

**TO
THE
NEW**™



CI/CD : Jenkins

Trainee Name : Chhavi Sharma

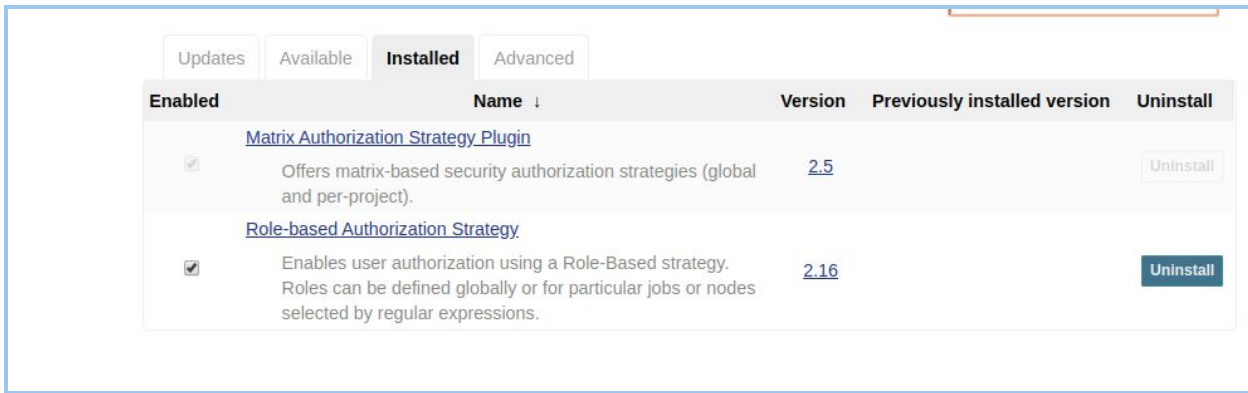
Newers ID : 4023

College : UPES

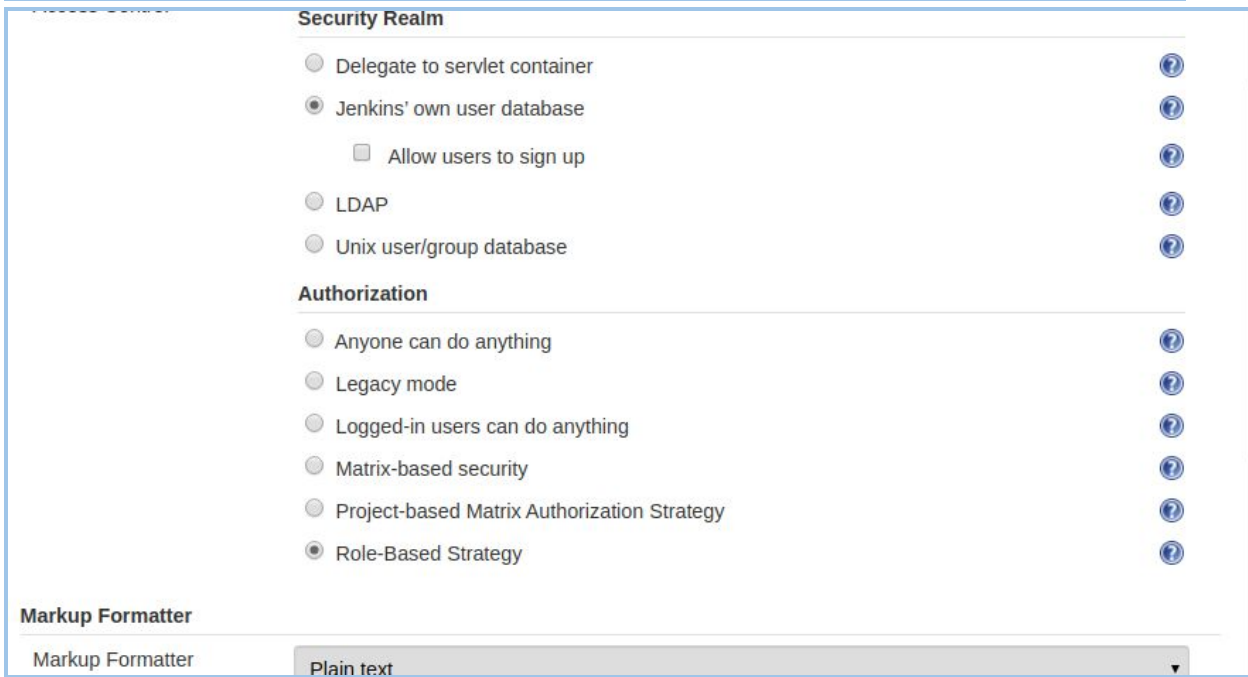
1. Create 2 users: developer1, developer2. The developer1 should be able to build job1 only and can't change the job configuration. The developer2 can configure and build the job2, also he is able to view job1 but can't build/configure it.

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	 



Manage and Assign Roles



[Manage Roles](#)
Manage Roles



[Assign Roles](#)
Assign Roles



[Role Strategy Macros](#)
Provides info about macro usage and available macros

Jenkins > Manage and Assign Roles

Manage Roles

Global roles

Role	Overall				Credentials				Agent								
	Administer	Read	Create	Delete	ManageDomains	Update	View	Build	Configure	Connect	Create	Delete	Disconnect	Provision	Build	Cancel	Configure
<input checked="" type="checkbox"/> admin	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> global read	<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 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"/>

Role to add

Add

Item roles

Role	Pattern	Credentials				Job								Run				
		Create	Delete	ManageDomains	Update	View	Build	Cancel	Configure	Create	Delete	Discover	Move	Read	Workspace	Delete	Replay	Up
<input checked="" type="checkbox"/> dev1	"testJob1"	<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 type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/> dev2	"testJob2"	<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 type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/> job1readonly	"testJob1"	<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 type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Assign Roles

Global roles

User/group	admin	global read
<input checked="" type="checkbox"/> Chhavi Sharma	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/> developer1	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> developer2	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> Anonymous	<input type="checkbox"/>	<input type="checkbox"/>

User/group to add

Add

Item roles

User/group	dev1	dev2	job1readonly
<input checked="" type="checkbox"/> developer1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/> developer2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> Anonymous	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Developer1 console

search

developer1 | log out

Jenkins

ENABLE AUTO REFRESH

People

Build History

My Views

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

All

S	W	Name ↓	Last Success	Last Failure	Last Duration
		testJob1	10 days - #2	N/A	27 ms

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

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

search

developer1 | log out

Jenkins

testJob1

ENABLE AUTO REFRESH

Back to Dashboard

Status

Changes

Build Now

Build History

find

#2

9 Mar 2020, 16:12

#1

5 Mar 2020, 11:25

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

Project testJob1

Recent Changes

Upstream Projects

Permalinks

Last build (#2), 10 days ago

Last stable build (#2), 10 days ago

Last successful build (#2), 10 days ago

Last completed build (#2), 10 days ago

Developer2 console

search

developer2 | log out

Jenkins

ENABLE AUTO REFRESH

People

Build History

My Views

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

All

S	W	Name ↓	Last Success	Last Failure	Last Duration
		testJob1	10 days - #2	N/A	27 ms
		testJob2	6 days 20 hr - #20	N/A	23 ms

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

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

He can build job 2 but not job 1.

The image displays two screenshots of the Jenkins web interface. The top screenshot is for 'Project testJob2'. It features a sidebar on the left with links: 'Back to Dashboard', 'Status', 'Changes', 'Build Now', 'Configure', and 'Rename'. The main content area has a title 'Project testJob2' and a 'Recent Changes' section with a 'Recent Changes' link. Below this is a 'Permalinks' section with four links: 'Last build (#20), 6 days 20 hr ago', 'Last stable build (#20), 6 days 20 hr ago', 'Last successful build (#20), 6 days 20 hr ago', and 'Last completed build (#20), 6 days 20 hr ago'. A 'Build History' table is also visible, showing builds #20 and #19 from March 13, 2020. The bottom screenshot is for 'Project testJob1'. It has a similar sidebar and main content area. The 'Recent Changes' section shows a 'Recent Changes' link. The 'Permalinks' section has four links: 'Last build (#2), 10 days ago', 'Last stable build (#2), 10 days ago', 'Last successful build (#2), 10 days ago', and 'Last completed build (#2), 10 days ago'. The 'Build History' table shows builds #2 and #1 from March 9 and 5, 2020. Both screenshots have a top navigation bar with 'Jenkins' and 'testJob2' or 'testJob1' breadcrumbs, and a right sidebar with 'ENABLE AUTO REFRESH' and 'add description'/'Disable Project' buttons.

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.

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

Description

[Plain text] [Preview](#)

☐ Discard old builds

☐ GitHub project

☐ This build requires lockable resources

☒ This project is parameterised

String Parameter

Name

Default Value

Description

[Plain text] [Preview](#)

Save

Apply

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

String Parameter

Name

Default Value

Description

[Plain text] [Preview](#)

☐ Trim the string

Choice Parameter

Name

Choices

Description

Save

Apply

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

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

Execute shell

Command

```
echo "Result is "
echo `expr $val1 $operator $val2`
echo output=`expr $val1 $operator $val2` > chhavi
```

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

Advanced...

Inject environment variables

Properties File Path chhavi

Properties Content

Configure the project to send email at every build.

Click "Add post-build action".

Click Editable Email Notification

Click Advanced Settings

Editable Email Notification

Disable Extended Email Publisher ☐

Allows the user to disable the publisher, while maintaining the settings

Project From caljob

Project Recipient List chhavi.sharma@tothenew.com

Comma-separated list of email address that should receive notifications for this project.

Project Reply-To List \$DEFAULT_REPLYTO
OUTPUT=\$output

Comma-separated list of email address that should be in the Reply-To header for this project.

Content Type Default Content Type

Default Subject

Save Apply

\$DEFAULT_SUBJECT

OUTPUT=\$output

Click Add Trigger

Click "Always."

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

Additional groovy classpath

Save to Workspace ☐

Triggers

Success

Send To

- Developers
- Recipient List

Add

Advanced...

Add Trigger

Add post-build action

Goto Manage Jenkins -> Configure Systems

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

E-mail Notification

SMTP server

Default user e-mail suffix

☒ Use SMTP Authentication

User Name

Password

Use SSL ☒

SMTP Port

Reply-To Address

Charset

☐ Test configuration by sending test e-mail

Next provide the parameters and build the job.

Project caljob

This build requires parameters:

val1
Value 1

val2
Value 2

operator
Operators

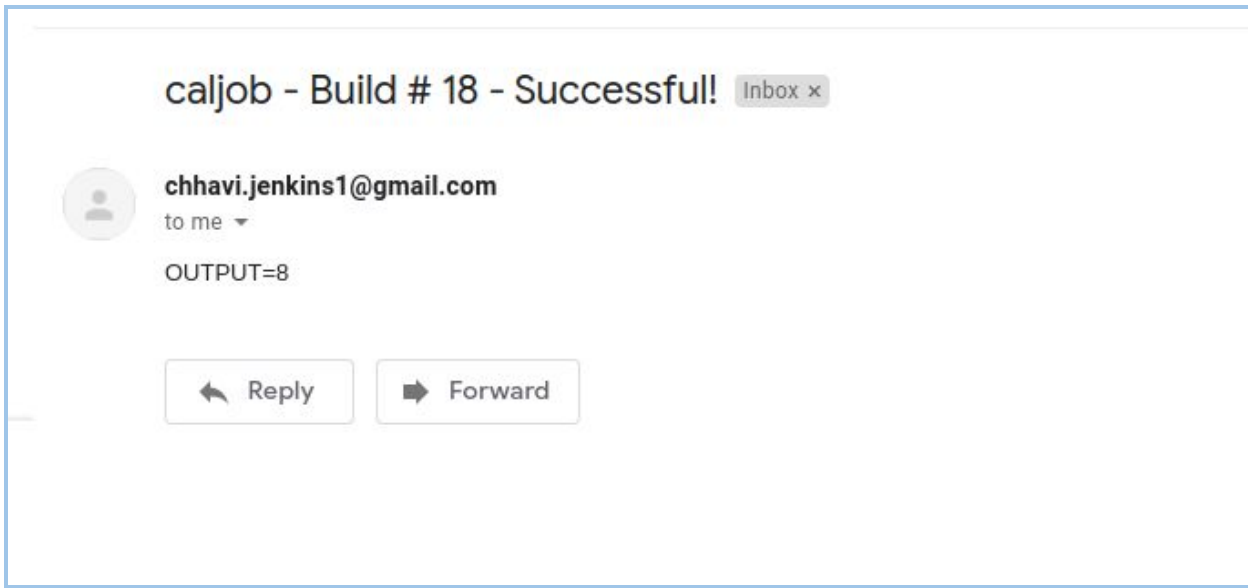
Check the console output for result.



Console Output

```
Started by user unknown or anonymous
Running as SYSTEM
[EnvInject] - Loading node environment variables.
Building in workspace /var/lib/jenkins/workspace/caljob
No emails were triggered.
[caljob] $ /bin/sh -xe /tmp/jenkins684379679457048271.sh
+ echo Result is
Result is
+ expr 4 + 4
+ echo 8
8
+ expr 4 + 4
+ echo output=8
[EnvInject] - Injecting environment variables from a build step.
[EnvInject] - Injecting as environment variables the properties file path 'chhavi'
[EnvInject] - Variables injected successfully.
Email was triggered for: Success
Sending email for trigger: Success
Sending email to: chhavi.sharma@tothenew.com
Finished: SUCCESS
```

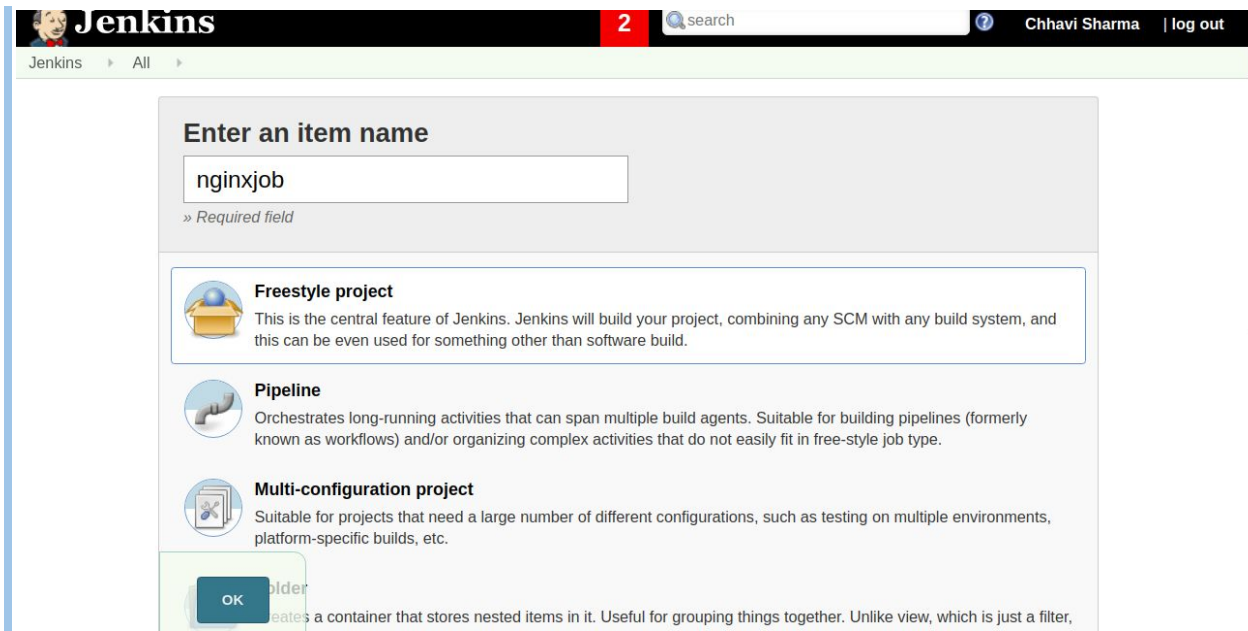
You must have also received the mail on the provided mail id.



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.



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

Description
 [Plain text] [Preview](#)

☐ Discard old builds

☐ GitHub project

☐ This build requires lockable resources

☒ This project is parameterised

Choice Parameter

Name

Choices

Description
 [Plain text] [Preview](#)

Add Parameter

Build

Execute shell

Command

```
FILE1=/var/www/html/ENV1.html
FILE2=/var/www/html/ENV2.html
FILE3=/var/www/html/ENV3.html
if test -f "$FILE1";
then
    sudo rm -rf /var/www/html/ENV1.html
fi
if test -f "$FILE2";
then
    sudo rm -rf /var/www/html/ENV2.html
fi
if test -f "$FILE3";
then
    sudo rm -rf /var/www/html/ENV3.html
fi
sudo cp -r /var/www/html/$ENV.html /var/www/html
```

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

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.

Project nginxjob

This build requires parameters:

ENV

Choose anyone from the provided options

Build

YOU CHOSE ENV 2

Project nginxjob

This build requires parameters:

ENV

Choose anyone from the provided options

Build

YOU CHOSE ENV 1

Project nginxjob

This build requires parameters:

ENV **ENV3** ▼

Choose anyone from the provided options

Build

← → ↻ ⓘ Not secure | nginxenv.com

YOU CHOSE ENV 3

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

Ans.

Install Downstream-Ext Plugin .

[Back to Dashboard](#)

[Manage Jenkins](#)

Filter:

Updates

Available

Installed

Advanced

Install	Name	Version
<input checked="" type="checkbox"/>	Downstream-Ext This plugin provides extended options for triggering downstream projects.	1.8

Install without restart

Download now and install after restart

Update information obtained: 49 sec ago


Check n


Now create a new job (upstream project).


Enter an item name

NewJob

» 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.

 **Pipeline**
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

 **Multi-configuration project**
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

In the Build Action , Write a script with error.

Build

 **Execute shell** ✖ ?

Command

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

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.

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

Project testJob1

[add description](#)
[Disable Project](#)

[Workspace](#)
[Recent Changes](#)

Upstream Projects

[NewJob](#)

Permalinks

- [Last build \(#2\) .14 sec ago](#)
- [Last stable build \(#2\) .14 sec ago](#)
- [Last successful build \(#2\) .14 sec ago](#)
- [Last completed build \(#2\) .14 sec ago](#)

Build History

[trend](#)

#2	9 Mar 2020, 16:12
#1	5 Mar 2020, 11:25

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

- 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
cronsript.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 private key.

Jenkins > Credentials > System > Global credentials (unrestricted) > ubuntu

Back to Global credentials (unrestricted)

Update

Delete

Move

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

ID: ssh_3.87.205.76

Description:

Username: ubuntu

Private Key: Enter directly

Key: Concealed for Confidentiality

Replace

Passphrase:

Save

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

SSH remote hosts

SSH sites

Hostname: 3.87.205.76

Port: 22

Credentials: ubuntu Add

Pty: ☐

serverAliveInterval:

timeout:

Successfull connection

Check connection

Delete

Add

SSH sites that projects will want to connect

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

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

[Plain text] [Preview](#)

Save **Apply**

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.

Build

Execute shell script on remote host using ssh

SSH site

Command

Execute each line ☐

Hide command from console output ☐

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

Project cronjob

This build requires parameters:

par1
Parameter 1

par2
Parameter 2

Step 7: Check the console output for any errors.



Console Output

```
Started by user Chhavi Sharma
Running as SYSTEM
[EnvInject] - Loading node environment variables.
Building in workspace /var/lib/jenkins/workspace/cronjob
[SSH] script:
par1="Good"
par2="Morning"

echo " * * * * /home/ubuntu/cronscript.sh $par1 $par2 > output.txt" | sudo tee /var/spool/cron/crontabs/ubuntu;
sudo chown -R ubuntu:ubuntu /var/spool/cron/crontabs
crontab /var/spool/cron/crontabs/ubuntu

[SSH] executing...
* * * * /home/ubuntu/cronscript.sh Good Morning > output.txt

[SSH] completed
[SSH] exit-status: 0

Finished: SUCCESS
```

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

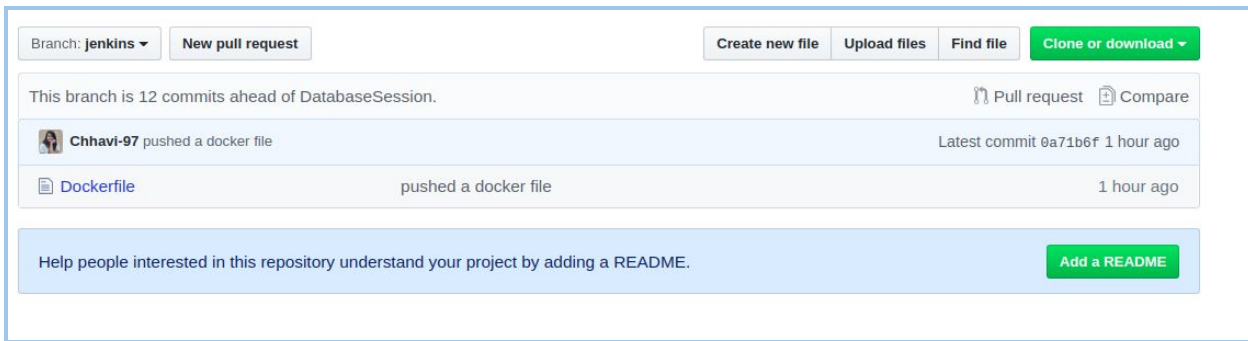
```
ubuntu@ip-172-31-179-174:~$ crontab -l
* * * * * /home/ubuntu/cronscript.sh Good Morning > output.txt
ubuntu@ip-172-31-179-174:~$ ls
cronscript.sh  output.txt
ubuntu@ip-172-31-179-174:~$ cat output.txt
Good
Morning
```

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.

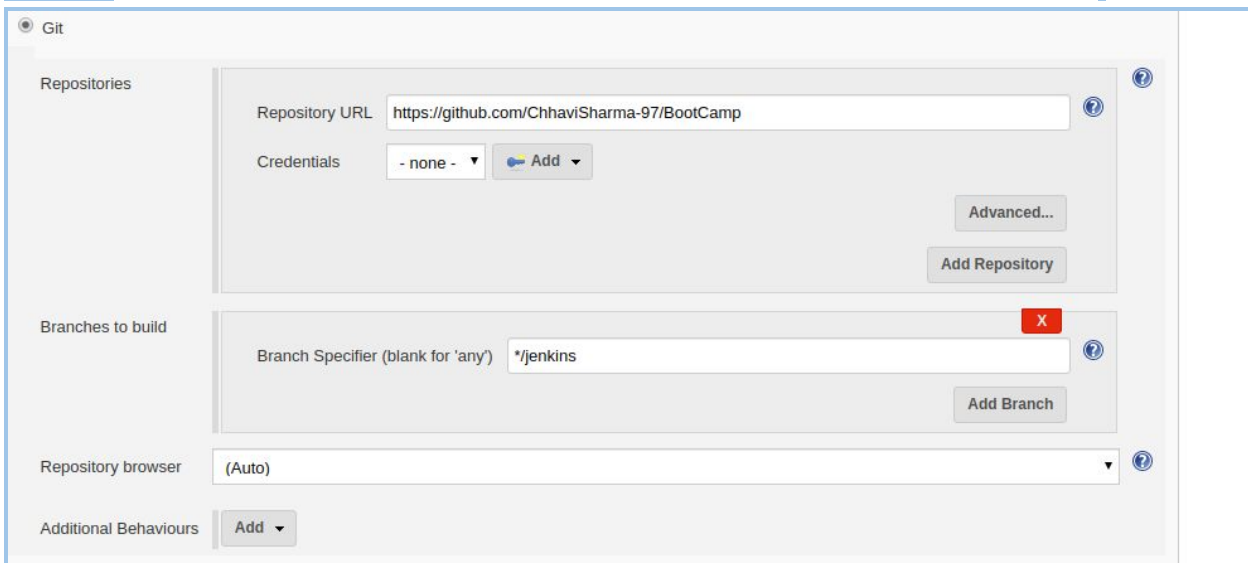
Ans.

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.

lii. 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

```
chhavi@chhavi:~$ sudo usermod -aG docker jenkins
chhavi@chhavi:~$ id jenkins
uid=125(jenkins) gid=129(jenkins) groups=129(jenkins),998(docker)
chhavi@chhavi:~$ docker --version
Docker version 19.03.6, build 369ce74a3c
chhavi@chhavi:~$ sudo service jenkins restart
chhavi@chhavi:~$ sudo su jenkins
jenkins@chhavi:/home/chhavi$ cd
jenkins@chhavi:~$ docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.
Username: chhavisharma97
Password:
WARNING! Your password will be stored unencrypted in /var/lib/jenkins/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
jenkins@chhavi:~$
```

Step 4: Build the project.

Status

Changes

Console Output

View as plain text

Edit Build Information

Delete build '#2'

Environment Variables

Git Build Data

No Tags

Previous Build

Console Output

Started by user [Chhavi Sharma](#)

Running as SYSTEM

[EnvInject] - Loading node environment variables.

Building in workspace /var/lib/jenkins/workspace/docker

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/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

[docker] \$ /bin/bash /tmp/jenkins15447613902898179972.sh

/var/lib/jenkins/workspace/docker

Dockerfile

0a71b6ff1562ceef2c31af5177a1297c54391aec

Building DockerFile

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

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
Tagging 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.

The screenshot shows the Docker Hub interface for the repository `chhavisharma97/jenkinsdocker`. The 'Tags' tab is selected, displaying a list of tags. The latest tag is `0a71b6ff1562ceef2c31af5177a1297c54391aec`, which was updated 7 minutes ago. The image details include the digest `921adec4e0d5`, the OS/ARCH `linux/amd64`, and the compressed size of `90.42 MB`. A 'docker pull' button is visible next to the tag.

IMAGE	DIGEST	OS/ARCH	COMPRESSED SIZE
<code>0a71b6ff1562ceef2c31af5177a1297c54391aec</code> Last updated 7 minutes ago by chhavisharma97	<code>921adec4e0d5</code>	<code>linux/amd64</code>	90.42 MB

VIA CLOUDBEES

Step 1: Install cloudbees plugin.

Updates Available Installed Advanced		
Install	Name	Version
<input type="checkbox"/>	CloudBees Disk Usage Simple Alternative to disk-usage plugin, to compute in the background per job disk usage on jenkins master	0.9
<input checked="" type="checkbox"/>	CloudBees Docker Build and Publish This plugin enables building Dockerfile based projects, as well as publishing of the built images/repos to the docker registry.	1.3.2
<input type="checkbox"/>	Bitbucket Branch Source Allows to use Bitbucket Cloud and Bitbucket Server as sources for multi-branch projects. It also provides the required connectors for Bitbucket Cloud Team and Bitbucket Server Project folder (also known as repositories auto-discovering).	2.7.0
<input type="checkbox"/>	CloudBees Docker Custom Build Environment	1.7.3

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

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

This branch is 12 commits ahead of DatabaseSession. [Pull request](#) [Compare](#)

Chhavi-97 pushed a docker file Latest commit 0a71b6f 1 hour ago

[Dockerfile](#) pushed a docker file 1 hour ago

Help people interested in this repository understand your project by adding a README. [Add a README](#)

Step 3: Create Dockerhub credentials

Kind: Username with password
Scope: Global (Jenkins, nodes, items, all child items, etc)
Username: chhavisharma97
Password: *****
ID: docker_hub
Description: docker hub credentials

OK

Global credentials (unrestricted)

Credentials that should be available irrespective of domain specification to requirements matching.

ID	Name	Kind	Description
docker_hub	chhavisharma97/***** (docker hub credentials)	Username with password	docker hub credentials

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

Step 4: Add a build step.

Build

Docker Build and Publish

X ?

Repository Namechhavisharma97/cloudbees?

Tag\${GIT_COMMIT}?

Docker Host URI?

Server credentials- none - Add ?

Docker registry URL?

Registry credentialschhavisharma97/***** (docker hub credentials) Add ?

Advanced...

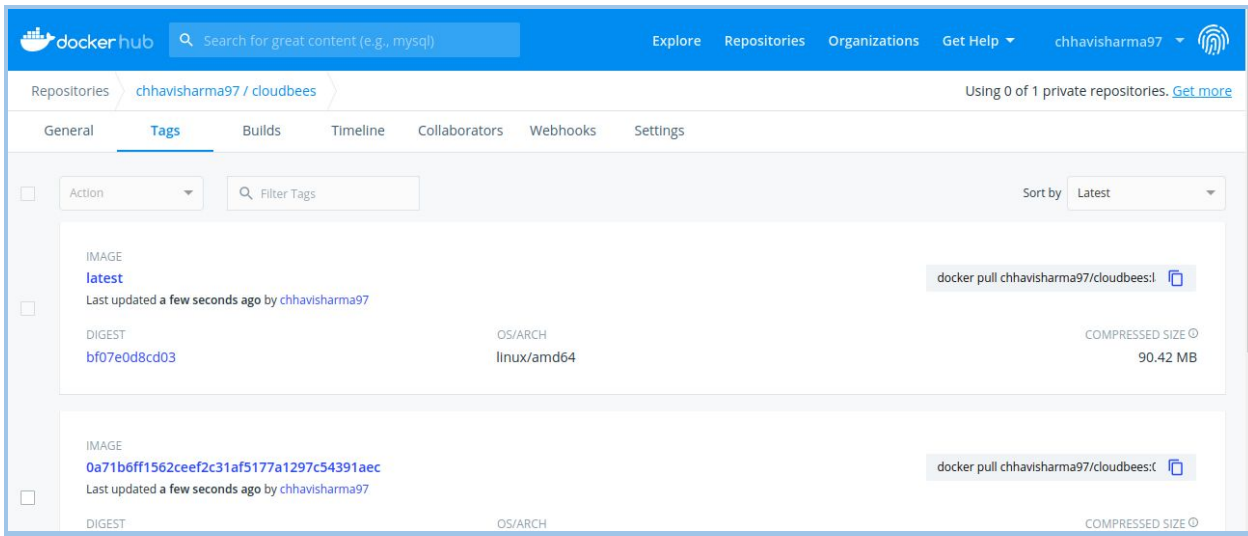
Add build step

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

Console Output

```
Started by user Chhavi Sharma
Running as SYSTEM
[EnvInject] - Loading node environment variables.
Building in workspace /var/lib/jenkins/workspace/dockerviaccloudbees
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
[dockerviaccloudbees] $ docker build -t chhavisharma97/cloudbees:0a71b6ff1562ceef2c31af5177a1297c54391aec --
pull=true /var/lib/jenkins/workspace/dockerviaccloudbees
Sending build context to Docker daemon 38.1MB

Step 1/7 : FROM ubuntu
latest: Pulling from library/ubuntu
5bd26d33875: Pulling fs layer
f11b29a9c730: Pulling fs layer
930bda195c84: Pulling fs layer
c80e10874d60: Layer already exists
977183d4e999: Layer already exists
latest: digest: sha256:bf07e0d8cd0341b6c0ea754c68de11f3cccd970ee96ddcc0c65b5840098b4d57 size: 1788
Finished: SUCCESS
```

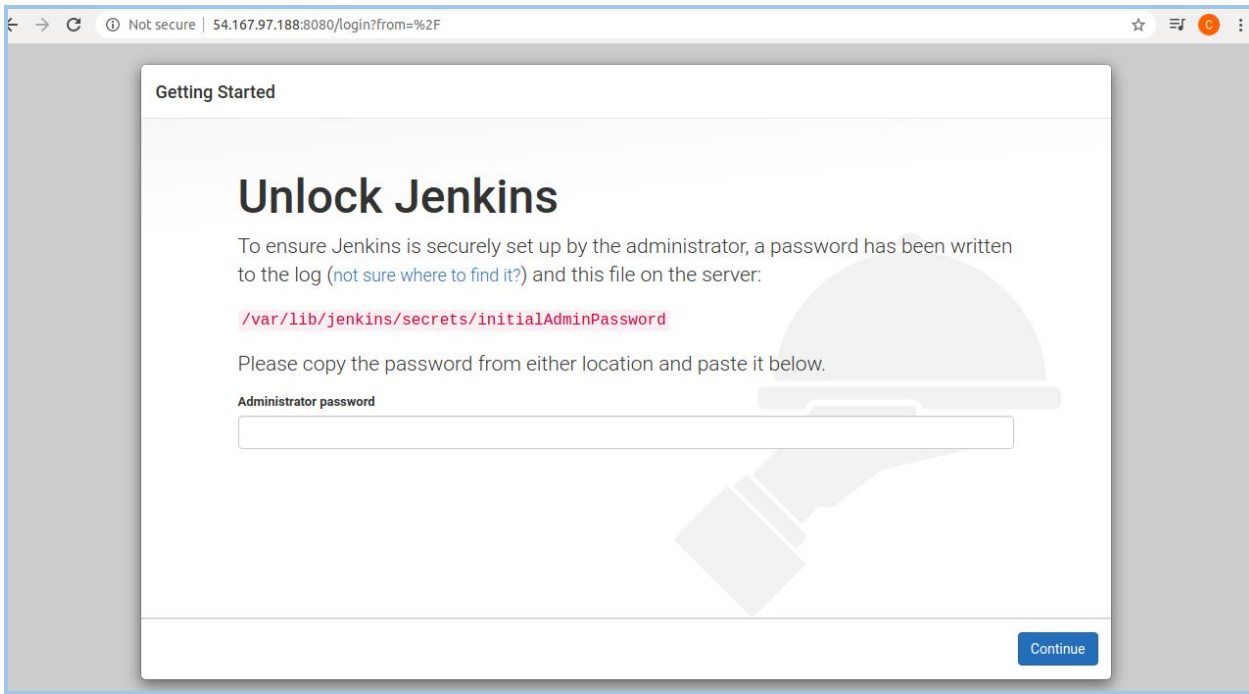


7. Host a static website on s3. Its static content should be in git repo. When a person commits any change in the repo, the job should automatically reflect the changes in the s3 website

Ans.

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

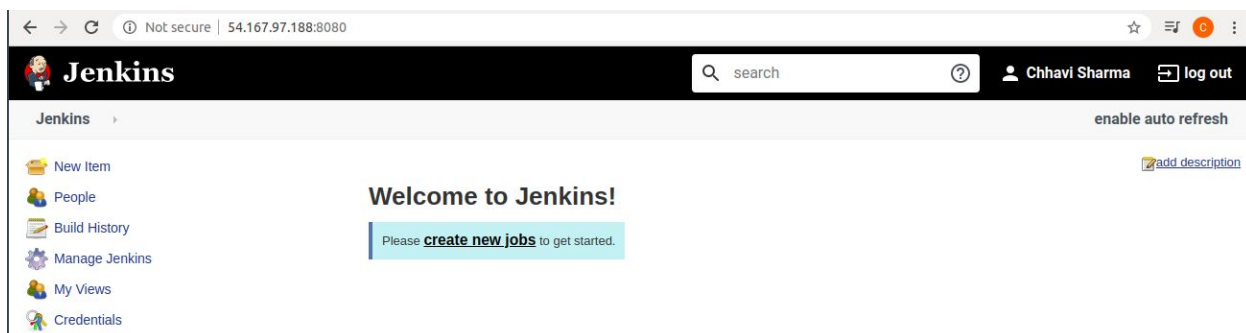
```
ubuntu@ip-172-31-37-88:~$ sudo apt-get install jenkins -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  daemon
The following NEW packages will be installed:
  daemon jenkins
0 upgraded, 2 newly installed, 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

ChhaviSharma-97 / jenkinsStaticWebsite

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](#)

8 commits 1 branch 0 packages 0 releases 1 contributor

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

Ubuntu changes Latest commit 7162fff 19 minutes ago

File	Commit Message	Time
error.html	Create error.html	5 days ago
index.html	changes	19 minutes ago

Help people interested in this repository understand your project by adding a README. [Add a README](#)

Step 4: Create a static website in s3 .

chhavis3bucket

Overview Properties Permissions Management Access points

Q Type a prefix and press Enter to search. Press ESC to clear.

Upload Create folder Download Actions Versions Hide Show US East (N. Virginia)

Viewing 1 to 3

Name	Last modified	Size	Storage class
error.html	Mar 11, 2020 9:05:59 PM GMT+0530	129.0 B	Standard
index.html	Mar 11, 2020 9:06:00 PM GMT+0530	270.0 B	Standard

Check if the website is working .

Not secure | chhavis3bucket.s3-website-us-east-1.amazonaws.com

```
<html>
<head>
  <title>This is the title of the webpage!</title>
</head>
<body>
  <p>This is an example paragraph. Anything in the <strong>body</strong> tag will appear on the page, just
like this <strong>p</strong> tag and its contents.</p>
</body>
</html>
```

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

Options

Manage access

Branches

Webhooks

Notifications

Integrations

Deploy keys

Secrets

Actions

Webhooks / Manage webhook

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in our [developer documentation](#).

Payload URL *

http://18.206.229.73:8080/github-webhook/

Content type

application/x-www-form-urlencoded

Secret

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://18.206.229.73:8080/github-webhook/ (push)

EditDelete

Step 6: Create a jenkins freestyle job with following configuration

Jenkins > staticwebsite >

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

☐ Trigger builds

☐ Disable this project

☐ Execute concurrent builds if necessary

Advanced...

Source Code Management

☐ None

☒ Git

Repositories

Repository URL

Credentials

Advanced...

Add Repository

Save Apply

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://chhavis3bucket/
aws s3 sync . s3://chhavis3bucket/
```

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

Advanced...

Add build step

Step 7: Build the job.

Console Output

```
Started by user chhavi\_sharma
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/staticwebsite
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/jenkinsStaticWebsite # timeout=10
Fetching upstream changes from https://github.com/ChhaviSharma-97/jenkinsStaticWebsite
> git --version # timeout=10
> git fetch --tags --progress -- https://github.com/ChhaviSharma-97/jenkinsStaticWebsite
+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 8ddaccf8d6246cc770c0e6567f76666a9a97a9d6 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 8ddaccf8d6246cc770c0e6567f76666a9a97a9d6 # timeout=10
Commit message: "changes made to index file"
> git rev-list --no-walk 8ddaccf8d6246cc770c0e6567f76666a9a97a9d6 # timeout=10
[staticwebsite] $ /bin/bash /tmp/jenkins1533392071988754758.sh

Unknown options: s3://chhavis3bucket/
Completed 117 Bytes/342 Bytes (2.0 KiB/s) with 2 file(s) remaining
upload: .git/FETCH_HEAD to s3://chhavis3bucket/.git/FETCH_HEAD
Completed 117 Bytes/342 Bytes (2.0 KiB/s) with 1 file(s) remaining
Completed 342 Bytes/342 Bytes (3.6 KiB/s) with 1 file(s) remaining
upload: .git/index to s3://chhavis3bucket/.git/index
Finished: SUCCESS
```

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.

Upload + Create folder Download Actions		Versions	Hide Show	US East (N. Virginia) 
<input type="checkbox"/>	 error.html	Apr 11, 2020 7:09:53 PM GMT+0530	88.0 B	Standard
<input type="checkbox"/>	 index.html	Apr 11, 2020 8:43:31 PM GMT+0530	119.0 B	Standard

INDEX

x

+



Not secure | chhavis3bucket.s3-website-us-east-1.amazonaws.com

CHANGE MADE TO INDEX FILE SUCCESSFULLY