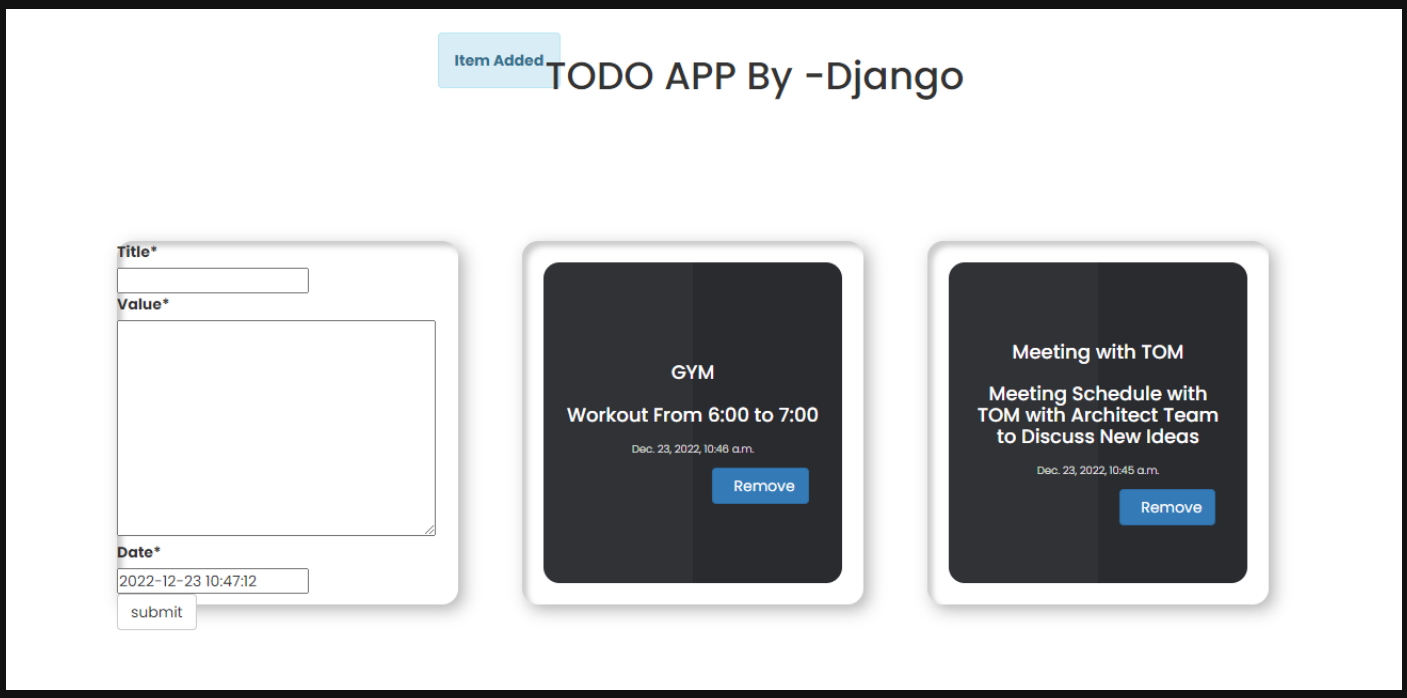
Project: Deploy Docker Provision Django Application on AWS with Default server & Include CI/CD with Jenkins.

Pre-requisites:

1. AWS Account
2. GitHub Account
3. Basic Knowledge About Docker & Jenkins

Steps:

i) Clone the Django TODO App



ii) Launch an EC2 Ubuntu Instance with access of HTTP & HTTPS
iii) Install Jenkins on Ubuntu Machine
iv) Setup Jenkins & connect with GIT
v) Finally Enable the Jenkins Script for CI/CD of Docker Provision Django Application

Step 1:

Clone the Django TODO app by running Below Command & push that Code to Your Personal Repository

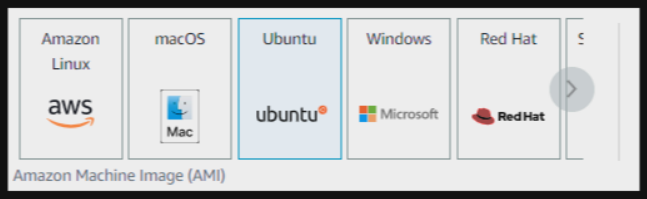
```
git clone https://github.com/tushargangurde2029/django-todo.git
```

Repository to use

Note:- You need to push this application to your Personal Repository so you can add ssh keys & add Webhooks to implement CI/CD Integration

Step 2:

Go to your AWS Account & Launch an Ubuntu EC2 Instance
Open EC2 -> Instances -> Launch an EC2 Instance
Select Ubuntu Image



Provide Following Details
Name = SampleDjangoApp
Instance Type = t2.micro
Allow HTTP & HTTPS Traffic From Internet

Once Done Launch Your EC2 Instance

Note: Please create a new key pair or use existing one to login

Step 3:

Connect to your EC2 Instance to Install Jenkins

RUN Below Commands One by One
Update Your System

```
sudo apt update
```

Install JAVA

```
sudo apt install openjdk-11-jre
```

Check JAVA Version by running below command

```
java -version
```

Now Install Jenkins

```
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
```

```
curl -fsSL https://pkg.jenkins.io/debian/jenkins.io.key | sudo tee \   /usr/share/keyrings/jenkins-
keyring.asc > /dev/null
```

```
sudo apt-get update
```

```
sudo apt-get install jenkins
```

Once Done Enable & start the Jenkins Service

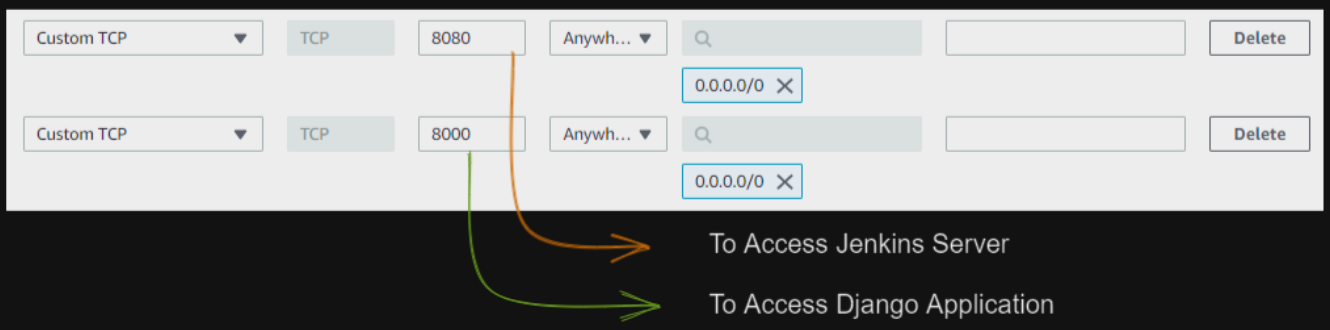
```
sudo systemctl enable jenkins
```

```
sudo systemctl start jenkins
```

Check Jenkins is successfully started by below command

```
sudo systemctl status jenkins
```

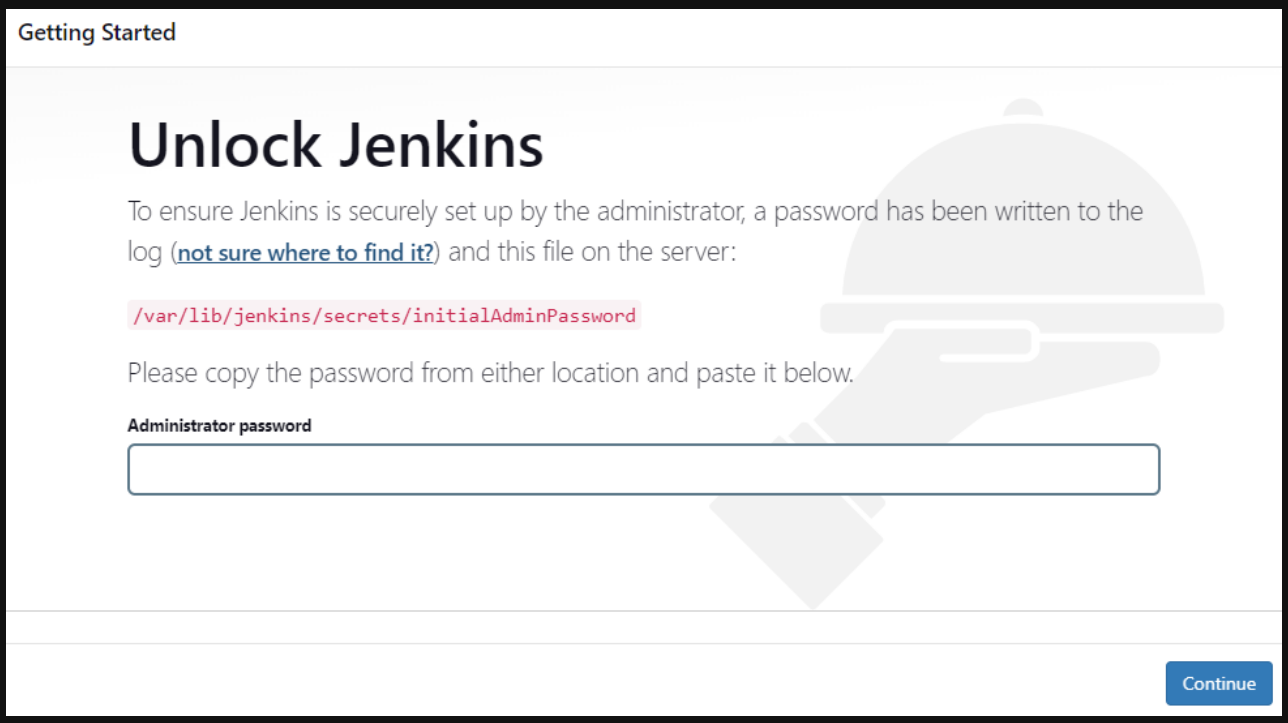
Now GO-TO Security Group of your EC2 Instance & Provide below Inbound Rules & Save Changes



Now Open your Jenkins Server by Below address

http://[public-ip]:8080/

You can see Below Screen



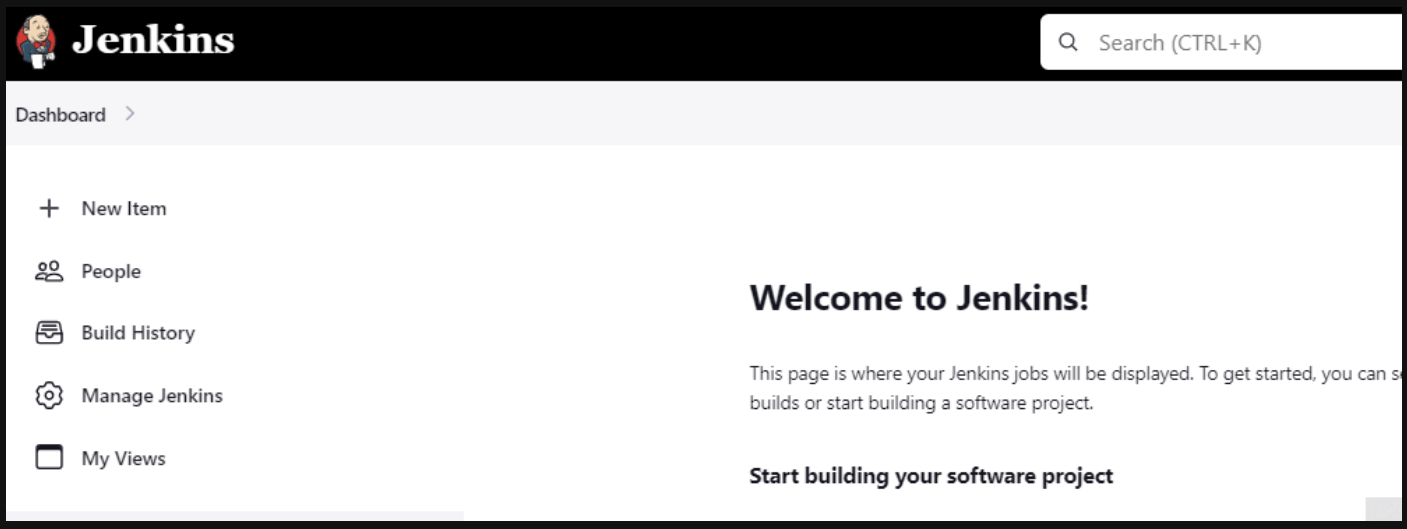
Step 4:

Now locate your Jenkins administrator password by command

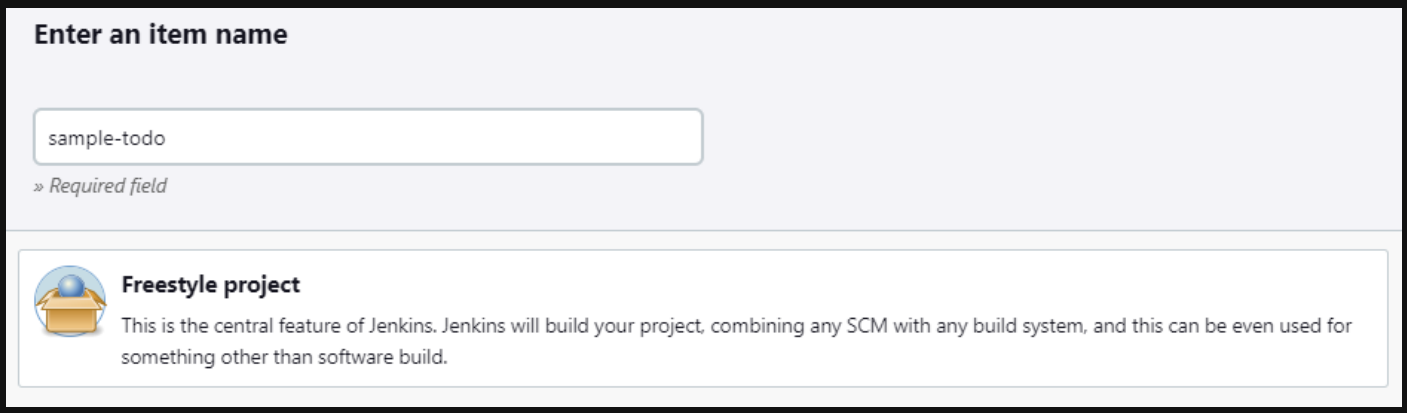
```
ubuntu@ip-172-31-34-232:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
e88fedcbc52a41dfb93f756d5c99d571
```

Enter that Password & select Install Suggested Plugins
Once Done Provide the Necessary Details & click on Save & Continue
Check the Jenkins URL & click on Save & Finish

Now click on Start Using Jenkins You can see below screen



Provide Item Name, we are using freestyle pipeline so choose freestyle project



Once Done click on Save

Before configuring lets connect Jenkins with git using SSH

GO to your EC2 Instance & run below command

```
ubuntu@ip-172-31-34-232:~$ ssh-keygen
```

It will generate public & private key provide public key for GitHub & provide private key for Jenkins
Access the keys by changing the directory to

```
ubuntu@ip-172-31-34-232:~$ cd .ssh
ubuntu@ip-172-31-34-232:~/.ssh$ ls
authorized_keys  id_rsa  id_rsa.pub
```

Now GO to your GitHub & provide the public key as

```
ubuntu@ip-172-31-34-232:~/.ssh$ cat id_rsa.pub
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQDC7t+X5gu9NK/hKaFV0UXaa7eHHZSsXR4uSEpkeP6vjt+MnDDhXnc67BxtT08/2tQFf0paPjm
7W+bMgxUJJMbpTfFQfjMEgGXHJ8sjvkZQRtL8i1ScXCLWNpV/LyCfzxqEg/zoOplVttbZi7cTCJjNLi6snW2xq5IbGBikCu0e4kbvFC
aXJ4mCaHB4NLWkw35qz+wmWfaDPLlyyYXYijVfxZ7LVOIVm1xGelFo42wmvaImiY6/oJP1didiVsrDOo1CNt5AtzjMQ0RVQkKh9pRab
HF2E+yW7ZS0jERXSOtiiFw1FC/MyVpLzGZwDoB8mR65VRxrKUEIzfp6JBZJmd/+EjqGa28kSfeKld3NKeHL7iEUmcAexOC3Cg3h4nUs
NVYyuOUsYgtqa0arRZdqmlD8gMJ2IwFL+R2z5vTQFzZOTMJJCUxxZAn9Ga4xvTYFgSXLqtO5iu0jSH7Gr/MmGlbwoZsFSBkGLgH2Pt
4jDJCoTGjgUUIGp4cEa0w+XY8V2k= ubuntu@ip-172-31-34-232
```

SSH keys / Add new

Title

Sample

Key type

Authentication Key

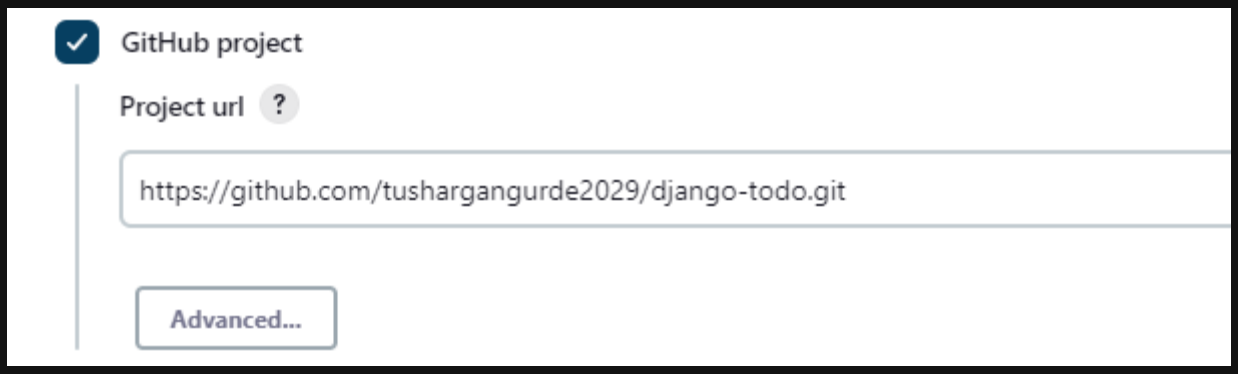
Key

ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQDC7t+X5gu9NK/hKaFV0UXaa7eHHZSsXR4uSEpkeP6vjt+MnDDhXnc67BxtT08/
2tQFf0paPjm7W+bMgxUJJMbpTfFQfjMEgGXHJ8sjvkZQRtL8i1ScXCLWNpV/LyCfzxqEg/zoOplVttbZi7cTCJjNLi6snW2xq5Ib
GBikCu0e4kbvFCaXJ4mCaHB4NLWkw35qz+wmWfaDPLlyyYXYijVfxZ7LVOIVm1xGelFo42wmvalmiY6/oJP1didiVsrDOo1CN
t5AtzjMQ0RVQkKh9pRabHF2E+yW7ZS0jERXSOtiiFw1FC/MyVpLzGZwDoB8mR65VRxrKUEIzfp6JBZJmd/+EjqGa28kSfeKld
3NKeHL7iEUmcAexOC3Cg3h4nUsNVYyuOUsYgtqa0arRZdqmID8gMJ2IwFL+R2z5vTQFzZOTMJJCUxxZAn9Ga4xvTYFgSXL
qtO5iu03jSH7Gr/MmGIbwoZsFSBkGLgH2Pt4jDJCoTGjgUUIGp4cEa0w+XY8V2k= ubuntu@ip-172-31-34-232

Add SSH key

As you can see i have added the SSH key for the GitHub

Now Similarly add private key to your Jenkins
GO to your Project --> Configure
Check GitHub Project



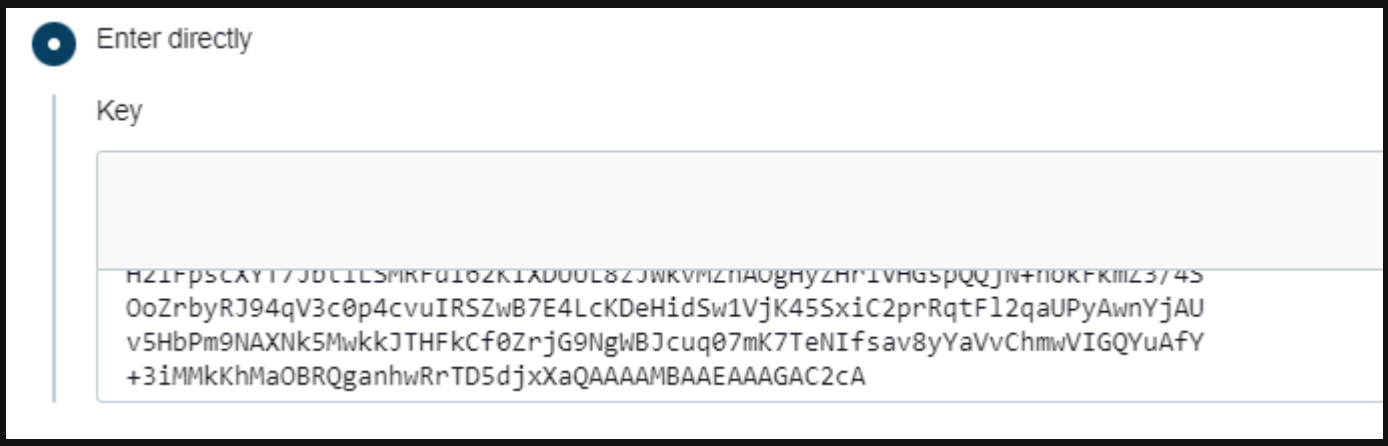
In Source Code Management select GIT and paste the repository URL

Now in Credentials Click on add

Provide the Details
In Kind select SSH with Private Key
Username select as Ubuntu --> Username of EC2 Instance

In Private Key select Entire Directly & paste your Private Key as we copied public key

```
ubuntu@ip-172-31-34-232:~/.ssh$ cat id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAwu7fl+YLvTSv4SmhVdFF2mu3hx2UrF0eLkhKZHj+r47fjJww4V53
OuwcbU9PP9rUBX9KWj45u1vmzIMVCSTG6U3xUH4zBIBlxyfLI75GUEbS/ItUnFwi1jaVfy
8gn88ahIP86DqZVbbW2Yu3EwiYzS4urJ1tsauSGxgYpArtHuJG7xQmlyeJgmhweDS1pMN+
as/sJln2gzy5csmF2Io1X8Wey1TiFZtcRnpRaONsJr2iJomOv6CT9XYnYlbKwzqNQjbeQL
c4zENEVUJCofaUWmxxdhPslu2UtIxEV0jrYohcNRQvzMlaS8xmcA6AfJkeuVUcaylBCM36
```



Once Done Check Specifier for me it's main

Now Click on Save

After that click on Build Now

You can see build is started Once Done open that build
Go to console Output & copy the address

Now open your Instance & change Directory with

```
ubuntu@ip-172-31-34-232:~/.ssh$ cd /var/lib/jenkins/workspace/sample-todo
ubuntu@ip-172-31-34-232:/var/lib/jenkins/workspace/sample-todo$ ls
Dockerfile  README.md  mainapp  manage.py  requirements.txt  todoZ
```

As you can see our project is present now lets install docker and build the docker image by following commands

```
ubuntu@ip-172-31-34-232: sudo apt install docker.io
```

Once Docker is installed build image by following command

```
ubuntu@ip-172-31-34-232: sudo docker build . -t todo
```

After successfully image is built run the image by

```
ubuntu@ip$ sudo docker run --name todo -d -p 8000:8000 todo
8751aae57f9b23e8f99642d182c8e02064e31873ff1c3d82ee56ccb067cad872
```

container name

port expose to run app

Image name

Verify the Application is running or not by below URL

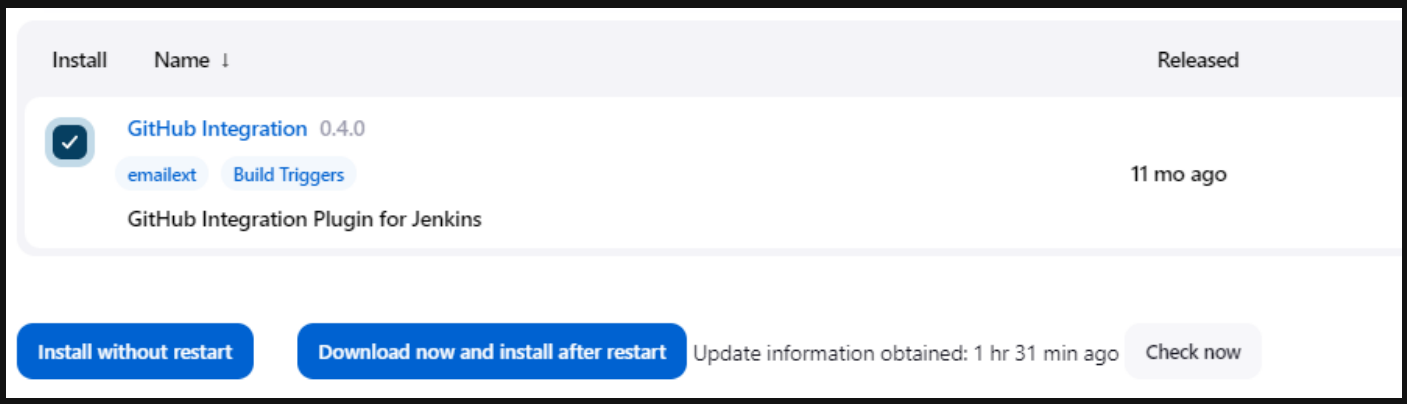
http:// [public-ip] :8000/

As you can see our application is running successfully

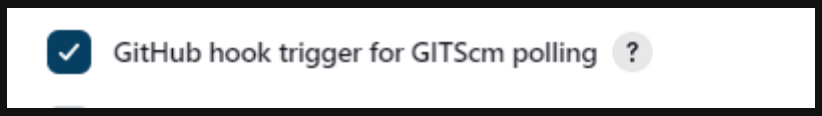
Step 5:-

Let's add a git webhook to implement Continuous Delivery

First install GitHub Integration Plugin for Jenkins
Go to Jenkins --> Manage Jenkins --> Manage Plugins
In Available Plugins Search for GitHub Integration
Select the plugin & Install without Restart



Once installation is done go to your Jenkins and configure
After opening Configuration check the dialog box as shown in the image

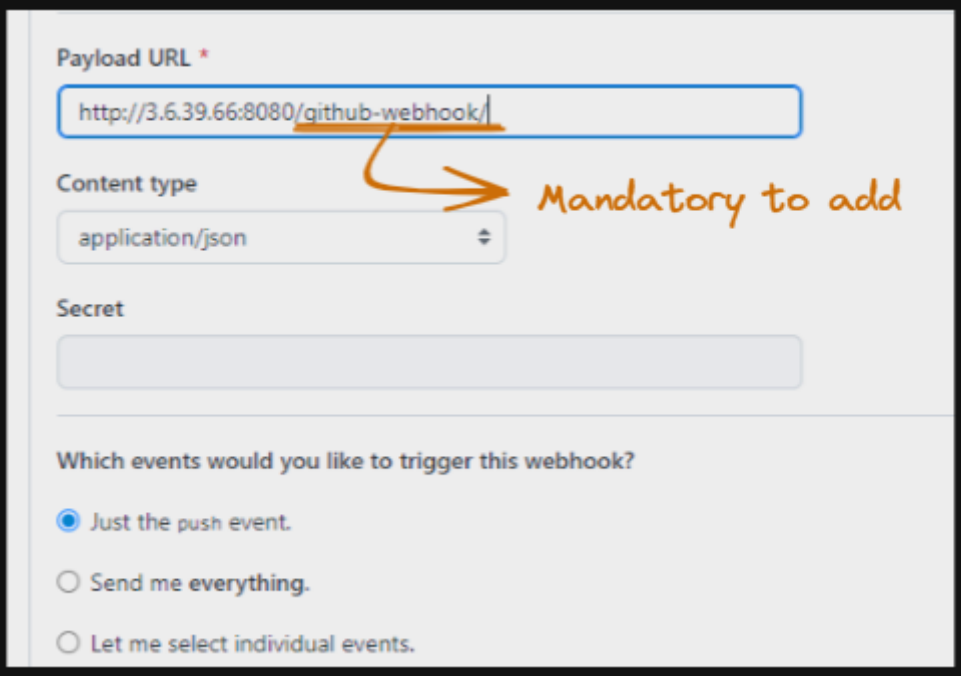


Check the Dialog box in build triggers section

Once Done go to Build Steps section click on add build step select execute shell
And add the below commands as shown in the image

After that click on Save

Now go to your GitHub open project repository click on setting
In Settings click on --> Webhook --> Add Webhook
Provide the Payload URL as
    Jenkis-URL / github-webhook /
Refer the below Image for configurations



After that click on Add Webhook

Once webhook is added go to your EC2 Instance give Jenkins access to run Docker commands by below command

```
sudo usermod -a -G docker jenkins
```

After that restart the jenkins server by below command

```
sudo systemctl restart jenkins
```

All the Steps are Successfully Completed.

Now try to change the code and push the changes to GIT
it will run a build and new docker image will be created &
it will be automatically going to deploy on your EC2 Instance

Console Output

```
Started by GitHub push by tushargangurde2029
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/sample-todo
The recommended git tool is: NONE
using credential ubuntu
 > git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/sample-todo/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/tushargangurde2029/django-todo.git # timeout=10
Fetching upstream changes from https://github.com/tushargangurde2029/django-todo.git
 > git --version # timeout=10
 > git --version # 'git version 2.34.1'
using GIT_SSH to set credentials lkjdsfl
Verifying host key using known hosts file
```

As you can see in the above image build successfully run by github after pushing code and also title has been change and new image is deployed successfully