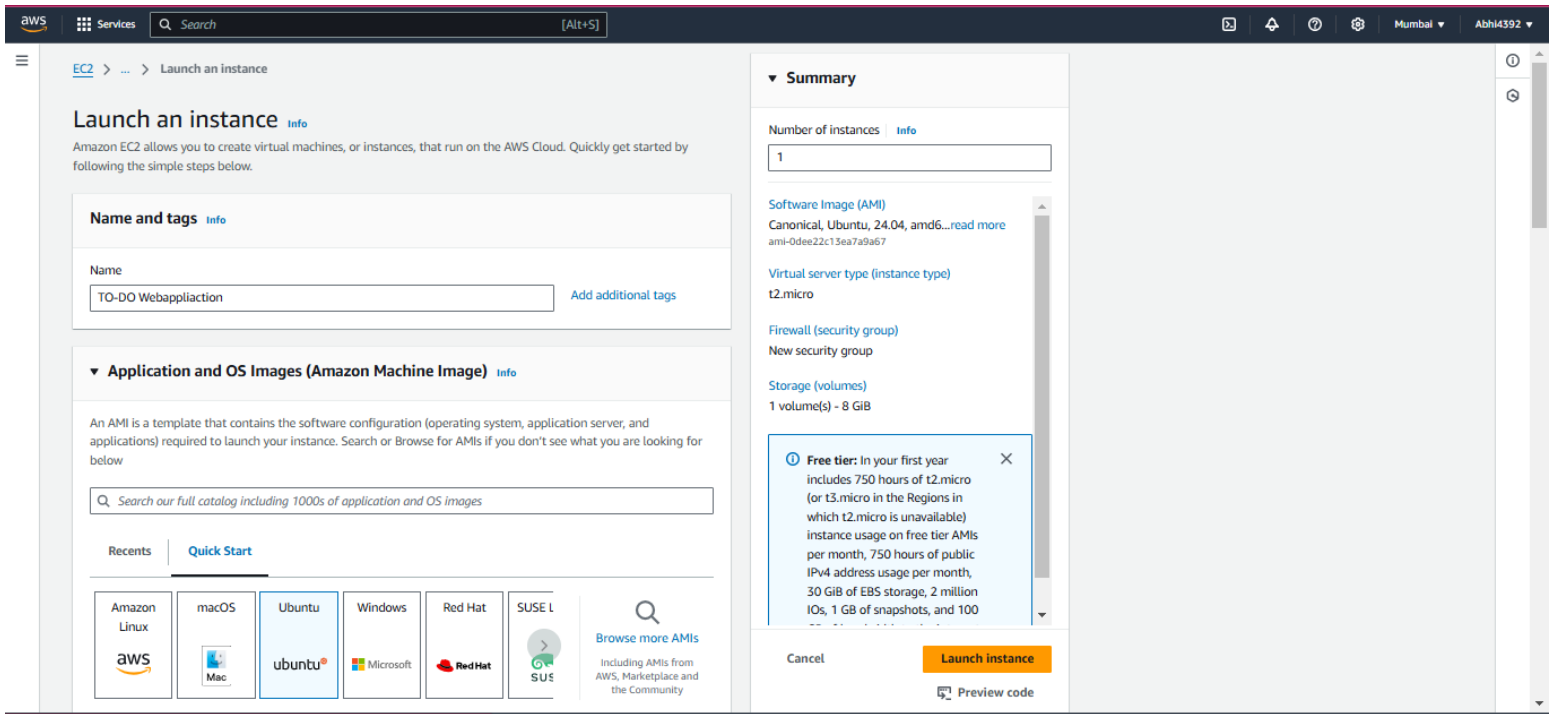


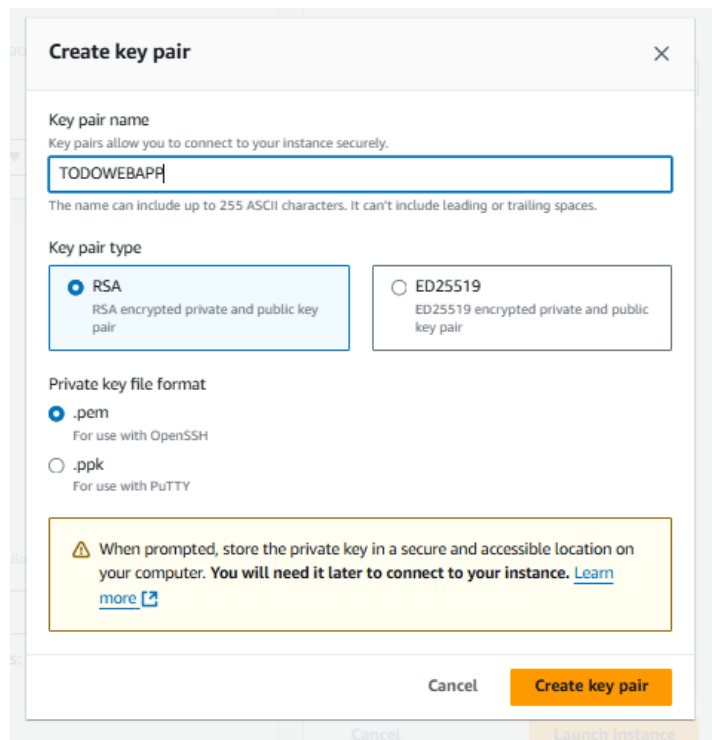
TO-DO WEBAPPLICTION

Steps:

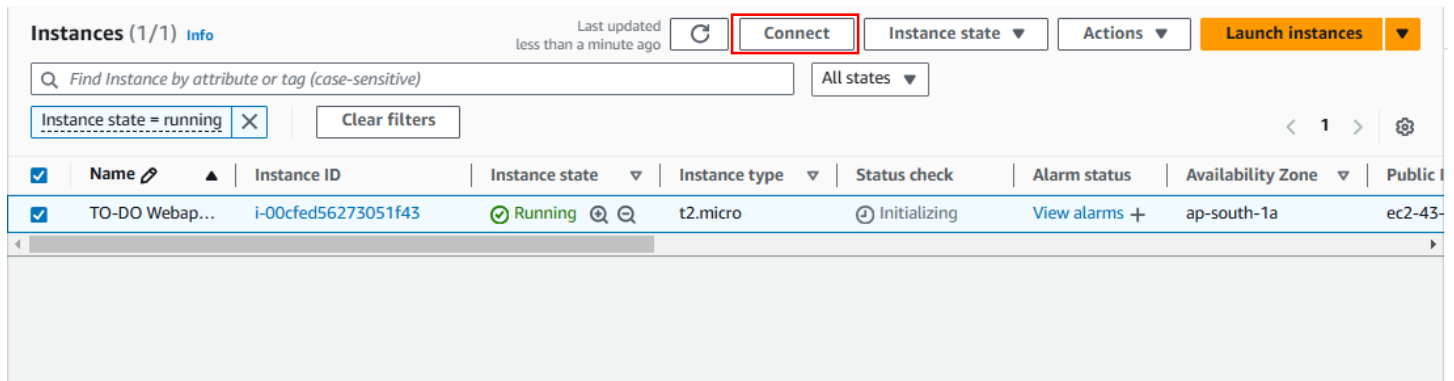
1) Create a virtual machine in AWS EC2 instance where OS is Ubuntu



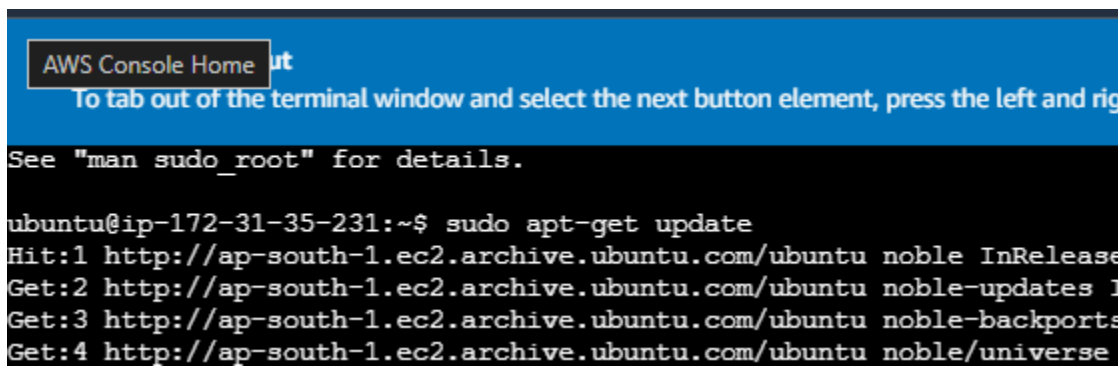
2) Create a Key Pair which will be later use to connect to the instance.



3) Instance is Created and connect the instance.



4) Update the system



5) Install Java in Ubuntu

sudo apt update

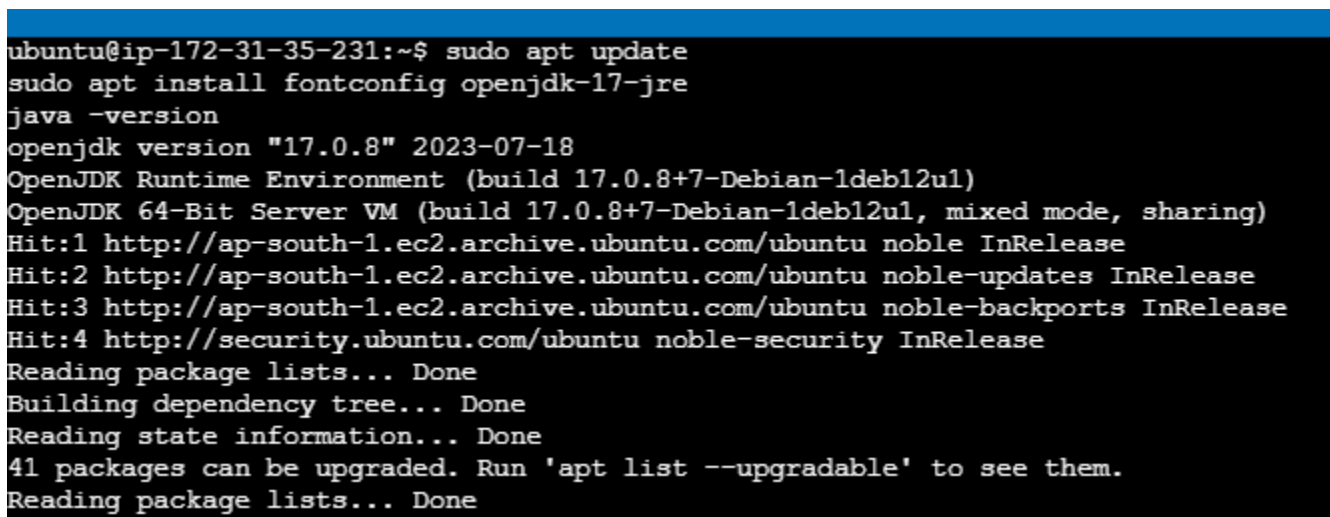
sudo apt install fontconfig openjdk-17-jre

java -version

openjdk version "17.0.8" 2023-07-18

OpenJDK Runtime Environment (build 17.0.8+7-Debian-1deb12u1)

OpenJDK 64-Bit Server VM (build 17.0.8+7-Debian-1deb12u1, mixed mode, sharing)



6) INSTALLATION OF JENKINS we first generate the key.

```
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
```

```
ubuntu@ip-172-31-35-231:~$ sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
```

7) Pushing key into devnull

```
echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
https://pkg.jenkins.io/debian-stable binary/" | sudo tee \
/etc/apt/sources.list.d/jenkins.list > /dev/null
```

```
ubuntu@ip-172-31-35-231:~$ echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]" \
https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
/etc/apt/sources.list.d/jenkins.list > /dev/null
```

8) Now command for installing

Sudo apt-get update (important to do after pushing)
sudo apt-get install jenkins

```
ubuntu@ip-172-31-35-231:~$ sudo apt-get update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Ign:4 https://pkg.jenkins.io/debian-stable binary/ InRelease
Get:5 https://pkg.jenkins.io/debian-stable binary/ Release [2044 B]
```

9) start the Jenkins

sudo systemctl enable Jenkins

```
ubuntu@ip-172-31-35-231:~$ sudo systemctl enable jenkins
Synchronizing state of jenkins.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable jenkins
ubuntu@ip-172-31-35-231:~$
```

Sudo systemctl start Jenkins

Sudo systemctl status Jenkins

```
ubuntu@ip-172-31-35-231:~$ sudo systemctl start jenkins
ubuntu@ip-172-31-35-231:~$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; preset: enabled)
   Active: active (running) since Fri 2024-11-15 05:56:21 UTC; 4min 26s ago
     Main PID: 4572 (java)
        Tasks: 38 (limit: 1130)
      Memory: 293.3M (peak: 338.3M)
         CPU: 18.904s
    CGroup: /system.slice/jenkins.service
            └─4572 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080

Nov 15 05:55:57 ip-172-31-35-231 jenkins[4572]: ef944f392fe94cad9969f1214e1a301f
Nov 15 05:55:57 ip-172-31-35-231 jenkins[4572]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Nov 15 05:55:57 ip-172-31-35-231 jenkins[4572]: *****
Nov 15 05:55:57 ip-172-31-35-231 jenkins[4572]: *****
Nov 15 05:55:57 ip-172-31-35-231 jenkins[4572]: *****
Nov 15 05:56:21 ip-172-31-35-231 jenkins[4572]: 2024-11-15 05:56:21.354+0000 [id=31] INFO jenkins.InitReactorRunner$1#onAt
Nov 15 05:56:21 ip-172-31-35-231 jenkins[4572]: 2024-11-15 05:56:21.383+0000 [id=23] INFO hudson.lifecycle.Lifecycle#onRea
Nov 15 05:56:21 ip-172-31-35-231 systemd[1]: Started jenkins.service - Jenkins Continuous Integration Server.
Nov 15 05:56:27 ip-172-31-35-231 jenkins[4572]: 2024-11-15 05:56:27.807+0000 [id=46] INFO h.m.DownloadService$Downloadable
Nov 15 05:56:27 ip-172-31-35-231 jenkins[4572]: 2024-11-15 05:56:27.808+0000 [id=46] INFO hudson.util.Retrier#start: Perfo
lines 1-20/20 (END)
```

10) Jenkins runs on port 8080 so if we are working with AWS EC2 instance we have to change the security group inbound rules and allow port 8080 for only My IP so it can be access by the root user only

Inbound rules

Outbound rules

Sharing - new

VPC associations - new

Tags

Inbound rules (3)

Manage tags

Edit inbound rules

Q Search

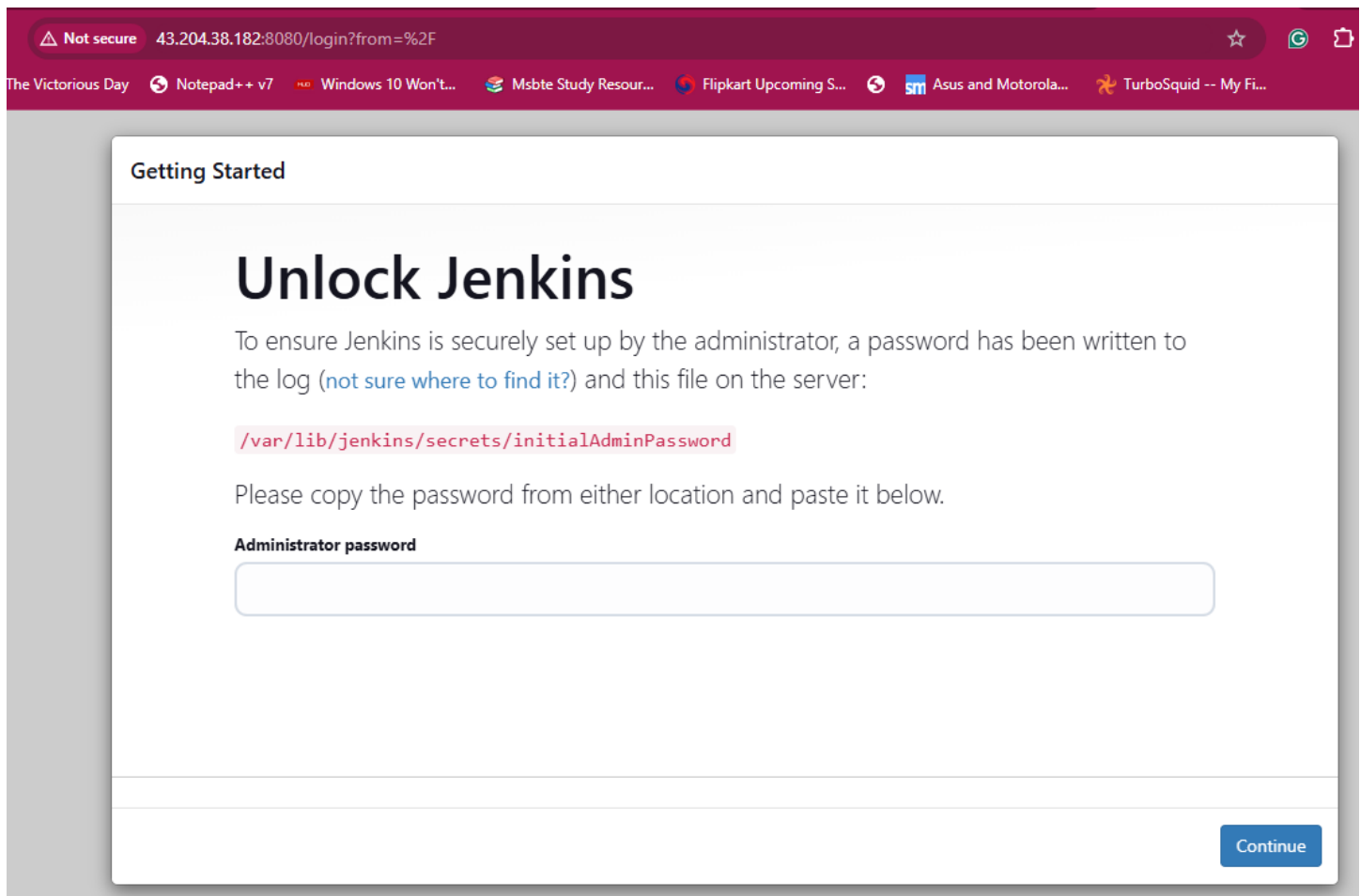
< 1 >

⚙

<input type="checkbox"/>	Name	Security group rule...	IP version	Type	Protocol	Port range
<input type="checkbox"/>	-	sgr-0074cce384f7fae16	IPv4	SSH	TCP	22
<input type="checkbox"/>	-	sgr-0251a0114c7d6d5...	IPv4	HTTPS	TCP	443
<input type="checkbox"/>	-	sgr-0bb374c71d2c76c67	IPv4	HTTP	TCP	80

Security group rule ID	Type	Protocol	Port range	Source	Description - optional	
sgr-0074cce384f7fae16	SSH	TCP	22	Custom	<div>Q</div> <div>0.0.0.0/0</div>	Delete
sgr-0251a0114c7d6d504	HTTPS	TCP	443	Custom	<div>Q</div> <div>0.0.0.0/0</div>	Delete
sgr-0bb374c71d2c76c67	HTTP	TCP	80	Custom	<div>Q</div> <div>0.0.0.0/0</div>	Delete
sgr-061c73e4ee26f6a21	Custom TCP	TCP	8080	My IP	<div>Q</div> <div>49.36.105.8/32</div>	Delete

11) Open Jenkins in the browser with the public ip and the port 8080

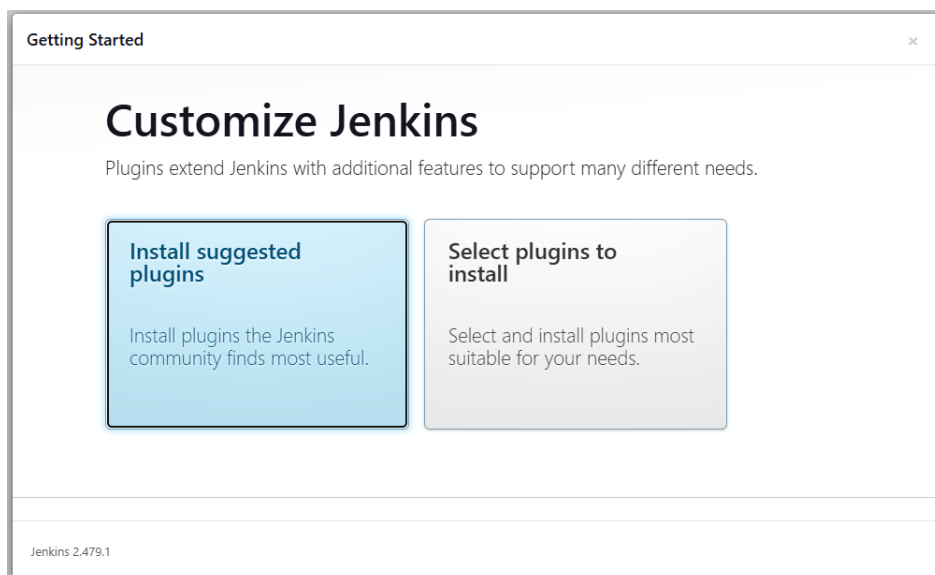


12)Getting the Admin password.

Sudo cat `/var/lib/jenkins/secrets/initialAdminPassword`

```
ubuntu@ip-172-31-35-231:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
ef944f392fe94cad9969f1214e1a301f
ubuntu@ip-172-31-35-231:~$
```

Now copy the code and paste it and continue



Create First Admin User

Username

Abhishek

Password

.....

Confirm password

.....

Full name

Abhishek Pujari

Jenkins 2.479.1

Skip and continue as admin

Save and Continue

Instance Configuration

Jenkins URL:

http://43.204.38.182:8080/

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 2.479.1

Not now

Save and Finish

Jenkins is ready!

Your Jenkins setup is complete.

Start using Jenkins

 **Jenkins**

Search (CTRL+K) ?



 1

 Abhishek Pujari

 log out

Dashboard >

+ New Item

 Build History Manage Jenkins My Views


Build Queue

v

No builds in the queue.

Build Executor Status

0/2 v

 Add description

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



13) Create a job in Jenkins from create new item.


✓ All > New Item


New Item


Enter an item name

TO-DO-APP

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.

14) Adding information in General

Description

This is a node.js TO DO app

Plain text [Preview](#)

☐ Discard old builds ?

☒ GitHub project

Project url ?

https://github.com/abhipujari76/TO-DO-CICD-USING-JENKINS.git

Advanced ▼

☐ This project is parameterized ?

Save

Apply

Source Code Management

☐ None

☒ Git ?

Repositories ?

Repository URL ?

Credentials ?

+ Add

Advanced ▾

Add Repository

15) Generating Private key pair using ssh

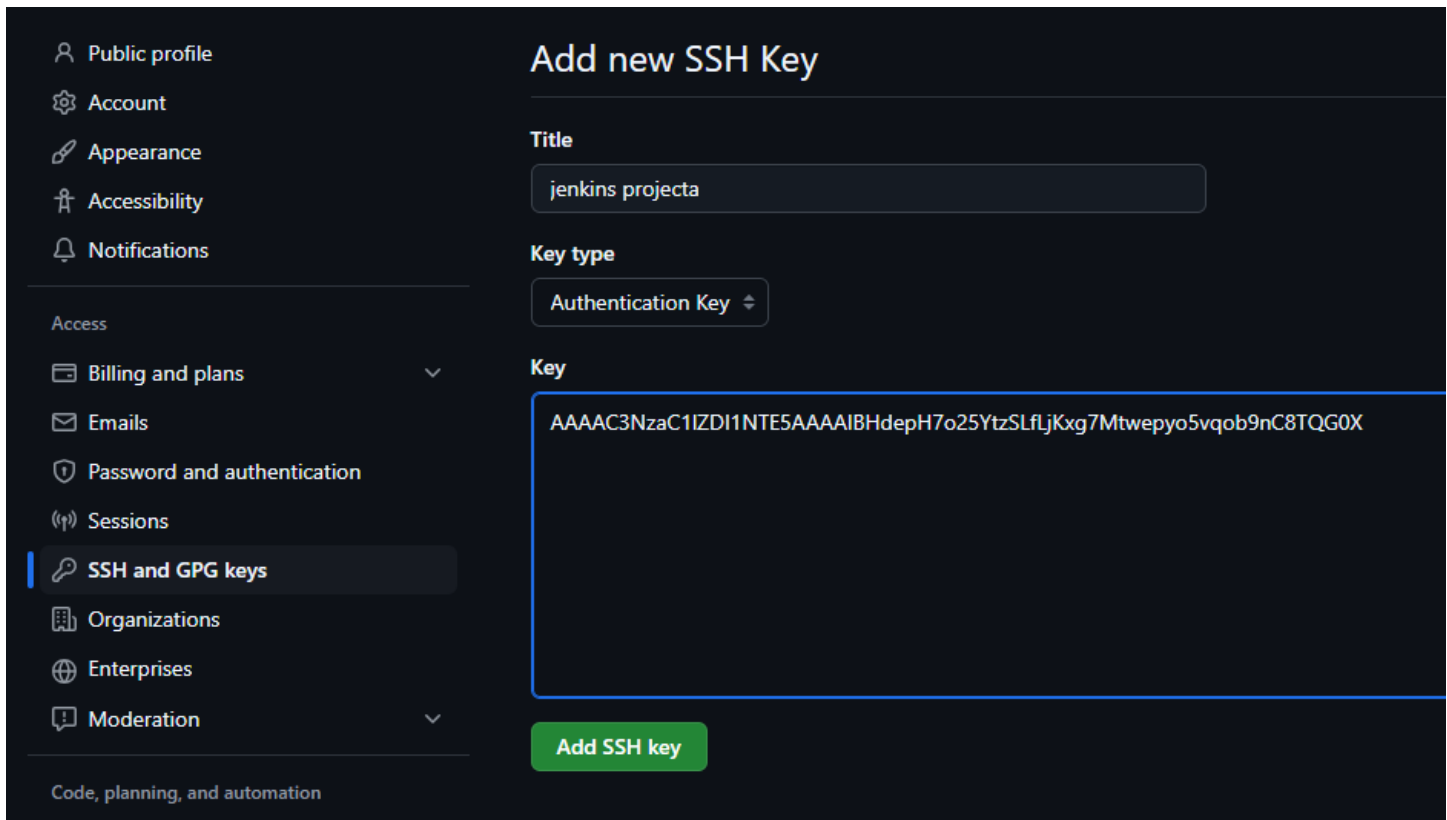
```
ubuntu@ip-172-31-35-231:~$ ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/ubuntu):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
```

16) Getting key pair location

Cd .ssh > ls > cat id_ed2559.pub(public key) >cat id_ed2559(private key)

```
ubuntu@ip-172-31-35-231:~$ cd.ssh
cd.ssh: command not found
ubuntu@ip-172-31-35-231:~$ cd .ssh
ubuntu@ip-172-31-35-231:~/.ssh$ ls
authorized_keys  id_ed25519  id_ed25519.pub
ubuntu@ip-172-31-35-231:~/.ssh$ cat id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIBHdeph7o25YtzSLfLjKxg7Mtwepyo5vqob9nC8TQG0X ubuntu@ip-172-31-35-231
ubuntu@ip-172-31-35-231:~/.ssh$ cat id_ed25519
-----BEGIN OPENSSH PRIVATE KEY-----
b3B1bnZaC1rZXktdjEAAAAABG5vbmUAAAAAEbm9uZQAAAAABAAAAABAAAAAMwAAAAAtzc2gtZW
QyNTUxOQAAACAR3XqR+6NuWLC0i3y4ysYOzLcHqcqOb6qG/ZwvE0BtFwAAAKBEEXklRBF5
JQAAAAAtzc2gtZWQyNTUxOQAAACAR3XqR+6NuWLC0i3y4ysYOzLcHqcqOb6qG/ZwvE0BtFw
AAAEKgZyUAJSbotg9/pc+nrp2zZbxd5mE62NPbn5Fqr2xcRRHdeph7o25YtzSLfLjKxg7M
twepyo5vqob9nC8TQG0XAAAAAF3VidW50dUBpcC0xNzItMzMtMzUtMjMxMzUxMzUxMzUxMzUx
-----END OPENSSH PRIVATE KEY-----
```


17) To connect Git hub to Jenkins we need to add Public key to the git hub account in the settings .



The screenshot shows the GitHub 'Add new SSH Key' form. On the left is a sidebar with navigation links: Public profile, Account, Appearance, Accessibility, Notifications, Access, Billing and plans, Emails, Password and authentication, Sessions, SSH and GPG keys (highlighted), Organizations, Enterprises, and Moderation. The main content area has the title 'Add new SSH Key'. It contains three fields: 'Title' with the value 'jenkins projecta', 'Key type' set to 'Authentication Key', and 'Key' containing a long alphanumeric string. A green 'Add SSH key' button is at the bottom right.

Public profile

Account

Appearance

Accessibility

Notifications

Access

Billing and plans

Emails

Password and authentication

Sessions

SSH and GPG keys

Organizations

Enterprises

Moderation

Code, planning, and automation

Add new SSH Key

Title

jenkins projecta

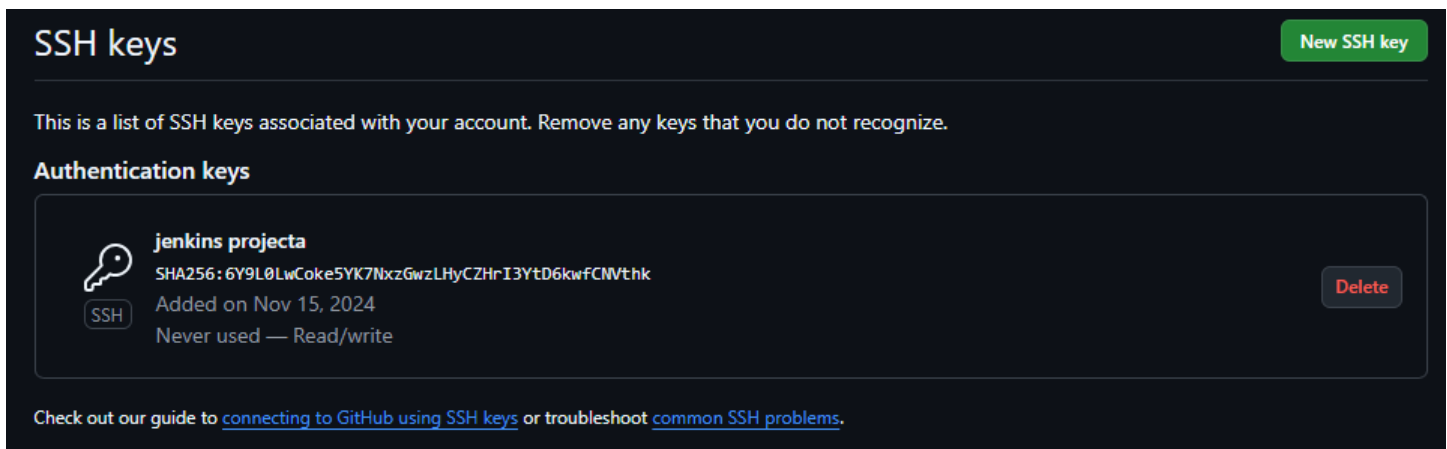
Key type

Authentication Key

Key

AAAAC3NzaC1lZDI1NTE5AAAAIBHdepH7o25YtzSLfjKxg7Mtwepyo5vqob9nC8TQG0X

Add SSH key




The screenshot shows the GitHub 'SSH keys' page. At the top right is a green 'New SSH key' button. Below the header, a message states: 'This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.' Under the 'Authentication keys' section, there is a card for 'jenkins projecta'. The card shows a key icon, the key type 'SSH', the SHA256 hash, the date 'Added on Nov 15, 2024', and the status 'Never used — Read/write'. A red 'Delete' button is on the right. At the bottom, a link points to a guide for connecting to GitHub using SSH keys.

SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Authentication keys



jenkins projecta

SHA256: 6Y9L0LwCoke5YK7NxcGwzLHyCZhrI3YtD6kwfCNVthk

Added on Nov 15, 2024

Never used — Read/write

Delete

Check out our guide to [connecting to GitHub using SSH keys](#) or troubleshoot [common SSH problems](#).

18) Adding Cradentials to the Jenkins

Source Code Management

☐ None

☒ Git ?

Repositories ?

Repository URL ?

https://github.com/abhipujari76/TO-DO-CICD-USING-JENKINS.git

Credentials ?

- none -

Jenkins Credentials Provider

Jenkins

Save

Apply

Jenkins Credentials Provider: Jenkins

Add Credentials

Domain

Global credentials (unrestricted)

Kind

SSH Username with private key

Scope ?

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

ID ?

Github-jenkins

Description ?

Jenkins and github intigration

Username

ubuntu

☐ Treat username as secret ?

Private Key

☒ Enter directly

Key

```
QyNTUXOQAAACAR3XqR+6NuWLC0i3y4ysY0ZLChqcQ0b6qG/ZwVE0BtFWAAAKBEEAKIRBF5
JQAAAAAtzc2gtZWQyNTUxOQAAACAR3XqR+6NuWLC0i3y4ysY0ZLChqcQ0b6qG/ZwVE0BtFW
AAAEHgZYUAJ5botg9/pc+nnp2zZbxd5mE62NPbn5Fqr2xcRRHdeph7o25Yt2SLfLjKxg7M
twepyo5vqob9nC8TQG8XAAAAF3V1dW50dUBpcC0xNzItMzEtMzUtMjMxMQIDBAUG
```

Passphrase

☐ None

☒ Git ?

Repositories ?

Repository URL ? ✕

`https://github.com/abhipujari76/TO-DO-CICD-USING-JENKINS.git`

Credentials ?

ubuntu (Jenkins and github intigration) ▼

+ Add

Advanced ▼

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ? ✕

`*/main`

19) Now we have to build it so it can get all the files from the git hub

Status

Changes

Workspace

Build Now

Configure

Delete Project

GitHub

Rename

Builds

Filter

Today

✓ #3 3:01 PM

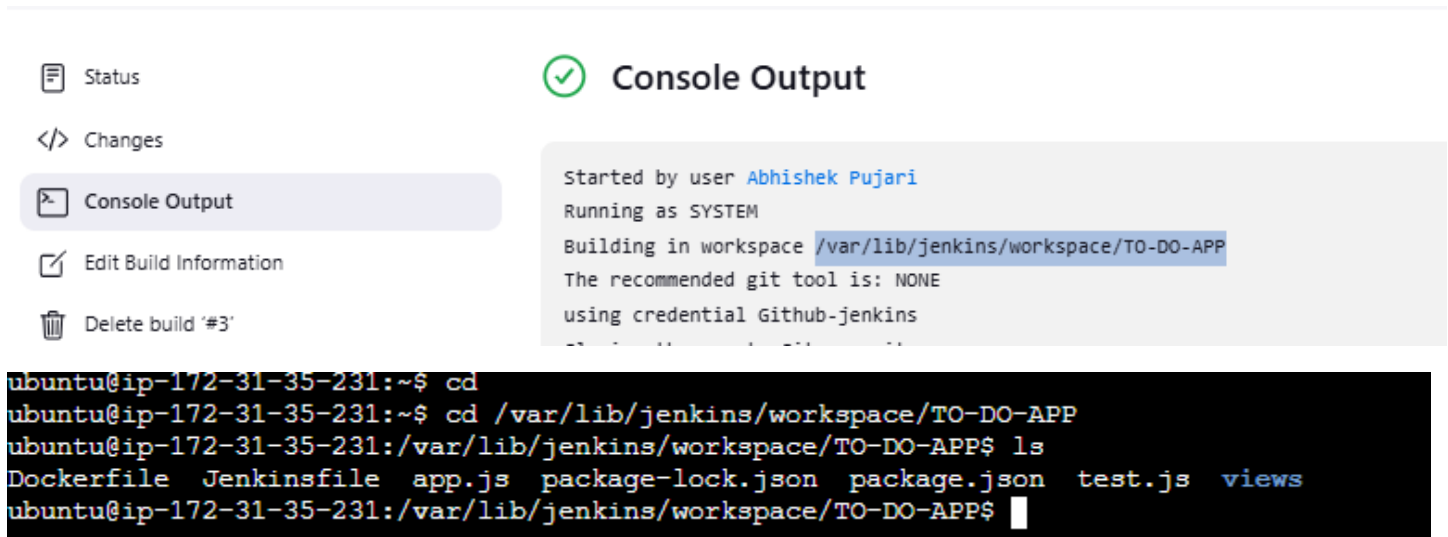
✓ TO-DO-APP

This is a node.js TO DO app

Permalinks

- Last build (#3), 27 sec ago
- Last stable build (#3), 27 sec ago
- Last successful build (#3), 27 sec ago
- Last completed build (#3), 27 sec ago

20) Check if the code has pull from the git hub



The screenshot shows the Jenkins interface. On the left, a sidebar contains links: Status, Changes, Console Output (selected), Edit Build Information, and Delete build '#3'. The main area is titled 'Console Output' with a green checkmark icon. It displays the following text:

```
Started by user Abhishek Pujari
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/TO-DO-APP
The recommended git tool is: NONE
using credential Github-jenkins
```

Below this, a terminal window shows the following commands and output:

```
ubuntu@ip-172-31-35-231:~$ cd
ubuntu@ip-172-31-35-231:~$ cd /var/lib/jenkins/workspace/TO-DO-APP
ubuntu@ip-172-31-35-231:/var/lib/jenkins/workspace/TO-DO-APP$ ls
Dockerfile Jenkinsfile app.js package-lock.json package.json test.js views
ubuntu@ip-172-31-35-231:/var/lib/jenkins/workspace/TO-DO-APP$
```

21) We need to install Nodejs and npm on the server

Sudo apt install nodejs

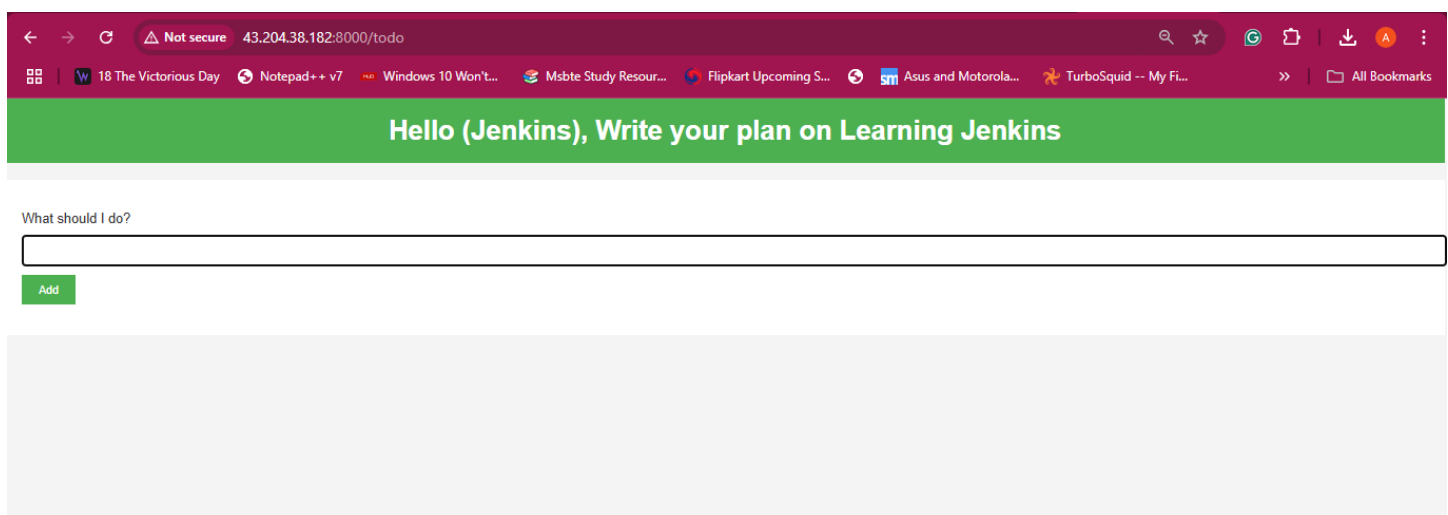
Sudo npm install

```
ubuntu@ip-172-31-35-231:/var/lib/jenkins/workspace/TO-DO-APP$ nodejs -v
v18.19.1
```

22) Now Run the web-app using command

node app.js

```
ubuntu@ip-172-31-35-231:/var/lib/jenkins/workspace/TO-DO-APP$ node app.js
Todolist running on http://0.0.0.0:8000
```



The screenshot shows a web browser window with the address bar displaying '43.204.38.182:8000/todo'. The page has a green header with the text 'Hello (Jenkins), Write your plan on Learning Jenkins'. Below the header, there is a form with the label 'What should I do?' and a text input field. A green 'Add' button is located below the input field. The browser's address bar and tabs are visible at the top.

23) Now we have to create a docker for virtualization

Why we do dockerize so that it can run on any environment.

Sudo apt install docker.io

sudo usermod -a -G docker \$USER(Give permission to the docker to use docker ps)

```
ubuntu@ip-172-31-35-231:~$ sudo apt install docker.io
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
```

24) Creating a Docker file

Create a file using command sudo vim Dockerfile

Node Base Image

FROM node:12.2.0-alpine

#Working Directry

WORKDIR /node

#Copy the Code

COPY . .

#Install the dependencies

RUN npm install

RUN npm run test

EXPOSE 8000

#Run the code

CMD ["node","app.js"]

```
# Node Base Image
FROM node:12.2.0-alpine

#Working Directry
WORKDIR /node

#Copy the Code
COPY . .

#Install the dependencies
RUN npm install
RUN npm run test
EXPOSE 8000

#Run the code
CMD ["node","app.js"]
~
~
~
~
~
~
~
```

25) Build the docker file to make the container

Docker build . -t todo-node-app

```
ubuntu@ip-172-31-35-231:~$ cd /var/lib/jenkins/workspace/TO-DO-APP
ubuntu@ip-172-31-35-231:/var/lib/jenkins/workspace/TO-DO-APP$ docker build . -t todo-node-app
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/
```

26) Make a container through docker to run the code

sudo docker run -d --name node-todo-app -p 8000:8000 todo-node-app

```
ubuntu@ip-172-31-35-231:/var/lib/jenkins/workspace/TO-DO-APP$ sudo docker run -d --name node-todo-app -p 8000:8000 todo-node-app
67a650ee931f3d2f00881c8b292fc28d2fa75412e661a78df15f9551d788badf
ubuntu@ip-172-31-35-231:/var/lib/jenkins/workspace/TO-DO-APP$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
67a650ee931f   todo-node-app  "node app.js"           7 seconds ago Up 6 seconds   0.0.0.0:8000->8000/tcp, :::8000->8000/tcp  node-todo-app
ubuntu@ip-172-31-35-231:/var/lib/jenkins/workspace/TO-DO-APP$
```

Now the code will be running even if it is stopped in the terminal using docker.

Running code through Jenkins

1) Kill the docker

Docker kill Container_id

```
ubuntu@ip-172-31-35-231:/var/lib/jenkins/workspace/TO-DO-APP$ docker kill 67a650ee93
67a650ee93
ubuntu@ip-172-31-35-231:/var/lib/jenkins/workspace/TO-DO-APP$
```

2) Go to Jenkins and select the job go to configuration and go to build steps

Go in the Build Environment

Click on build steps using shell

Do the steps which use to create the container and run the container.

docker build . -t todo-node-app

docker run -d --name node-app -p 8000:8000 todo-node-app

Dashboard > TO-DO-APP > Configuration

Configure

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

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
- ☐ Terminate a build if it's stuck
- ☐ With Ant ?

Build Steps

Execute shell ?

Command

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

```
docker build . -t todo-node-app
docker run -d --name node-app -p 8000:8000 todo-node-app
```

Advanced ▾

Save Apply

Important Note : Before running the build in the Jenkins give permission to the Jenkins

sudo chmod 777 /var/lib/jenkins/workspace/TO-DO-APP

sudo usermod -a -G docker jenkins

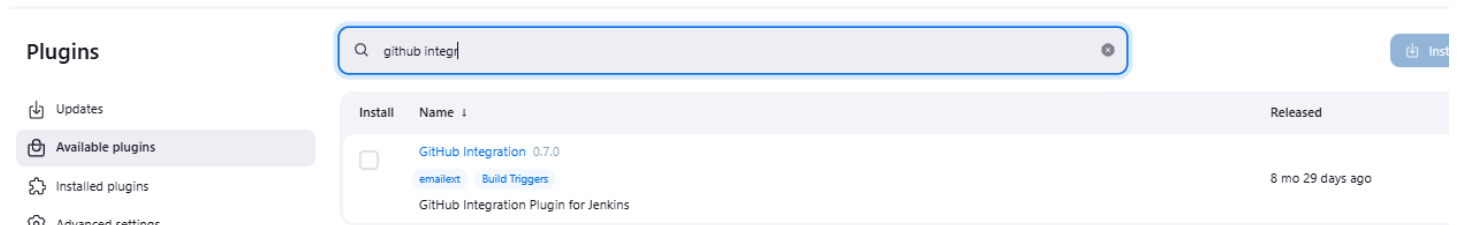
```
ubuntu@ip-172-31-35-231:/var/lib/jenkins/workspace/TO-DO-APP$ sudo chmod 777 /var/lib/jenkins/workspace/TO-DO-APP
ubuntu@ip-172-31-35-231:/var/lib/jenkins/workspace/TO-DO-APP$
```

Now click on the Build button

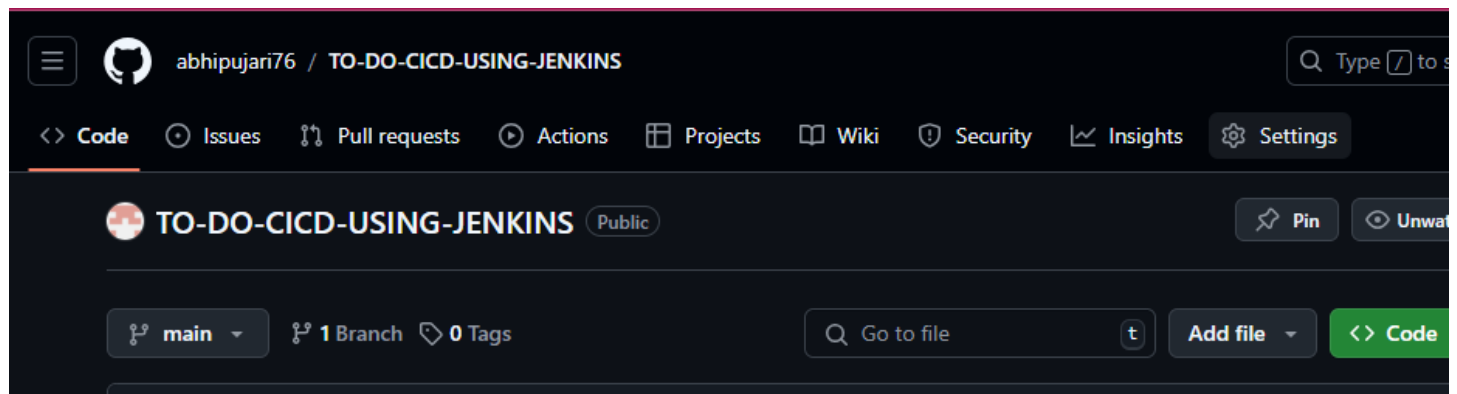
```
ubuntu@ip-172-31-35-231:/var/lib/jenkins/workspace/TO-DO-APP$ sudo usermod -a -G docker jenkins
ubuntu@ip-172-31-35-231:/var/lib/jenkins/workspace/TO-DO-APP$ sudo systemctl restart jenkins
^C
ubuntu@ip-172-31-35-231:/var/lib/jenkins/workspace/TO-DO-APP$ sudo systemctl restart jenkins
ubuntu@ip-172-31-35-231:/var/lib/jenkins/workspace/TO-DO-APP$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; preset: enabled)
   Active: active (running) since Fri 2024-11-15 19:02:22 UTC; 7s ago
     Main PID: 2701 (java)
    Tasks: 46 (limit: 1130)
```

To auto-mate all the process of CICD we will be using WEBHOOKS which will be in the middle of git hub and Jenkins

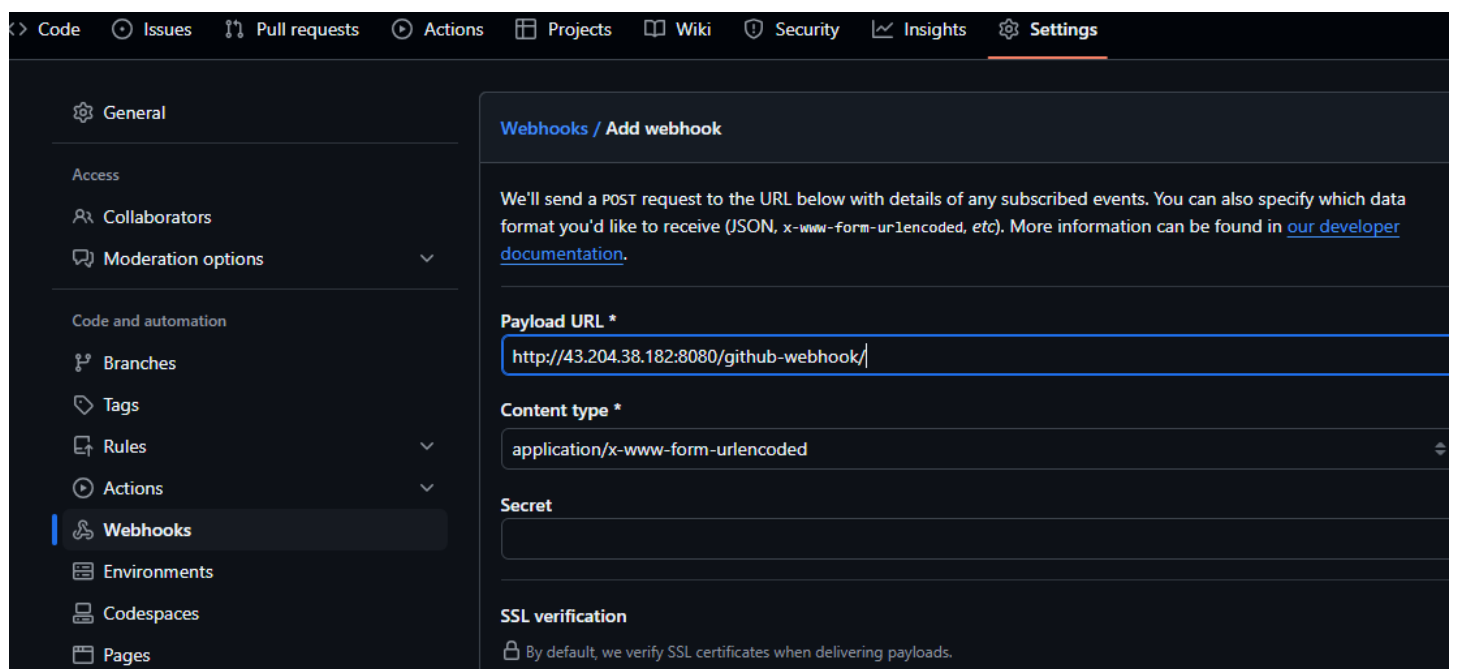
1) Install a plugin call github Integration



2) Now we have to create a webhook in the GitHub through the repo setting



Add your Jenkins payload URL in the github Webhook



Important : Make Changes for instance security group where Jenkins is for MyIP to anywhere so that github can access it from any where.

Make content type to application/json and then add webhook

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

Payload URL *

Content type *

Secret

SSL verification
☐ By default, we verify SSL certificates when delivering payloads.
☒ **Enable SSL verification** ☐ **Disable (not recommended)**

✓ <http://43.204.38.182:8080/github-w...> (push) Edit Delete

Last delivery was successful.

Dashboard > TO-DO-APP > 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 Branches
- ☐ GitHub Pull Requests ?
- ☒ **GitHub hook trigger for GITScm polling** ?

When Jenkins receives a GitHub push hook, GitHub Plugin checks to see if the push matches the SCM/Git section of this job. If they match and this option is enabled, Jenkins will check for change and initiates a build. The last sentence describes the behavior of the hook trigger.

- ☐ Poll SCM ?

3) When you make some Changes in the Github it will automatic build trigger and make the build

Dashboard > TO-DO-APP >

Build Now

Configure

Delete Project

GitHub Hook Log

GitHub

Rename

- Last build (#8), 28 min ago
- Last stable build (#8), 28 min ago
- Last successful build (#8), 28 min ago
- Last failed build (#7), 31 min ago
- Last unsuccessful build (#7), 31 min ago
- Last completed build (#8), 28 min ago

Builds

Filter

Pending

#9

In the quiet period. Expires in 2.4 sec

Today

43.204.38.182:8000/todo

18 The Victorious Day | Notepad++ v7 | Windows 10 Won't... | Msbte Study Resour... | Flipkart Upcoming S... | Asus and Motorola... | TurboSquid -- My Fi... | All Bookmarks

Hello USER(Jenkins), Write your plan on Learning Jenkins

To do Homework

What should I do?

Add