

# Devops Capstone Project1

Create 3 instance as attached below

**Launch an instance** [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

**Name and tags** [Info](#)

Name  
sandeep-master [Add additional tags](#)

**Application and OS Images (Amazon Machine Image)** [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

**Recent** | **Quick Start**

Amazon Linux | macOS | **Ubuntu** | Windows | Red Hat | SUSE L

**Amazon Machine Image (AMI)**

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type Free tier eligible

ami-0a0e5d9c7acc336f1 (64-bit (x86)) / ami-070f589e4b4a3fccc (64-bit (Arm))  
Virtualization: hvm ENA enabled: true Root device type: ebs

**Summary**

Number of instances [Info](#)  
3

When launching more than 1 instance, consider EC2 Auto Scaling

**Software Image (AMI)**  
Canonical, Ubuntu, 22.04 LTS, ...[read more](#)  
ami-0a0e5d9c7acc336f1

**Virtual server type (instance type)**  
t2.micro

**Firewall (security group)**  
default

**Storage (volumes)**  
1 volume(s) - 8 GiB

**Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

[Cancel](#) [Launch instance](#) [Review commands](#)

**Instance type** [Info](#) [Get advice](#)

Instance type  
t2.micro Free tier eligible

Family: t2 1 vCPU 1 GiB Memory Current generation: true  
On-Demand Windows base pricing: 0.0162 USD per Hour  
On-Demand SUSE base pricing: 0.0116 USD per Hour  
On-Demand RHEL base pricing: 0.026 USD per Hour  
On-Demand Linux base pricing: 0.0116 USD per Hour

☒ All generations [Compare instance types](#)

**Additional costs apply for AMIs with pre-installed software**

**Key pair (login)** [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*  
ubuntu [Create new key pair](#)

**Network settings** [Info](#)

VPC - *required* [Info](#)  
vpc-0a832994d2ed7744d (project-vpc)  
10.0.0.0/16

Subnet [Info](#)  
subnet-0bd26c1914c4637f project-subnet-public1-us-east-1a  
VPC: vpc-0a832994d2ed7744d Owner: 767398084852  
Availability Zone: us-east-1a Zone type: Availability Zone  
IP addresses available: 4091 CIDR: 10.0.0.0/20 [Create new subnet](#)

Auto-assign public IP [Info](#)  
Enable

**Additional charges apply when outside of free tier allowance**

**Firewall (security groups)** [Info](#)  
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your

**Summary**

Number of instances [Info](#)  
3

When launching more than 1 instance, consider EC2 Auto Scaling

**Software Image (AMI)**  
Canonical, Ubuntu, 22.04 LTS, ...[read more](#)  
ami-0a0e5d9c7acc336f1

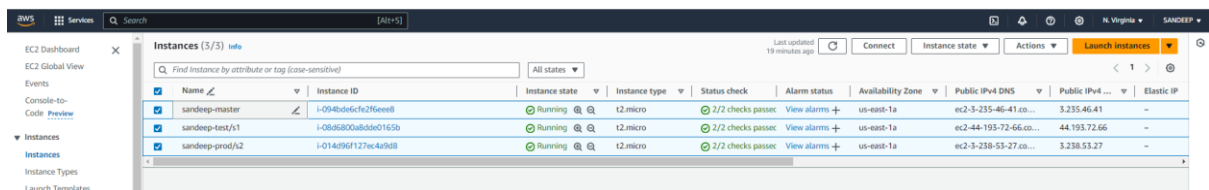
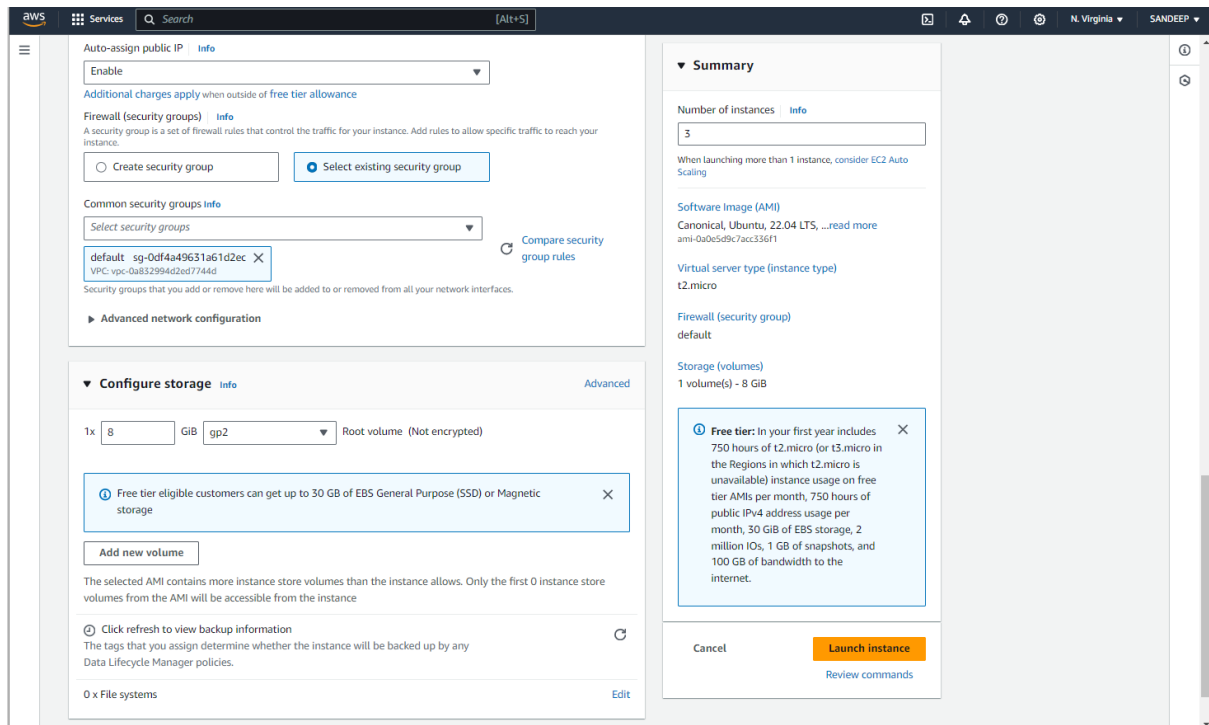
**Virtual server type (instance type)**  
t2.micro

**Firewall (security group)**  
default

**Storage (volumes)**  
1 volume(s) - 8 GiB

**Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

[Cancel](#) [Launch instance](#) [Review commands](#)



## 1- In the master machine install below commands and install Ansible

sudo apt update

sudo apt install software-properties-common

sudo add-apt-repository --yes --update ppa:ansible/ansible

sudo apt install ansible

Now Connect both machine through master

cd .ssh

ls

ssh.keygen

sudo cat id\_rsa.pub (it will reflect id\_rsa.pub code)

ssh-rsa

```
AAAAB3NzaC1yc2EAAAADAQABAAQGCirGTPFRSSmyKW/DcG04KWZhD6mSdBlmaT7YDN+CGqINuZnv/vP00uhvSGYFFEHPPEPU
FDz3nVoU30zoVZnKO4B2mhfMeaZPqAftBu5wMznNwxGabG1DeuW0EJ6kGphfPwvmoqnM5zFqahNbuKyOVcOo/kVc/IlkwHJfJDIg
BDSFT38bSEN9fknrdM9E5bsLk7JuaX5MASGAm/4SYPaKb8XbwEQJvs4A+gCjFYlcEEmlggbvn6CduDQr3j8/49X2qHHsVyU93ARb4s
FOZTzgLkAt3xC0txYU5OUGRIdaoAstrgrXr/5k0oygTVpvXLO6Ok+ijQEP2nRZxUV8ypsFQhHln2LC1wTkVJkMe7m29hXfvj/ILZCdSwSwZ
1k9dFgDozrYbe+kWw3HHtnog5bC3vdK+DS+ms7Y7lof8O/KLjscw4gCE8HgHy1Tcl1W8M0/5yWA4RI6BsIH7MNVBaaWXaLYic9mJ6P
ZzrUhmPAIKDg3tF5aGzb4V6Ojzogs= ubuntu@ip-10-0-14-135
```

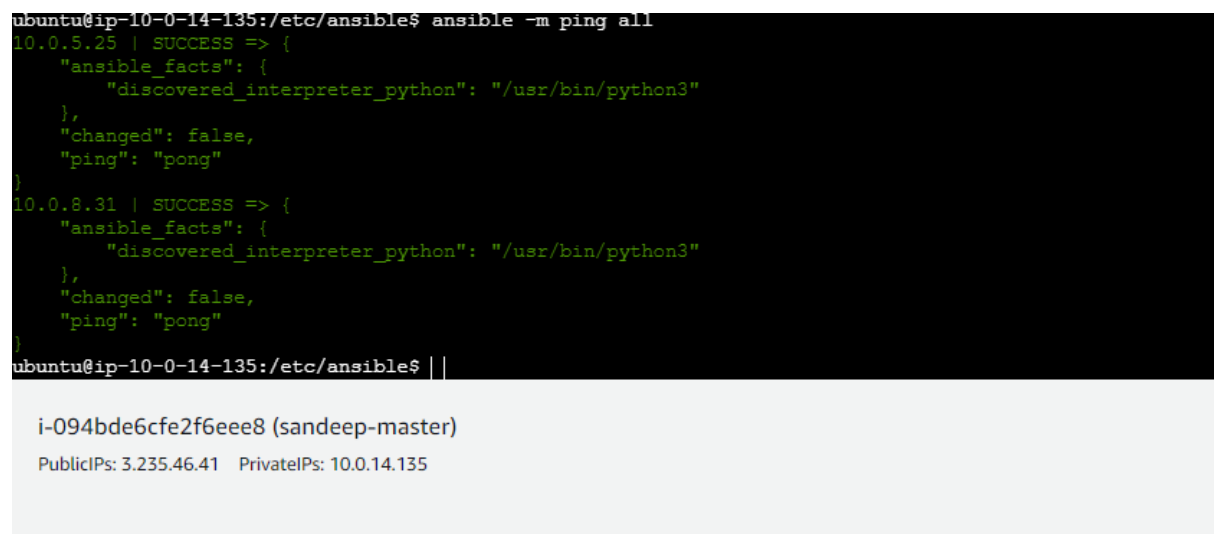
In master

```
cd .ssh
ls
sudo cat id_rsa.pub
cd
cd /etc/ansible
ls
sudo nano hosts
copy paste private ip of both prod and test here
ansible -m all
ansible -m ping all
press yes
```

Now Go to Sandeep-test/s1 and Sandeep -prod/s2  
and run below command

```
ls
sudo nano authorized_keys
copy paste the code here of id_rsa.pub
```

Now checked in master machine both machine connected

A terminal window with a black background and green text. The prompt is 'ubuntu@ip-10-0-14-135:/etc/ansible\$'. The command 'ansible -m ping all' is entered. The output shows two successful ping results for hosts 10.0.5.25 and 10.0.8.31, both returning 'pong'. Below the terminal output, a light gray box contains the text 'i-094bde6cfe2f6eee8 (sandeep-master)' and 'PublicIPs: 3.235.46.41 PrivateIPs: 10.0.14.135'.

```
ubuntu@ip-10-0-14-135:/etc/ansible$ ansible -m ping all
10.0.5.25 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
10.0.8.31 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
ubuntu@ip-10-0-14-135:/etc/ansible$ | |
```

i-094bde6cfe2f6eee8 (sandeep-master)  
PublicIPs: 3.235.46.41 PrivateIPs: 10.0.14.135

Now both machine connected to master

Install below command in

type

1-sudo nano cp1.yaml

# play.yaml

---

```
- name: master tasks
  hosts: localhost
  become: true
  tasks:
    - name: execute tasks for master
      script: master.sh
- name: slave tasks
  hosts: all
  become: true
  tasks:
    - name: execute tasks for slave
      script: slave.sh
```

2-sudo nano master.sh

(install jenkins on linux)

```
sudo apt update
sudo apt install openjdk-17-jre -y
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
  https://pkg.jenkins.io/debian/jenkins.io-2023.key
echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
  https://pkg.jenkins.io/debian binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null"
sudo apt-get update
sudo apt-get install jenkins -y
```

3- sudo nano slave.sh

```
sudo nano slave.sh
sudo apt update
sudo apt install openjdk-17-jre -y
sudo apt install docker.io -y
```

run below commands to check

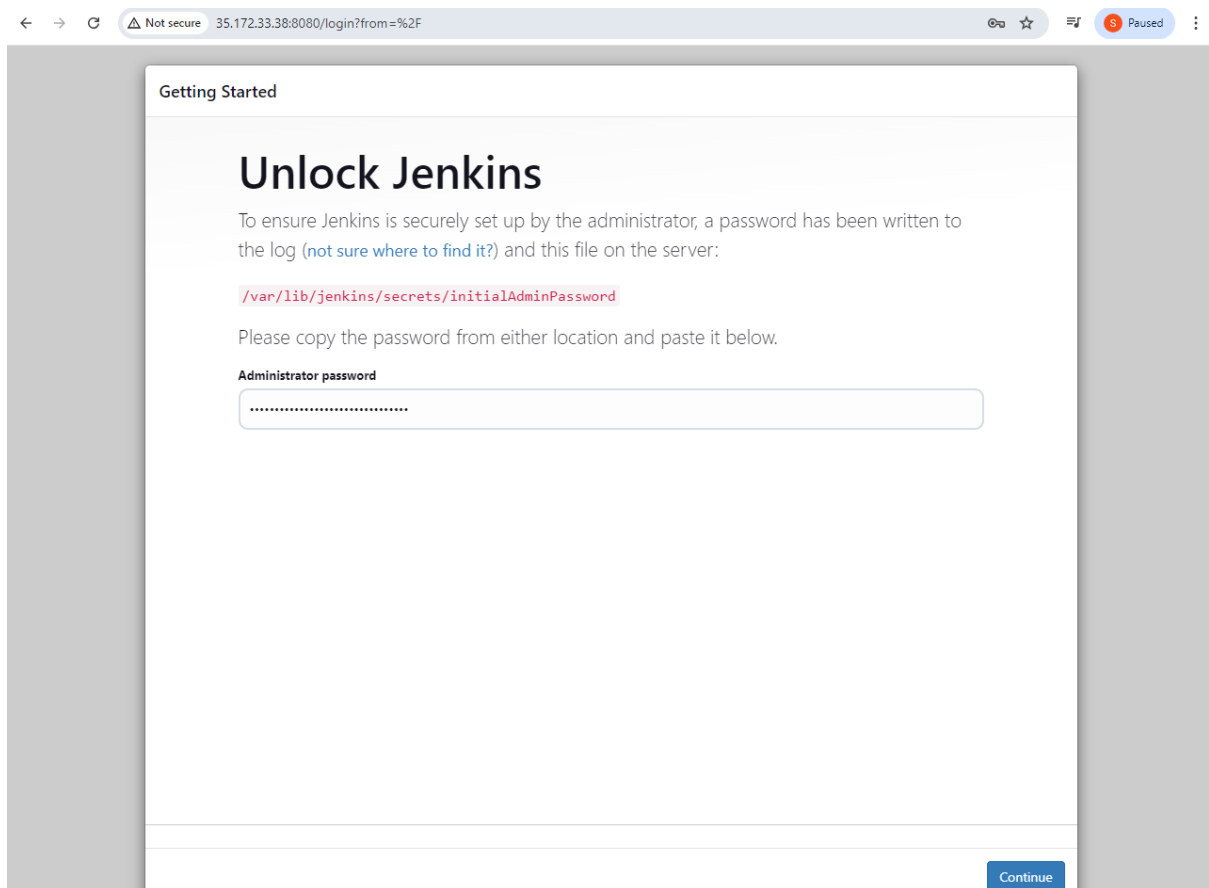
```
ansible-playbook cp1.yaml --syntax-check
ansible-playbook cp1.yaml --check
ansible-playbook cp1.yaml
```

GO to Sandeep test/s1 and in master check below s/w installed

```
java --version
and in master
which Jenkins
```

configuration part done

Take public ip of master and paste on browser with 8080



Sudo cat /var/lib/Jenkins/secrets/initialAdminPassword

7157022984f84cefb4d26d4ede52adfc

copy unlock jenkins link password and paste on master

sudo cat /var/lib/Jenkins/secrets/initialAdminPassword

Copy token and paste on browser to login Jenkins

1-install suggested plugins

2-make user and password

3- start using Jenkins

← → ↻ ⚠ Not secure 35.172.33.38:8080 🔍 ☆ 📄

### Getting Started

## Create First Admin User

Username

Password

Confirm password

Full name

E-mail address

Jenkins 2.474

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

⚠ Not secure 35.172.33.38:8080 🔍 ☆ 📄 S

### Getting Started

## Instance Configuration

Jenkins URL:

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 2.474

Not now [Save and Finish](#)

## Plugins

- Updates
- Available plugins
- Installed plugins
- Advanced settings
- Download progress

ssh-agent		
Install	Name ↓	Released
<input checked="" type="checkbox"/>	<a href="#">SSH Agent</a> 376.v8933585c69d3 This plugin allows you to provide SSH credentials to builds via a ssh-agent in Jenkins.	1 mo 4 days ago

Name ?

Test

Description ?

Test Node

Plain text [Preview](#)

Number of executors ?

1

Remote root directory ?

/home/ubuntu/jenkins/

Labels ?

Usage ?

Use this node as much as possible

Launch method ?

Dashboard > Manage Jenkins > Nodes

Launch method ?

Launch agents via SSH

Host ?

10.0.13.38

Credentials ?

ubuntu

+ Add

Host Key Verification Strategy ?

Non verifying Verification Strategy

Advanced

Availability ?

Keep this agent online as much as possible

Node Properties

☐ Disable deferred wipeout on this node ?

☐ Disk Space Monitoring Thresholds

☐ Environment variables

☐ Tool Locations

Save

Created two node as test and prod

Dashboard > 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	4.41 GiB	0 B	4.41 GiB	0ms
	Prod	Linux (amd64)	In sync	5.06 GiB	0 B	5.06 GiB	38ms
	test	Linux (amd64)	In sync	4.97 GiB	0 B	4.97 GiB	21ms
Data obtained			0.24 sec	0.24 sec	0.23 sec	0.23 sec	0.23 sec

Build Queue: No builds in the queue.

Build Executor Status:

- Built-In Node: 0/4
- Prod: 0/1
- test: 0/1

Icon: S M L

Legend

Jenkins id and password-

username - Sandeep\_24

password- \*ongc123

http://3.235.46.41 :8080/



Go to GitHub

website

fork- create fork

will be the owner of website

create branch -develop

#Add file create new file

Dockerfile

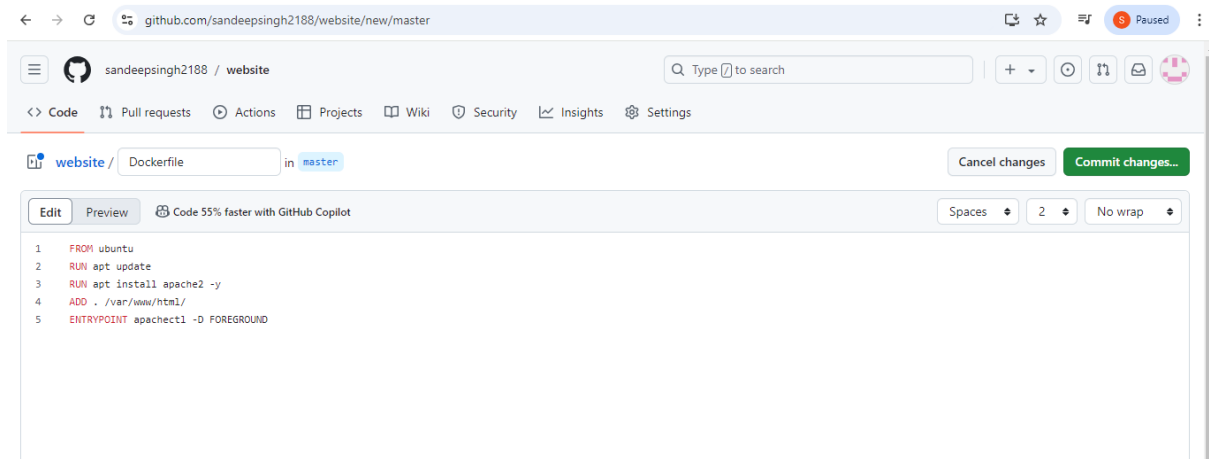
FROM ubuntu

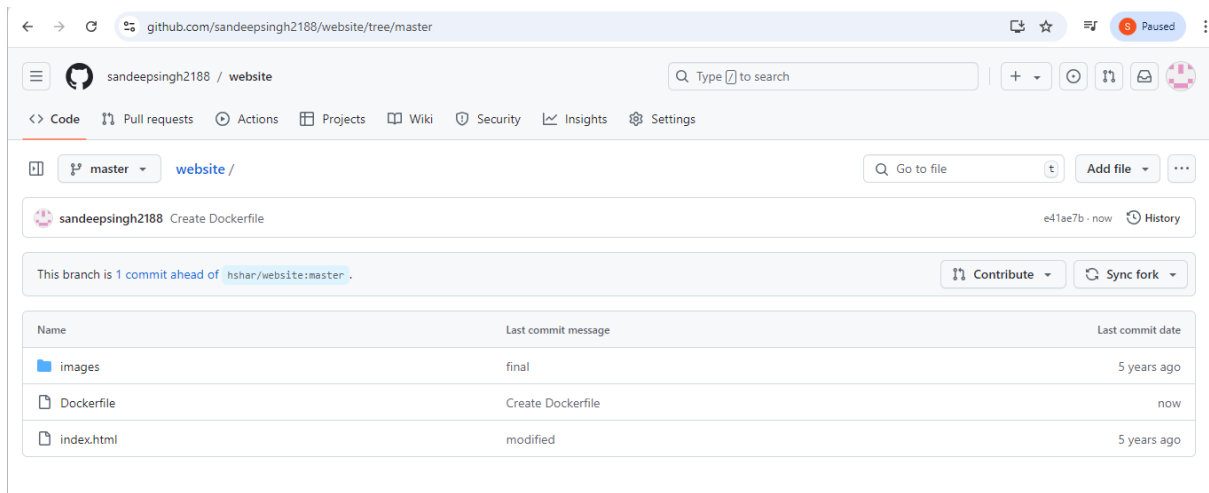
RUN apt update

RUN apt install apache2 -y

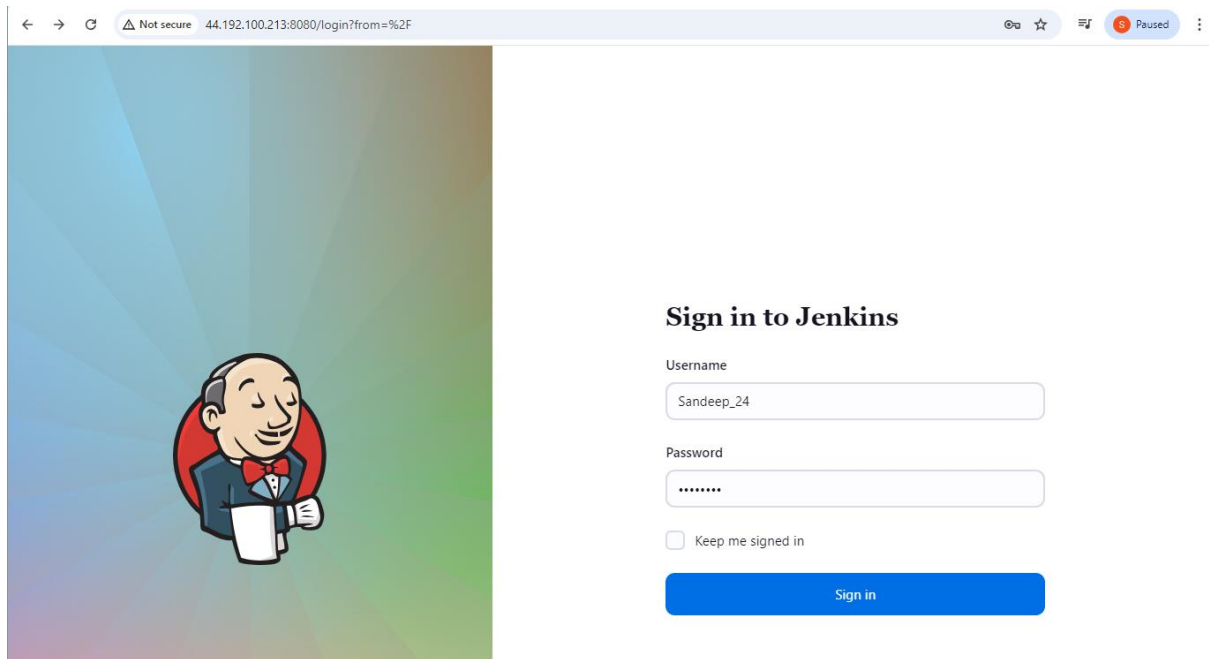
ADD . /var/www/html/

ENTRYPOINT apachectl -D FOREGROUND






## Login to Jenkins



## Create a item with freestyle project

← → ↻ ⚠ Not secure 44.192.100.213:8080/view/all/newJob ☆ ⚙ Paused ⋮

 **Jenkins**

🔍 Search (CTRL+K) ⓘ ⚠ 1 👤 sandeep kumar ▾ 🚪 log out


Dashboard > All > New Item


## New Item


Enter an item name


Job1


Select an item type


 **Freestyle project**  
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

 **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.

OK


← → ↻ ⚠ Not secure 44.192.100.213:8080/job/Job1/configure ☆ ⚙ Paused ⋮


 **Jenkins**


🔍 Search (CTRL+K) ⓘ ⚠ 1 👤 sandeep kumar ▾ 🚪 log out


Dashboard > Job1 > Configuration


## Configure


 General

 Source Code Management


 Build Triggers

 Build Environment

 Build Steps

 Post-build Actions

## General

Enabled 

Description

Plain text [Preview](#)

☐ Discard old builds ⓘ

☐ GitHub project

☐ This project is parameterized ⓘ

☐ Throttle builds ⓘ

☐ Execute concurrent builds if necessary ⓘ

☒ Restrict where this project can be run ⓘ

Label Expression ⓘ

test

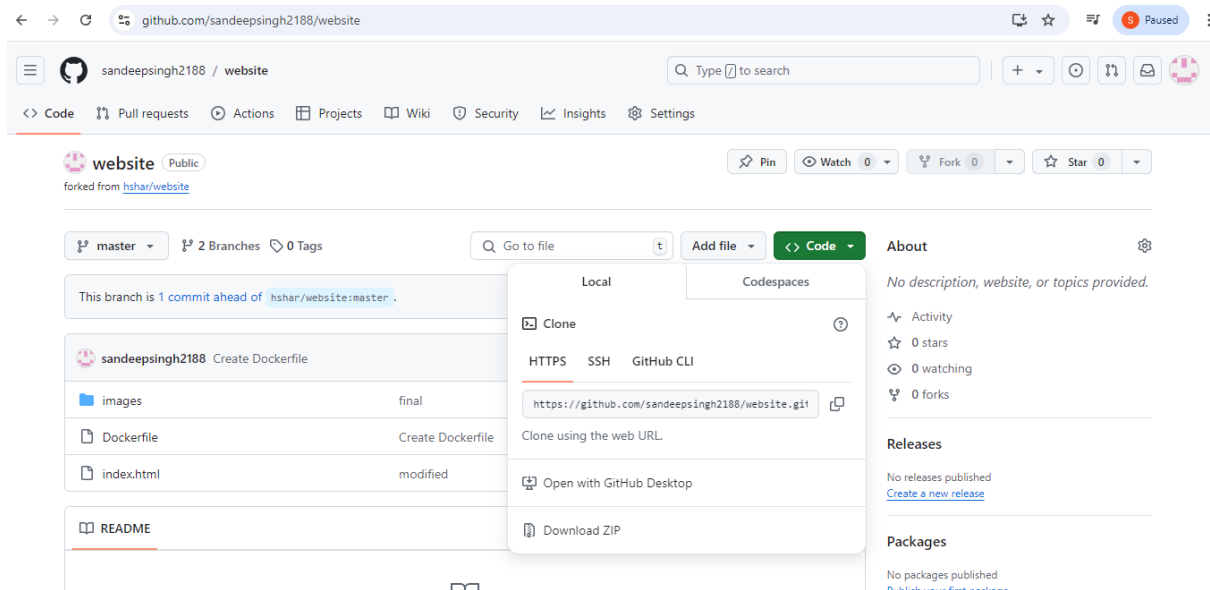
Label **test** matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.

Advanced ▾

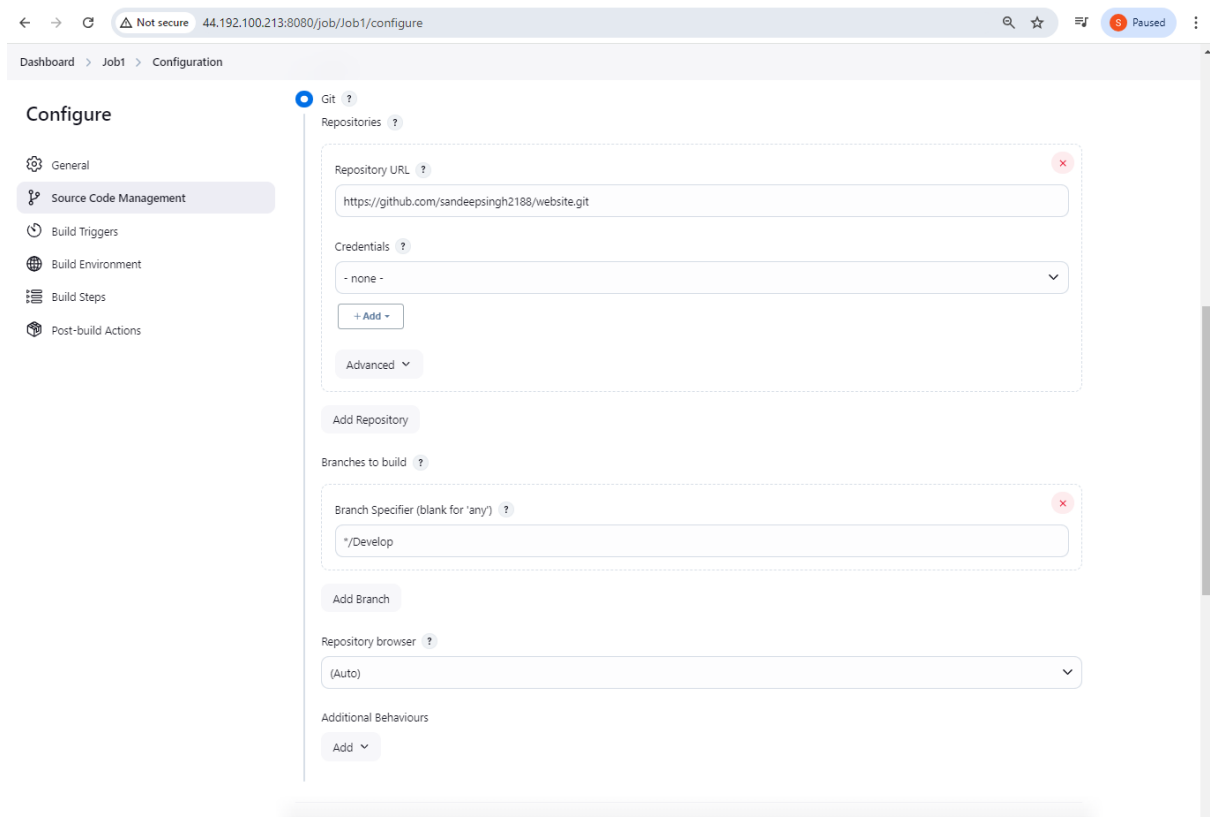
### Source Code Management

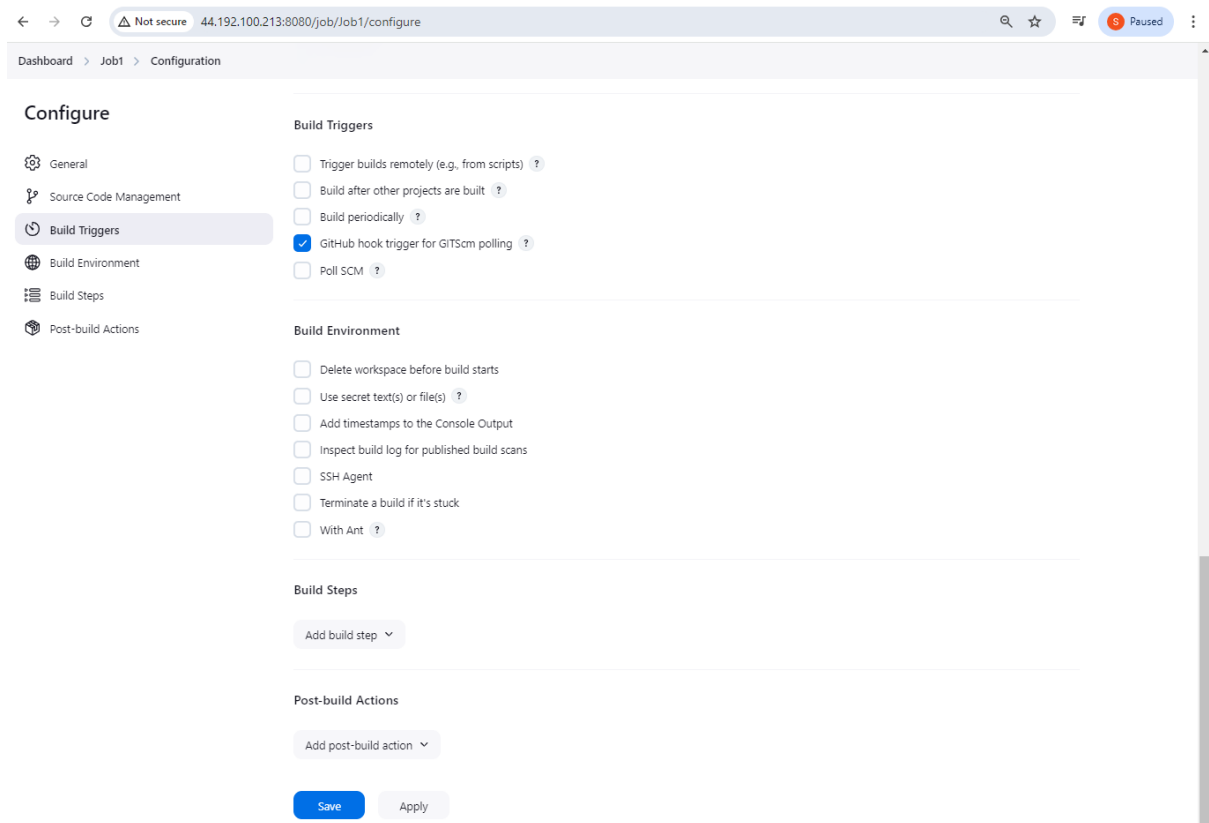
☐ None

☒ Git ⓘ



Copied github link to Job1





Created the item

Now check on Test machine

```
Last login: Tue Sep 3 12:14:18 2024 from 10.0.14.135
ubuntu@ip-10-0-5-25:~$ cd jenkins
ubuntu@ip-10-0-5-25:~/jenkins$ cd workspace
ubuntu@ip-10-0-5-25:~/jenkins/workspace$ cd job1
-bash: cd: job1: No such file or directory
ubuntu@ip-10-0-5-25:~/jenkins/workspace$ cd Job1
ubuntu@ip-10-0-5-25:~/jenkins/workspace/Job1$ ls
Dockerfile  images  index.html
ubuntu@ip-10-0-5-25:~/jenkins/workspace/Job1$ pwd
/home/ubuntu/jenkins/workspace/Job1
ubuntu@ip-10-0-5-25:~/jenkins/workspace/Job1$ ||
```

i-08d6800a8dde0165b (sandeep-test/s1)  
PublicIPs: 98.80.174.48 PrivateIPs: 10.0.5.25

Come to build Steps in Jenkins job1

and inside Execute shell paste the present directory of Test machine

`sudo docker build /home/ubuntu/jenkins/workspace/Job1 -t Job1`

Execute shell ?

Command

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

```
sudo docker build /home/ubuntu/jenkins/workspace/Job1 -t job1
sudo docker run -itd -p 88:80 job1
```

Advanced ▾

Add build step ▾

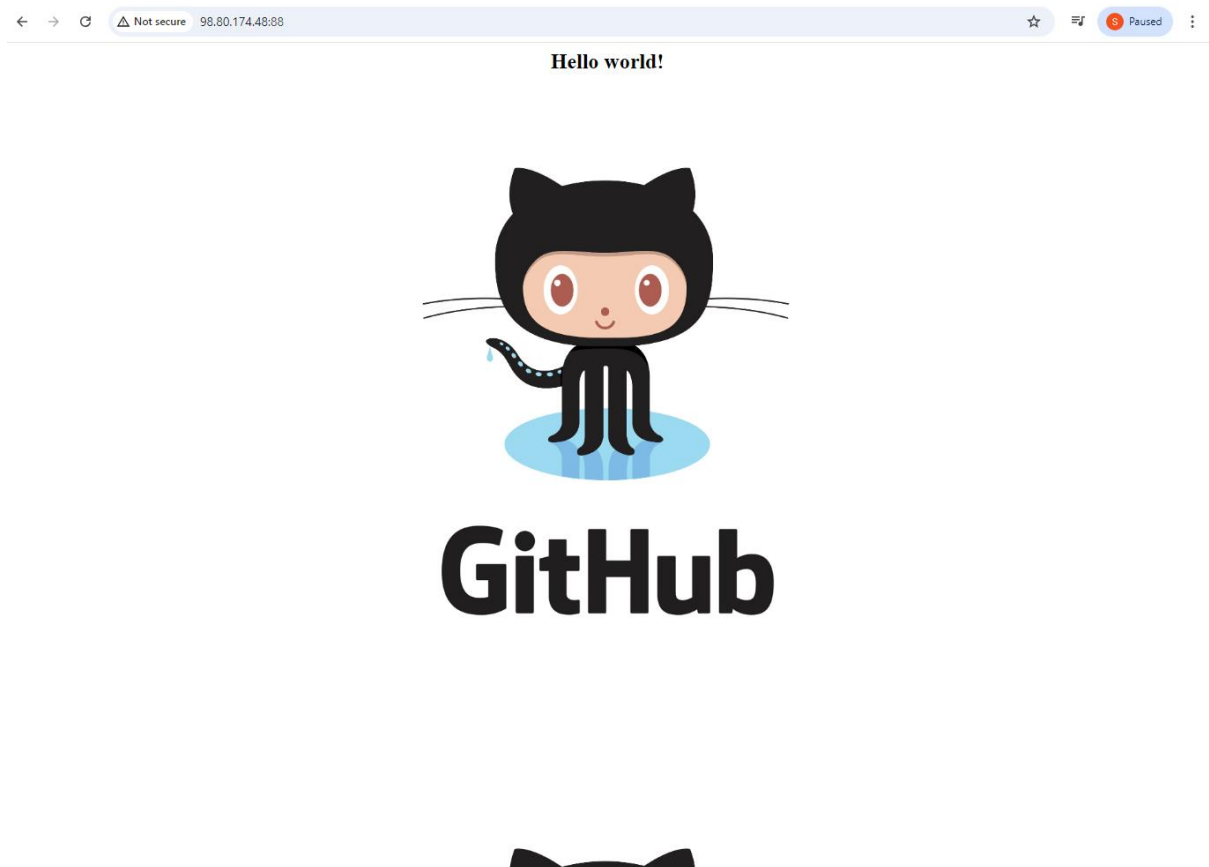
Post-build Actions

Add post-build action ▾

Save Apply

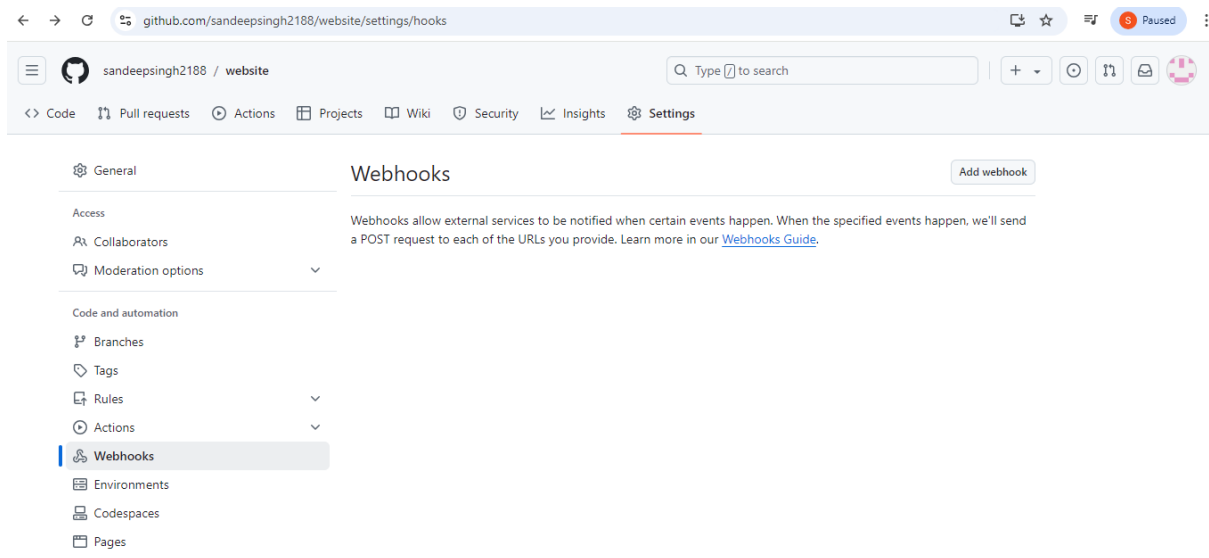
Copy paste public ip of test on browser as below

Output attached



## First Job Done

## Add Webhook on github

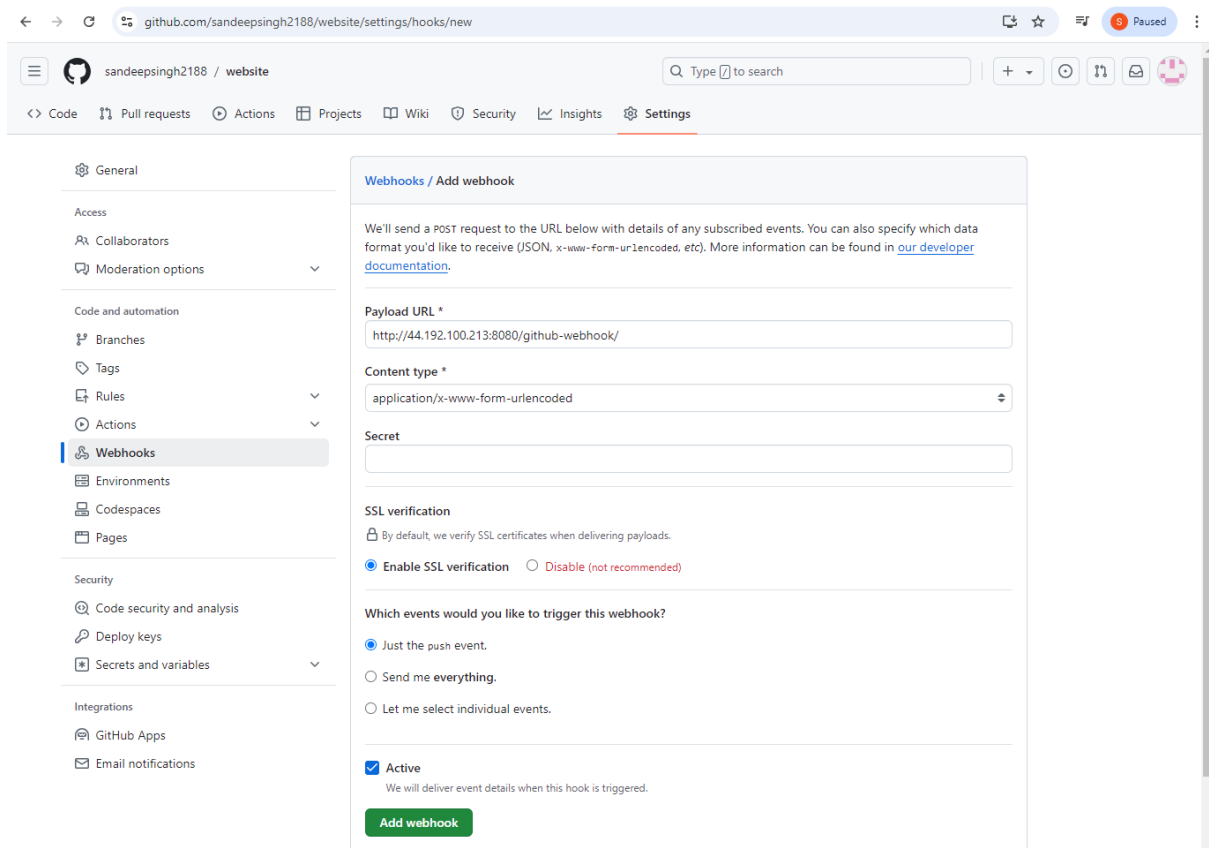


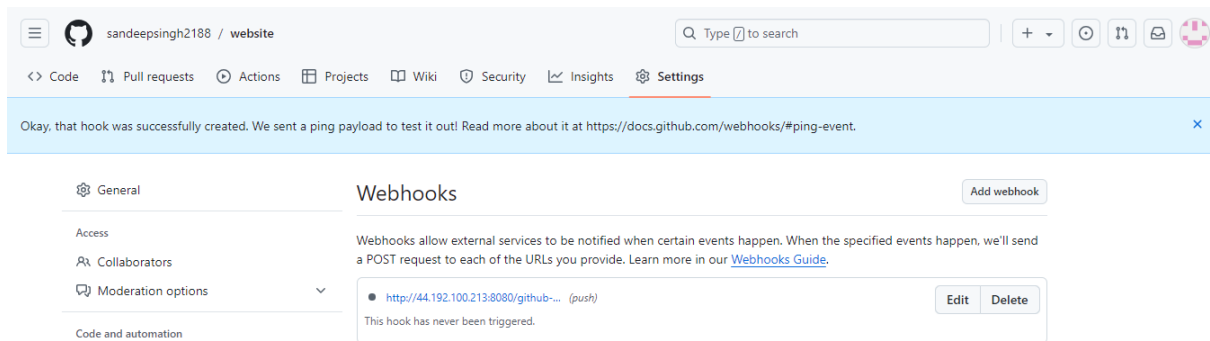
## Now on GitHub create Webhook

## Payload URL

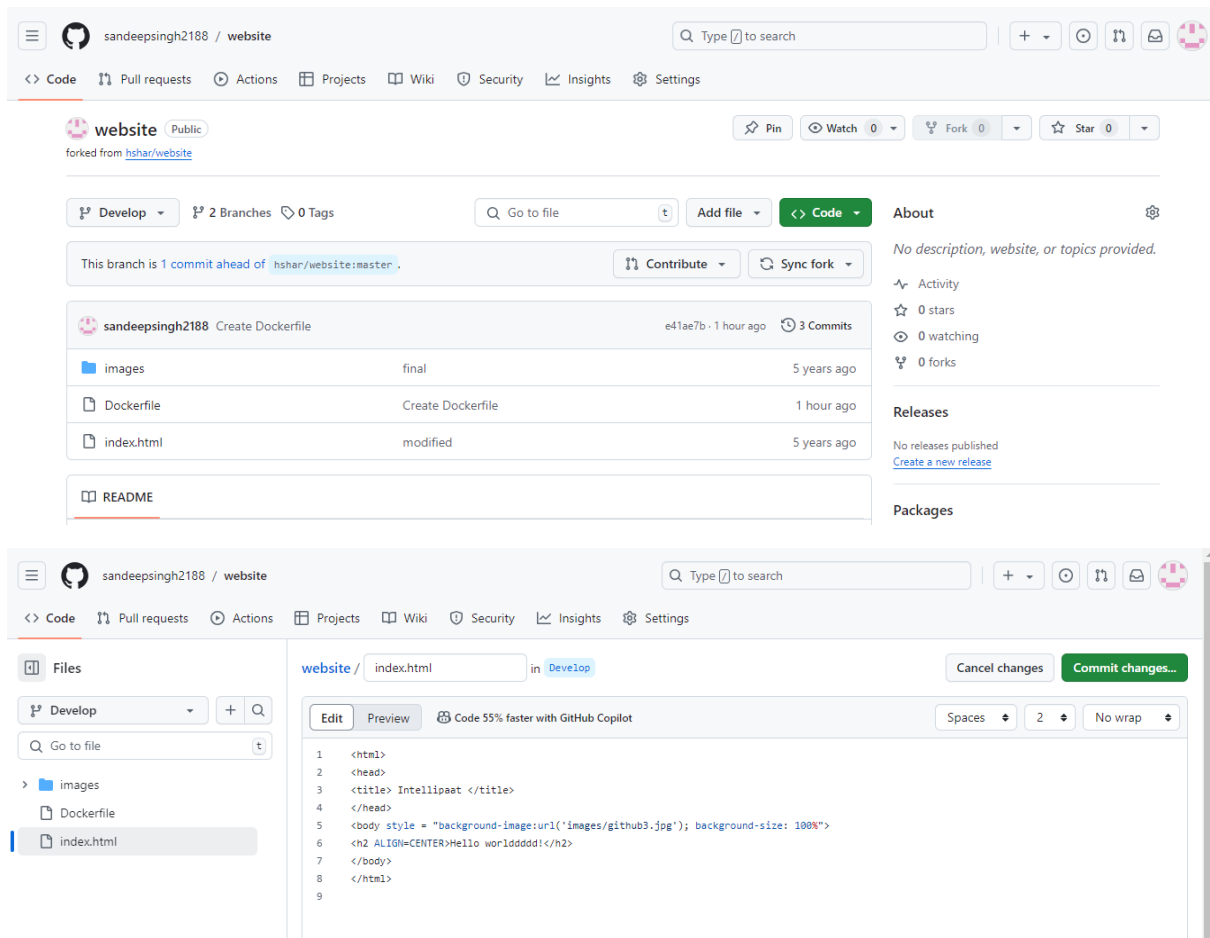
## Paste Public ip of master

<https://44.192.100.213:8080/github-webhook/>





Now go to Develop branch



Commit changes

Output attached below



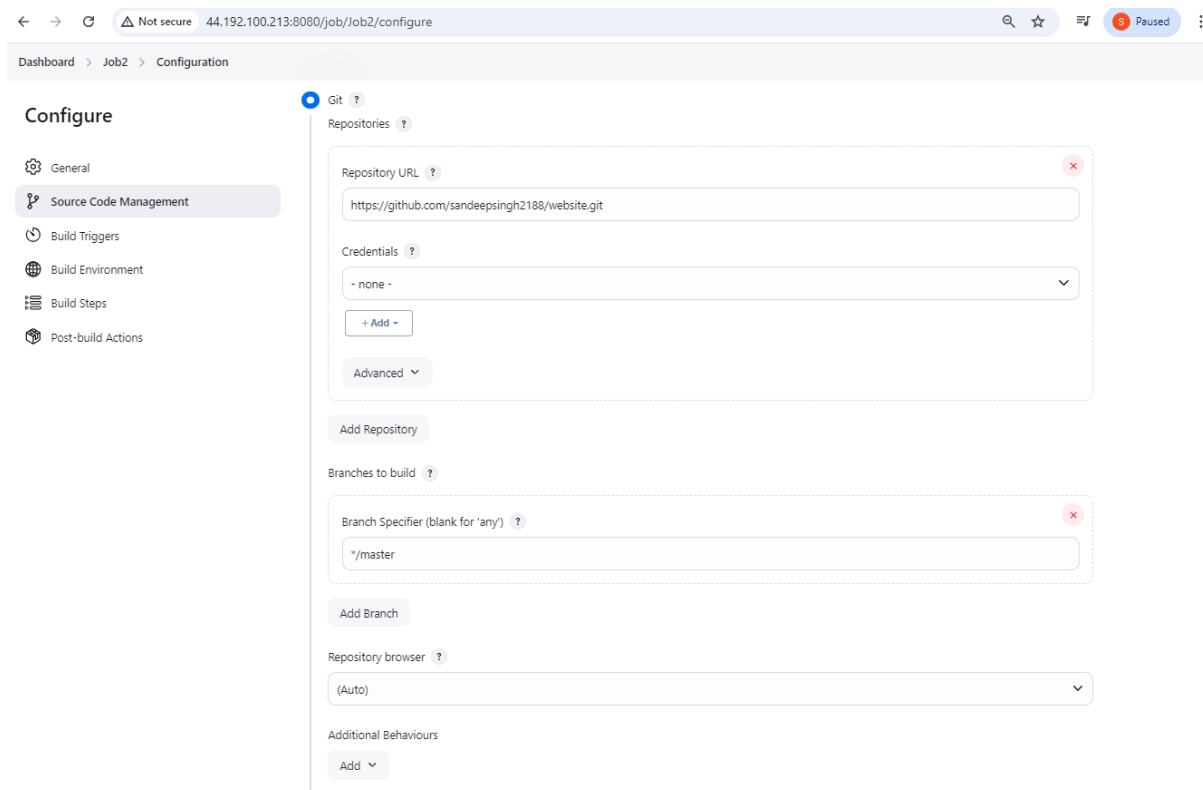
Hello worldddd!



# GitHub

— —

Now create Job 2



← → ↻ ⚠ Not secure 44.192.100.213:8080/job/Job2/configure 🔍 ☆ 📄 ⏸ Paused ⋮

Dashboard > Job2 > Configuration

### Configure

⚙ General

🔑 Source Code Management

**🕒 Build Triggers**

🌐 Build Environment

⚙ Build Steps

🔧 Post-build Actions

**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) ?

☐ Add timestamps to the Console Output

☐ Inspect build log for published build scans

☐ SSH Agent

☐ Terminate a build if it's stuck

☐ With Ant ?

**Build Steps**

Add build step ▾

**Post-build Actions**

Add post-build action ▾

Save

Apply

In Test machine-

```
ubuntu@ip-10-0-5-25:~$ cd jenkins
ubuntu@ip-10-0-5-25:~/jenkins$ cd workspace
ubuntu@ip-10-0-5-25:~/jenkins/workspace$ ls
Job1  Job2
ubuntu@ip-10-0-5-25:~/jenkins/workspace$ cd Job2
ubuntu@ip-10-0-5-25:~/jenkins/workspace/Job2$ ls
Dockerfile  images  index.html
ubuntu@ip-10-0-5-25:~/jenkins/workspace/Job2$ | |
```

i-08d6800a8dde0165b (sandeep-test/s1)

PublicIPs: 98.80.174.48 PrivateIPs: 10.0.5.25

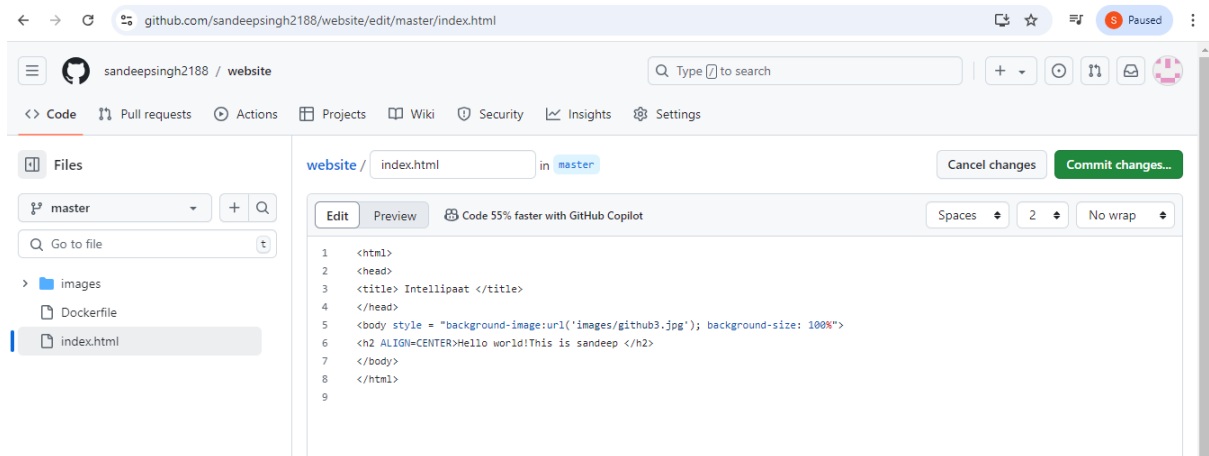
☰ Execute shell ?

✕

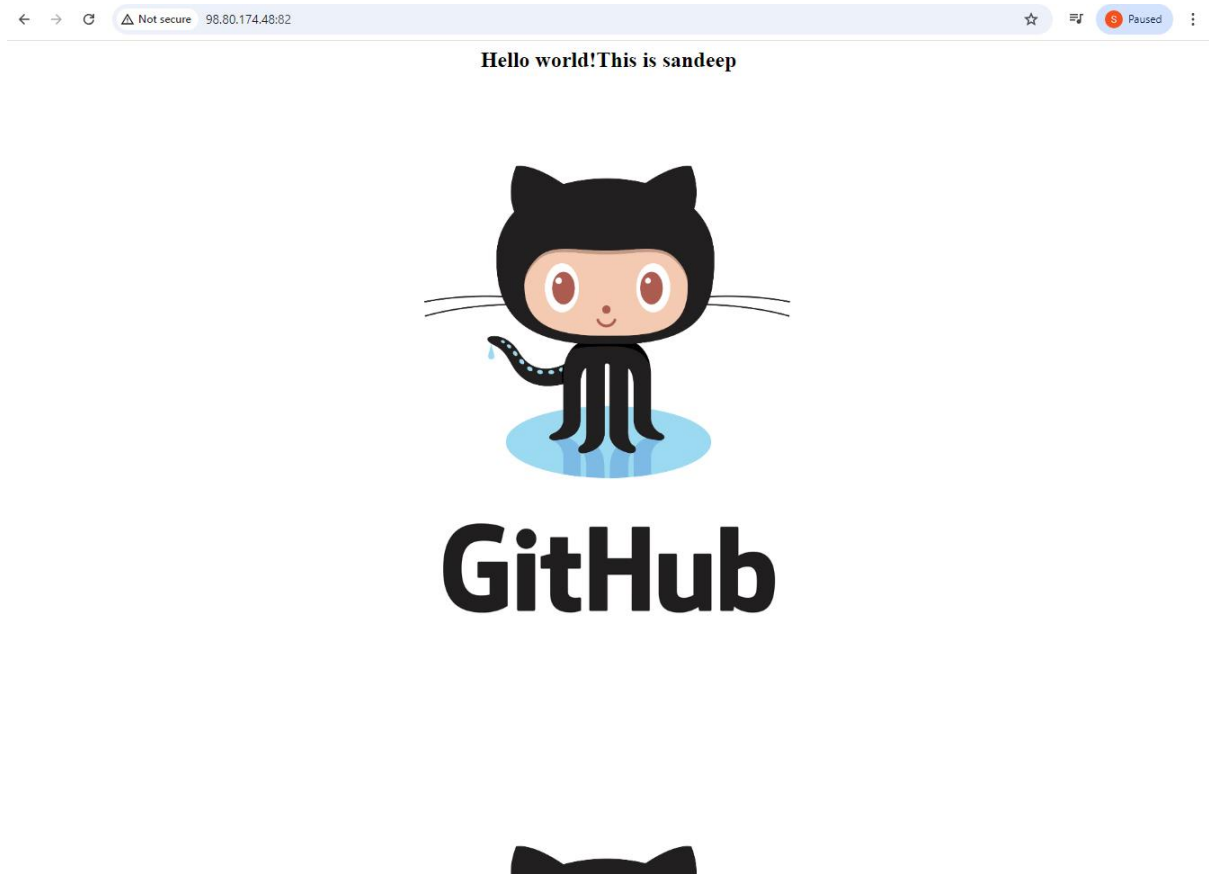
Command

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

```
sudo docker rm -f $(sudo docker ps -aq)
sudo docker build /home/ubuntu/jenkins/workspace/Job2 -t job2
sudo docker run -itd -p 82:80 job2
```



Job 2 also created and output attached below



### Create Job3

←

→

🔄


⚠️ Not secure 44.192.100.213:8080/job/Job3/configure

🔍

☆

☰

⏸ Paused

 Jenkins

🔍 Search (CTRL+K) ?

🚨 1

👤 sandeep kumar

🚪 log out

Dashboard > Job3 > Configuration

Configure

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

General

Description

Plain text [Preview](#)

☐ Discard old builds ?

☐ GitHub project

☐ This project is parameterized ?

☐ Throttle builds ?

☐ Execute concurrent builds if necessary ?

☐ Restrict where this project can be run ?

Advanced ▾

Source Code Management

☐ None

☒ Git ?

Repositories ?

Repository URL ?

https://github.com/sandeepsingh2188/website.git

← → ↻ Not secure 44.192.100.213:8080/job/Job3/configure 🔍 ☆ 📄 ⏸ Paused

Dashboard > Job3 > Configuration

### Configure

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

Advanced ▾

Add Repository

Branches to build ?

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 ?

Build Environment

- ☐ Delete workspace before build starts

Save Apply

← → ↻ github.com/sandeepsingh2188/website/edit/master/index.html 🔍 ☆ 📄 ⏸ Paused

sanadeepsingh2188 / website 🔍 Type to search + ▾ ⌂ 🔍 📧 🔄

<> Code 🔍 Pull requests ⌚ Actions 📁 Projects 📖 Wiki 🛡 Security 📈 Insights ⚙ Settings

Files

master ▾ + 🔍

Go to file 🔍

images

Dockerfile

index.html

website / index.html in master

Cancel changes Commit changes...

Edit Preview 🧠 Code 55% faster with GitHub Copilot Spaces 2 No wrap

```
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!This is Job3 </h2>
7 </body>
8 </html>
9
```

Job 3 also created and output attached below

Hello world! This is Job3



# GitHub



Project completed by

Sandeep kumar

9453743921