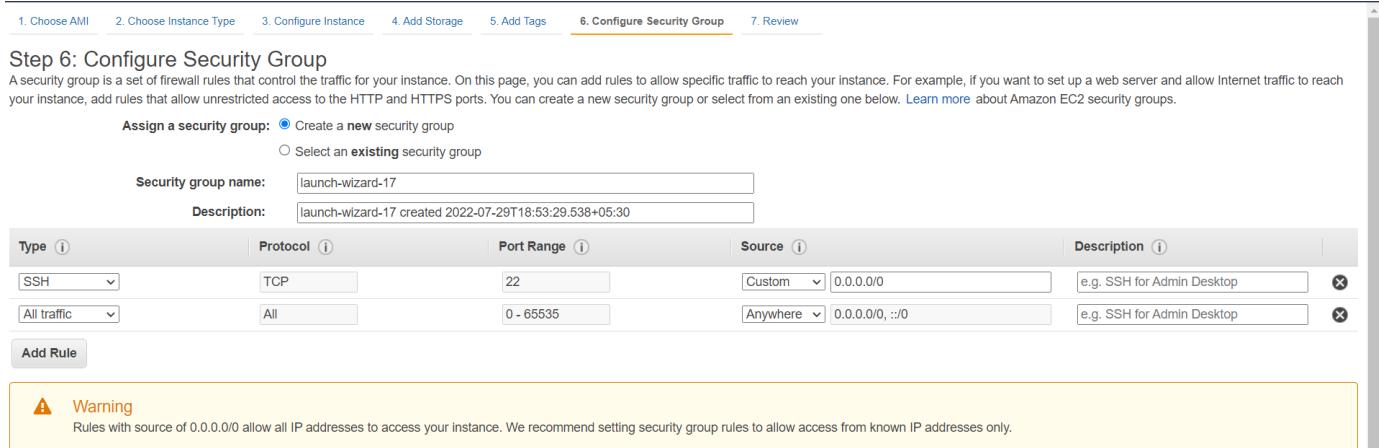


Jenkins LAB

To install Jenkins:

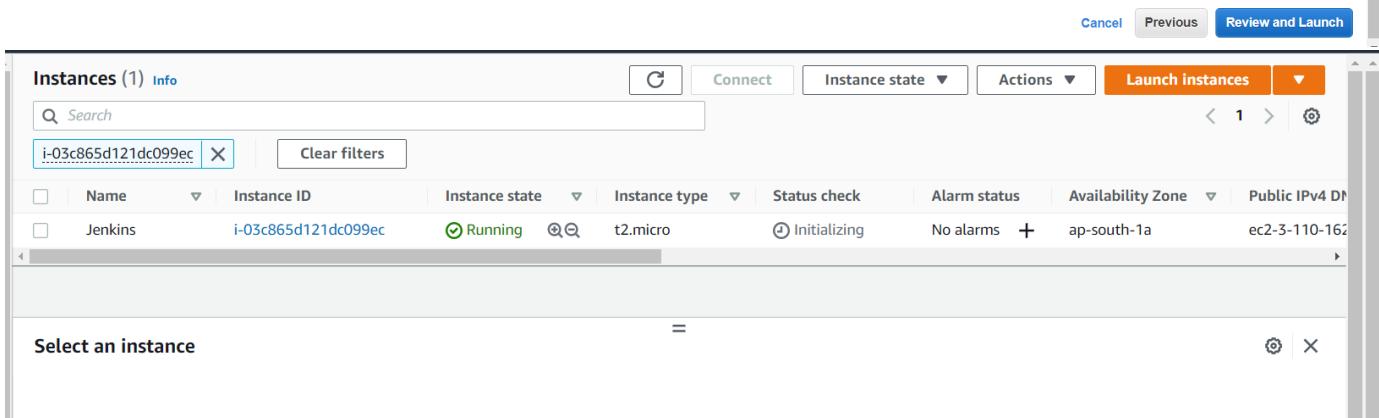
1. Launch a ec2-instance with amazon-ami image. (security group - all traffic open)



The screenshot shows the 'Configure Security Group' step of a CloudFormation wizard. It displays a table of security group rules. Two rules are present: one for SSH (TCP port 22) and another for all traffic (All port 0-65535). A warning message at the bottom cautions against using 0.0.0.0/0 as the source for security group rules.

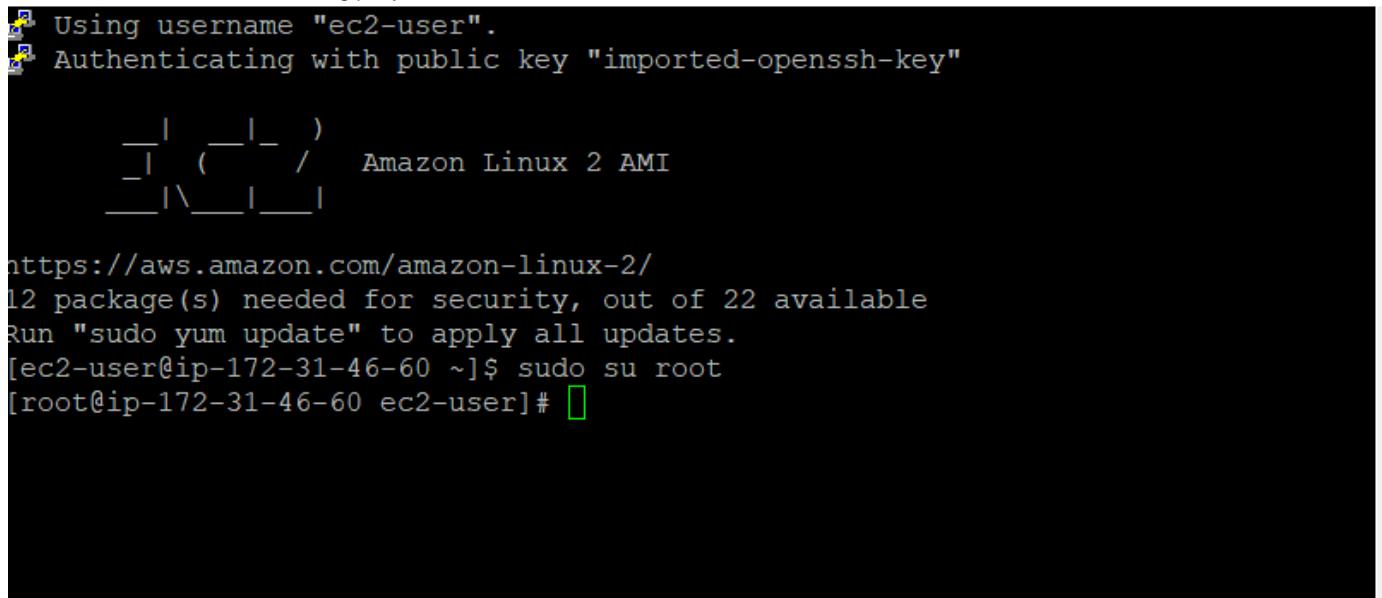
Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
All traffic	All	0 - 65535	Anywhere 0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop

Warning: Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.



The screenshot shows the 'Instances (1)' section of the CloudFormation console. It lists a single instance named 'Jenkins' with the ID 'i-03c865d121dc099ec'. The instance is running, of type 't2.micro', and is in the 'Initializing' status. It is located in the 'ap-south-1a' availability zone with a public IP of 'ec2-3-110-162'. A 'Launch instances' button is visible at the top.

2. connect to this ec2-instance using putty



```
Using username "ec2-user".
Authenticating with public key "imported-ssh-key"

_____|_____|_) 
_____| (_____| /  Amazon Linux 2 AMI
_____| \_____|_|

https://aws.amazon.com/amazon-linux-2/
12 package(s) needed for security, out of 22 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-46-60 ~]$ sudo su root
[root@ip-172-31-46-60 ec2-user]#
```

3. To install jenkins:- reference <https://www.jenkins.io/doc/tutorials/tutorial-for-installing-jenkins-on-AWS/>

Downloading and installing Jenkins

Completing the previous steps enables you to download and install Jenkins on AWS. To download and install Jenkins:

1. Ensure that your software packages are up to date on your instance by using the following command to perform a quick software update:

```
[ec2-user ~]$ sudo yum update -y
```

2. Add the Jenkins repo using the following command:

```
[ec2-user ~]$ sudo wget -O /etc/yum.repos.d/jenkins.repo \
https://pkg.jenkins.io/redhat-stable/jenkins.repo
```

3. Import a key file from Jenkins-CI to enable installation from the package:

```
[ec2-user ~]$ sudo rpm --import https://pkg.jenkins.io/redhat-
stable/jenkins.io.key
```

```
[ec2-user ~]$ sudo yum upgrade
```

4. Install Java:

```
[ec2-user ~]$ sudo amazon-linux-extras install java-openjdk11 -y
```

5. Install Jenkins:

```
[ec2-user ~]$ sudo yum install jenkins -y
```

6. Enable the Jenkins service to start at boot:

```
[ec2-user ~]$ sudo systemctl enable jenkins
```

7. Start Jenkins as a service:

```
[ec2-user ~]$ sudo systemctl start jenkins
```

You can check the status of the Jenkins service using the command:

```
[ec2-user ~]$ sudo systemctl status jenkins
```

```
[root@ip-172-31-46-60 ec2-user]# sudo systemctl enable jenkins
Created symlink from /etc/systemd/system/multi-user.target.wants/jenkins.service
to /usr/lib/systemd/system/jenkins.service.
[root@ip-172-31-46-60 ec2-user]# sudo systemctl start jenkins
[root@ip-172-31-46-60 ec2-user]# sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
  Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; vendor pres
  et: disabled)
  Active: active (running) since Fri 2022-07-29 13:36:38 UTC; 8s ago
    Main PID: 7812 (java)
      CGroup: /system.slice/jenkins.service
              └─7812 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java...
Jul 29 13:36:07 ip-172-31-46-60.ap-south-1.compute.internal jenkins[7812]: Th...
Jul 29 13:36:07 ip-172-31-46-60.ap-south-1.compute.internal jenkins[7812]: **...
Jul 29 13:36:07 ip-172-31-46-60.ap-south-1.compute.internal jenkins[7812]: **...
Jul 29 13:36:07 ip-172-31-46-60.ap-south-1.compute.internal jenkins[7812]: **...
Jul 29 13:36:38 ip-172-31-46-60.ap-south-1.compute.internal jenkins[7812]: 20...
Jul 29 13:36:38 ip-172-31-46-60.ap-south-1.compute.internal jenkins[7812]: 20...
Jul 29 13:36:38 ip-172-31-46-60.ap-south-1.compute.internal systemd[1]: Start...
Jul 29 13:36:39 ip-172-31-46-60.ap-south-1.compute.internal jenkins[7812]: 20...
Jul 29 13:36:39 ip-172-31-46-60.ap-south-1.compute.internal jenkins[7812]: 20...
Jul 29 13:36:39 ip-172-31-46-60.ap-south-1.compute.internal jenkins[7812]: 20...
Hint: Some lines were ellipsized, use -l to show in full.
[root@ip-172-31-46-60 ec2-user]#
```

Configuring Jenkins

Jenkins is now installed and running on your EC2 instance. To configure Jenkins:

1. Connect to `http://<your_server_public_DNS>:8080` from your browser. You will be able to access Jenkins through its management interface:
2. As prompted, enter the password found in `/var/lib/jenkins/secrets/initialAdminPassword`.
 - a. Use the following command to display this password:

```
[ec2-user ~]$ sudo cat /var/lib/jenkins/secrets
/initialAdminPassword
```

Not secure | http://3.110.162.5:8080/login?from=%2F

Getting Started

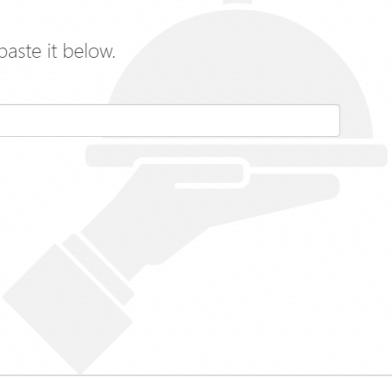
Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:

`/var/lib/jenkins/secrets/initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password



Continue

```
[root@ip-172-31-46-60 ec2-user]# cat /var/lib/jenkins/secrets/initialAdminPasswo
rd
0c94def50a964c82ae27cc0c34f430f7
[root@ip-172-31-46-60 ec2-user]# 
```

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:

```
/var/lib/jenkins/secrets/initialAdminPassword
```

Please copy the password from either location and paste it below.

Administrator password

```
.....
```

Continue

1. The Jenkins installation script directs you to the **Customize Jenkins page**. Click **Install suggested plugins**.
2. Once the installation is complete, the **Create First Admin User** will open. Enter your information, and then select **Save and Continue**.

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins

Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

Getting Started

Folders	OWASP Markup Formatter	Build Timeout	Credentials Binding	** JavaBeans Activation Framework (JAF) API
Timestamper	Workspace Cleanup	Ant	Gradle	
Pipeline	GitHub Branch Source	Pipeline: GitHub Groovy Libraries	Pipeline: Stage View	
Git	SSH Build Agents	Matrix Authorization Strategy	PAM Authentication	
LDAP	Email Extension	Mailer		

** - required dependency

Jenkins 2.346.2

Create First Admin User

Username:

Password:

Confirm password:

Full name:

E-mail address:

Jenkins 2.346.2

[Skip and continue as admin](#)[Save and Continue](#)

Create First Admin User

Username:

Password:

Confirm password:

Full name:

E-mail address:

Jenkins 2.346.2

[Skip and continue as admin](#)

[Save and Continue](#)

Instance Configuration

Jenkins URL:

`http://3.110.162.5:8080/`

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the `BUILD_URL` environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Jenkins 2,346.2

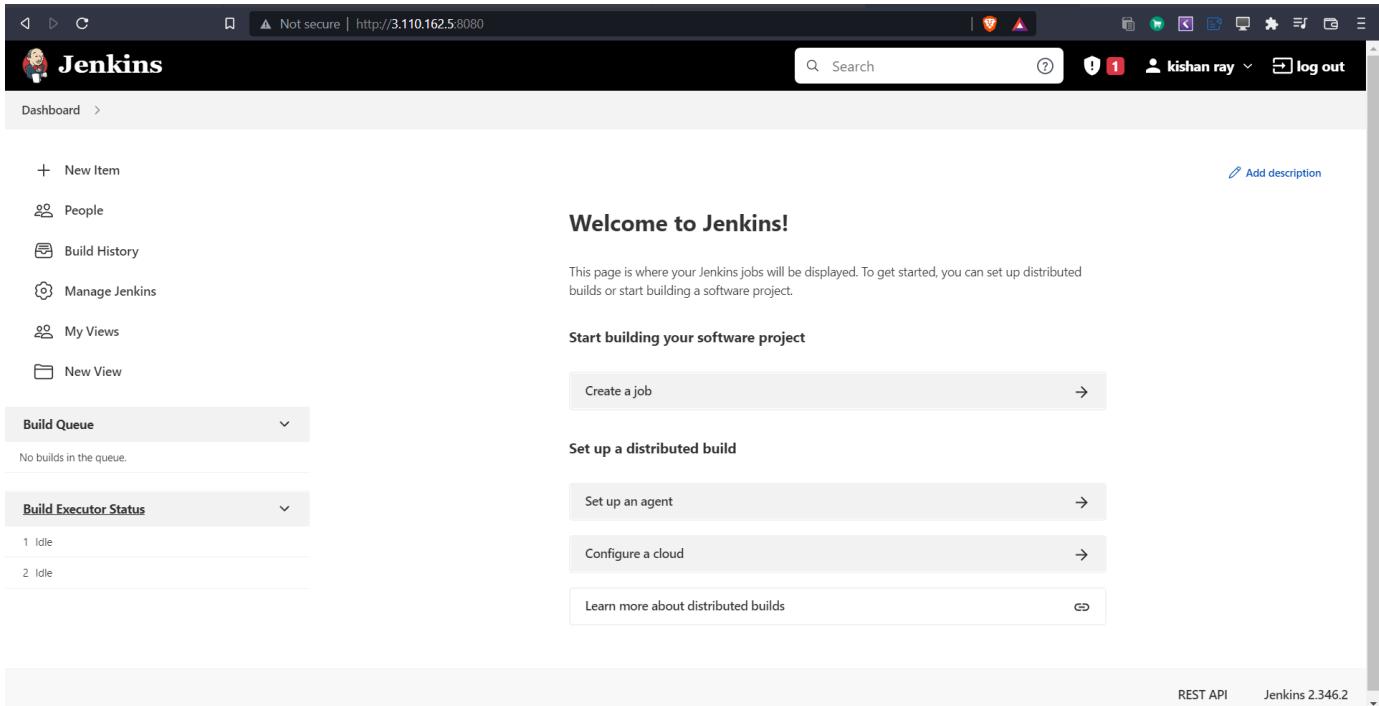
Not now

Save and Finish

Jenkins is ready!

Your Jenkins setup is complete.

[Start using Jenkins](#)



Not secure | http://3.110.162.5:8080

Jenkins

Dashboard >

+ New Item

People

Build History

Manage Jenkins

My Views

New View

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Create a job →

Set up a distributed build

Set up an agent →

Configure a cloud →

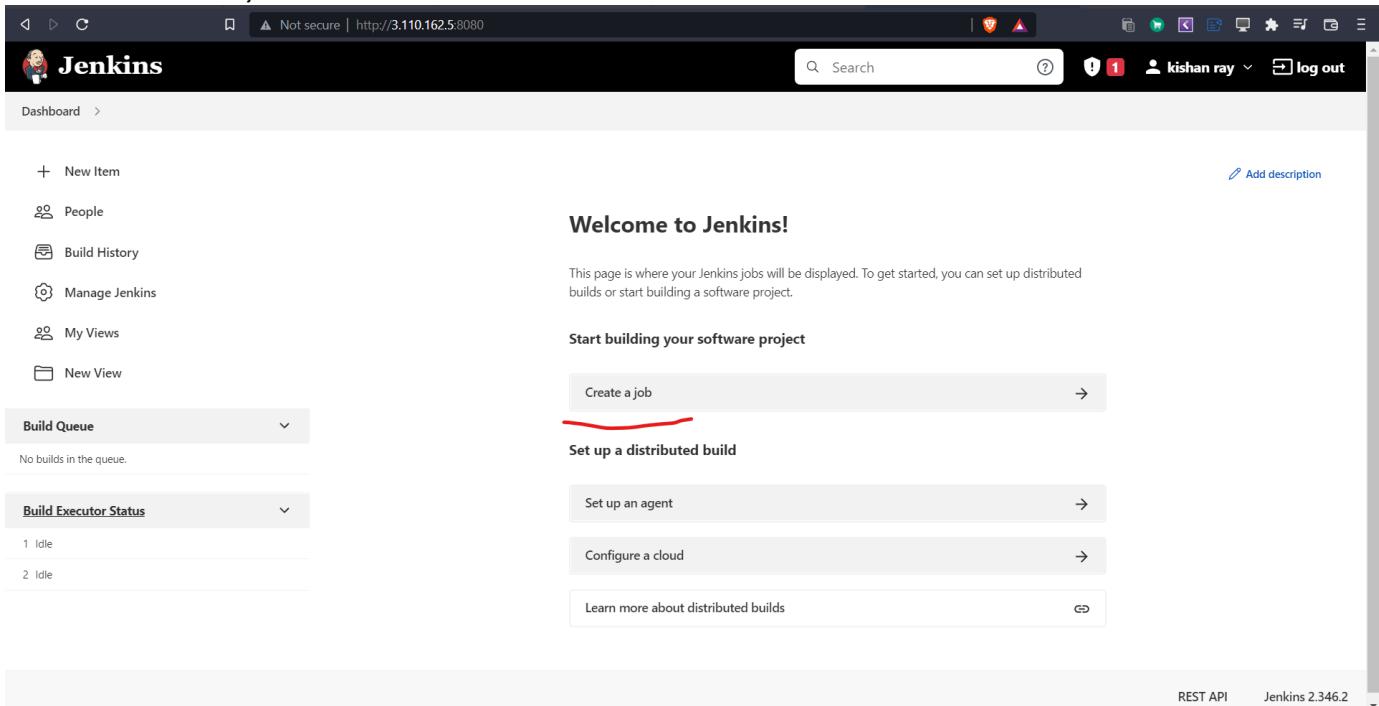
Learn more about distributed builds →

REST API Jenkins 2.346.2

Now jenkins is up and ready for use.

To create a first simple Job

1. Click on create a job



Not secure | http://3.110.162.5:8080

Jenkins

Dashboard >

+ New Item

People

Build History

Manage Jenkins

My Views

New View

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Create a job →

Set up a distributed build

Set up an agent →

Configure a cloud →

Learn more about distributed builds →

REST API Jenkins 2.346.2

2. Create a test job of type freestyle project

Enter an item name

TestJob
» 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.

 **Folder**
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

OK **Cancel Pipeline** Creates a new pipeline job. Jenkins will automatically create pipeline projects according to detected branches in your SCM repository.

3. Add description

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

Description

This project will run date and `cal` command simply.

[Plain text] **Preview**

Discard old builds ?
 GitHub project
 This project is parameterized ?
 Throttle builds ?
 Disable this project ?
 Execute concurrent builds if necessary ?

Advanced...

Save **Apply**

4. In build add some command which you want to run

Build

≡ Execute shell [?](#)

Command

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

```
date
cal
echo "hello, this is my first job"
```

[Advanced...](#)

[Add build step ▾](#)

Save **Apply**

5. Then click apply and save

Jenkins

Dashboard > TestJob >

[Back to Dashboard](#)

Project TestJob

This project will run date and cal command simply.

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

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

Permalinks

Build History [trend ▾](#)

No builds

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

6. click on build now

↑ Back to Dashboard

Project TestJob

This project will run date and cal command simply.

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

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

[Build History](#) [trend](#) [Filter builds...](#)

No builds

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

7. check build history- green color means build successfully happened and click on #1

↑ Back to Dashboard

Project TestJob

This project will run date and cal command simply.

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

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

[Build History](#) [trend](#) [Filter builds...](#)

[#1 Jul 29, 2022, 2:00 PM](#)

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

8. To see output of command click on console output

↑ Back to Project

Build #1 (Jul 29, 2022, 2:00:36 PM)

[Keep this build forever](#)

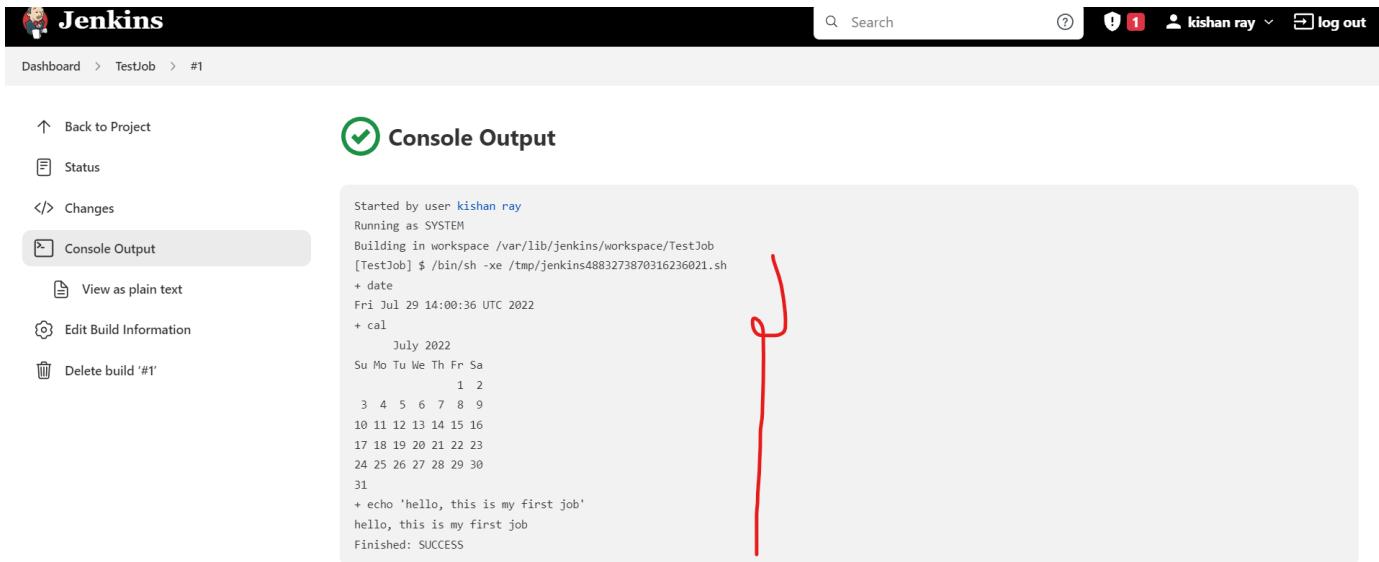
Started 59 sec ago
Took 0.11 sec

[Status](#) [Changes](#) [Console Output](#) [Edit Build Information](#) [Delete build '#1'](#)

</> No changes.

Started by user [kishan ray](#)

9. Output of date,cal and echo command can be seen.

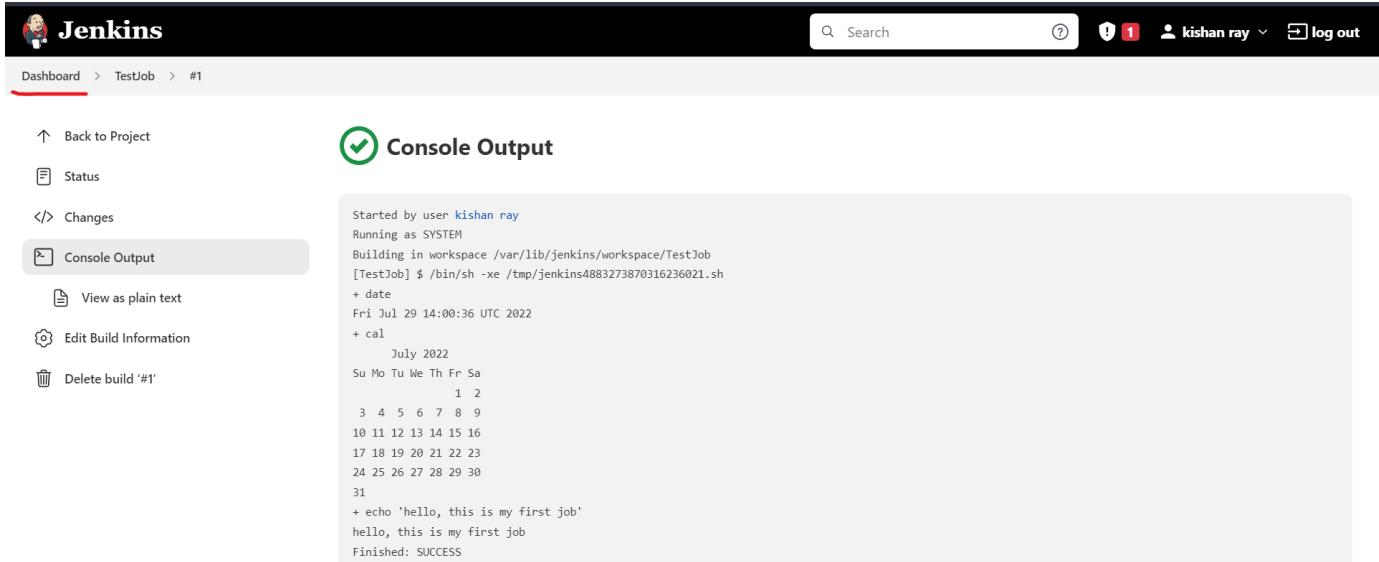


Dashboard > TestJob > #1

Console Output

Started by user **kishan ray**
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/TestJob
[TestJob] \$ /bin/sh -xe /tmp/jenkins4883273870316236021.sh
+ date
Fri Jul 29 14:00:36 UTC 2022
+ cal
July 2022
Su Mo Tu We Th Fr Sa
1 2
3 4 5 6 7 8 9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
+ echo 'hello, this is my first job'
hello, this is my first job
Finished: SUCCESS

10. To go on dashboard click on dashboard



Dashboard > TestJob > #1

Console Output

Started by user **kishan ray**
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/TestJob
[TestJob] \$ /bin/sh -xe /tmp/jenkins4883273870316236021.sh
+ date
Fri Jul 29 14:00:36 UTC 2022
+ cal
July 2022
Su Mo Tu We Th Fr Sa
1 2
3 4 5 6 7 8 9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
+ echo 'hello, this is my first job'
hello, this is my first job
Finished: SUCCESS

Dashboard >

+ New Item All + Add description

People Build History Manage Jenkins My Views New View

Build Queue No builds in the queue.

Build Executor Status 1 Idle 2 Idle

Icon legend: S M L Atom feed for all Atom feed for failures Atom feed for just latest builds

11. to check workspace

```
[root@ip-172-31-46-60 ec2-user]# cd /var/lib/jenkins/workspace
[root@ip-172-31-46-60 workspace]# ls
TestJob
```

To create a github Integrated webserver Job

1. Create a public git repo and in that add a file with index.html or else use by github repo url.

<https://github.com/devopskroy/Simplewebserver.git>

devopskroy / Simplewebserver Public

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags

Go to file Add file Code Gitpod

devopskroy Create index.html 674067 19 minutes ago 1 commit

index.html Create index.html 19 minutes ago

About No description, website

0 stars 1 watching 0 forks

Releases No releases published Create a new release

Packages

Add a README

```

1 <body style="background-color:powderblue;">
2
3 <h1>Welcome to my webpage</h1>
4 <p>This webpage deployed by jenkins</p>
5
6 </body>

```

2. Launch a ec2-instance with “kishanray-webserver-image” and login into that and name it as jenkins webserver. (port 80 add in security group)

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: Create a new security group

Select an existing security group

Security group name: launch-wizard-19

Description: launch-wizard-19 created 2022-07-29T22:00:07.281+05:30

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Anywhere 0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop

Add Rule



Warning

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Instances (1/1) Info		C	Connect	Instance state	Actions	Launch instances	▼
<input type="checkbox"/> i-07c88ec3b6c9592c1 X		Clear filters		Running	Initializing	No alarms	ap-south-1a
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
Jenkinsnewwe...	i-07c88ec3b6c9592c1	Running	t2.micro	Initializing	No alarms	ap-south-1a	ec2-15-206-12-18

Instance: i-07c88ec3b6c9592c1 (Jenkinsnewwebserver)

[Details](#) [Security](#) [Networking](#) [Storage](#) [Status checks](#) [Monitoring](#) [Tags](#)

[▼ Instance summary](#) [Info](#)

Instance ID: [i-07c88ec3b6c9592c1](#) | Public IPv4 address: [ec2-15-206-12-18](#) | Private IPv4 address: [172.31.17.10](#)

3. Login to this machine:-

```
sudo su root
```

Now create a user with name “webserver” and set password for that

```
useradd jenkinsweb
passwd jenkinsweb
```

```
[ec2-user@ip-172-31-46-212 ~]$ sudo su root
[root@ip-172-31-46-212 ec2-user]# useradd jenkinsweb
[root@ip-172-31-46-212 ec2-user]# passwd jenkinsweb
Changing password for user jenkinsweb.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@ip-172-31-46-212 ec2-user]# █
```

Now we have to provide sudo power to this user "jenkinsweb" like we done in ansible.

```
visudo
```

```
root@ip-172-31-46-212:/home/ec2-user
#
# Defaults env_keep += "HOME"
#
Defaults    secure_path = /sbin:/bin:/usr/sbin:/usr/bin
#
## Next comes the main part: which users can run what software on
## which machines (the sudoers file can be shared between multiple
## systems).
## Syntax:
##
##      user      MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root    ALL=(ALL)      ALL
jenkinsweb ALL=(ALL)      ALL █
## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOC
ATE, DRIVERS
#
## Allows people in group wheel to run all commands
```

```
##  
## The COMMANDS section may have other options as  
##  
## Allow root to run any commands anywhere  
root    ALL=(ALL)        ALL  
jenkinsweb ALL=(ALL)      ALL  
## Allows members of the 'sys' group to run netwo  
## service management apps and more.  
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STOI  
# ATE, DRIVERS  
  
## Allows people in group wheel to run all comman  
:wq
```

Now we have to enable password login as we done in ansible

```
vi /etc/ssh/sshd_config  
# uncomment permitrootlogin yes  
#uncomment passwordauthentication yes  
#comment passwordauthentication no  
systemctl restart sshd
```

```
# Authentication:

#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# The default is to check both .ssh/authorized_keys and .ssh/authorized_
# but this is overridden so installations will only check .ssh/authorized_
AuthorizedKeysFile .ssh/authorized_keys

#AuthorizedPrincipalsFile none

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes
#PermitEmptyPasswords no
#PasswordAuthentication no

# Change to no to disable s/key passwords
#ChallengeResponseAuthentication yes
ChallengeResponseAuthentication no

# Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#-----[root@ip-172-31-46-212 ec2-user]# visudo
[root@ip-172-31-46-212 ec2-user]# vi /etc/ssh/sshd_config
[root@ip-172-31-46-212 ec2-user]# systemctl restart sshd
[root@ip-172-31-46-212 ec2-user]# 
```

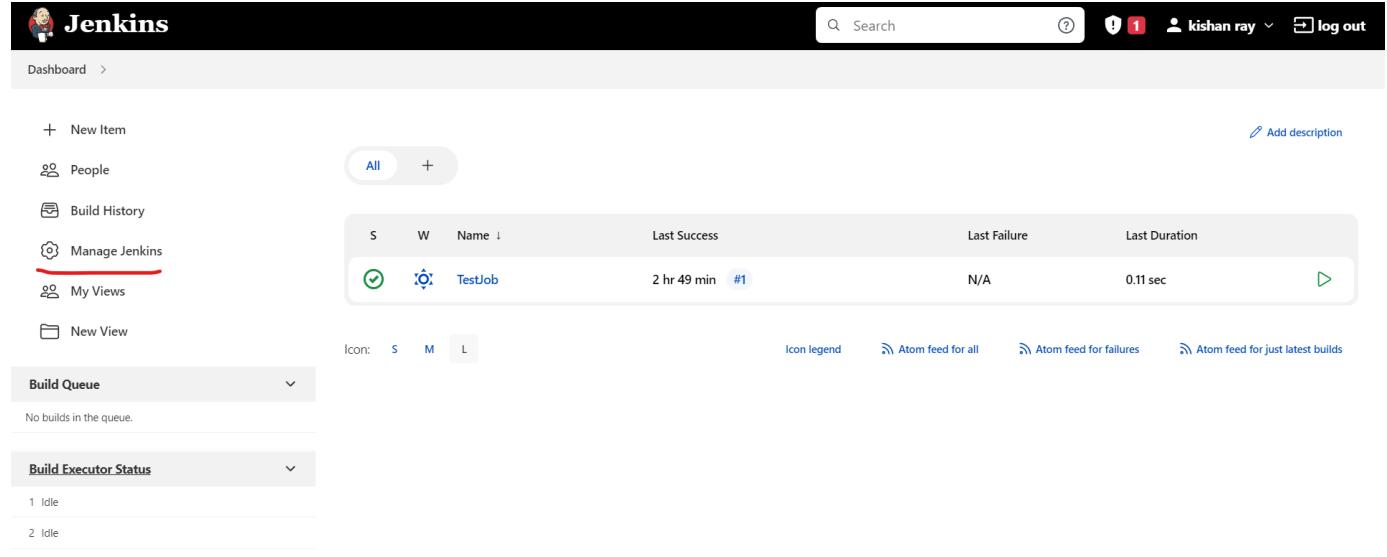
Now make user "jenkinsweb" as owner of directory /var/www/html

```
chown jenkinsweb /var/www/html
cd /var/www/
ls -l
```

```
[root@ip-172-31-46-212 ec2-user]# chown jenkinsweb /var/www/html
[root@ip-172-31-46-212 ec2-user]# cd /var/www/
[root@ip-172-31-46-212 www]# ls -l
total 0
drwxr-xr-x 2 root      root   6 Apr 12 12:01 cgi-bin
drwxr-xr-x 2 jenkinsweb root 23 Jul 12 06:28 html
[root@ip-172-31-46-212 www]# █
```

3. Now come to jenkins dashboard

Click on Manage Jenkins



The screenshot shows the Jenkins dashboard. On the left, there is a sidebar with links: 'New Item', 'People', 'Build History', 'Manage Jenkins' (which is highlighted with a red underline), and 'My Views'. The main content area shows a table for a job named 'TestJob'. The table has columns for Status (S), Warning (W), Name, Last Success, Last Failure, and Last Duration. The 'TestJob' row shows: S (green circle), W (blue circle), Name: TestJob, Last Success: 2 hr 49 min #1, Last Failure: N/A, and Last Duration: 0.11 sec. Below the table, there are buttons for 'Icon legend', 'Atom feed for all', 'Atom feed for failures', and 'Atom feed for just latest builds'. At the bottom of the sidebar, there are dropdowns for 'Build Queue' (showing 'No builds in the queue.') and 'Build Executor Status' (showing '1 Idle' and '2 Idle').

click on manage plugins

Jenkins

Dashboard >

+ New Item

People

Build History

Manage Jenkins

My Views

New View

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

Manage Jenkins

Building on the built-in node can be a security issue. You should set up distributed builds. See [the documentation](#).

[Set up agent](#) [Set up cloud](#) [Dismiss](#)

System Configuration

Configure System: Configure global settings and paths.

Global Tool Configuration: Configure tools, their locations and automatic installers.

Manage Nodes and Clouds: Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

Manage Plugins: Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

Security

Configure Global Security: Secure Jenkins; define who is allowed to

Manage Credentials: Configure credentials

Configure Credential Providers: Configure the credential providers and types

Now click on available and search “publish over ssh plugin”

Dashboard > Plugin Manager

↑ Back to Dashboard

Manage Jenkins

Updates Available Installed Advanced

Q publish over ssh

Install	Name	Released
<input type="checkbox"/>	Publish Over SSH 1.24 Artifact Uploaders Build Tools Send build artifacts over SSH	5 mo 7 days ago

Install without restart Download now and install after restart Update information obtained: 3 hr 19 min ago Check now

select and click on install without restart

Updates Available Installed Advanced

Q publish over ssh

Install	Name	Released
<input checked="" type="checkbox"/>	Publish Over SSH 1.24 Artifact Uploaders Build Tools Send build artifacts over SSH	5 mo 7 days ago

Install without restart Download now and install after restart Update information obtained: 3 hr 19 min ago Check now

Installing Plugins/Upgrades

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

Publish Over SSH



Success

Loading plugin extensions



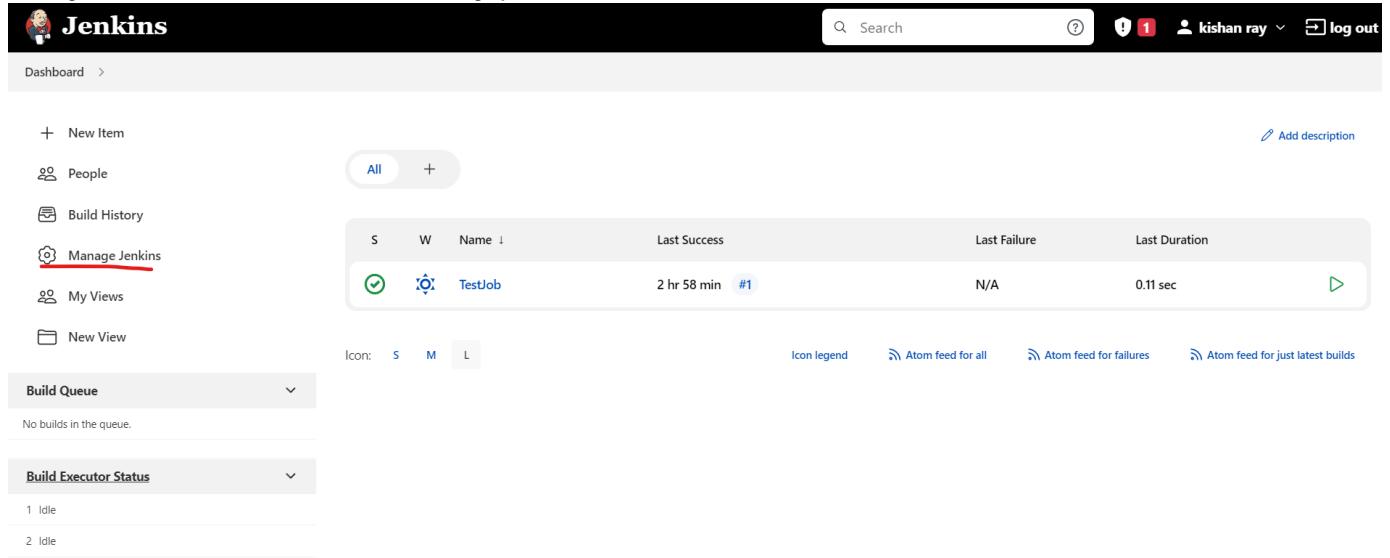
Success

→ [Go back to the top page](#)

(you can start using the installed plugins right away)

→ Restart Jenkins when installation is complete and no jobs are running

Now again come to dashboard and click on manage jenkins

A screenshot of the Jenkins dashboard. The top navigation bar is black with the Jenkins logo and the word 'Jenkins'. The main content area has a light gray background. On the left, there is a sidebar with links: 'New Item', 'People', 'Build History', 'Manage Jenkins' (which is underlined in red), 'My Views', 'New View', 'Build Queue' (which is expanded to show 'No builds in the queue.'), and 'Build Executor Status' (which is expanded to show '1 Idle' and '2 Idle'). The main area shows a table of build jobs. The table has columns: 'S' (Status), 'W' (Work), 'Name' (sorted by name), 'Last Success', 'Last Failure', and 'Last Duration'. There is one job listed: 'TestJob' with a green status icon, a blue work icon, and a red failure icon. The 'Last Success' column shows '2 hr 58 min #1'. The 'Last Failure' column shows 'N/A'. The 'Last Duration' column shows '0.11 sec'. At the bottom of the table, there are links for 'Icon legend', 'Atom feed for all', 'Atom feed for failures', and 'Atom feed for just latest builds'.

S	W	Name ↓	Last Success	Last Failure	Last Duration
		TestJob	2 hr 58 min #1	N/A	0.11 sec

Now click on configure system

 **Jenkins**

Dashboard >

+ New Item

People

Build History

Manage Jenkins

My Views

New View

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

Manage Jenkins

Building on the built-in node can be a security issue. You should set up distributed builds. See [the documentation](#).

[Set up agent](#) [Set up cloud](#) [Dismiss](#)

System Configuration

-  **Configure System**
Configure global settings and paths.
-  **Global Tool Configuration**
Configure tools, their locations and automatic installers.
-  **Manage Plugins**
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

 **Manage Nodes and Clouds**
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

Security

Now go to publish over ssh and fill the details like webserver name, private ip of our new ec2-instance and username "jenkinsweb" and the password whatever we set.

SSH Servers

SSH Server

Name

Hostname

Username

Remote Directory

Use password authentication, or use a different key

Passphrase / Password

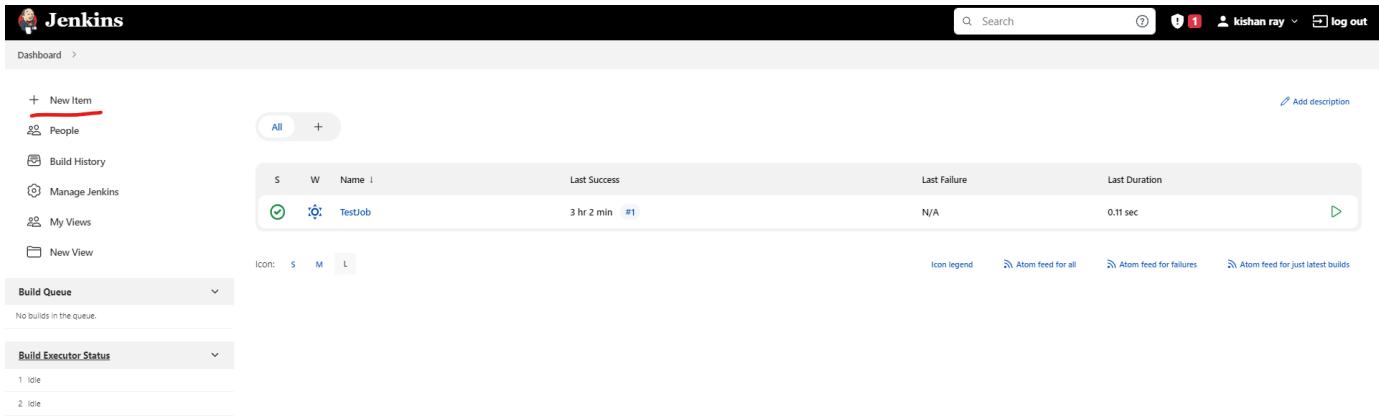
Path to key

Key

[Save](#) [Apply](#)

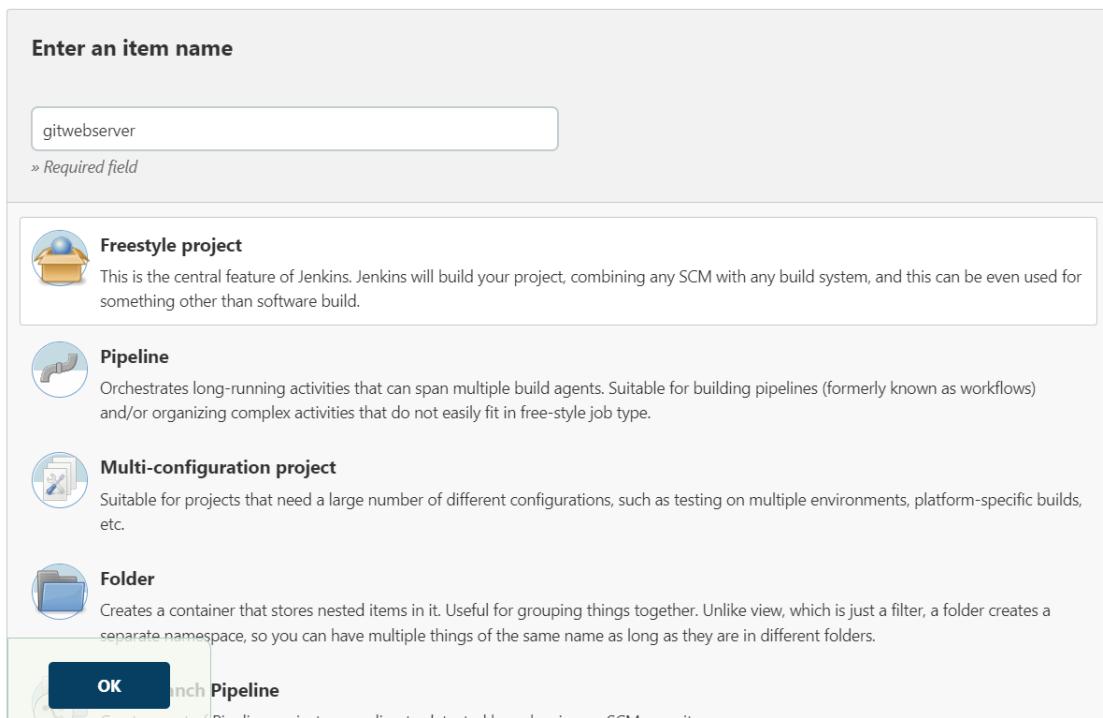
Now click apply and save.

Now again come to dashboard and click on new item



The screenshot shows the Jenkins dashboard. On the left, there is a sidebar with links: 'New Item' (highlighted with a red box), 'People', 'Build History', 'Manage Jenkins', 'My Views', and 'New View'. The main area has a search bar, a user icon for 'kishan ray', and a 'log out' link. Below the search bar is a 'Build Queue' section with the message 'No builds in the queue.' and a 'Build Executor Status' section showing '1 Idle' and '2 Idle'. The central part of the dashboard is a table with columns 'S', 'W', 'Name', 'Last Success', 'Last Failure', and 'Last Duration'. It shows one entry: 'TestJob' with a green circle icon, '3 hr 2 min' last success, 'N/A' last failure, and '0.11 sec' last duration. There are buttons for 'All', '+', 'Icon legend', and links for 'Atom feed for all', 'Atom feed for failures', and 'Atom feed for just latest builds'.

Now give the name, select freestyle project and click ok



The screenshot shows a 'Enter an item name' dialog box. The input field contains 'gitwebserver' with the note '» Required field'. Below the input field are four project selection options: 'Freestyle project' (selected), 'Pipeline', 'Multi-configuration project', and 'Folder'. Each option has a description and an icon. At the bottom, there are 'OK' and 'Cancel' buttons, with 'Pipeline' also being a selectable option.

Add description

General [Source Code Management](#) [Build Triggers](#) [Build Environment](#) [Build](#) [Post-build Actions](#)

Description

This is job to run webserver over [jenkins](#).

[\[Plain text\]](#) [Preview](#)

Discard old builds [?](#)

GitHub project

This project is parameterized [?](#)

Throttle builds [?](#)

Disable this project [?](#)

Execute concurrent builds if necessary [?](#)

[Advanced...](#)

[Save](#) [Apply](#)

Add github central repo url whichever we create in first step

Source Code Management

None

Git [?](#)

Repositories [?](#)

Repository URL [?](#) ✖

https://github.com/devopskroy/Simplewebserver.git

Credentials [?](#)

- none - ▼

[+ Add](#)

[Advanced...](#)

select correct branch

Branches to build [?](#)

Branch Specifier (blank for 'any') [?](#)

*/main X

[Add Branch](#)

Repository browser [?](#)

In Build trigger choose github hook trigger

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 [?](#)

In build choose send file or execute commands over ssh

Build

Add build step ▾

Filter

Execute Windows batch command

Execute shell

Invoke Ant

Invoke Gradle script

Invoke top-level Maven targets

Run with timeout

Send files or execute commands over SSH

Set build status to "pending" on GitHub commit

Choose ssh server and select verbose option in advance

SSH Publishers

SSH Server

Name [?](#)

webserver



Verbose output in console [?](#)

Credentials [?](#)

Retry [?](#)

Label [?](#)

Transfers

enter file name and the target location where we want to copy these files.

Transfers

Transfer Set

Source files ?

Remove prefix ?

Remote directory ?

Exec command ?

Either Source files, Exec command or both must be supplied

All of the transfer fields (except for Exec timeout) support substitution of [Jenkins environment variables](#)

[Advanced...](#)

Now click on apply and save

Jenkins

Dashboard > gitwebserver >

[Back to Dashboard](#)

Project gitwebserver

This is job to run webserver over jenkins.

[Edit description](#)

[Disable Project](#)

[Status](#)

[Changes](#)

[Workspace](#)

[Build Now](#)

[Configure](#)

[Delete Project](#)

[GitHub Hook Log](#)

[Rename](#)

Permalinks

Build History [trend](#)

No builds

[↑](#) [↓](#)

Now copy the public ip of our ec2-machine and enter it in browser

Instances (1/1) [Info](#)

Search [Clear filters](#)

<input checked="" type="checkbox"/> Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
<input checked="" type="checkbox"/> Jenkinsnewwe...	i-07c88ec3b6c9592c1	Running	t2.micro	Initializing	No alarms	ap-south-1a	ec2-15-206-12...

Instance: i-07c88ec3b6c9592c1 (Jenkinsnewwebserever)

[Details](#) [Security](#) [Networking](#) [Storage](#) [Status checks](#) [Monitoring](#) [Tags](#)

[Instance summary](#) [Info](#)

Instance ID i-07c88ec3b6c9592c1 (Jenkinsnewwebserever)	Public IPv4 address 15.206.125.32 open address	Private IPv4 addresses 172.31.46.212
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-15-206-125-32.ap-south-...

you will see this webpage

Not secure | <http://15.206.125.32>

```
welcome to devops training
eth0: flags=4163 mtu 9001
    inet 172.31.46.212 netmask 255.255.240.0 broadcast 172.31.47.255
    inet6 fe80::70:6ff:fe0a:316a prefixlen 64 scopeid 0x20
    ether 02:70:06:0a:31:6a txqueuelen 1000 (Ethernet)
    RX packets 60293 bytes 82069875 (78.2 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 6929 bytes 538178 (525.5 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73 mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Now go to jenkins dashboard and click build now

 **Jenkins** Search

Dashboard > gitwebserver >

[↑ Back to Dashboard](#)

[Status](#)

[Changes](#)

[Workspace](#)

[Build Now](#)

[Configure](#)

[Delete Project](#)

[GitHub Hook Log](#)

[Rename](#)

Project gitwebserver

This is job to run webserver over jenkins.

 [Workspace](#)

 [Recent Changes](#)

Permalinks

Build History [trend](#) [Filter builds...](#)

No builds

 **Jenkins** Search

Dashboard > gitwebserver >

[↑ Back to Dashboard](#)

[Status](#)

[Changes](#)

[Workspace](#)

[Build Now](#)

[Configure](#)

[Delete Project](#)

[GitHub Hook Log](#)

[Rename](#)

Project gitwebserver

This is job to run webserver over jenkins.

 [Workspace](#)

 [Recent Changes](#)

Permalinks

Build History [trend](#) [Filter builds...](#)

[#1 Jul 29, 2022, 5:13 PM](#) [Atom feed for all](#) [Atom feed for failures](#)

[↑ Back to Project](#)



Build #1 (Jul 29, 2022, 5:13:58 PM)

[Status](#)

[Changes](#)

[Console Output](#)



No changes.

[Edit Build Information](#)



Started by user [kishan ray](#)

[Delete build '#1'](#)



Revision: 5b3facd6784c678176894ca539d9f7a0db19affb

Repository: <https://github.com/devopskroy/Simplewebserver.git>

[Git Build Data](#)

• refs/remotes/origin/main

Now go to browser and again refresh the page you will see new webpage there

The screenshot shows a web browser window with the following details:

- Address bar: Not secure | http://15.206.125.32
- Page content:

Welcome to my webpage

This webpage deployed by jenkins

Now add one more line in index.html in github central repo and commit it.

devopskroy / Simplewebserver · Public

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Simplewebserver / index.html in main

Cancel changes

Edit file Preview changes Spaces 2 No wrap

```
1 <body style="background-color:powderblue;">
2
3 <h1>Welcome to my webpage</h1>
4 <p>This webpage deployed by jenkins</p>
5 <br><br>
6 <p>This is new line added <p>
7
8 </body>
9
```

Now go to jenkins dashboard and again click build now

 Jenkins

Search

Dashboard > gitwebserver >

[↑ Back to Dashboard](#)

Project gitwebserver

This is job to run webserver over jenkins.

[⟨/⟩ Changes](#)

[Workspace](#)

[Build Now](#)



[Configure](#)

[⟨/⟩ Recent Changes](#)

[Delete Project](#)

[GitHub Hook Log](#)

[Rename](#)

[Build History](#)

trend ▾

Filter builds...

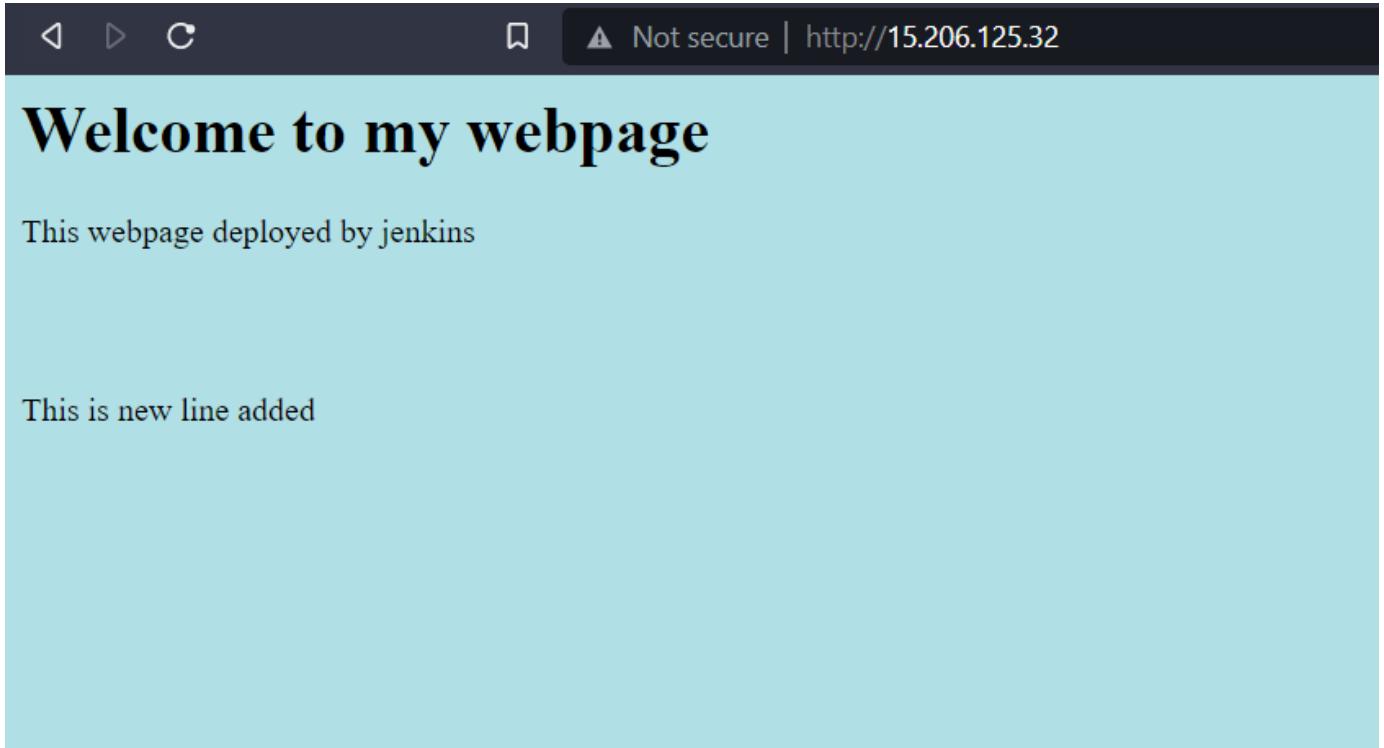
 #2 Jul 29, 2022, 5:17 PM

 #1 Jul 29, 2022, 5:13 PM



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

Now go to browser and refresh the page, we will be able to see new updated content.



To schedule a Job Periodically

1. Login to jenkins dashboard

S	W	Name	Last Success	Last Failure	Last Duration
✓	⌚	gitwebserver	2 days 19 hr #2	N/A	0.93 sec
✓	⌚	TestJob	2 days 22 hr #1	N/A	0.11 sec

2. We will schedule job "Testjob" to run periodically

Click on "Testjob" and then click on configure

 **Jenkins**

Dashboard > TestJob >

[↑ Back to Dashboard](#)

Project TestJob

This project will run date and cal command simply.

[Status](#) [Edit description](#) [Disable Project](#)

[Changes](#) [Workspace](#) [Build Now](#) [Configure](#) [Delete Project](#) [Rename](#)

[Build History](#) [trend](#) [Filter builds...](#)

[#1 Jul 29, 2022, 2:00 PM](#) [Atom feed for all](#) [Atom feed for failures](#)

Permalinks

- [Last build \(#1\), 2 days 22 hr ago](#)
- [Last stable build \(#1\), 2 days 22 hr ago](#)
- [Last successful build \(#1\), 2 days 22 hr ago](#)
- [Last completed build \(#1\), 2 days 22 hr ago](#)

3. To trigger job every minute , select build periodically and in schedule write as per expression

To schedule your build every 5 minutes, this will do the job : */5 * * *
* * OR H/5 * * * *

To the job every 5min past every hour(5th Minute of every Hour) 5 * * * *
*

To schedule your build every day at 8h00, this will do the job : 0 8 * * *
* *

To schedule your build for 4, 6, 8, and 10 o'clock PM every day - 0
16,18,20,22 * * *

To schedule your build at 6:00PM and 1 AM every day - 0 1,18 * * *

To schedule your build start daily at morning - 03 09 * * 1-5

To schedule your build start daily at lunchtime - 00 12 * * 1-5

To schedule your build start daily in the afternoon - 00 14 * * 1-5

To schedule your build start daily in the late afternoon - 00 16 * * 1-5

To schedule your build start at midnight - 59 23 * * 1-5 OR @midnight

To run a job on 9.30p.m. (at night) on 3rd of May then I'll write or
21.30 on 3/5/2011 - 21 30 3 5 *

Every fifteen minutes (perhaps at :07, :22, :37, :52) 0 - H/15 * * * *

Every ten minutes in the first half of every hour (three times, perhaps
at :04, :14, :24) - H(0-29)/10 * * * *

Once every two hours every weekday (perhaps at 10:38 AM, 12:38 PM, 2:38
PM, 4:38 PM) - H 9-16/2 * * 1-5

Once a day on the 1st and 15th of every month except December - H H
1,15 1-11 *

Trigger builds remotely (e.g., from scripts) ?

Build after other projects are built ?

Build periodically ?

Schedule ?

```
*/1 * * * *
```

⚠ Do you really mean "every minute" when you say "*/1 * * * *"? Perhaps you meant "H * * * *" to poll once per hour
Would last have run at Monday, August 1, 2022 at 12:50:06 PM Coordinated Universal Time; would next run at Monday, August 1, 2022 at 12:50:06 PM Coordinated Universal Time.

GitHub hook trigger for GITScm polling ?

4. click on apply and save

5. Now you will see that every minute automatically build will happen.

Dashboard > TestJob >

Project TestJob

Back to Dashboard

Status

This project will run date and cal command simply.

Changes

Workspace

Build Now

Configure

Delete Project

Rename

Build History trend ▾

Filter builds...

#4 Aug 1, 2022, 12:57 PM

#3 Aug 1, 2022, 12:56 PM

#2 Aug 1, 2022, 12:55 PM

Recent Changes

Edit description

Disable Project

Permalinks

- Last build (#3), 55 sec ago
- Last stable build (#3), 55 sec ago
- Last successful build (#3), 55 sec ago
- Last completed build (#3), 55 sec ago

6. To stop this again click on configure, untick build periodically and then apply and save.

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 Environment

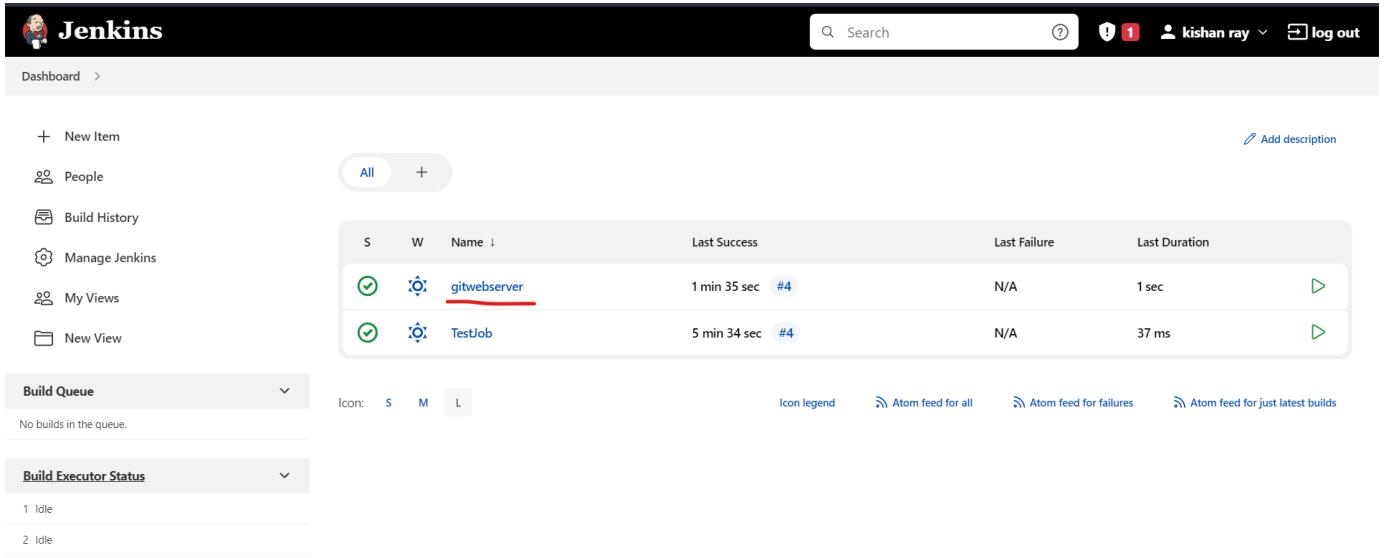
Delete workspace before build starts

Use secret text(s) or file(s) ?

Send files or execute commands over SSH before the build starts ?

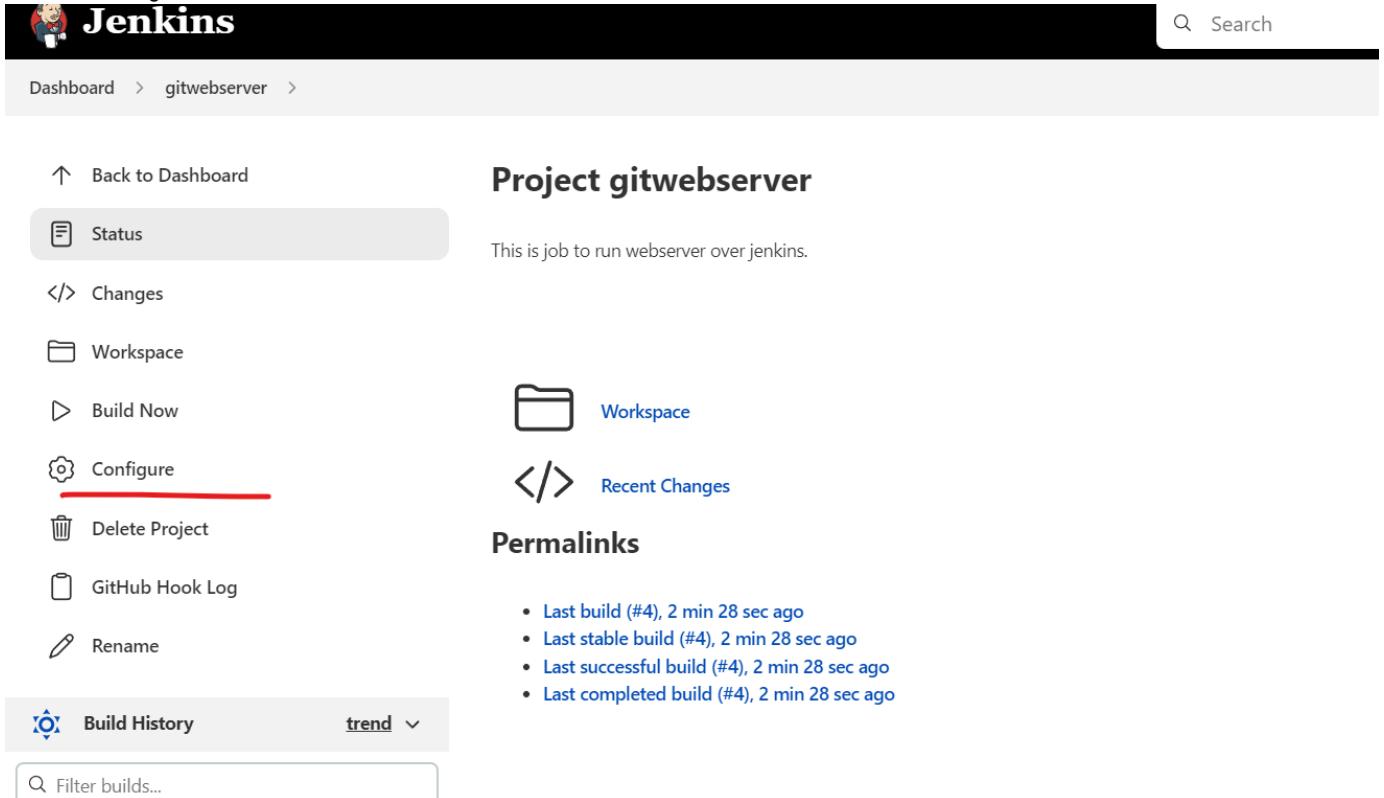
To Trigger a job when new code pushed to Github repo

1. Go to jenkins dashboard and click on gitwebserver which we created.



The screenshot shows the Jenkins dashboard. On the left, there's a sidebar with links like 'New Item', 'People', 'Build History', 'Manage Jenkins', 'My Views', and 'New View'. The main area has a table titled 'All' showing build status. The 'gitwebserver' job is listed with a green checkmark, a blue Jenkins icon, and the name 'gitwebserver'. It shows 'Last Success' at '1 min 35 sec #4', 'Last Failure' at 'N/A', and 'Last Duration' at '1 sec'. Below the table, there's a 'Build Queue' section stating 'No builds in the queue.' and a 'Build Executor Status' section showing 1 idle and 2 idle executors. At the top right, there are search, help, notifications, user info, and a 'log out' button.

2. click on configure



The screenshot shows the 'Project gitwebserver' configuration page. On the left, a sidebar lists options: 'Status' (selected), 'Changes', 'Workspace', 'Build Now', 'Configure' (underlined), 'Delete Project', 'GitHub Hook Log', 'Rename', 'Build History' (selected), and a search bar for 'Filter builds...'. The main content area is titled 'Project gitwebserver' and contains the text 'This is job to run webserver over jenkins.' Below this, there are two sections: 'Workspace' (with a folder icon) and 'Recent Changes' (with a diff icon). The 'Recent Changes' section lists the last four builds. At the bottom, there's a 'Permalinks' section with a bulleted list of build links.

S	W	Name	Last Success	Last Failure	Last Duration
Green	Blue	gitwebserver	1 min 35 sec #4	N/A	1 sec
Green	Blue	TestJob	5 min 34 sec #4	N/A	37 ms

Icon: S M L

Icon legend: Atom feed for all Atom feed for failures Atom feed for just latest builds

1 Idle
2 Idle

Project gitwebserver

This is job to run webserver over jenkins.

Workspace

</> Recent Changes

Permalinks

- Last build (#4), 2 min 28 sec ago
- Last stable build (#4), 2 min 28 sec ago
- Last successful build (#4), 2 min 28 sec ago
- Last completed build (#4), 2 min 28 sec ago

3. Now check whether github webhook trigger poll scm is selected or not , if not then select and click on apply and save.

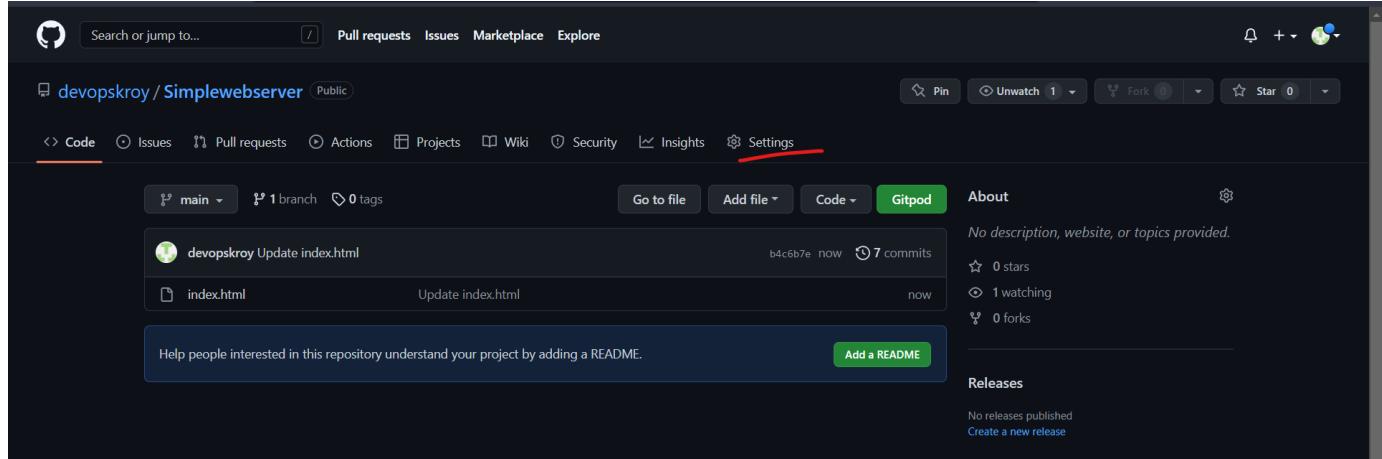
Build Triggers

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

Build Environment

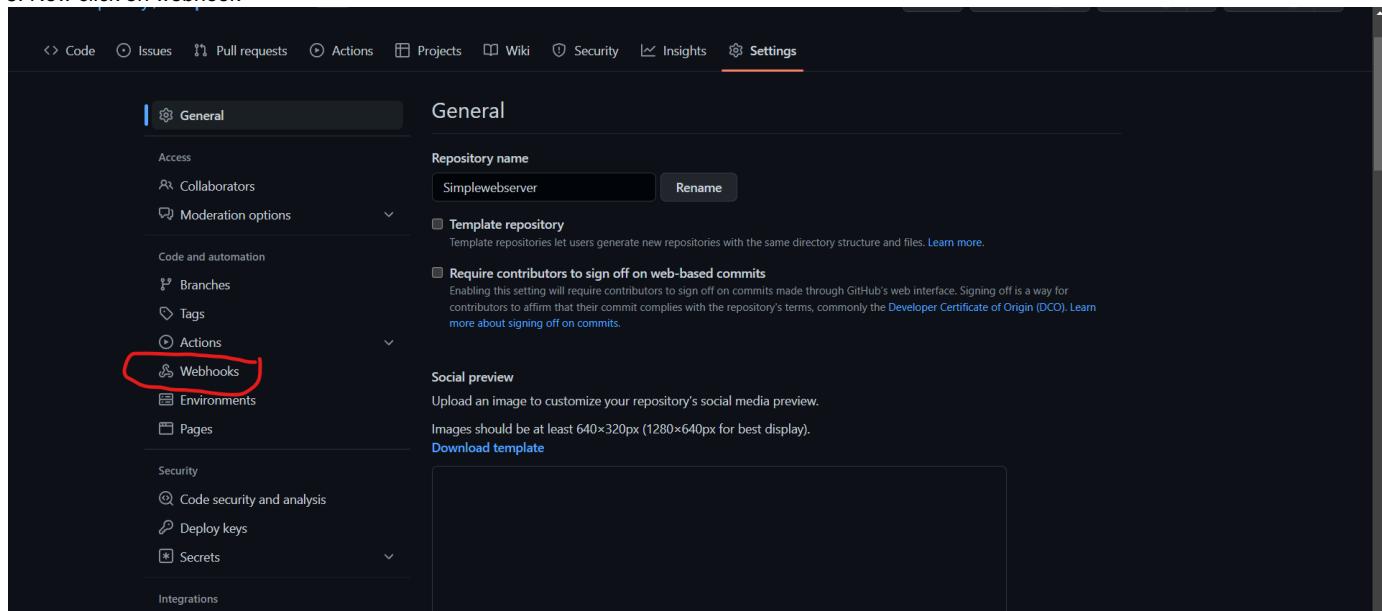
- Delete workspace before build starts
- Use secret text(s) or file(s) [?](#)
- Send files or execute commands over SSH before the build starts [?](#)
- Send files or execute commands over SSH after the build runs [?](#)

4. Now go to github account and click on central repo and then select settings.



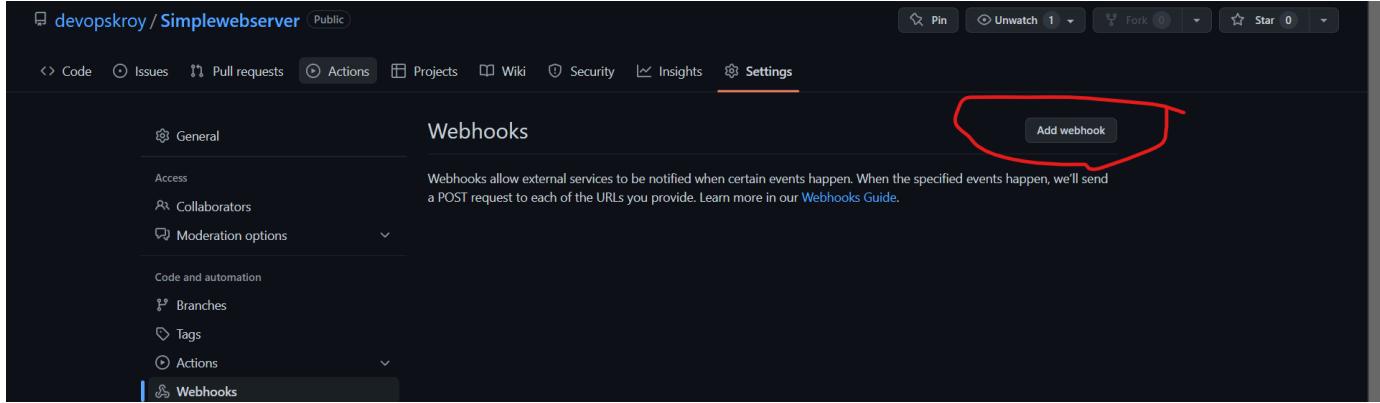
The screenshot shows a GitHub repository page for 'devopskroy / Simplewebserver'. The 'Settings' tab is active, indicated by a red box. The page displays a list of commits, including one from 'devopskroy' that updated 'index.html'. It also shows repository statistics: 0 stars, 1 watching, and 0 forks. A 'README' button is visible.

5. Now click on webhook



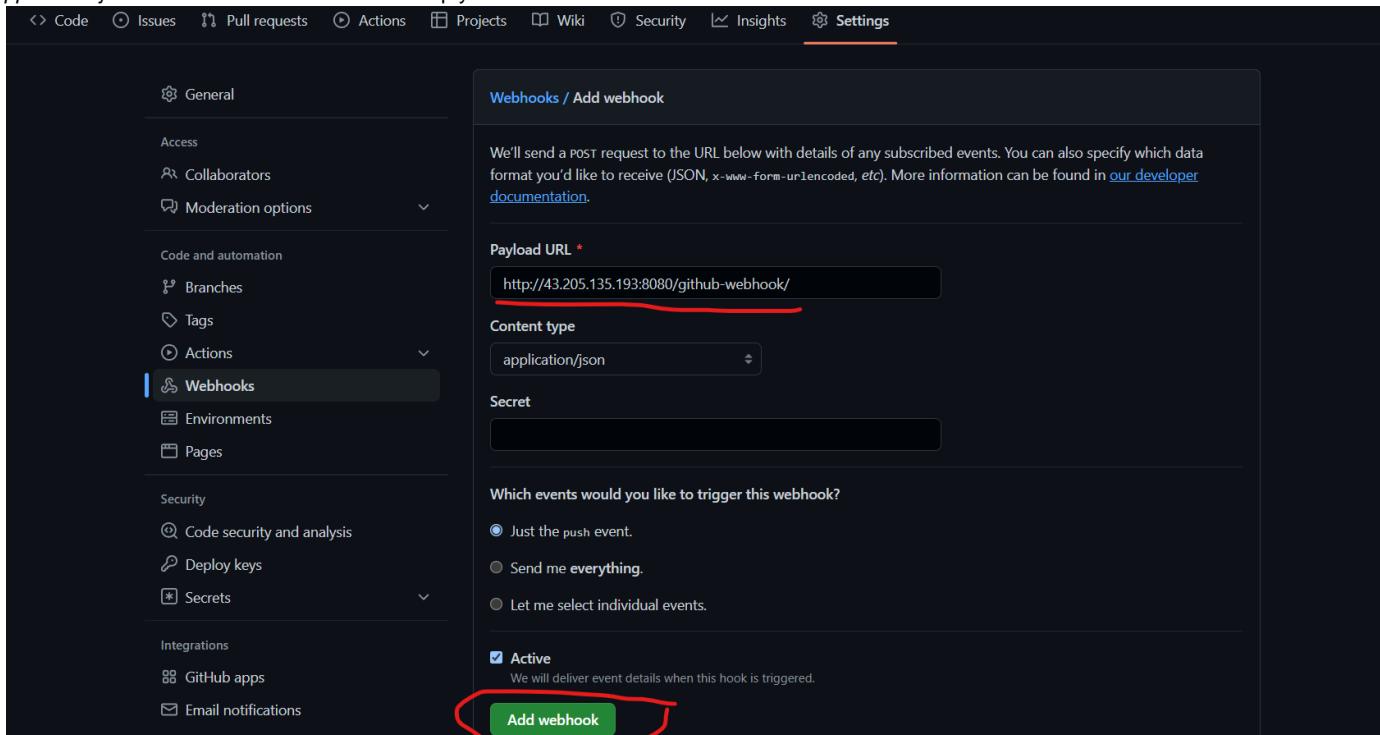
The screenshot shows the 'General' tab of the GitHub repository settings. The 'Webhooks' section is highlighted with a red box. The page includes fields for 'Repository name' (Simplewebserver) and 'Template repository'. It also features a 'Social preview' section for customizing repository social media preview images.

6. click on add webhook



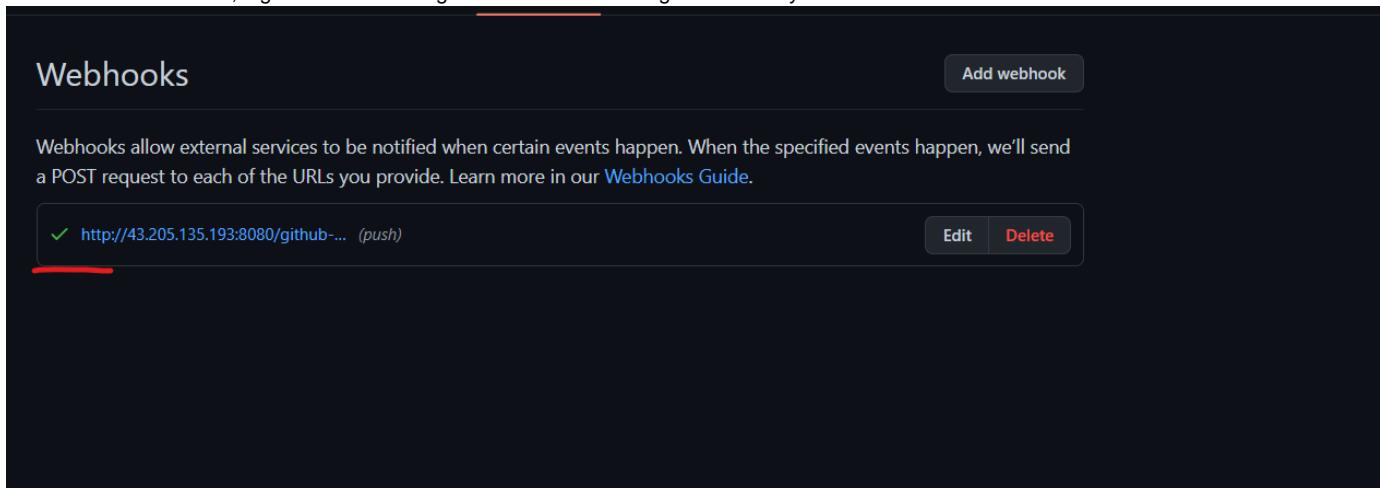
The screenshot shows the GitHub Settings page for a repository named 'Simplewebserver'. The 'Webhooks' section is highlighted. A red circle is drawn around the 'Add webhook' button at the top right of the section.

7. In the 'Payload URL' field, paste your Jenkins environment URL. At the end of this URL add /github-webhook/. In the 'Content type' select: 'application/json' and leave the 'Secret' field empty



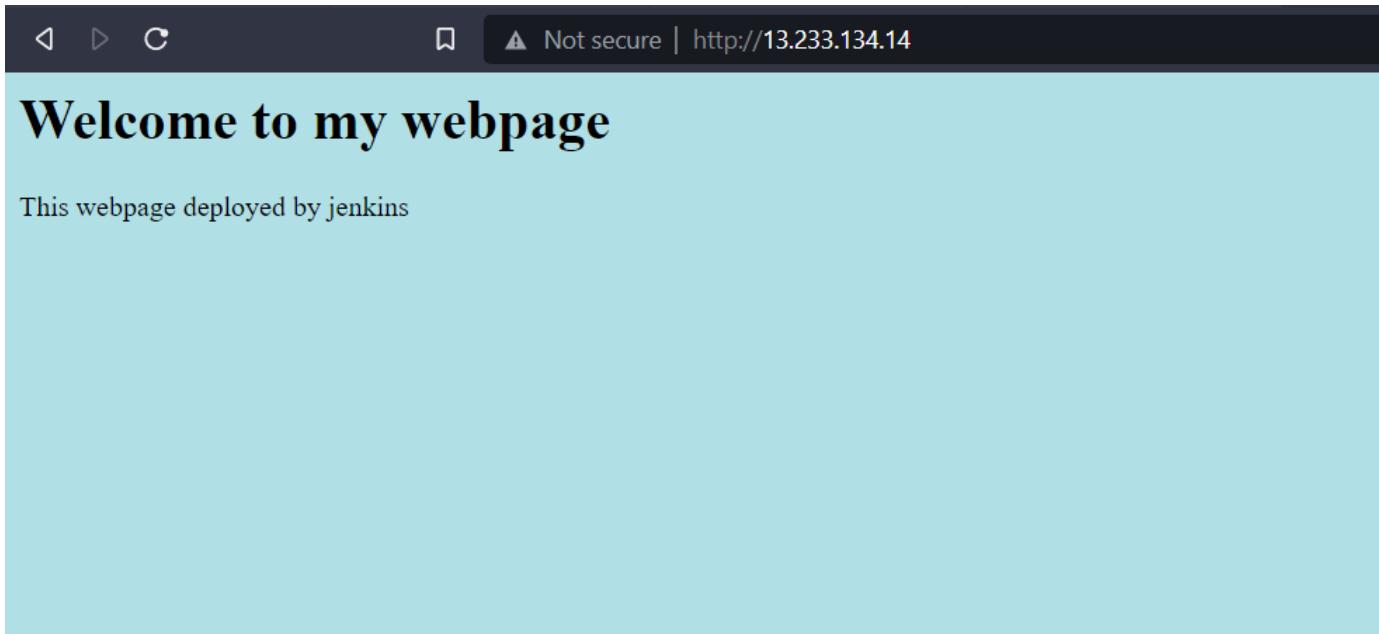
The screenshot shows the 'Add webhook' configuration page. The 'Payload URL' field contains 'http://43.205.135.193:8080/github-webhook/'. The 'Content type' is set to 'application/json'. The 'Secret' field is empty. The 'Add webhook' button at the bottom is circled in red.

8. click on add webhook, if green tick is coming then webhook is configured correctly.

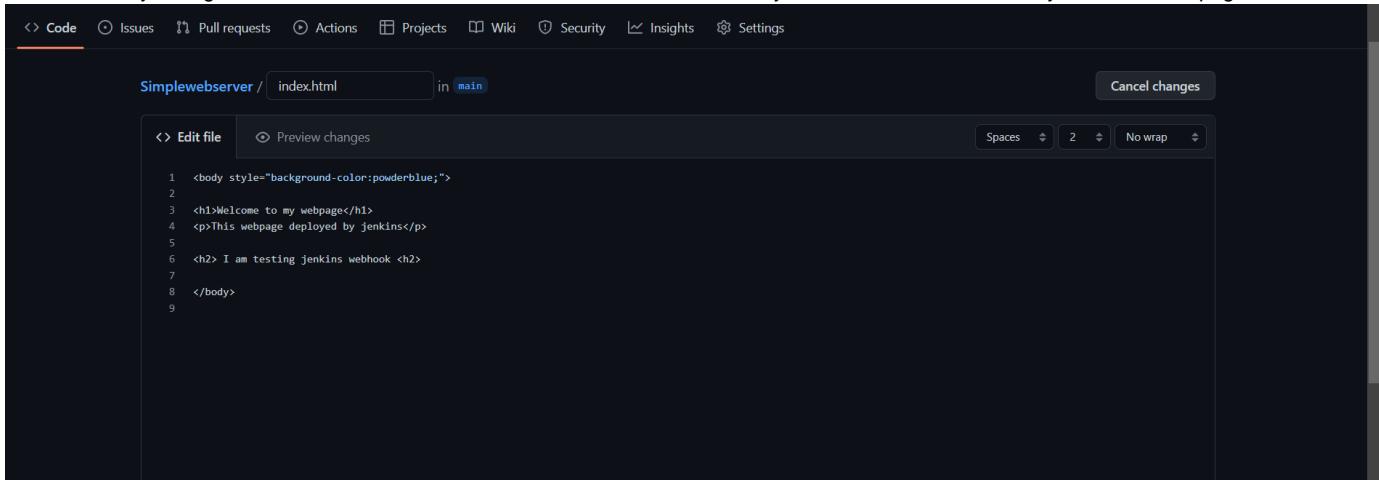


The screenshot shows the 'Webhooks' list page. A single webhook is listed with the URL 'http://43.205.135.193:8080/github-... (push)'. To the right of the URL are 'Edit' and 'Delete' buttons. A red line highlights the URL in the list.

9. Now check content of webserver, by typing url of ec2-instance jenkinswebserevr.



10. Now do any changes in index.html file and commit it. You will see automatically that content will be seen in your webserver page.



11. check jenkins dashboard , you will automatically new build happens.

Dashboard > gitwebserver >
Back to Dashboard

Project gitwebserver

Status This is job to run webserver over jenkins.

Changes

Workspace

Build Now

Configure

Delete Project

GitHub Hook Log

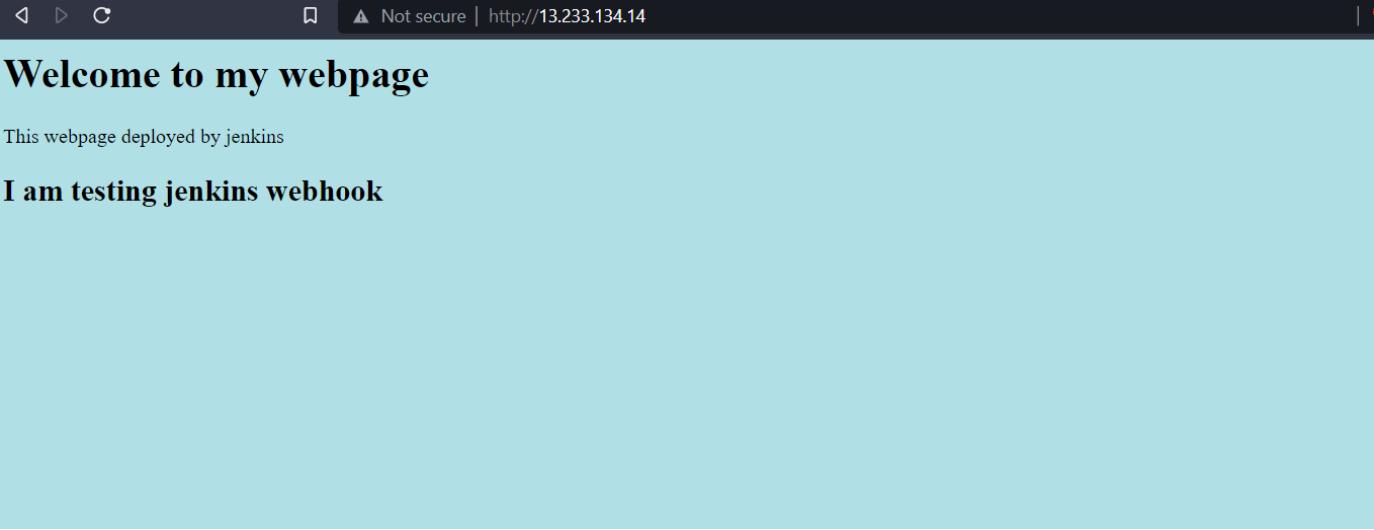
Rename

Build History trend ▾

Filter builds...

#7 Aug 1, 2022, 1:20 PM

12. Refresh the webserver page on browser



Welcome to my webpage

This webpage deployed by jenkins

I am testing jenkins webhook

Webhook trigger is successfully implemented.

To create a simple pipeline Hello world

1. click on New Item

Jenkins

Dashboard >

[+ New Item](#) New Item

[People](#) [All](#) [+](#)

[Build History](#)

[Manage Jenkins](#)

[My Views](#)

[New View](#)

S	W	Name ↓	Last Success	Last Failure
		gitwebserver	1 day 22 hr #7	N/A
		TestJob	1 day 22 hr #4	N/A

Build Queue ▼

No builds in the queue.

Icon: S M L

[Icon legend](#) [Atom feed for all](#) [Atom feed f](#)

Build Executor Status ▼

1 Idle

2 Idle

2. choose pipeline with name “testpipeline”

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.

Maven project Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

Pipeline A red curly arrow points to this item. 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.

Folder Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

OK Pipeline Pipeline projects according to detected branches in one SCM repository.

General Build Triggers Advanced Project Options Pipeline

Description

Test pipeline

[Plain text] Preview

Discard old builds ?
 Do not allow concurrent builds
 Do not allow the pipeline to resume if the controller restarts
 GitHub project
 Pipeline speed/durability override ?
 Preserve stashes from completed builds ?
 This project is parameterized ?

Save **Apply**

choose Hello world

Dashboard > testpipeline >

General Build Triggers **Advanced Project Options** Pipeline

[Advanced...](#)

Pipeline

Definition

Pipeline script

Script ?

```

1  pipeline {
2      agent any
3
4      stages {
5          stage('Hello') {
6              steps {
7                  echo 'Hello World'
8              }
9          }
10     }
11 }
12

```

Use Groovy Sandbox ?

Add more stage in this pipeline

```

pipeline {
    agent any

    stages {
        stage('Hello') {
            steps {
                echo 'Hello World'
            }
        }
        stage('Show date') {
            steps {
                sh 'date'
            }
        }
        stage('Show calendar') {
            steps {
                sh 'cal'
            }
        }
    }
}

```

Pipeline

Definition

Pipeline script

Script

```

4  stages {
5    stage('Hello') {
6      steps {
7        echo 'Hello World'
8      }
9    }
10   stage('Show date') {
11     steps {
12       sh 'date'
13     }
14   }
15   stage('Show calendar') {
16     steps {
17       sh 'cal'
18     }
19   }
20 }

```

Use Groovy Sandbox [?](#)

Save

Apply

click on apply and save

Click on Build Now

Stage View

Delete Pipeline

Full Stage View

Rename

Pipeline Syntax

Average stage times:
(Average full run time: ~3s)

Hello	Show date	Show calendar
213ms	264ms	245ms
111ms	419ms	392ms

Build History

Aug 3, 2022, 12:06 PM

#4 Aug 03 17:36 No Changes

#3 Aug 03 17:36 No Changes

By using console output you can see the output of each steps.

Dashboard > testpipeline > #4

View as plain text

Edit Build Information

Delete build '#4'

Restart from Stage

Replay

Pipeline Steps

Workspaces

Previous Build

```

[Pipeline] {
[Pipeline] stage
[Pipeline] { (Hello)
[Pipeline] echo
Hello World
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Show date)
[Pipeline] sh
+ date
Wed Aug 3 12:06:41 UTC 2022
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Show calendar)
[Pipeline] sh
+ cal
      August 2022
      Su Mo Tu We Th Fr Sa
      1 2 3 4 5 6
      7 8 9 10 11 12 13
      14 15 16 17 18 19 20
      21 22 23 24 25 26 27
      28 29 30 31

[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```

To create a MultiBranch Pipeline

Sometimes it may be necessary to create a pipeline on Jenkins for each Git branch. In this case it could be difficult to create independent pipeline for each branch. Besides that what if we create or delete a branch in future? So someone has to take care of the pipelines on Jenkins whenever there is change in branches. that's where Multibranch pipeline comes into picture.

Pre-Requisites:

Install a plugin nodejs (without restart)

Plugin Manager

Updates Available Installed Advanced

Q node

Install	Name	Released
<input type="checkbox"/>	Role-based Authorization Strategy 552.v14cb_85499b_89	3 days 15 hr ago
<input type="checkbox"/>	Security Authentication and User Management	
Enables user authorization using a Role-Based strategy. Roles can be defined globally or for particular jobs or nodes selected by regular expressions.		
<input checked="" type="checkbox"/>	NodeJS 1.5.1	6 mo 20 days ago
<input checked="" type="checkbox"/>	npm	
NodeJS Plugin executes NodeJS script as a build step.		
<input type="checkbox"/>	built-on-column 1.1	11 yr ago
<input type="checkbox"/>	List view columns	
Shows the actual node that a job was built on		
<input type="checkbox"/>	Node and Label parameter 1.11.0	1 mo 12 days ago
<input type="checkbox"/>	Agent Management Build Parameters Build Triggers	
The node and label parameter plugin allows the node for a job to be selected dynamically.		
<input type="checkbox"/>	Throttle Concurrent Builds 2.8	3 mo 12 days ago
<input type="checkbox"/>	Build Wrappers Cluster Management Agent Management	
This plugin allows for throttling the number of concurrent builds of a project running per node or globally.		

Install without restart **Download now and install after restart** Update information obtained: 22 hr ago **Check now**

Go to manage jenkins global tool configuration add node

It should be exact same as shown below

NodeJS

NodeJS installations

List of NodeJS installations on this system

Add NodeJS

NodeJS
Name

nodejs

Install automatically ?

Install from nodejs.org

Version

NodeJS 16.15.0

For the underlying architecture, if available, force the installation of the 32bit package. Otherwise the build will fail

Force 32bit architecture

Global npm packages to install

Specify list of packages to install globally -- see npm install -g. Note that you can fix the packages version by using the syntax 'packageName@version'

Save

Apply

Create a Multibranch Pipeline

1. Login to Jenkins GUI
2. Click on "New Item" Specify a job name Select "Multibranch Pipeline option"

multibranch-pipeline-example

» 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.



Maven project

Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.



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.



Folder

Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.



Multibranch Pipeline

Creates a set of Pipeline projects according to detected branches in one SCM repository.



Organization Folder

Creates a set of multibranch project subfolders by scanning for repositories.

If you want to create a new item from other existing, you can use this option:



Copy from

Type to autocomplete

OK

3. Give Display Name and Description

4. Under Branch Sources Add source Chose Git and provide GitHub URL

<https://github.com/devopskroy/test-nodejs-app.git>

Branch Sources

Git

Project Repository ?

<https://github.com/devopskroy/test-nodejs-app.git>

Credentials ?

- none -

+ Add

Behaviors

Discover branches

?

Add ▾

Build Configuration

Mode

by Jenkinsfile

Script Path ?

Jenkinsfile

Scan Multibranch Pipeline Triggers

5. Apply and Save the job

Now Jenkins automatically scans the repository and create a job for each branch wherever it finds a Jenkinsfile and initiate first build.



Configure

Scan Multibranch Pipeline Now

Scan Multibranch Pipeline Log

Multibranch Pipeline Events

Delete Multibranch Pipeline

People

Build History

Rename

Config Files

Pipeline Syntax

Credentials

New View

Build Queue (3)[multibranch-pipeline-example » master](#)[multibranch-pipeline-example » stage](#)[multibranch-pipeline-example » dev](#)**Build Executor Status**

1 Idle

2 Idle



Jenkins

Dashboard > multibranch-pipeline-example >

Status Configure Scan Multibranch Pipeline Now Scan Multibranch Pipeline Log Multibranch Pipeline Events Delete Multibranch Pipeline People Build History Rename Config Files Pipeline Syntax Credentials New View

multibranch-pipeline-example

Branches (3)

S	W	Name	Last Success	Last Failure	Last Duration
...	...	dev	N/A	N/A	N/A
...	...	master	N/A	N/A	N/A
...	...	stage	N/A	N/A	N/A

Icon: S M L

Icon legend: Atom feed for all Atom feed for failures Atom feed for just latest builds

Disable Multibranch Pipeline

Build Queue

No builds in the queue.

Build Executor Status

1 multibranch-pipeline-example_	dev	2 (Test)
2 multibranch-pipeline-example_

Jenkins

Dashboard > multibranch-pipeline-example >

Status Configure Scan Multibranch Pipeline Now Scan Multibranch Pipeline Log Multibranch Pipeline Events Delete Multibranch Pipeline People Build History Rename Config Files Pipeline Syntax Credentials New View

multibranch-pipeline-example

Branches (3)

S	W	Name	Last Success	Last Failure	Last Duration
✓	...	dev	2 min 38 sec #1	N/A	10 sec
✓	...	master	2 min 38 sec #1	N/A	13 sec
✓	...	stage	2 min 38 sec #1	N/A	9.2 sec

Icon: S M L

Icon legend: Atom feed for all Atom feed for failures Atom feed for just latest builds

Build Queue

No builds in the queue.

Build Executor Status

1 Idle
2 Idle

↑ Up

Pipeline master

Status Full project name: multibranch-pipeline-example/master

Changes </> Recent Changes

Build Now

View Configuration

Full Stage View

Pipeline Syntax

Build History trend ▾

Filter builds... #1 Aug 3, 2022, 12:55 PM

Atom feed for all Atom feed for failures

Average stage times: (Average full run time: ~13s)

Declarative: Checkout SCM	Declarative: Tool Install	Install Dependencies	Test	Deploy application
1s	95ms	1s	470ms	469ms

Stage View

Permalinks

- Last build (#1), 3 min 16 sec ago
- Last stable build (#1), 3 min 16 sec ago
- Last successful build (#1), 3 min 16 sec ago
- Last completed build (#1), 3 min 16 sec ago

Using Webhook

If you wish to automate the build process in the multibranch pipeline we can use Webhook. This feature is not enabled until we install "Multibranch Scan Webhook Trigger". This enables an option "scan by webhook" under "Scan Multibranch Pipeline Triggers". Here we should give a token. I am giving it as "mytoken". by this time your job looks something like below.

Plugin Manager

Updates Available Installed Advanced

Q multibranch scan

Install Name ↴ Released

Multibranch Scan Webhook Trigger 1.0.9 1 yr 1 mo ago

Trigger that can receive any HTTP request and trigger a multibranch job scan when token matched.

Install without restart **Download now and install after restart** Update information obtained: 22 hr ago **Check now**

Configure pipeline again

 Jenkins

Dashboard > multibranch-pipeline-example >

Status

Configure

Scan Multibranch Pipeline Now

Scan Multibranch Pipeline Log

Multibranch Pipeline Events

Delete Multibranch Pipeline

People

Build History

Rename

Config Files

Pipeline Syntax

Credentials

New View

multibranch-pipeline-example

Branches (3)

S	W	Name ↓	Last Success	Last Failure
		dev	4 min 54 sec #1	N/A
		master	4 min 54 sec #1	N/A
		stage	4 min 54 sec #1	N/A

Icon: S M L

Icon legend Atom feed for all

Scan Multibranch Pipeline Triggers

Build whenever a SNAPSHOT dependency is built ?

Periodically if not otherwise run ?

Scan by webhook ?

Trigger token ?

mytoken

The token to match with webhook token. Receive any HTTP request `JENKINS_URL/multibranch-webhook-trigger/invoke?token=[Trigger token]` if a token match, than a multibranch scan will be triggered.

(from [Multibranch Scan Webhook Trigger](#))

Now to enable auto build process we should provide Jenkins URL with token in the GitHub. in this case like should be

```
http://3.109.62.172:8080/multibranch-webhook-trigger/invoke?  
token=mytoken
```

for this log into GitHub settings Webhooks Add webhook

Screenshot of the GitHub Settings page, specifically the Webhooks section. The left sidebar shows the 'Webhooks' option is selected. The main content area is titled 'Webhooks / Add webhook' and contains fields for 'Payload URL' (set to 'http://3.109.62.172:8080/multibranch-webhook-trigger/invoke?token'), 'Content type' (set to 'application/x-www-form-urlencoded'), and a 'Secret' field. Below these are options for triggering events: 'Just the push event.' (radio button selected), 'Send me everything.', and 'Let me select individual events.' There is also a checked 'Active' checkbox with the note 'We will deliver event details when this hook is triggered.' A green 'Add webhook' button is at the bottom.

In this demonstration, I am going to create a new branch called "test" and push the changes onto remote repo.

Screenshot of the GitHub branches dropdown menu. It shows 'master' is the current branch, and there are 3 branches and 0 tags. A modal window titled 'Switch branches/tags' is open, showing a list of branches. The 'test' branch is selected. Below the list are buttons for 'Branches' and 'Tags', and a link to 'Create branch: test from 'master''. A red arrow points to this link. Other options in the modal include 'View all branches', 'Update Jenkinsfile', and 'updated readme file'.

devopskroy / test-nodejs-app Public

forked from [ravdy/test-nodejs-app](#)

[Code](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

[test](#) [4 branches](#) [0 tags](#)

This branch is 1 commit ahead of ravdy:master.

[Go to file](#) [Add file](#) [Code](#) [Gitpod](#)

[About](#)

No description

[Readme](#) [Star 0 stars](#) [Watch 0 watching](#) [Fork 339 forks](#)

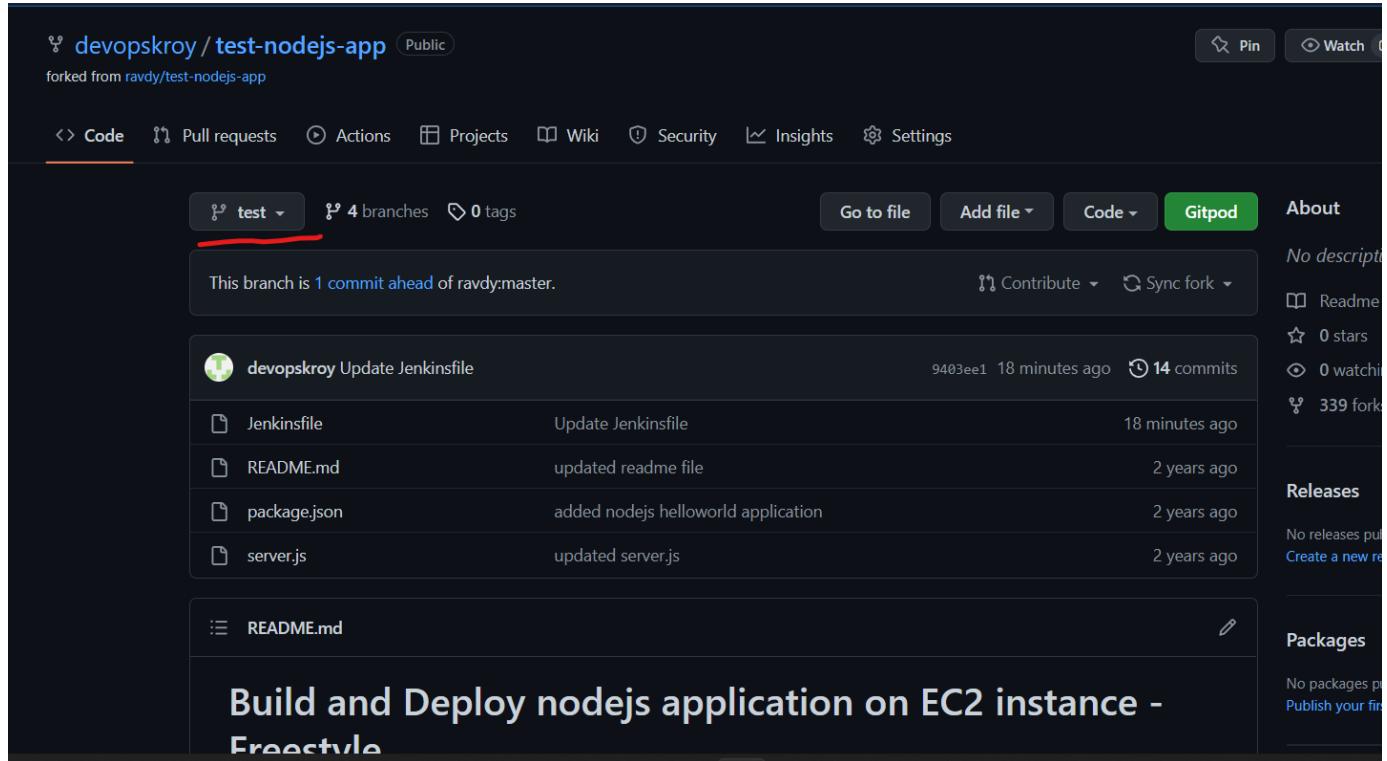
[Releases](#)

No releases published. [Create a new release](#)

[Packages](#)

No packages published. [Publish your first package](#)

Build and Deploy nodejs application on EC2 instance - Freestyle



Jenkins

Dashboard > multibranch-pipeline-example >

Status [Search](#) [Help](#) [2](#) [kishan ray](#) [log out](#)

[Configure](#) [Scan Multibranch Pipeline Now](#) [Scan Multibranch Pipeline Log](#)

[Multibranch Pipeline Events](#) [Delete Multibranch Pipeline](#)

[People](#) [Build History](#) [Rename](#) [Config Files](#) [Pipeline Syntax](#) [Credentials](#) [New View](#)

multibranch-pipeline-example

[Branches \(4\)](#) [Disable Multibranch Pipeline](#)

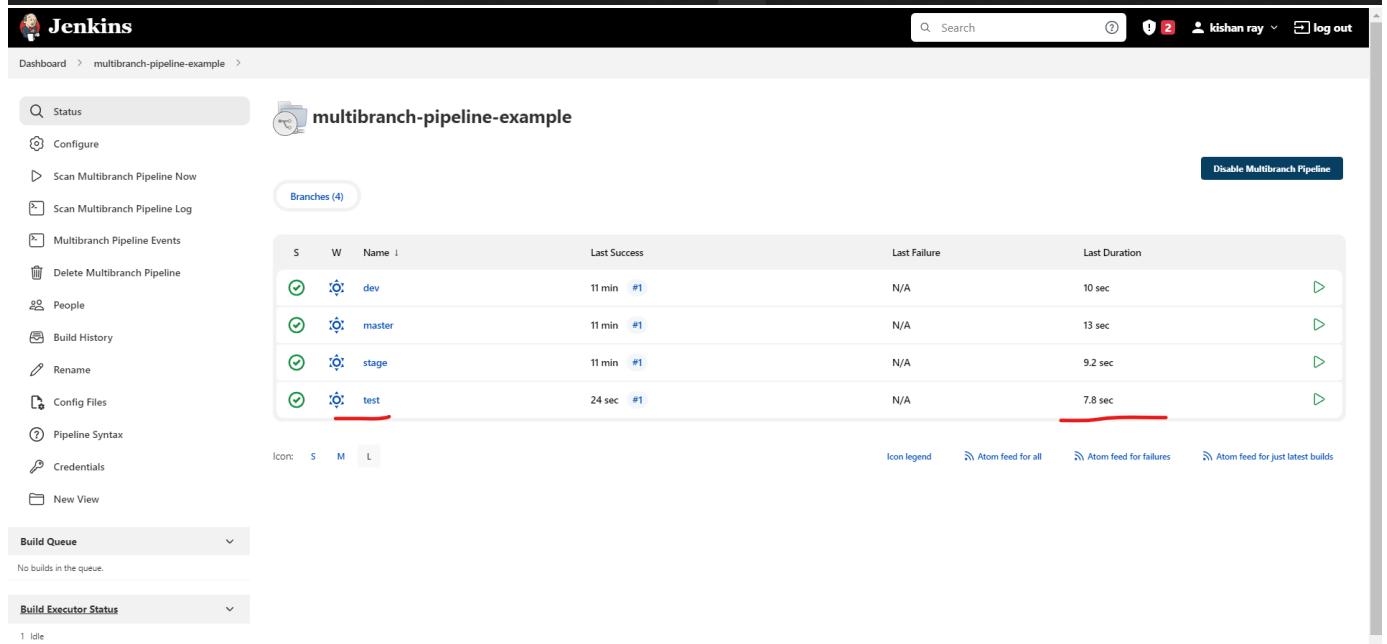
S	W	Name	Last Success	Last Failure	Last Duration
		dev	11 min #1	N/A	10 sec
		master	11 min #1	N/A	13 sec
		stage	11 min #1	N/A	9.2 sec
		test	24 sec #1	N/A	7.8 sec

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

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

Build Queue [▼](#)
No builds in the queue.

Build Executor Status [▼](#)
1 Idle



Multibranch pipeline successfully implemented.

To create a Maven Project

You have to install maven plugin first

Dashboard > Plugin Manager

Plugin Manager

Back to Dashboard | Manage Jenkins | Update Center

Updates Available Installed Advanced

Q: maven

Name	Enabled
Config File Provider Plugin 3.11.1	<input checked="" type="checkbox"/> Report an issue with this plugin
Maven Integration plugin 3.19	<input checked="" type="checkbox"/> Report an issue with this plugin
Pipeline Maven Integration Plugin 1161.v89a_7dcec5d31	<input checked="" type="checkbox"/> Report an issue with this plugin

Now go to Manage Jenkins Global Tool Configuration and configure it and click apply and save

Maven

Maven installations

List of Maven installations on this system

[Add Maven](#)

Maven	X
Name	<input type="text" value="maven"/>
<input checked="" type="checkbox"/> Install automatically ?	
Install from Apache	
Version	<input type="text" value="3.8.6"/>
Add Installer ▾	

[Add Maven](#)

Create a Maven project

Dashboard > All >

Enter an item name

Maventest-project 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.

Maven project Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

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.

Folder Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

Multibranch Pipeline Creates a set of Pipeline projects according to detected branches in one SCM repository.

Organization Folder Creates a set of multibranch project subfolders by scanning for repositories.

If **ok** Create a new item from other existing, you can use this option:

Add git source

`https://github.com/devopskroy/java-hello-world-with-maven.git`

Source Code Management

None

Git ?

Repositories ?

Repository URL ? ×
`https://github.com/devopskroy/java-hello-world-with-maven.git`

Credentials ?
- none - ▼

+ Add

Advanced...

Add Repository

Add pre-build step ▾

Build

Root POM ?

pom.xml

Goals and options ?

clean install

Specifies the goals to execute, such as "clean install" or "deploy". This field can also accept any other command line options to Maven, such as "-e" or "-Djava.net.preferIPv4Stack=true".

Advanced...

Post Steps

Run only if build succeeds

click on apply and save, we are going to only build here.

Now click on Build Now

Dashboard > Maventest-project >

↑ Back to Dashboard

Maven project Maventest-project

 Status

</> Changes

 Workspace

 Build Now

 Configure

 Delete Maven project

 Modules

 Rename

 Workspace

 Recent Changes

Permalinks

 Build History

trend ▾

 Filter builds...

No builds

 Atom feed for all  Atom feed for failures

Build is success

Dashboard > Maventest-project >

[↑ Back to Dashboard](#)

Maven project Maventest-project

[Status](#)

[Changes](#)

[Workspace](#)

[Build Now](#)

[Configure](#)

[Delete Maven project](#)

[Modules](#)

[Rename](#)



Workspace



Recent Changes

Permalinks

[Build History](#) [trend](#) [▼](#)

[#1 Aug 3, 2022, 3:40 PM](#)

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

Dashboard > Maventest-project > #1

```

[INFO] --- maven-resources-plugin:2.0:testResources (default-testResources) @ jb-hello-world-maven ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory /var/lib/jenkins/workspace/Maventest-project/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ jb-hello-world-maven ---
[INFO] No sources to compile
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ jb-hello-world-maven ---
[INFO] No tests to run.
[JENKINS] Recording test results
[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ jb-hello-world-maven ---
[INFO] Building jar: /var/lib/jenkins/workspace/Maventest-project/target/jb-hello-world-maven-0.2.0.jar
[INFO]
[INFO] --- maven-shade-plugin:2.1:shade (default) @ jb-hello-world-maven ---
[INFO] Including joda-time:joda-time:2.2 in the shaded jar.
[INFO] Replacing original artifact with shaded artifact.
[INFO] Replacing /var/lib/jenkins/workspace/Maventest-project/target/jb-hello-world-maven-0.2.0.jar with /var/lib/jenkins/workspace/Maventest-project/target/jb-hello-world-maven-0.2.0-shaded.jar
[INFO] Dependency-reduced POM written at: /var/lib/jenkins/workspace/Maventest-project/dependency-reduced-pom.xml
[INFO]
[INFO] --- maven-install-plugin:2.4:install (default-install) @ jb-hello-world-maven ---
[INFO] Installing /var/lib/jenkins/workspace/Maventest-project/target/jb-hello-world-maven-0.2.0.jar to /var/lib/jenkins/.m2/repository/org/springframework/jb-hello-world-maven/0.2.0/jb-hello-world-maven-0.2.0.jar
[INFO] Installing /var/lib/jenkins/workspace/Maventest-project/dependency-reduced-pom.xml to /var/lib/jenkins/.m2/repository/org/springframework/jb-hello-world-maven/0.2.0/jb-hello-world-maven-0.2.0.pom
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 5.562 s
[INFO] Finished at: 2022-08-03T15:40:55Z
[INFO] -----
Waiting for Jenkins to finish collecting data
[JENKINS] Archiving /var/lib/jenkins/workspace/Maventest-project/dependency-reduced-pom.xml to org.springframework/jb-hello-world-maven/0.2.0/jb-hello-world-maven-0.2.0.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/Maventest-project/target/jb-hello-world-maven-0.2.0.jar to org.springframework/jb-hello-world-maven/0.2.0/jb-hello-world-maven-0.2.0.jar
channel stopped
Finished: SUCCESS

```

Now click on workspace

[↑ Back to Dashboard](#)

Maven project Maventest-project

[Status](#)[Changes](#)[Workspace](#)[Build Now](#)[Configure](#)[Delete Maven project](#)[Modules](#)[Rename](#)

Permalinks

- Last build (#1), 1 min 4 sec ago
- Last stable build (#1), 1 min 4 sec ago
- Last successful build (#1), 1 min 4 sec ago
- Last completed build (#1), 1 min 4 sec ago

[Build History](#) [trend](#) ▾

Filter builds...

[#1 Aug 3, 2022, 3:40 PM](#)

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

click on target

[↑ Back to Dashboard](#)

Workspace of Maventest-project on Built-In Node

Maventest-project /		→
📁	.git	
📁	src/main/java/hello	
📁	target	
📄	.gitignore	Aug 3, 2022, 3:40:44 PM 83 B
📄	dependency-reduced-pom.xml	Aug 3, 2022, 3:40:54 PM 1.19 KB
📄	Jenkinsfile	Aug 3, 2022, 3:40:44 PM 496 B
📄	pom.xml	Aug 3, 2022, 3:40:44 PM 1.90 KB
📄	README.md	Aug 3, 2022, 3:40:44 PM 10.82 KB

[\(all files in zip\)](#)

[Build History](#) [trend](#) ▾

Filter builds...

[#1 Aug 3, 2022, 3:40 PM](#)

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

Click on all files in zip, it will be downloaded to local system

Workspace of Maventest-project on Built-In Node

Maventest-project / target /

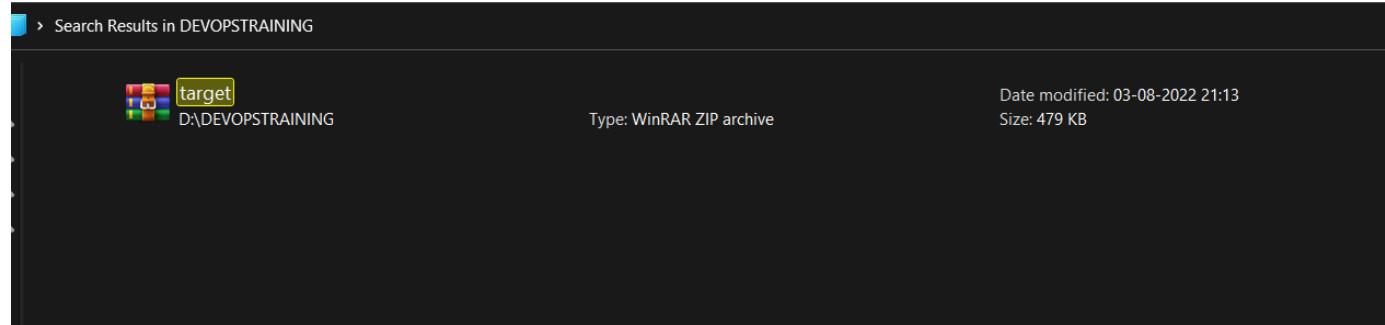
- classes/hello
- generated-sources/annotations
- maven-archiver
- maven-status/maven-compiler-plugin/compile/default-compile
 - jb-hello-world-maven-0.2.0.jar
 - original-jb-hello-world-maven-0.2.0.jar

Aug 3, 2022, 3:40:54 PM 574.06 KB

Aug 3, 2022, 3:40:54 PM 3.00 KB

(all files in zip)

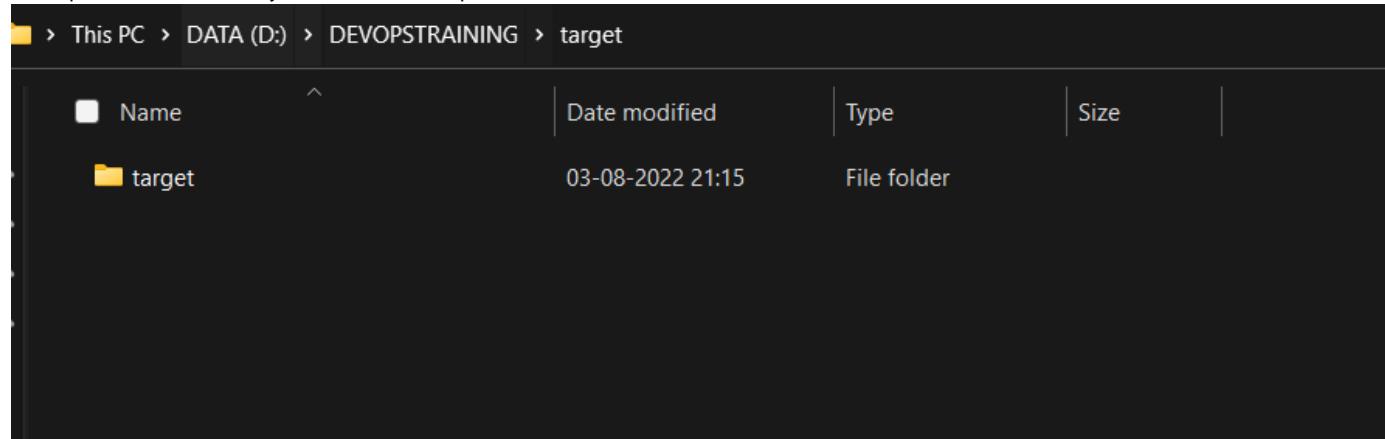
Now extract that zip file



Make sure java is installed in your local machine. Follow this doc to install java.

<https://phoenixnap.com/kb/install-java-windows>

Now open the folder where you extracted that zip file



Open this directory in cmd

```
Microsoft Windows [Version 10.0.22000.832]
(c) Microsoft Corporation. All rights reserved.
```

```
D:\DEVOPSTRAINING\target>
```

To check whether java is installed or not

```
java -version
```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22000.832]
(c) Microsoft Corporation. All rights reserved.

D:\DEVOPSTRAINING\target>java -version
java version "13.0.2" 2020-01-14
Java(TM) SE Runtime Environment (build 13.0.2+8)
Java HotSpot(TM) 64-Bit Server VM (build 13.0.2+8, mixed mode, sharing)

D:\DEVOPSTRAINING\target>
```

To run the executable jar file of our java application

```
java -cp target/jb-hello-world-maven-0.2.0.jar hello.HelloWorld
```

```
D:\DEVOPSTRAINING\target>java -cp target/jb-hello-world-maven-0.2.0.jar hello.HelloWorld
The current local time is: 21:18:15.670
DevOps Project for AchiStar Technologies

D:\DEVOPSTRAINING\target>
```

Maven completed.....

Setup Tomcat server

Reference: <https://blog.devops4me.com/aws-tutorial-how-to-install-tomcat-in-aws-ec2-install/>

1. Launch a ec2-instance with security group all traffic.

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: Create a new security group Select an existing security group

Security group name: launch-wizard-24

Description: launch-wizard-24 created 2022-08-04T20:23:31.112+05:30

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0	e.g. SSH for Admin Desktop
All traffic	All	0 - 65535	Anywhere 0.0.0.0/::0	e.g. SSH for Admin Desktop

[Add Rule](#)

Warning
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Instances (1/2) [Info](#)

Search [Clear filters](#)

Instance state = running [Actions](#) [Launch instances](#)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 D
Jenkins	i-03c865d121dc099ec	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	ec2-52-66-13-
<input checked="" type="checkbox"/> TomcatServer	i-01925e203cabdd781	Running	t2.micro	Initializing	No alarms	ap-south-1a	ec2-15-206-8C

2. Go in your in your EC2 instance and remember log as "sudo" 1st before you execute below command:

```
sudo yum install java-1.8.0-openjdk
wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.65/bin/apache-tomcat-9.0.65.tar.gz
```

```
Last login: Thu Aug  4 14:55:26 2022 from ec2-13-233-177-5.ap-south-1.compute.amazonaws.com
[ec2-user@ip-172-31-41-190 ~]$ sudo su root
[ec2-user@ip-172-31-41-190 ec2-user]# wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.65/bin/apache-tomcat-9.0.65.tar.gz
--2022-08-04 14:56:55-- https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.65/bin/apache-tomcat-9.0.65.tar.gz
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42:644
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 11593900 (11M) [application/x-gzip]
Saving to: 'apache-tomcat-9.0.65.tar.gz'

100%[=====] 11,593,900  --K/s  in 0.1s

2022-08-04 14:56:56 (113 MB/s) - 'apache-tomcat-9.0.65.tar.gz' saved [11593900/11593900]

[ec2-user@ip-172-31-41-190 ec2-user]# ls
apache-tomcat-9.0.65.tar.gz
[ec2-user@ip-172-31-41-190 ec2-user]# 
```

3. Extract TAR file

```
tar -xvf apache-tomcat-9.0.65.tar.gz
```

Rename the Tomcat

Rename tomcat to simplest name so easy for us to maintain in later stage.

```
mv apache-tomcat-9.0.65 /usr/local/tomcat9
[ec2-user@ip-172-31-41-190 ~]# ls
apache-tomcat-9.0.65  apache-tomcat-9.0.65.tar.gz
[ec2-user@ip-172-31-41-190 ~]# mv apache-tomcat-9.0.65 /usr/local/tomcat9
[ec2-user@ip-172-31-41-190 ~]# cd /usr/local
[ec2-user@ip-172-31-41-190 local]# ls
bin  etc  games  include  lib  lib64  libexec  sbin  share  src  tomcat9
[ec2-user@ip-172-31-41-190 local]# 
```

After that your "tomcat9" will look like below;

```
cd tomcat9/
ll -la
```

```
[root@ip-172-31-41-190 ec2-user]# cd /usr/local
[root@ip-172-31-41-190 local]# ls
bin  etc  games  include  lib  lib64  libexec  sbin  share  src  tomcat9
[root@ip-172-31-41-190 local]# cd tomcat9/
[root@ip-172-31-41-190 tomcat9]# ll -la
total 128
drwxr-xr-x  9 root root  220 Aug  4 14:59 .
drwxr-xr-x 13 root root  146 Aug  4 15:01 ..
drwxr-x---  2 root root 4096 Aug  4 14:59 bin
-rw-r----  1 root root 19992 Jul 14 12:28 BUILDING.txt
drwxr----- 2 root root  238 Jul 14 12:28 conf
-rw-r----- 1 root root 6210 Jul 14 12:28 CONTRIBUTING.md
drwxr-x---  2 root root 4096 Aug  4 14:59 lib
-rw-r----- 1 root root 57092 Jul 14 12:28 LICENSE
drwxr-x---  2 root root  6 Jul 14 12:28 logs
-rw-r----- 1 root root 2333 Jul 14 12:28 NOTICE
-rw-r----- 1 root root 3398 Jul 14 12:28 README.md
-rw-r----- 1 root root 6901 Jul 14 12:28 RELEASE-NOTES
-rw-r----- 1 root root 16505 Jul 14 12:28 RUNNING.txt
drwxr-x---  2 root root  30 Aug  4 14:59 temp
drwxr-x---  7 root root  81 Jul 14 12:28 webapps
drwxr-x---  2 root root  6 Jul 14 12:28 work
[root@ip-172-31-41-190 tomcat9]# ]
```

1. Add User for Tomcat

```
useradd -r tomcat
```

1. Give User Permission

Give permission user to tomcat9 folder:

```
chown -R tomcat:tomcat /usr/local/tomcat9
```

1. Create Tomcat Service

Copy below script for your Tomcat Service file

```
sudo tee /etc/systemd/system/tomcat.service<<EOF
[Unit]
Description=Tomcat Server
After=syslog.target network.target

[Service]
Type=forking
User=tomcat
Group=tomcat

Environment=CATALINA_HOME=/usr/local/tomcat9
Environment=CATALINA_BASE=/usr/local/tomcat9
Environment=CATALINA_PID=/usr/local/tomcat9/temp/tomcat.pid

ExecStart=/usr/local/tomcat9/bin/catalina.sh start
ExecStop=/usr/local/tomcat9/bin/catalina.sh stop

RestartSec=12
Restart=always

[Install]
WantedBy=multi-user.target
EOF
```

```
[root@ip-172-31-41-190 tomcat9]# useradd -r tomcat
[root@ip-172-31-41-190 tomcat9]# chown -R tomcat:tomcat /usr/local/tomcat9
[root@ip-172-31-41-190 tomcat9]# sudo tee /etc/systemd/system/tomcat.service<<EOF
> [Unit]
> Description=Tomcat Server
> After=syslog.target network.target
>
> [Service]
> Type=forking
> User=tomcat
> Group=tomcat
>
> Environment=CATALINA_HOME=/usr/local/tomcat9
> Environment=CATALINA_BASE=/usr/local/tomcat9
> Environment=CATALINA_PID=/usr/local/tomcat9/temp/tomcat.pid
>
> ExecStart=/usr/local/tomcat9/bin/catalina.sh start
> ExecStop=/usr/local/tomcat9/bin/catalina.sh stop
>
> RestartSec=12
> Restart=always
>
> [Install]
> WantedBy=multi-user.target
> EOF
[Unit]
Description=Tomcat Server
After=syslog.target network.target
After=syslog.target network.target
```

```
[Service]
Type=forking
User=tomcat
Group=tomcat

Environment=CATALINA_HOME=/usr/local/tomcat9
Environment=CATALINA_BASE=/usr/local/tomcat9
Environment=CATALINA_PID=/usr/local/tomcat9/temp/tomcat.pid

ExecStart=/usr/local/tomcat9/bin/catalina.sh start
ExecStop=/usr/local/tomcat9/bin/catalina.sh stop

RestartSec=12
Restart=always

[Install]
WantedBy=multi-user.target
[root@ip-172-31-41-190 tomcat9]# ]
```

Enable and start tomcat service:

```
systemctl daemon-reload
systemctl start tomcat
systemctl enable tomcat
systemctl status tomcat
```

```
[root@ip-172-31-41-190 tomcat9]# systemctl daemon-reload
[root@ip-172-31-41-190 tomcat9]# systemctl start tomcat
[root@ip-172-31-41-190 tomcat9]# systemctl status tomcat
● tomcat.service - Tomcat Server
   Loaded: loaded (/etc/systemd/system/tomcat.service; disabled; vendor preset: disabled)
     Active: active (running) since Thu 2022-08-04 15:11:56 UTC; 24s ago
       Main PID: 4581 (java)
      CGroup: /usr/lib/system.slice/tomcat.service
              └─4581 /usr/bin/java -Djava.util.logging.config.file=/usr/local/tomcat9/conf/logging.properties -Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -Djdk.tls.epe...
```

Aug 04 15:11:55 ip-172-31-41-190.ap-south-1.compute.internal systemd[1]: tomcat.service holdoff time over, scheduling restart.
Aug 04 15:11:55 ip-172-31-41-190.ap-south-1.compute.internal systemd[1]: Stopped Tomcat Server.
Aug 04 15:11:55 ip-172-31-41-190.ap-south-1.compute.internal systemd[1]: Starting Tomcat Server...
Aug 04 15:11:56 ip-172-31-41-190.ap-south-1.compute.internal systemd[1]: Started Tomcat Server.
[root@ip-172-31-41-190 tomcat9]#

Check Tomcat Service and Host-Manager(GUI)

Finally we check on our browser the Tomcat Server GUI via [http://\[AWS EC2 Public IP\]:8080](http://[AWS EC2 Public IP]:8080)

Go to tomcat ec2-instance

webapps directory : we keep war files here

```
[root@ip-172-31-41-190 tomcat9]# pwd
/usr/local/tomcat9
[root@ip-172-31-41-190 tomcat9]# ls
bin BUILDING.txt conf CONTRIBUTING.md lib LICENSE logs NOTICE README.md RELEASE-NOTES RUNNING.txt temp webapps work
[root@ip-172-31-41-190 tomcat9]#
```

To deploy a war file on tomcat server

1. install ssh agent plugin (install without restart)

Plugin Manager

Updates Available Installed Advanced

Q ssh age

Install Name ↓

Released

SSH Agent 295.v9ca_a_1c7cc3a_a_

This plugin allows you to provide SSH credentials to builds via a ssh-agent in Jenkins.

2 mo 25 days ago

Install without restart

Download now and install after restart

Update information obtained: 53 min ago

Check now

2. Create a pipeline Job

Enter an item name

tomcatpipeline

» 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.



Maven project

Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.



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.



OK

Container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a

click on pipeline syntax

Definition

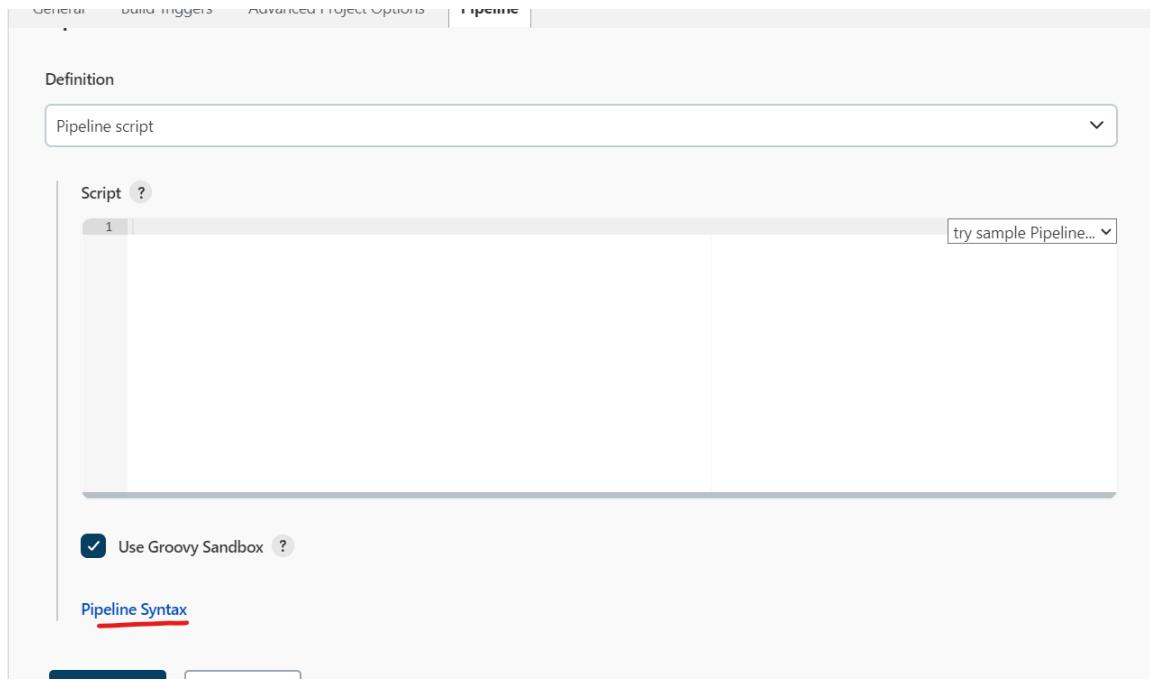
Pipeline script

Script ?

1 try sample Pipeline... ▾

Use Groovy Sandbox ?

Pipeline Syntax



Overview

This **Snippet Generator** will help you learn the Pipeline Script code which can be used to define various steps. Pick a step you are interested in from the list, configure it, click **Generate Pipeline Script**, and you will see a Pipeline Script statement that would call the step with that configuration. You may copy and paste the whole statement into your script, or pick up just the options you care about. (Most parameters are optional and can be omitted in your script, leaving them at default values.)

Steps

Sample Step

ssagent: SSH Agent

sshagent ?

ec2-user (deploy_user)

+ Add

Ignore missing credentials ?

Generate Pipeline Script

Kind: SSH Username with private key

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

ID: deploy_user

Description: deploy_user

Username: ec2-user

Treat username as secret

Generate Pipeline Script

Click on generate pipeline script

ec2-user (deploy_user) ?

+ Add

Ignore missing credentials ?

Generate Pipeline Script

```
sshagent(['deploy_user']) {
    // some block
}
```

Global Variables

There are many features of the Pipeline that are not steps. These are often exposed via global variables, which are not supported by the snippet generator. See the [Global Variables Reference](#) for details.

The project we are going to deploy , file structure of the project will be like this



Jenkins

Dashboard > tomcatpipeline > #4 > Allocate node : Start > Workspace

Up

Workspace

Status

Console Output

Workspace

/ webapp / target /

└── maven-archiver
 └── surefire
 └── webapp
 └── webapp.war

Aug 4, 2022, 3:49:00 PM 2.00 KB

↳ (all files in zip)

Go to tomcat server ec2-instance and do changes by changing the owner of tomcat9 directory

```
usermod -a -G tomcat ec2-user #will add ec2-user in tomcat group
chmod 771 /user/local/tomcat9/webapps #will give write permission to
ec2-user
```

```
-rw-r----- 1 tomcat tomcat 16505 Jul 14 12:28 RUNNING.txt
drwxr-x--- 2 tomcat tomcat     48 Aug  4 16:30 temp
drwxr-x--- 7 tomcat tomcat    81 Aug  4 16:27 webapps
drwxr-x--- 3 tomcat tomcat    22 Aug  4 15:11 work
[root@ip-172-31-41-190 tomcat9]# chmod 771 webapps/
[root@ip-172-31-41-190 tomcat9]# ls -l
total 128
drwxr-x--- 2 tomcat tomcat  4096 Aug  4 14:59 bin
-rw-r----- 1 tomcat tomcat 19992 Jul 14 12:28 BUILDING.txt
drwx----- 3 tomcat tomcat   254 Aug  4 15:11 conf
-rw-r----- 1 tomcat tomcat  6210 Jul 14 12:28 CONTRIBUTING.md
drwxr-x--- 2 tomcat tomcat  4096 Aug  4 14:59 lib
-rw-r----- 1 tomcat tomcat 57092 Jul 14 12:28 LICENSE
drwxr-x--- 2 tomcat tomcat   197 Aug  4 15:11 logs
-rw-r----- 1 tomcat tomcat  2333 Jul 14 12:28 NOTICE
-rw-r----- 1 tomcat tomcat  3398 Jul 14 12:28 README.md
-rw-r----- 1 tomcat tomcat  6901 Jul 14 12:28 RELEASE-NOTES
-rw-r----- 1 tomcat tomcat 16505 Jul 14 12:28 RUNNING.txt
drwxr-x--- 2 tomcat tomcat     48 Aug  4 16:30 temp
drwxrwx--x 7 tomcat tomcat    81 Aug  4 16:27 webapps
drwxr-x--- 3 tomcat tomcat    22 Aug  4 15:11 work
[root@ip-172-31-41-190 tomcat9]#
```

> Pipeline Syntax

The options you can add to these parameters are optional and can be omitted in your script leaving them at default values.

Implementation Steps

Sample Step

sshagent: SSH Agent

sshagent ?

```
node {
    sshagent (credentials: ['deploy-dev']) {
        sh 'ssh -o StrictHostKeyChecking=no -l cloudbuild 192.168.1.106 uname -a'
    }
}
```

Multiple credentials could be passed in the array but it is not supported using Snippet Generator.

(from [SSH Agent Plugin](#))

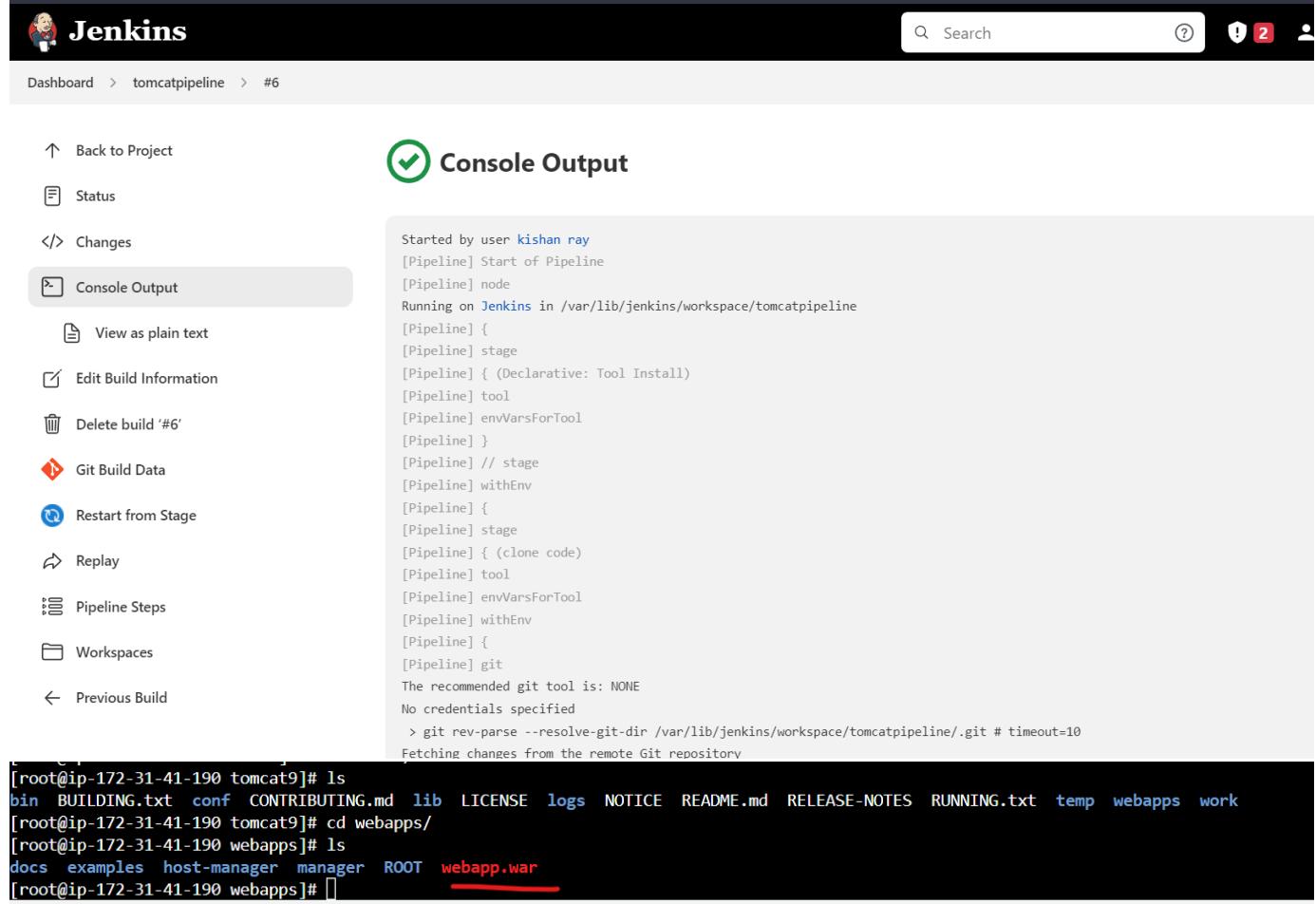
ec2-user (deploy user)

Now configure that pipeline job

```
pipeline{
    agent any

    tools {
        maven 'maven'
    }
    stages{
        stage('clone code'){
            steps{
                git url: 'https://github.com/devopskroy/hello-world.git'
            }
        }
        stage('build code'){
            steps{
                sh 'mvn clean install'
            }
        }
        stage('deploy code'){
            steps{
                sshagent(['deploy_user']) {
                    sh "scp -o StrictHostKeyChecking=no webapp/target/webapp.war ec2-user@15.206.80.81:/usr/local/tomcat9/webapps"
                }
            }
        }
    }
}
```

Save and apply

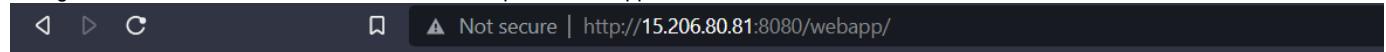


The screenshot shows the Jenkins interface for a pipeline job named 'tomcatpipeline'. The 'Console Output' tab is selected, displaying the build logs. The logs show the pipeline starting, cloning code from a Git repository, and copying a 'webapp.war' file to the Tomcat 'webapps' directory. The 'Console Output' tab is highlighted with a red box.

```
Started by user kishan ray
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/tomcatpipeline
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Tool Install)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (clone code)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] withEnv
[Pipeline] {
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/tomcatpipeline/.git # timeout=10
Fetching changes from the remote Git repository
[root@ip-172-31-41-190 tomcat9]# ls
bin BUILDING.txt conf CONTRIBUTING.md lib LICENSE logs NOTICE README.md RELEASE-NOTES RUNNING.txt temp webapps work
[root@ip-172-31-41-190 tomcat9]# cd webapps/
[root@ip-172-31-41-190 webapps]# ls
docs examples host-manager manager ROOT webapp.war
[root@ip-172-31-41-190 webapps]#
```

Webapp.war is successfully copied to tomcat server

Now go to tomcat server url on browser : <ec2-user ip:8080/webapp>



A screenshot of a web browser showing the Tomcat 'Hello, Welcome to Simple DevOps Project !!' welcome page. The URL is http://15.206.80.81:8080/webapp. The browser interface includes back, forward, and search buttons.

Hello, Welcome to Simple DevOps Project !!

Deploying on a Tomcat server

Glad to see you here

Thanks Kishan Ray

Successfully deployed our webserver on tomcat using jenkins.