

Integration of Devops tools with Jenkins

You have been Hired as a Devops Engineer in xyz software company. They want to implement CI/CD pipeline in their company. You have been asked to implement this lifecycle as fast as possible. As this is a product-based company, their product is available on this GitHub link.

<https://github.com/hshar/website.git>

Following are the specifications of the Continuous integration:

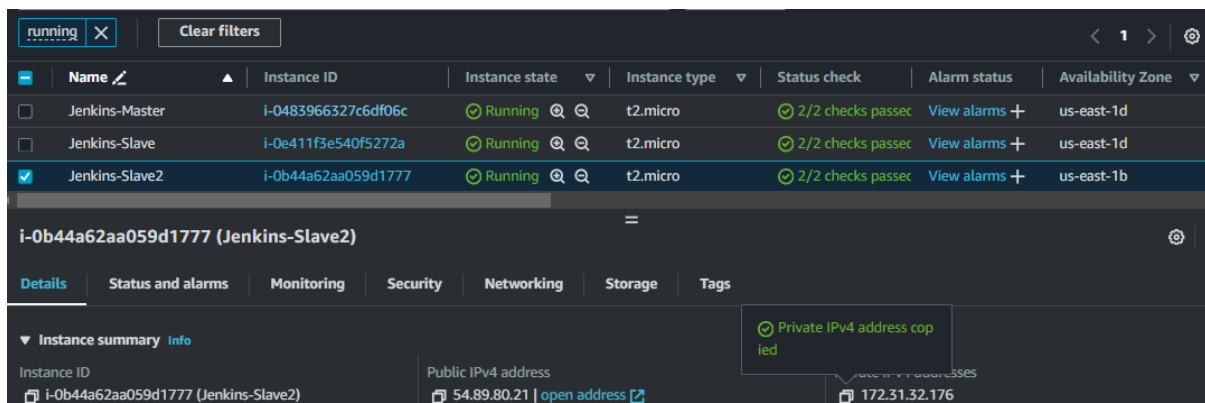
1. Git Workflow has to be implemented
2. Code Build should automatically be triggered once commit is made to master branch or develop branch.

If commit is made to master branch, build and publish website on

port 82. If commit is made to develop branch, just build the product, do not publish.

3. Create a pipeline for the above tasks.
4. Create a container with Ubuntu and apache installed in it and use that container to build the code and the code should be on '/var/www/html'.

Continuation of After Completing 3 Assignment



The screenshot displays the AWS Management Console interface. At the top, there's a filter bar with 'running' selected and a 'Clear filters' button. Below this is a table of EC2 instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Jenkins-Master	i-0483966327c6df06c	Running	t2.micro	2/2 checks passed	View alarms	us-east-1d
Jenkins-Slave	i-0e411f3e540f5272a	Running	t2.micro	2/2 checks passed	View alarms	us-east-1d
Jenkins-Slave2	i-0b44a62aa059d1777	Running	t2.micro	2/2 checks passed	View alarms	us-east-1b

Below the table, the details for the selected instance 'i-0b44a62aa059d1777 (Jenkins-Slave2)' are shown. The 'Details' tab is active, displaying the instance summary:

- Instance ID: i-0b44a62aa059d1777 (Jenkins-Slave2)
- Public IPv4 address: 54.89.80.21 | open address
- Private IPv4 address: 172.31.32.176

A tooltip indicates that the 'Private IPv4 address' has been copied.

Jenkins Search (CTRL+K) SalmanFarsi123 log out

Dashboard > Manage Jenkins > Nodes >

Nodes [+ New Node](#) [Configure Monitors](#)

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	4.58 GiB	0 B	4.58 GiB	0ms
	Prod	Linux (amd64)	In sync	5.26 GiB	0 B	5.26 GiB	41ms
	Test	Linux (amd64)	In sync	5.34 GiB	0 B	5.34 GiB	28ms
	Data obtained	22 min	22 min	22 min	22 min	22 min	22 min

Build Queue: No builds in the queue.

Build Executor Status: Built-In Node

Dockerfile

SalmanFarsi123 / website

Code Pull requests Actions Projects Wiki Security Insights Settings

Files: master Dockerfile index.html

website / Dockerfile

SalmanFarsi123 Create Dockerfile 230199 · last week History

```

1 FROM ubuntu
2 RUN apt-get update -y
3 RUN apt-get install apache2 -y
4 COPY . /var/www/html/
5 ENTRYPOINT apache2ctl -D FOREGROUND

```

Copy the Path to change

SalmanFarsi123 / website

Code Pull requests Actions Projects Wiki Security Insights Settings

website Public forked from hshar/website

master 2 Branches 0 Tags

This branch is 1 commit ahead of hshar/website:master.

SalmanFarsi123 Create Dockerfile

images final

Dockerfile Create Dockerfile

index.html modified

README

Local Codespaces

Clone

HTTPS SSH GitHub CLI

<https://github.com/SalmanFarsi123/website.git>

Clone using the web URL

Open with GitHub Desktop

Download ZIP

About: No description, website

Activity: 0 stars, 0 watching, 0 forks

Releases: No releases published, Create a new release

Packages

I renamed Job Names, Changing Git Repo Here

Dashboard > Test-Job > Configuration

Configure

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

☒ Restrict where this project can be run ?

Label Expression ?

Test

Label Test matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.

Advanced ▾

Source Code Management

☐ None

☒ Git ?

Repositories ?

Repository URL ?

https://github.com/SalmanFarsi123/website.git

Credentials ?

- none -

+ Add ▾

Save Apply

Dashboard > Test-Job > Configuration

Configure

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

Branch Specifier (blank for 'any') ?

*/develop

Add Branch

Repository browser ?

(Auto)

Additional Behaviours

Add ▾

Build Triggers

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

☐ Build after other projects are built ?

☐ Build periodically ?

☒ GitHub hook trigger for GITScm polling ?

Save Apply

Now Prod Job, Changing Git Name here also

Dashboard > Prod-Job > Configuration

Configure

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

☒ Restrict where this project can be run ?

Label Expression ?

Prod

Label Prod matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.

Advanced ▾

Source Code Management

☐ None

☒ Git ?

Repositories ?

Repository URL ?

https://github.com/SalmanFarsi123/website.git

Credentials ?

- none - ▾

And Changing Branch Name here

Dashboard > Prod-Job > Configuration

Configure

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

Branch Specifier (blank for 'any') ?

*/master

Add Branch

Repository browser ?

(Auto)

Additional Behaviours

Add ▾

Build Triggers

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

☐ Build after other projects are built ?

☐ Build periodically ?

☒ GitHub hook trigger for GITScm polling ?

☐ Poll SCM ?

Save Apply

Now Installing docker in master, slave1 and slave2

```
ubuntu@Jenkins-Master: ~  
ubuntu@Jenkins-Master:~$ sudo apt install docker.io -y  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following additional packages will be installed:  
  bridge-utils containerd dns-root-data dnsmasq-base pigz runc ubuntu-fan  
Suggested packages:
```

```
ubuntu@Jenkins-Master: ~  
ubuntu@Jenkins-Slave: ~  
ubuntu@Jenkins-Slave:~$ sudo apt install docker.io  
Reading package lists... Done  
Building dependency tree... Done
```

```
ubuntu@Jenkins-Master: ~  
ubuntu@Jenkins-Slave: ~  
ubuntu@Jenkins-Slave2: ~  
ubuntu@Jenkins-Slave2:~$ sudo apt install docker.io
```

Now Configuring Prod-Job and it will trigger to slave2 and pull data from master branch and save it

Dashboard > Prod-Job > Configuration

Configure

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

☐ Terminate a build if it's stuck

☐ With Ant ?

Build Steps

Execute shell ?

Command

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

```
#sudo docker rm -f c1  
sudo docker build . -t image1  
sudo docker run -itd -p 82:80 --name=c1 image1
```

Advanced ▾

Add build step ▾

Post-build Actions

Save Apply

Build Job1 and it will trigger job2 also

Jenkins Search (CTRL+K) SalmanFarsi log out

Dashboard > Test-Job

Status **Test-Job** Add description Disable Project

Changes Workspace Build Now Configure Delete Project GitHub Hook Log Rename

Downstream Projects

Prod-Job

Permalinks

- Last build (#3), 35 min ago
- Last stable build (#3), 35 min ago
- Last successful build (#3), 35 min ago
- Last completed build (#3), 35 min ago

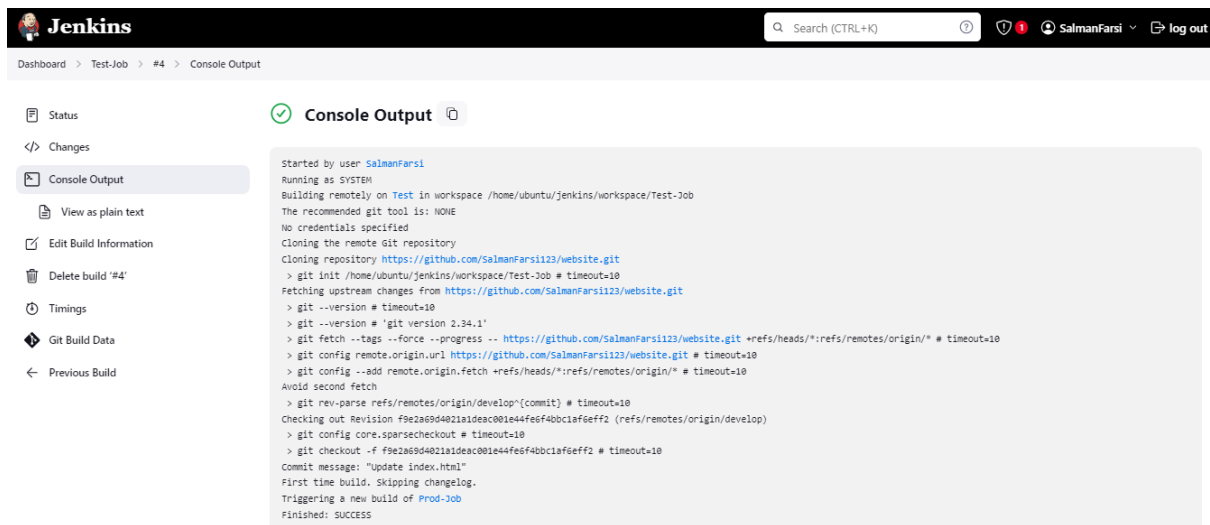
Build History

trend ▾

Filter...

#4 Jun 19, 2024, 1:04 PM

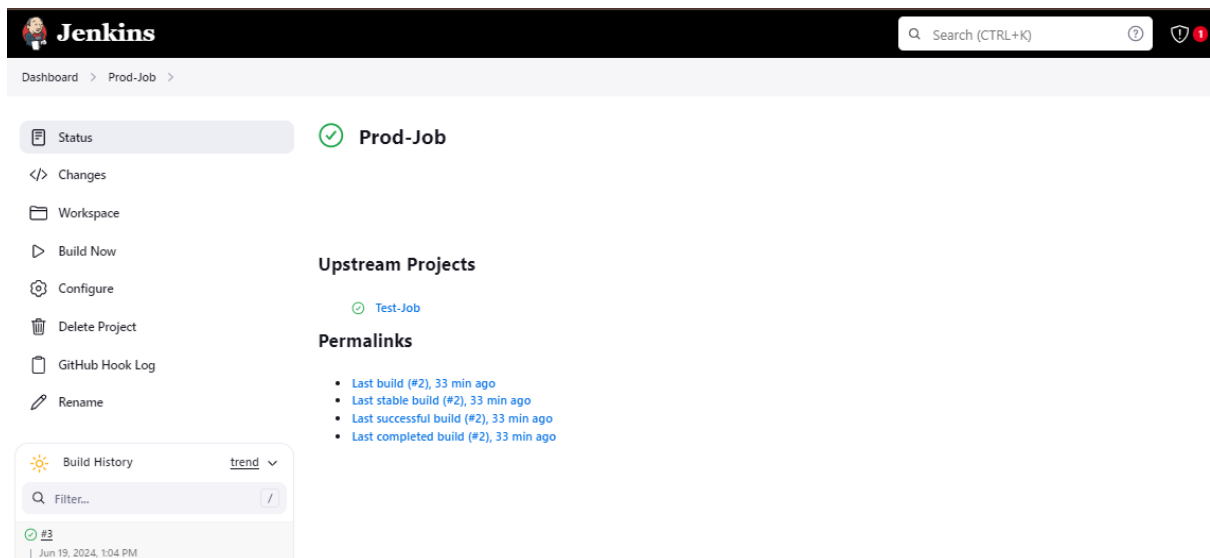
Successfully ran



The screenshot shows the Jenkins web interface. The top navigation bar includes the Jenkins logo, a search bar, and user information for 'SalmanFarsi'. The breadcrumb trail is 'Dashboard > Test-Job > #4 > Console Output'. On the left sidebar, the 'Console Output' tab is selected. The main content area displays the console output for a build, which is a successful Git checkout and build process. The output includes details about the user, workspace, repository cloning, and the final build status.

```
Started by user SalmanFarsi
Running as SYSTEM
Building remotely on Test in workspace /home/ubuntu/jenkins/workspace/Test-Job
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/SalmanFarsi123/website.git
> git init /home/ubuntu/jenkins/workspace/Test-Job # timeout=10
Fetching upstream changes from https://github.com/SalmanFarsi123/website.git
> git --version # timeout=10
> git --version # 'git version 2.34.1'
> git fetch --tags --force --progress -- https://github.com/SalmanFarsi123/website.git # timeout=10
> git config remote.origin.url https://github.com/SalmanFarsi123/website.git # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> git rev-parse refs/remotes/origin/develop^{commit} # timeout=10
Checking out Revision f9e2a69d4821a1deac001e44fe6f4bcb1af6eff2 (refs/remotes/origin/develop)
> git config core.sparsecheckout # timeout=10
> git checkout -f f9e2a69d4821a1deac001e44fe6f4bcb1af6eff2 # timeout=10
Commit message: "Update index.html"
First time build. Skipping changelog.
Triggering a new build of Prod-Job
Finished: SUCCESS
```

And Also triggers Job2 and ran successfully



The screenshot shows the Jenkins web interface for the 'Prod-Job'. The top navigation bar is the same as the previous screenshot. The breadcrumb trail is 'Dashboard > Prod-Job >'. On the left sidebar, the 'Status' tab is selected. The main content area displays the 'Prod-Job' overview, including 'Upstream Projects' (Test-Job) and 'Permalinks' for the last build. A 'Build History' widget is visible at the bottom left, showing the last build (#2) as successful.

Prod-Job

Upstream Projects

- Test-Job

Permalinks

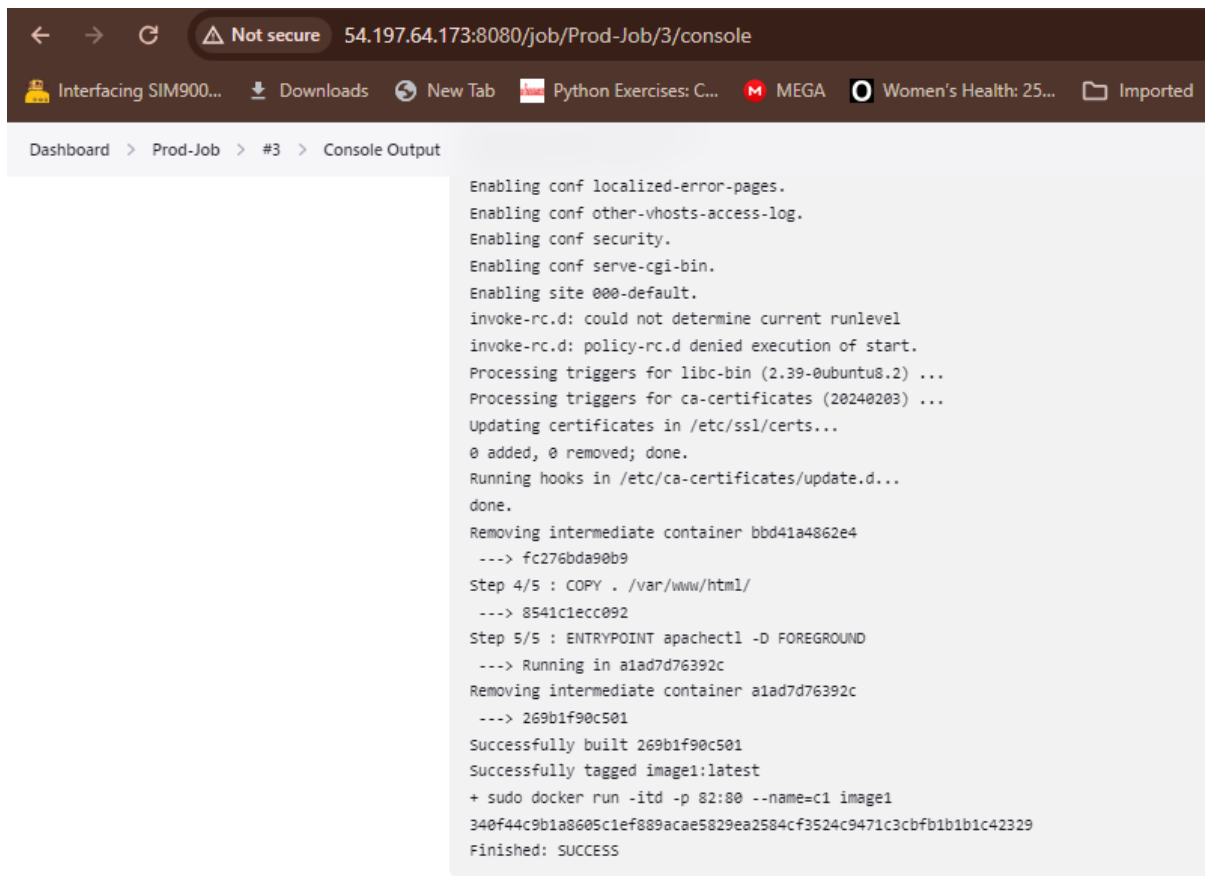
- Last build (#2), 33 min ago
- Last stable build (#2), 33 min ago
- Last successful build (#2), 33 min ago
- Last completed build (#2), 33 min ago

Build History

Filter... /

#2 | Jun 19, 2024, 1:04 PM

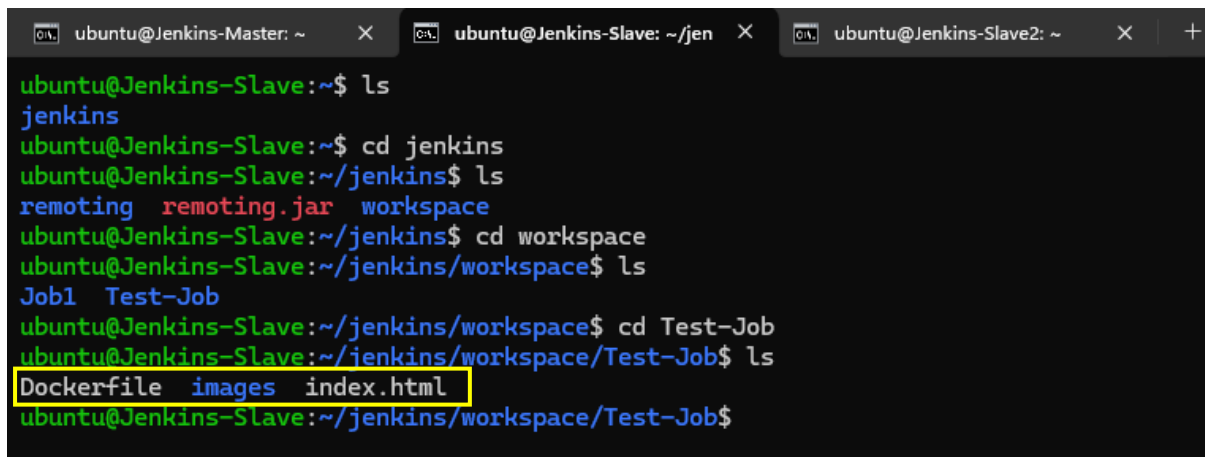
This Console output of Prod-Job



The screenshot shows a web browser window with the address bar displaying '54.197.64.173:8080/job/Prod-Job/3/console'. The browser's address bar also shows 'Not secure'. The browser's tabs include 'Interfacing SIM900...', 'Downloads', 'New Tab', 'Python Exercises: C...', 'MEGA', 'Women's Health: 25...', and 'Imported'. The browser's address bar also shows 'Dashboard > Prod-Job > #3 > Console Output'. The console output shows the following steps:

```
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
invoke-rc.d: could not determine current runlevel
invoke-rc.d: policy-rc.d denied execution of start.
Processing triggers for libc-bin (2.39-0ubuntu8.2) ...
Processing triggers for ca-certificates (20240203) ...
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
Removing intermediate container bbd41a4862e4
--> fc276bda90b9
Step 4/5 : COPY . /var/www/html/
--> 8541c1ecc092
Step 5/5 : ENTRYPOINT apachectl -D FOREGROUND
--> Running in a1ad7d76392c
Removing intermediate container a1ad7d76392c
--> 269b1f90c501
Successfully built 269b1f90c501
Successfully tagged image1:latest
+ sudo docker run -itd -p 82:80 --name=c1 image1
340f44c9b1a8605c1ef889acae5829ea2584cf3524c9471c3cbfb1b1b1c42329
Finished: SUCCESS
```

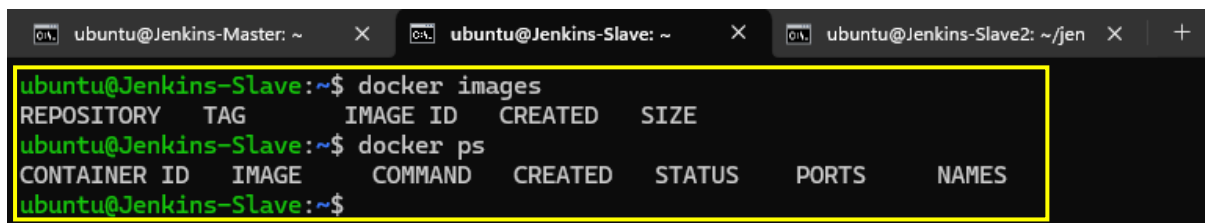
Slave1 Its pull the data



The screenshot shows a terminal window with the following command sequence:

```
ubuntu@Jenkins-Master: ~
ubuntu@Jenkins-Slave: ~/jen
ubuntu@Jenkins-Slave2: ~
ubuntu@Jenkins-Slave:~$ ls
jenkins
ubuntu@Jenkins-Slave:~$ cd jenkins
ubuntu@Jenkins-Slave:~/jenkins$ ls
remoting remoting.jar workspace
ubuntu@Jenkins-Slave:~/jenkins$ cd workspace
ubuntu@Jenkins-Slave:~/jenkins/workspace$ ls
Job1 Test-Job
ubuntu@Jenkins-Slave:~/jenkins/workspace$ cd Test-Job
ubuntu@Jenkins-Slave:~/jenkins/workspace/Test-Job$ ls
Dockerfile images index.html
ubuntu@Jenkins-Slave:~/jenkins/workspace/Test-Job$
```

But here there is no images and there is no running container we are only testing for this server



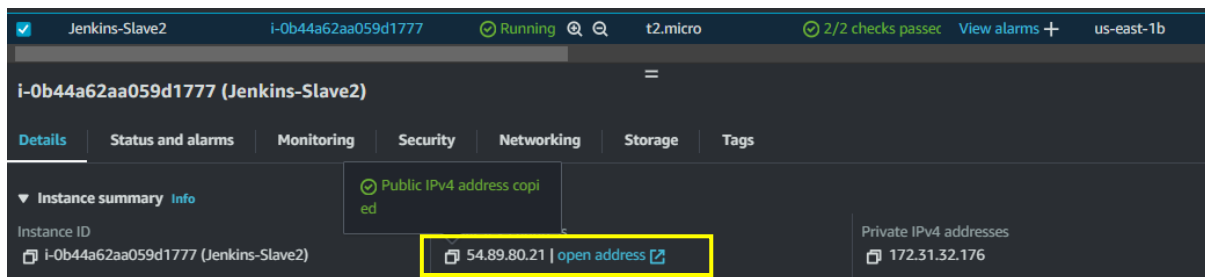
The screenshot shows a terminal window with the following command sequence:

```
ubuntu@Jenkins-Master: ~
ubuntu@Jenkins-Slave: ~
ubuntu@Jenkins-Slave2: ~/jen
ubuntu@Jenkins-Slave:~$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
ubuntu@Jenkins-Slave:~$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
ubuntu@Jenkins-Slave:~$
```

Slave2

```
ubuntu@Jenkins-Master: ~  
ubuntu@Jenkins-Slave: ~/jenk  
ubuntu@Jenkins-Slave2: ~/je  
ubuntu@Jenkins-Slave2:~$ cd jenkins/workspace  
ubuntu@Jenkins-Slave2:~/jenkins/workspace$ ls  
Job2  Prod-Job  
ubuntu@Jenkins-Slave2:~/jenkins/workspace$ cd Prod-Job  
ubuntu@Jenkins-Slave2:~/jenkins/workspace/Prod-Job$ ls  
Dockerfile  images  index.html  
ubuntu@Jenkins-Slave2:~/jenkins/workspace/Prod-Job$
```

Copy the Public Ip of slave2



We can see the web-browser, its worked



Hello world!



Docker images and running containers are there in slave2

```
ubuntu@Jenkins-Master: ~  
ubuntu@Jenkins-Slave: ~/jenk  
ubuntu@Jenkins-Slave2: ~/je  
ubuntu@Jenkins-Slave2:~/jenkins/workspace/Prod-Job$ docker images  
REPOSITORY TAG IMAGE ID CREATED SIZE  
image1 latest 269b1f90c501 6 minutes ago 222MB  
ubuntu latest 35a88802559d 12 days ago 78.1MB  
ubuntu@Jenkins-Slave2:~/jenkins/workspace/Prod-Job$ docker ps  
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES  
340f44c9b1a8 image1 "/bin/sh -c 'apachec..." 6 minutes ago Up 6 minutes 0.0.0.0:82->80/tcp, :::82->80/tcp c1  
ubuntu@Jenkins-Slave2:~/jenkins/workspace/Prod-Job$ docker
```


Now I am removing Hash and now it will remove c1 and create it Again

Dashboard > Prod-Job > Configuration

Configure

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

☐ SSH Agent

☐ Terminate a build if it's stuck

☐ With Ant ?

Build Steps

Execute shell ?

Command

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

```
sudo docker rm -f c1
sudo docker build . -t image1
sudo docker run -itd -p 82:80 --name=c1 image1
```

Advanced ▾

Add build step ▾

Save Apply

Go to Settings for webhook > Add Webhook

SalmanFarsi123 / website

Code Pull requests Actions Projects Wiki Security Insights **Settings**

Okay, that hook was successfully deleted.

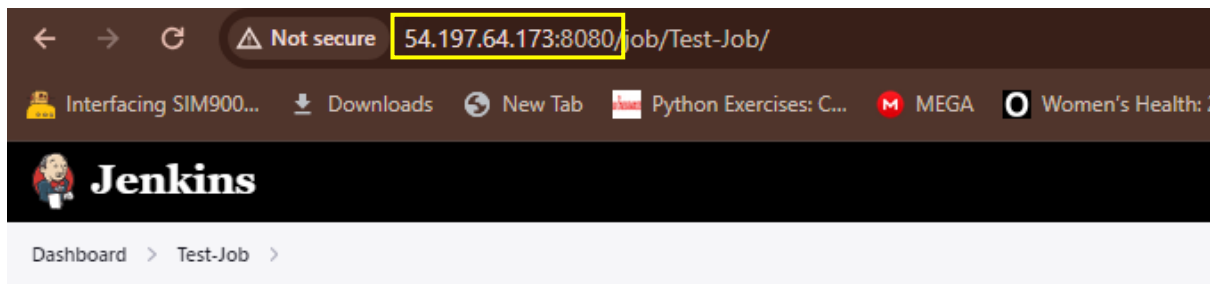
Webhooks

Add webhook

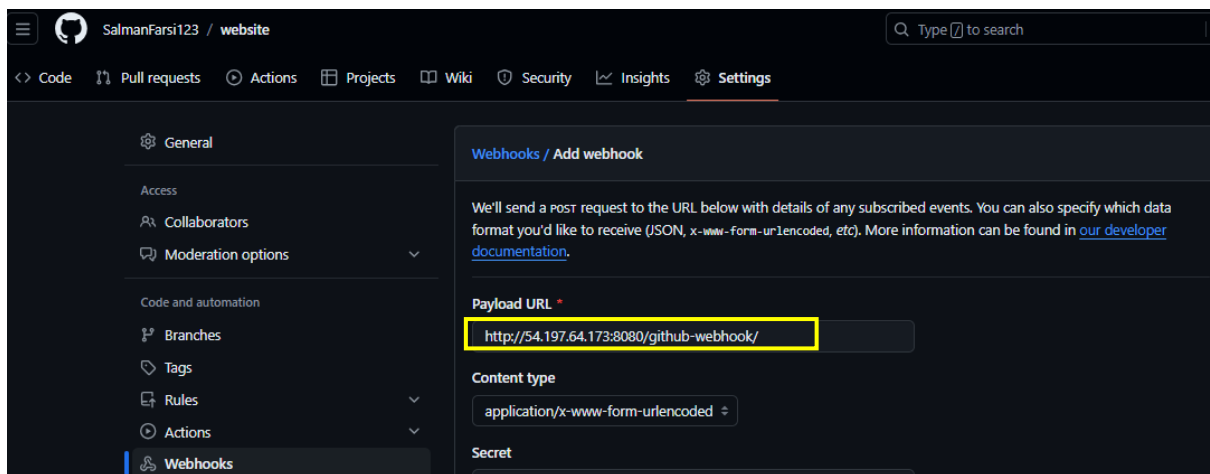
Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. [Learn more in our Webhooks Guide.](#)

- General
- Access
 - Collaborators
 - Moderation options ▾
- Code and automation
 - Branches
 - Tags
 - Rules ▾
 - Actions ▾
 - Webhooks**

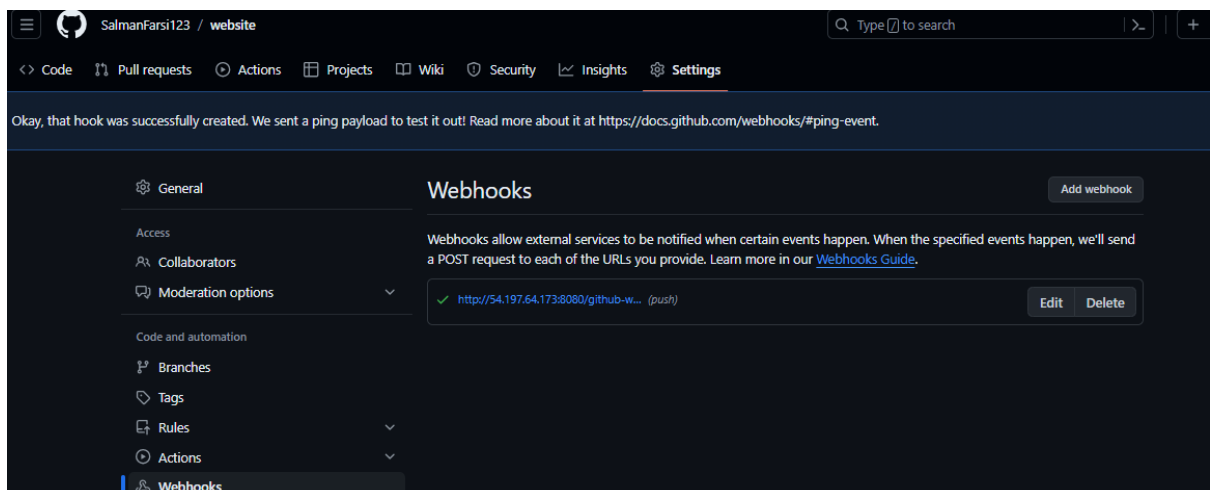
Copy the Public Ip of Jenkins-Master



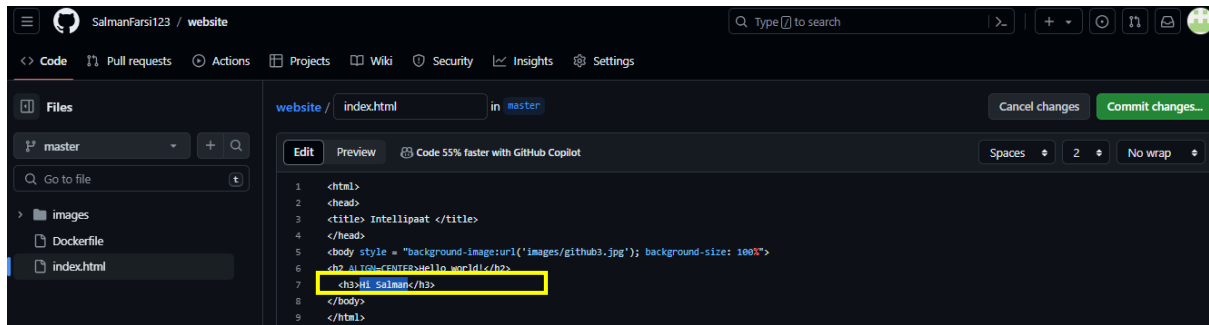
Paste it and type extension `github-webhook`



Refresh it

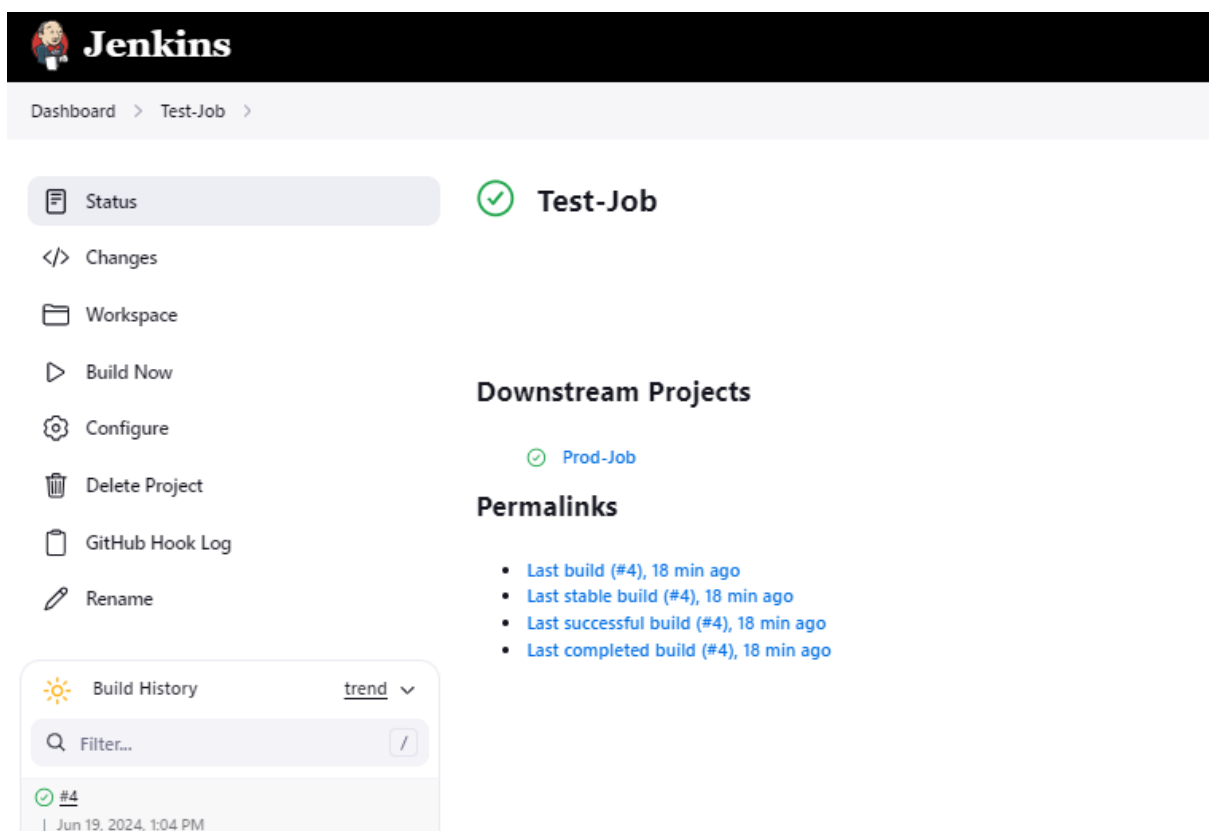


Now modifying index file on master branch and after Committing it




```
1 <html>
2 <head>
3 <title> Intellipaat </title>
4 </head>
5 <body style = "background-image:url('images/github3.jpg'); background-size: 100%">
6 <h2 align=CENTER>Hello world</h2>
7 <h3>Hi Salman</h3>
8 </body>
9 </html>
```


It will not Auto Triggering to Test-Job Because here test Job we are selected develop Branch not Master Branch





Jenkins


Dashboard > Test-Job >


 Status


 Changes


 Workspace


 Build Now


 Configure


 Delete Project

 GitHub Hook Log


 Rename

 Build History


 Filter...

 #4

Jun 19, 2024, 1:04 PM

 **Test-Job**

Downstream Projects

 [Prod-Job](#)

Permalinks

- [Last build \(#4\), 18 min ago](#)
- [Last stable build \(#4\), 18 min ago](#)
- [Last successful build \(#4\), 18 min ago](#)
- [Last completed build \(#4\), 18 min ago](#)

Its Triggered Prod-Job and ran successfully

Dashboard > Prod-Job >

</> Changes

Workspace

Build Now

Configure

Delete Project

GitHub Hook Log

Rename

Upstream Projects

Test-Job

Permalinks

- Last build (#3), 13 min ago
- Last stable build (#3), 13 min ago
- Last successful build (#3), 13 min ago
- Last completed build (#3), 13 min ago

Build History

trend

Filter...

#4
Jun 19, 2024, 1:22 PM

#3
Jun 19, 2024, 1:04 PM

Now the container replaced with new integrate data

Dashboard > Prod-Job > #4 > Console Output

Console Output

View as plain text

Edit Build Information

Delete build '#4'

Polling Log

Timings

Git Build Data

Previous Build

```
Started by GitHub push by SalmanFarsi123
Running as SYSTEM
Building remotely on Prod in workspace /home/ubuntu/jenkins/workspace/Prod-Job
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /home/ubuntu/jenkins/workspace/Prod-Job/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/SalmanFarsi123/website.git # timeout=10
Fetching upstream changes from https://github.com/SalmanFarsi123/website.git
> git --version # timeout=10
> git --version # 'git version 2.34.1'
> git fetch --tags --force --progress -- https://github.com/SalmanFarsi123/website.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 490b3bc10f08bf2c8b18613ac61b0b76cf018a6d (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 490b3bc10f08bf2c8b18613ac61b0b76cf018a6d # timeout=10
Commit message: "Update index.html"
> git rev-list --no-walk 23b019912dd20028eb6c0e55fda4bc6a90eaebd # timeout=10
[Prod-Job] $ /bin/sh -xe /tmp/jenkins12951544726623549064.sh
+ sudo docker rm -f c1
c1
+ sudo docker build . -t image1
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
Install the buildx component to build images with Buildkit:
https://docs.docker.com/go/buildx/

Sending build context to Docker daemon 305.2kB

Step 1/5 : FROM ubuntu
--> 35a08802559d
Step 2/5 : RUN apt-get update -y
```

```
Dashboard > Prod-Job > #4 > Console Output

+ sudo docker build . -t image1
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
Install the buildx component to build images with BuildKit:
https://docs.docker.com/go/buildx/

Sending build context to Docker daemon 305.2kB

Step 1/5 : FROM ubuntu
--> 35a88802559d
Step 2/5 : RUN apt-get update -y
--> Using cache
--> 472268a70078
Step 3/5 : RUN apt-get install apache2 -y
--> Using cache
--> fc276bda90b9
Step 4/5 : COPY . /var/www/html/
--> 8bd0e4ed4e07
Step 5/5 : ENTRYPOINT apachectl -D FOREGROUND
--> Running in 46c7821e411b
Removing intermediate container 46c7821e411b
--> 9c75d1c60f6e
Successfully built 9c75d1c60f6e
Successfully tagged image1:latest
+ sudo docker run -itd -p 82:80 --name=c1 image1
7373ef2a9623ee6863bc3ff964c9368e5f33b4bf2d55fccbdf250e295b5b05a6
Finished: SUCCESS
```

We can see the same container updated data and this is CI/CD



hi Salman



Docker images and docker ps

```
ubuntu@Jenkins-Master: ~
ubuntu@Jenkins-Slave: ~
ubuntu@Jenkins-Slave2: ~

ubuntu@Jenkins-Slave2:~$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
image1 latest 9c75d1c60f6e About a minute ago 222MB
<none> <none> 269b1f90c501 19 minutes ago 222MB
ubuntu latest 35a88802559d 12 days ago 78.1MB

ubuntu@Jenkins-Slave2:~$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
7373ef2a9623 image1 "/bin/sh -c 'apachec..." About a minute ago Up About a minute 0.0.0.0:82->80/tcp, :::82->80/tcp c1

ubuntu@Jenkins-Slave2:~$
```