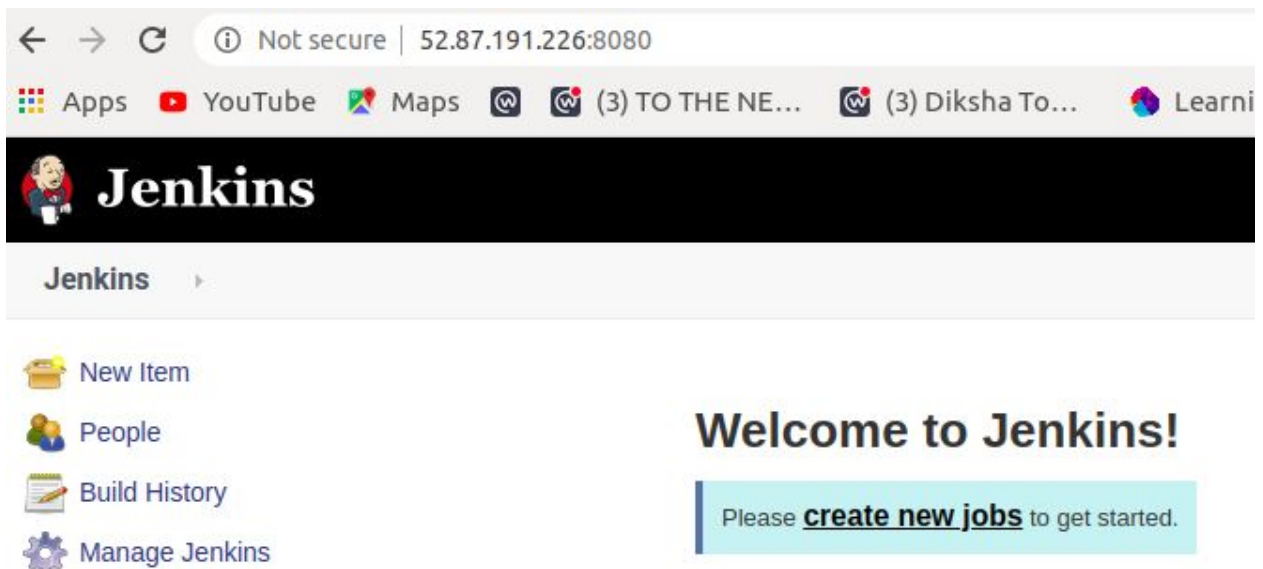
Its static content should be in git repo. When a person commits any change in the repo, the job should automatically reflect the changes in the s3 website.

STEP 1: Install jenkins on ec2 server.

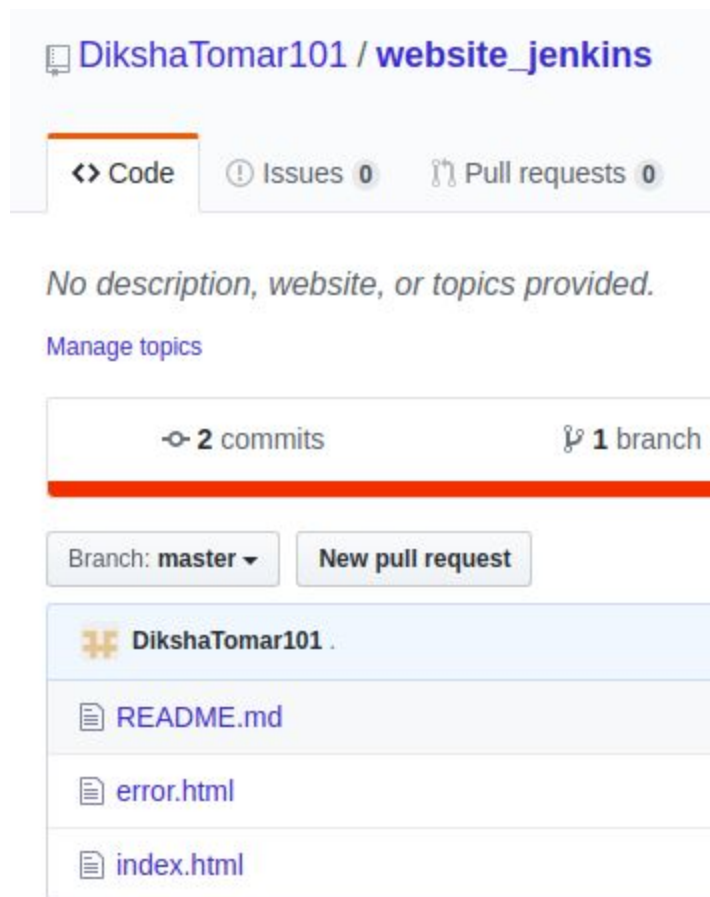
```
ubuntu@ip-172-31-181-104:~$ sudo apt install jenkins
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  daemon
The following NEW packages will be installed:
  daemon jenkins
0 upgraded, 2 newly installed, 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:

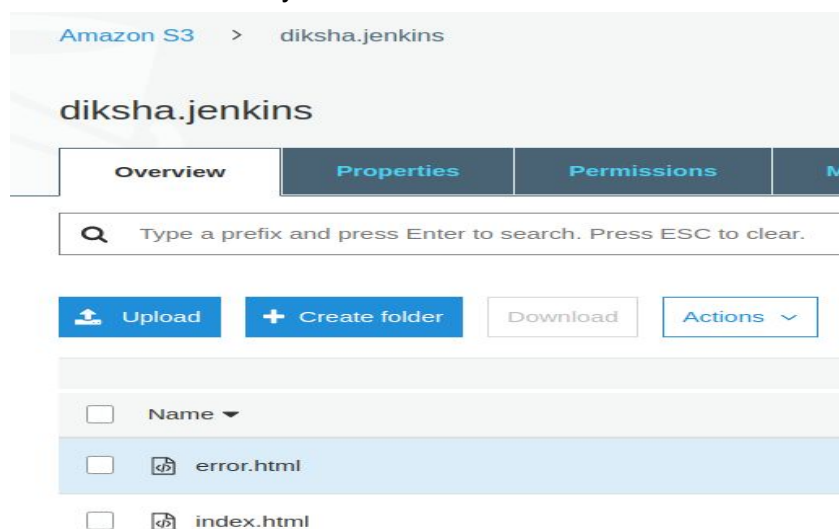
```
ubuntu@ip-172-31-225-166:~$ systemctl status jenkins
● jenkins.service - LSB: Start Jenkins at boot time
   Loaded: loaded (/etc/init.d/jenkins; generated)
   Active: active (exited) since Fri 2020-03-27 08:59:08 UTC; 10min ago
     Docs: man:systemd-sysv-generator(8)
    Tasks: 0 (limit: 1152)
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



### STEP 3 : Create a new repository



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



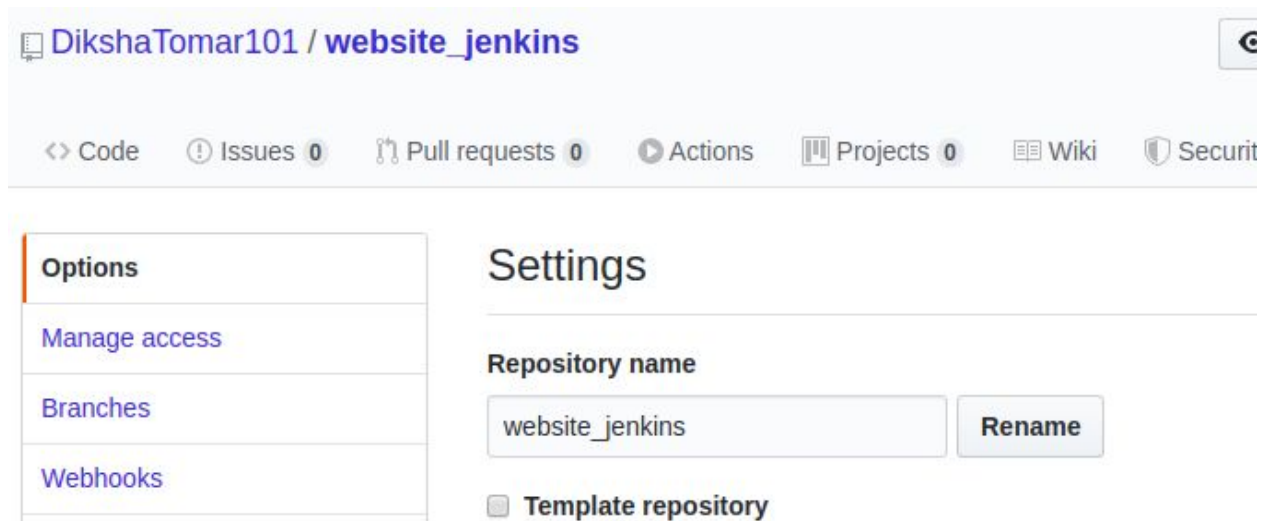


**STEP 5:** Check your website url if its working or not

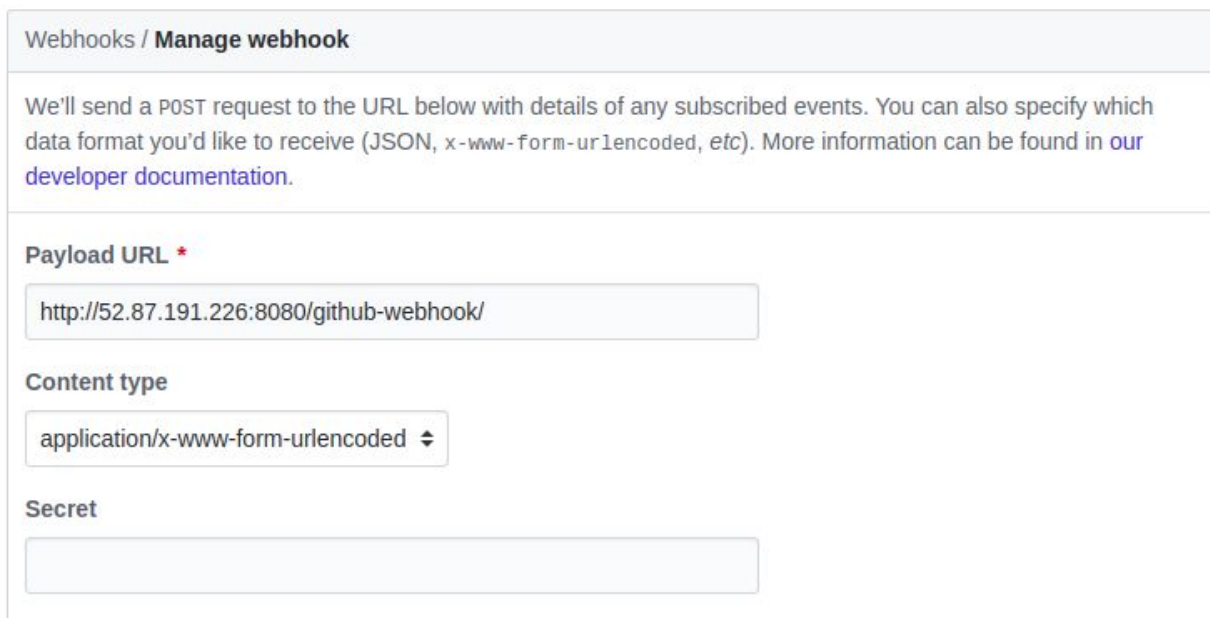


# This is my index page

**STEP 6:** Make a repository in git



**STEP 7:** Apply webhooks to it



## Webhooks

[Add webhook](#)

Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our [Webhooks Guide](#).

✓ <http://52.87.191.226:8080/github-webhook/> (push)

[Edit](#)[Delete](#)

### STEP 8: Make a new job

[Jenkins](#)[All](#)

#### Enter an item name

» Required field



#### Freestyle project

This is the central feature of Jenkins. Jenkins will build your project, compile something other than software build.

#### Source Code Management

☐ None☒ Git

Repositories

Repository URL



⊘ Please enter Git repository.

Credentials

- none -

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

## Build

### Execute shell

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

See [the list of available environment variables](#)


Override Hook URL

☒ 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" >index.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