

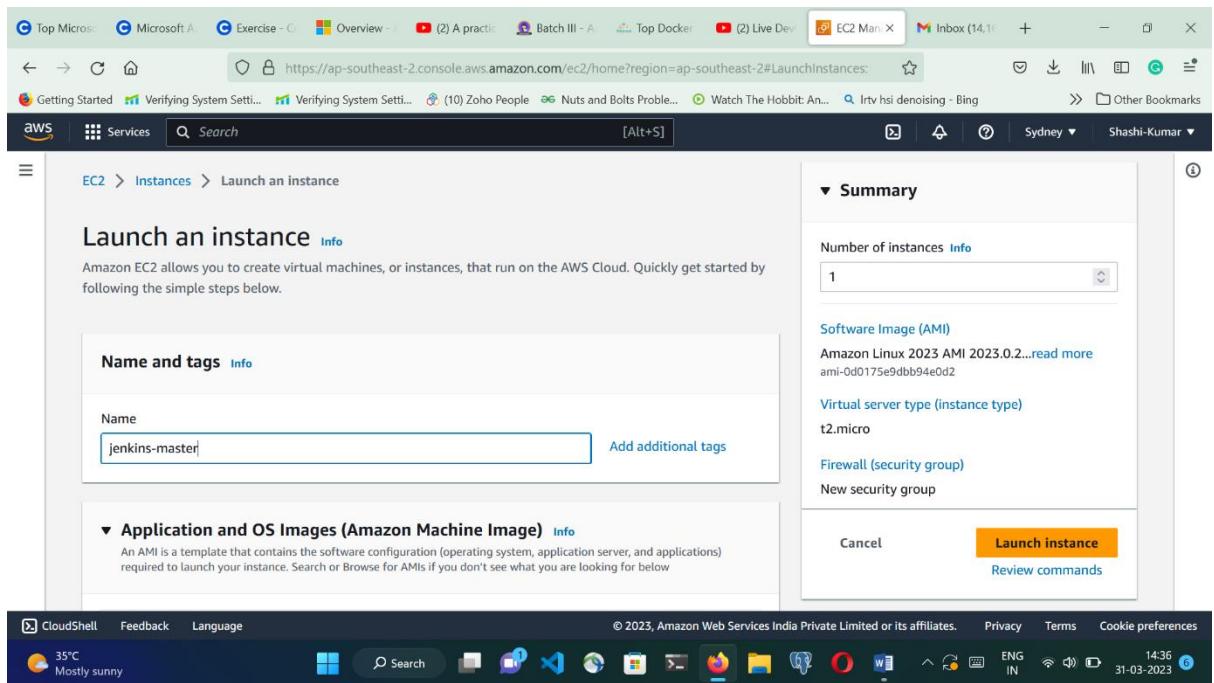
## TASK – “Create a CI/CD pipeline and deploy a web application”

- It should deploy a simple web application to a server on a code push to a repository.
- The deployed web application should be reachable on any web browser.
- Setup to be done using AWS, Jenkins, GitHub
- Jenkins should not be on the same server as the application being deployed to.

Tools – AWS EC2, GitHub, Jenkins, Docker, Webhook

## Solution-

### Step-1. Create an EC2 Instance



required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

**Quick Start**

Amazon Linux macOS Ubuntu Windows Red Hat

ubuntu® Microsoft Red Hat

Amazon Machine Image (AMI)

**Ubuntu Server 22.04 LTS (HVM, SSD Volume Type)**

ami-05f998315cca9bfe3 (64-bit (x86)) / ami-0a9cfaf40cedba2d5a (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

Description

Canonical, Ubuntu, 22.04 LTS, and 64-bit x86 image (ami-05f998315cca9bfe3) build on 2023-07-25

Number of instances **Info**

1

Software Image (AMI)

Canonical, Ubuntu, 22.04 LTS, ...[read more](#)

ami-05f998315cca9bfe3

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

Cancel **Launch instance** Review commands

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

Instance type

**t2.micro**

Family: t2 1 vCPU 1 GB Memory

On-Demand Linux pricing: 0.0146 USD per Hour

On-Demand Windows pricing: 0.0192 USD per Hour

On-Demand SUSE pricing: 0.0146 USD per Hour

On-Demand RHEL pricing: 0.0746 USD per Hour

Free tier eligible

Compare instance types

**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

Create new key pair

Number of instances **Info**

1

Software Image (AMI)

Canonical, Ubuntu, 22.04 LTS, ...[read more](#)

ami-05f998315cca9bfe3

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

Cancel **Launch instance** Review commands

Screenshot of the AWS EC2 Create Key Pair dialog box:

**Create key pair**

Key pairs allow you to connect to your instance securely.

Enter the name of the key pair below. When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more](#)

**Key pair name**  
project-key

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

**Key pair type**  
 RSA RSA encrypted private and public key pair  
 ED25519 ED25519 encrypted private and public key pair (Not supported for Windows instances)

**Private key file format**  
 .pem For use with OpenSSL

**Launch Instance** **Review commands**

Screenshot of the AWS EC2 Configure Storage dialog box:

**Configure storage**

1x 8 GiB gp2 Root volume (Not encrypted)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

Add new volume

The selected AMI contains more instance store volumes than the instance allows. Only the first 0 instance store volumes from the AMI will be accessible from the instance

0 x File systems

**Advanced details**

**Summary**

Number of instances: 1

Software Image (AMI): Canonical, Ubuntu, 22.04 LTS, ami-05f998315cca9bfe3

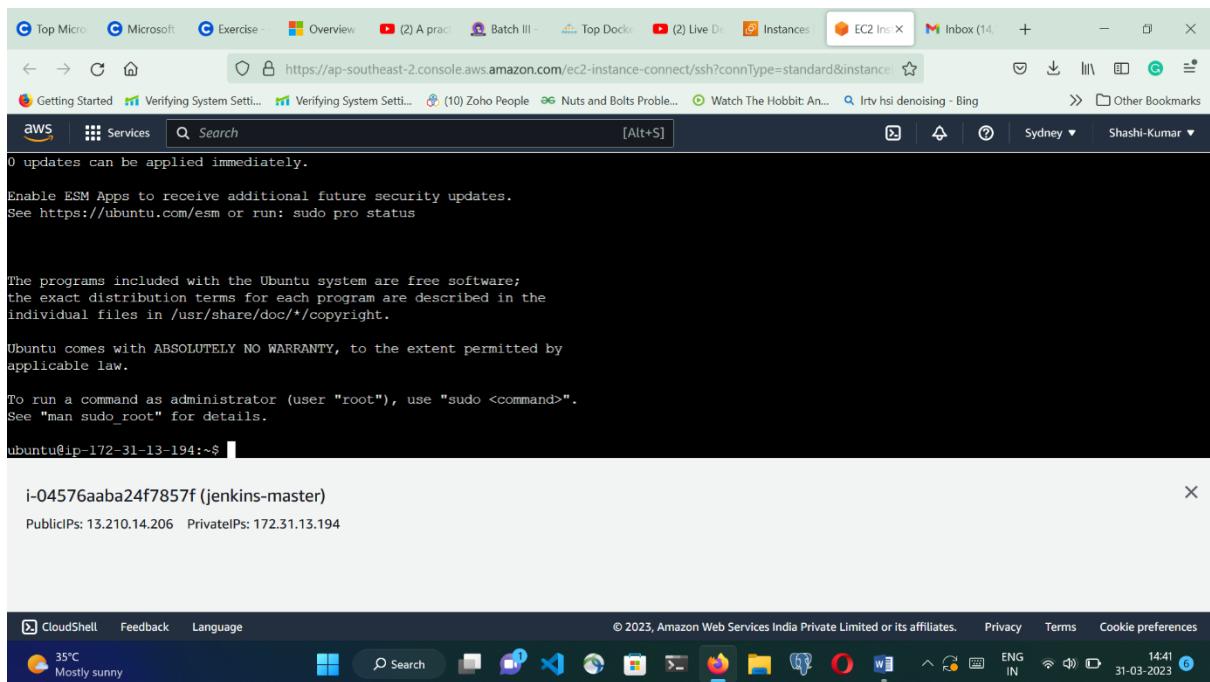
Virtual server type (instance type): t2.micro

Firewall (security group): New security group

**Launch instance** **Review commands**

The screenshot shows the AWS EC2 Instances Launch an instance page. A success message box displays: "Success Successfully initiated launch of instance (i-04576aaba24f7857f)". Below this, a "Launch log" link is visible. The main content area is titled "Next Steps" with a search bar and numbered steps: 1. Create billing and free tier usage alerts, 2. Connect to your instance, 3. Connect an RDS database, and 4. Create EBS snapshot policy. The bottom navigation bar includes CloudShell, Feedback, Language, and links to various AWS services like Overview, Batch, Docker, and Live Dev.

The screenshot shows the AWS EC2 Instances page. The left sidebar is expanded to show the "Instances" section, with "Instances" selected. The main content area displays "Instances (1/1) Info" for one instance: jenkins-master, i-04576aaba24f7857f. The instance is listed as "Running" with the type "t2.micro". The "Actions" dropdown menu is open, showing options like "Launch instances", "Stop", "Start", "Reboot", "Terminate", and "Edit". The bottom navigation bar includes CloudShell, Feedback, Language, and links to various AWS services like Overview, Batch, Docker, and Live Dev.



## Step-2. Install Jenkins on EC2 Instance from official Documentation

1. sudo apt update
2. sudo apt install openjdk-11-jre
3. curl -fsSL https://pkg.jenkins.io/debian/jenkins.io-2023.key | sudo tee \ /usr/share/keyrings/jenkins-keyring.asc > /dev/null
4. 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
5. sudo apt-get update
6. sudo apt-get install fontconfig openjdk-11-jre
7. sudo apt-get install Jenkins
8. sudo systemctl enable Jenkins
9. sudo systemctl start Jenkins
10. sudo systemctl status Jenkins

```

ubuntu@ip-172-31-13-194:~$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
     Active: active (running) since Fri 2023-03-31 09:22:14 UTC; 23min ago
       Main PID: 5553 (java)
          Tasks: 36 (limit: 1141)
        Memory: 306.5M
         CPU: 46.712s
        CGroup: /system.slice/jenkins.service
                  └─5553 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080

Mar 31 09:21:38 ip-172-31-13-194 jenkins[5553]: fad8alba86604d41a56c81631a456c65
Mar 31 09:21:38 ip-172-31-13-194 jenkins[5553]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Mar 31 09:21:38 ip-172-31-13-194 jenkins[5553]: ****
Mar 31 09:21:38 ip-172-31-13-194 jenkins[5553]: ****
Mar 31 09:21:38 ip-172-31-13-194 jenkins[5553]: ****
Mar 31 09:22:14 ip-172-31-13-194 jenkins[5553]: 2023-03-31 09:22:14.632+0000 [id=28]      INFO      jenkins.InitReactorRunner$1#onAttained: C
Mar 31 09:22:14 ip-172-31-13-194 jenkins[5553]: 2023-03-31 09:22:14.657+0000 [id=22]      INFO      hudson.lifecycle.Lifecycle$1#onReady: Jenkins

```

i-04576aaba24f7857f (jenkins-master)

PublicIPs: 13.210.14.206 PrivateIPs: 172.31.13.194

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences 35°C Mostly sunny 15:16 31-03-2023

### Step-3. Jenkins run at port 8080, so open this port in EC2 security group

New EC2 Experience Tell us what you think

EC2 Dashboard EC2 Global View Events Tags Limits Instances Instances Instance Types Launch Templates Spot Requests Savings Plans Reserved Instances Dedicated Hosts

EC2 > Security Groups > sg-06936f3045c150800 - launch-wizard-1

**sg-06936f3045c150800 - launch-wizard-1**

**Details**

Security group name	Security group ID	Description	VPC ID
launch-wizard-1	sg-06936f3045c150800	launch-wizard-1 created 2023-03-31T09:06:49.867Z	vpc-0e477674f88c7fc8a
Owner	Inbound rules count	Outbound rules count	
532806123370	3 Permission entries	1 Permission entry	

**Inbound rules**    **Outbound rules**    **Tags**

You can now check network connectivity with Reachability Analyzer

Run Reachability Analyzer

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences 35°C Mostly sunny 15:20 31-03-2023

The screenshot shows the AWS Management Console with the URL <https://ap-southeast-2.console.aws.amazon.com/ec2/home?region=ap-southeast-2#ModifyInboundSecurityGroup>. The page title is "Edit inbound rules". The navigation path is EC2 > Security Groups > sg-06936f3045c150800 - launch-wizard-1 > Edit inbound rules. The main content area displays a table of inbound rules:

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-0deec7ecd433ecc1	HTTP	TCP	80	Custom 0.0.0.0/0	
sgr-00960628cbcf97ef9	SSH	TCP	22	Custom 0.0.0.0/0	
sgr-0214f5c8d204b1f01	HTTPS	TCP	443	Custom 0.0.0.0/0	

At the bottom, there are links for CloudShell, Feedback, Language, © 2023, Amazon Web Services India Private Limited or its affiliates., Privacy, Terms, and Cookie preferences.

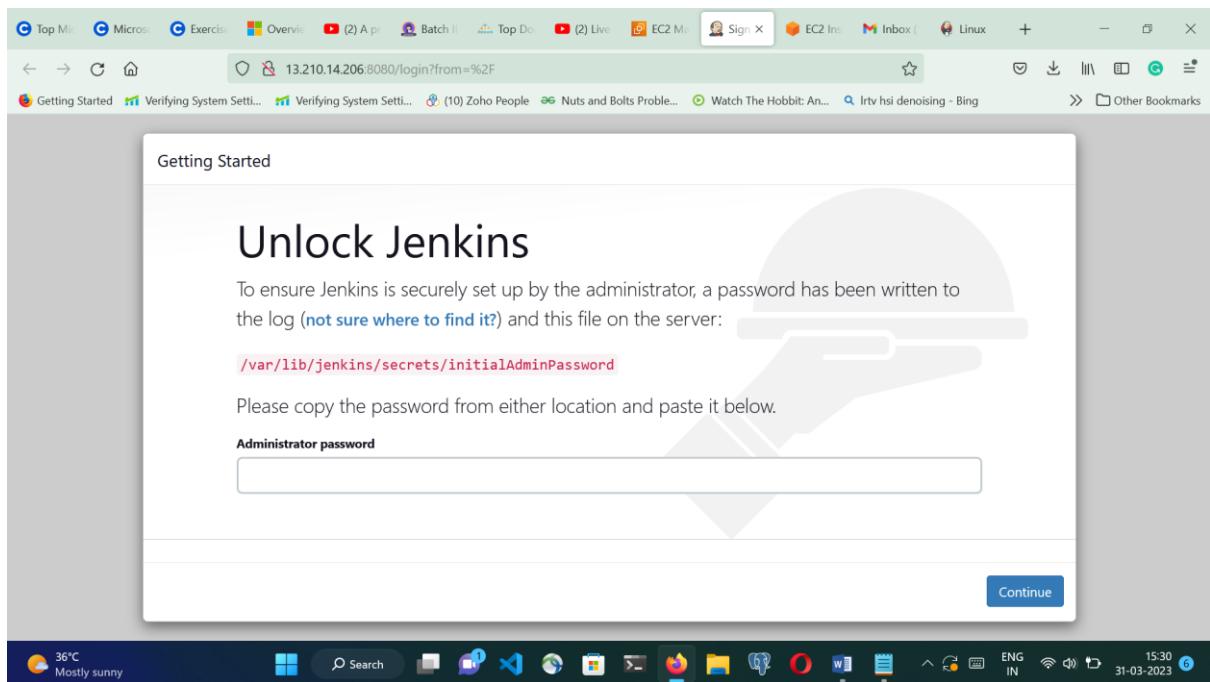
Screenshot of the AWS EC2 Security Groups page showing inbound rules for three security groups:

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-0dee7c7ecd433ecc1	HTTP	TCP	80	Custom 0.0.0.0/0	Delete
sgr-00960628cbc9f7ef9	SSH	TCP	22	Custom 0.0.0.0/0	Delete
sgr-0214f5c8d204b1f01	HTTPS	TCP	443	Custom 0.0.0.0/0	Delete
-	Custom TCP	TCP	8080	My IP 115.99.194.30/32	Add rule

**Add rule** button is visible at the bottom left.

#### **Step-4. Check the PORT is opened and working for Jenkins**

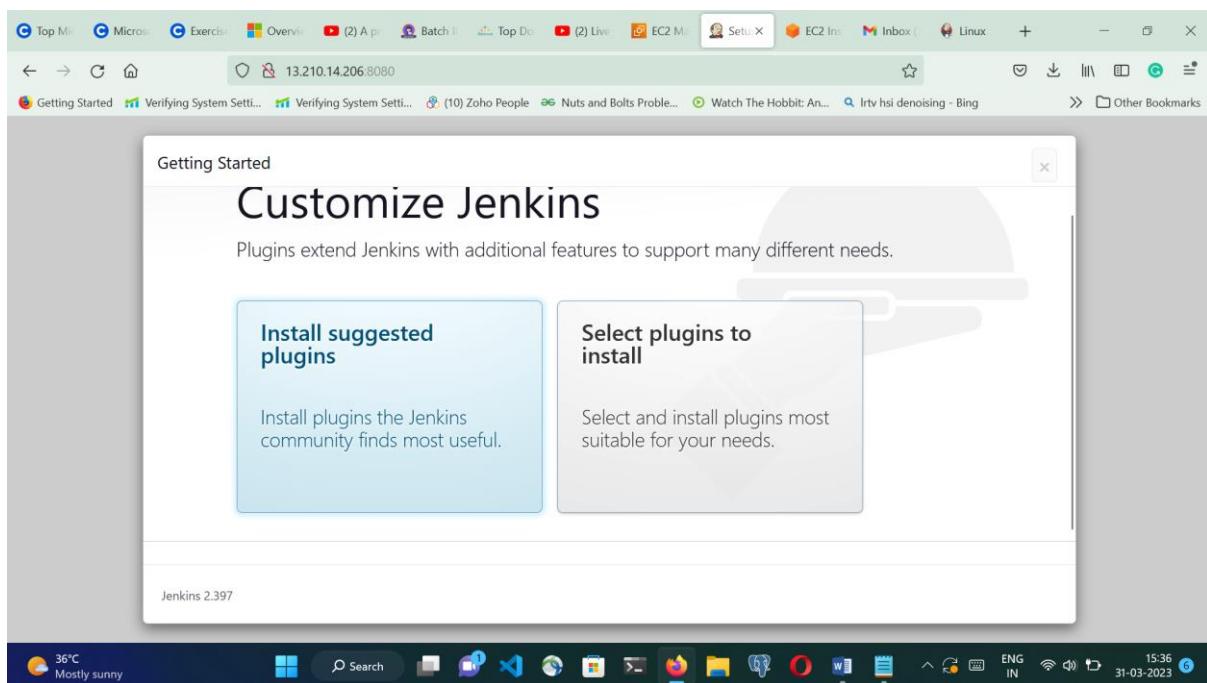
Use URL – “VM-IP:8080” (e.g – “13.210.14.206:8080”)

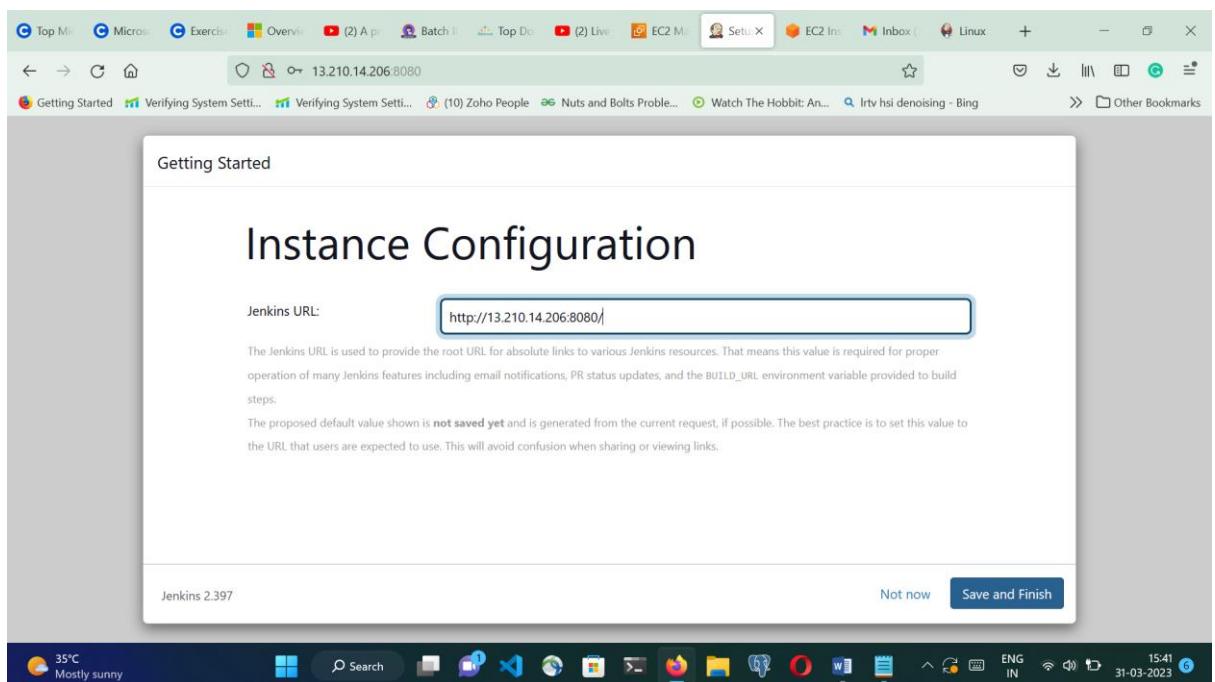
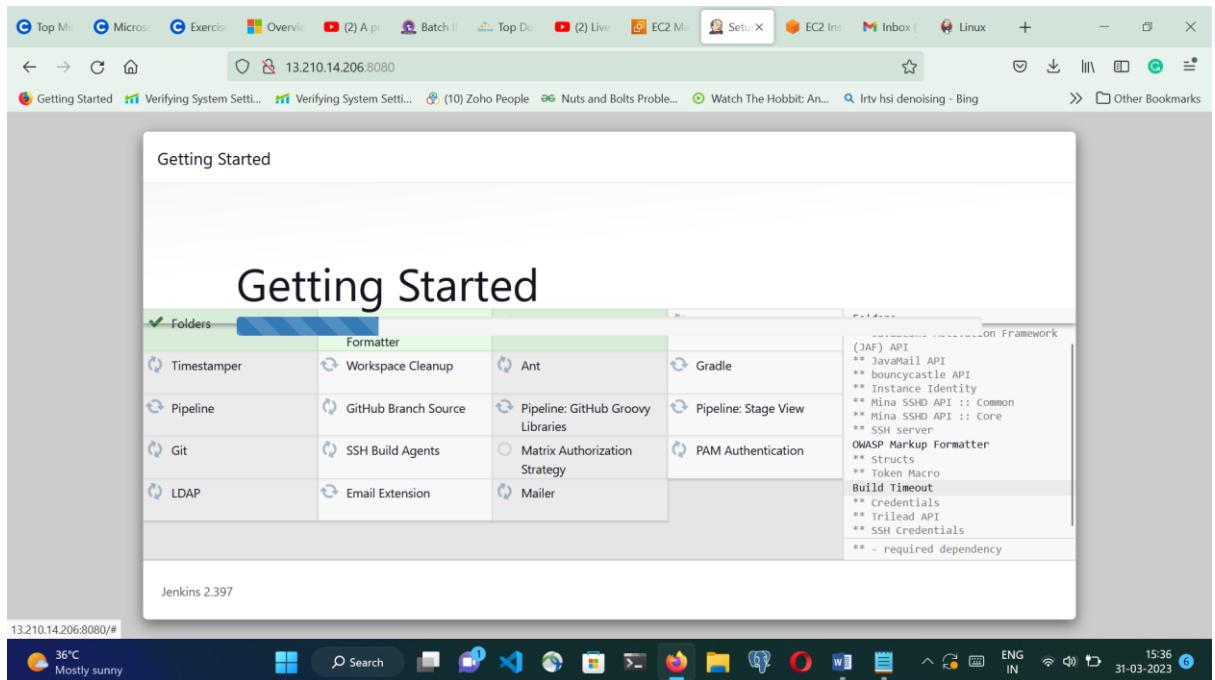


## Step-5. Get password using below command

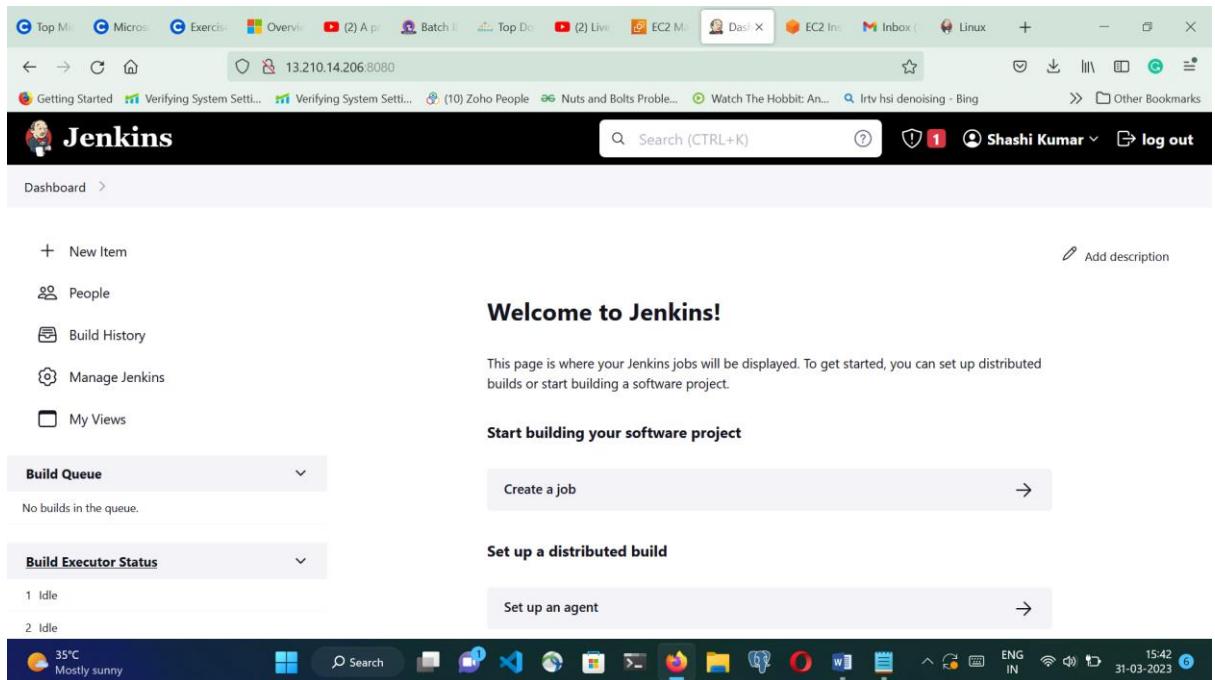
```
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

Use this password to login and install plugins.





Open Jenkins in web browser-



## Step-6. Create CI/CD pipeline and get code from GitHub

- From Jenkins page “create a new job”

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**

**Build Queue**

No builds in the queue.

**Create a job**

**Build Executor Status**

1 Idle

2 Idle

**Set up a distributed build**

Set up an agent

Configure a cloud

Learn more about distributed builds



Choose Free Style Project with name my-app or anything:

Enter an item name

my-app

» 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.

OK



**Configure**

**General**

Description  
My First App

Discard old builds

GitHub project

Project url  
`https://github.com/LondheShubham153/node-todo-cicd.git`

Save Apply

## GitHub want access to Jenkins server through SSH public and private key-

Command –

1. Go to Jenkins workspace directory :
- `cd /var/lib/jenkins/workspace/my-app`

## 2. ssh-keygen

```

Mar 31 09:21:38 ip-172-31-13-194 jenkins[5553]: fad8a1ba86604d41a56c81631a456c65
Mar 31 09:21:38 ip-172-31-13-194 jenkins[5553]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Mar 31 09:21:38 ip-172-31-13-194 jenkins[5553]: ****
Mar 31 09:21:38 ip-172-31-13-194 jenkins[5553]: ****
Mar 31 09:21:38 ip-172-31-13-194 jenkins[5553]: ****
Mar 31 09:22:14 ip-172-31-13-194 jenkins[5553]: 2023-03-31 09:22:14.632+0000 [id=28]      INFO      jenkins.InitReactorRunner$1#onAttained: C
>
Mar 31 09:22:14 ip-172-31-13-194 jenkins[5553]: 2023-03-31 09:22:14.657+0000 [id=22]      INFO      hudson.lifecycle.Lifecycle#onReady: Jenki
>
Mar 31 09:22:14 ip-172-31-13-194 systemd[1]: Started Jenkins Continuous Integration Server.
Mar 31 09:22:15 ip-172-31-13-194 jenkins[5553]: 2023-03-31 09:22:15.623+0000 [id=44]      INFO      h.m.DownloadService$Downloadable#load: Ob
>
Mar 31 09:22:15 ip-172-31-13-194 jenkins[5553]: 2023-03-31 09:22:15.628+0000 [id=44]      INFO      hudson.util.Retrier#start: Performed the
>

ubuntu@ip-172-31-13-194:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
fad8a1ba86604d41a56c81631a456c65
ubuntu@ip-172-31-13-194:~$ ssh-keygen
i-04576aab24f7857f (jenkins-master)
Public IPs: 13.210.14.206 Private IPs: 172.31.13.194

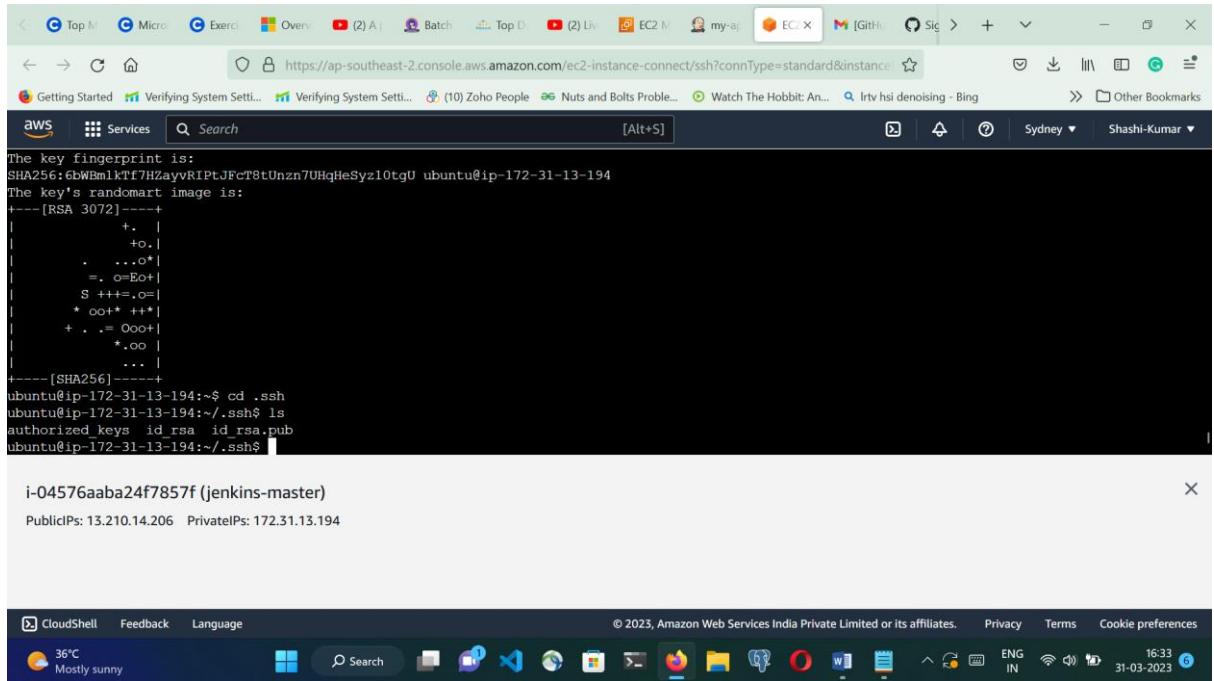
```

CloudShell   Feedback   Language   © 2023, Amazon Web Services India Private Limited or its affiliates.   Privacy   Terms   Cookie preferences

36°C   Mostly sunny   16:30   31-03-2023

3. cd .ssh

4. ls



```
The key fingerprint is:  
SHA256:6bWhmlkTf7HZayvRIPtJFcT8tUnz7UHQHeSyz10tg0 ubuntu@ip-172-31-13-194  
The key's randomart image is:  
+---[RSA 3072]---+  
| + . |  
| +o. |  
| . ...o*|  
| =. o=O+|  
| S +++=.o=|  
| * oo* ++*|  
| + .-= Ooo+|  
| *..oo |  
| ... |  
+---[SHA256]---+  
ubuntu@ip-172-31-13-194:~$ cd .ssh  
ubuntu@ip-172-31-13-194:~/ssh$ ls  
authorized_keys id_rsa id_rsa.pub  
ubuntu@ip-172-31-13-194:~/ssh$
```

i-04576aab24f7857f (jenkins-master)  
Public IPs: 13.210.14.206 Private IPs: 172.31.13.194

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences  
36°C Mostly sunny Search ENG IN 16:33 31-03-2023 6

id\_rsa → private key

id\_rsa.pub → public key

Check using cat command

cat id\_rsa and cat id\_rsa.pub

## Step-6.1 Now connect GitHub with Jenkins using public key

**Copy the public key and create new SSH key:**

The screenshot shows a GitHub repository page for 'node-todo-cicd/views/todo.ejs'. The code editor displays the contents of 'todo.ejs' with 37 lines and 881 bytes. A context menu is open on the right, with 'Settings' selected. Other options in the menu include 'Your repositories', 'Your projects', 'Your stars', 'Your gists', 'Your sponsors', 'Upgrade', 'Try Enterprise', 'Feature preview', 'Help', and 'Sign out'.

The screenshot shows the GitHub settings page for SSH keys. It lists an existing SSH key named 'jenkins-project' with the SHA256 fingerprint 'SHA256:6bwBm1kTf7HZayvRIPtJFcT8tUnzn7UHqHeSz10tgU'. The key was added on March 31, 2023, and has never been used. A 'Delete' button is available for this key. A 'New SSH key' button is located at the top right. On the left, there is a sidebar with links for Public profile, Account, Appearance, Accessibility, Notifications, Access, Billing and plans, Emails, Password and authentication, Sessions, SSH and GPG keys (which is currently selected), Organizations, and Moderation.

Click on new SSH key-

The screenshot shows a user profile for Shashi Kumar on GitHub. The left sidebar lists account settings like Public profile, Account, Appearance, Accessibility, and Notifications. The main area is titled 'SSH keys / Add new'. It has fields for 'Title' (set to 'jenkins-project') and 'Key type' (set to 'Authentication Key'). Below these is a large text input field with the placeholder 'copy the public key here...'. A tooltip 'copy the public key here...' is overlaid on this field. The bottom of the screen shows a Windows taskbar with various icons and system status.

## Step-7. Add GitHub credentials to Jenkins and verify

The screenshot shows a Jenkins job configuration page for 'my-app'. The left sidebar under 'Configure' includes tabs for General, Source Code Management, Build Triggers, Build Environment, Build Steps, and Post-build Actions. The 'General' tab is selected. It contains a 'Description' field with the text 'This is cicd project'. Below it is a 'GitHub project' section with a checked checkbox and a 'Project url' field containing 'https://github.com/shashiipt19/node-todo-cicd.git'. A 'Save' button is at the bottom. A tooltip 'copy the public key here...' is overlaid on the 'Key' input field in the previous screenshot. The bottom of the screen shows a Windows taskbar with various icons and system status.

The screenshot shows the Jenkins configuration interface for a job named "my-app". The "Source Code Management" section is selected, displaying a "Repository URL" input field containing "https://github.com/shashiipt19/node-todo-cicd.git" and a dropdown menu for "Jenkins Credentials Provider" set to "Jenkins".

**Configure**

**Source Code Management**

General

Build Triggers

Build Environment

Build Steps

Post-build Actions

Repository URL ?  
https://github.com/shashiipt19/node-todo-cicd.git

Jenkins Credentials Provider  
Jenkins

Save Apply

**Configure 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)

Save Apply

The screenshot shows two consecutive screenshots of the Jenkins configuration interface for a job named "my-app".

**Screenshot 1:** The "Source Code Management" section is displayed. The "Scope" dropdown is set to "Global (Jenkins, nodes, items, all child items, etc)". The "ID" field contains "github-jenkins", the "Description" field contains "integrating github with jenkins", and the "Username" field contains "ubuntu". The "Treat username as secret" checkbox is unchecked. Buttons for "Save" and "Apply" are visible at the bottom.

**Screenshot 2:** The "Source Code Management" section is displayed again, but the "Key" field under "Private Key" is now populated with a long string of characters, and the "Passphrase" field is empty. The "Treat username as secret" checkbox is also unchecked. Buttons for "Save" and "Apply" are visible at the bottom.

```

-----BEGIN OPENSSH PRIVATE KEY-----
b3B1brNzaClrZXktdjEAAAABG5bmUAAAEBm9uZQAAAAAAAAAAABlwAAAAdzc2gtcn
NhAAAIAwEAAQAAAYEAtd558WXYzOhaDsp7011T0aV6CW9AUlclqrjoc-swbhKzoPmcsg9d
wmteIW/aZGnrW1zayFps7cdpVdWSNXsmFeAnVg0yHrmSYql8eo9Mc5SH3SRxqHmZxIG
ZL0eyBioymrPy4hm9CEreILRhyP3w5Iai533/nRjotuVy/W2Bn5yuXdwSqCapPHIRvTDL7
DaQvZaGRV3mp1CsFz+ijJpaK3hAWM91DokJz/kx5IK3y22CEn4bctQ4019w7K7zqJa8
s49XCYcpBfuA/lpmCDR2eIys/f8MzjPhQ2JijRYykzBJFLUAUujPyjzG0tvyl3uNVCD
zAe/Xjw+tWuKrim+jWd6mBMZiP8uj0rqDQHCh/Kzwym2H0NTg8fcjw8K1XHVBBdpRxFS1
boZH2202cHCwTBceESF2u43gS04aQwrP1SLbJeygs0pRVb+2qUtdfvcBrcjs+bpdP6GU
92YVF2ceh6h0VQVs6D111LL62Bwoh/1973M0oin9AAAFKLKWAmaylgDGAAAAAB3NzaC1yc2
EAAAGBALXeeff12mszoWg7KeziNU96leg1vQFC3jaq46PrMc4Sma05nLPxcrX1Fv2vGR
p61tc2shT703A6Vm1kjV7jhXgj72INMh02kmKnZfOgPTHOuh90kcUR5mcSBms9HsgSKMpq
z8u1ZvKhK3iCOYcj98Ps2oud9+50Y6l6blcv1tmzeclr3VkgggKzxyEb0wy+w2kLWhkVd5
p6ZQrbC/o8yT2gCt4QFjPdQ6uCc/5MeSt8ttghJ+G3IJUOnJfcOyu2aiWviOPVwmHKQX7
-----END OPENSSH PRIVATE KEY-----

```

i-04576aaba24f7857f (jenkins-master)

PublicIPs: 13.210.14.206 PrivateIPs: 172.31.13.194

**Add this private key to jenkins.**

The screenshot shows two consecutive screenshots of the Jenkins configuration interface for a job named "my-app".

**Screenshot 1: Private Key Configuration**

The "Source Code Management" section is selected in the sidebar. The "Private Key" tab is active, showing a "None" radio button. The "Enter directly" option is selected, and a text input field contains the following OpenSSH private key:

```
-----BEGIN OPENSSH PRIVATE KEY-----  
b3B1bnzaC1rZXktdjEAAAAABG5vbmJAAAEbm9uZQAAAAAAAAABAAIBwAAAAdzc2gtcn  
NhAAAAAwEAAQAAAYEAtd558wXYZzOhaDsp7OI1T0aV6Cw9AULcIarjoc+swbhKzoPmc9d
```

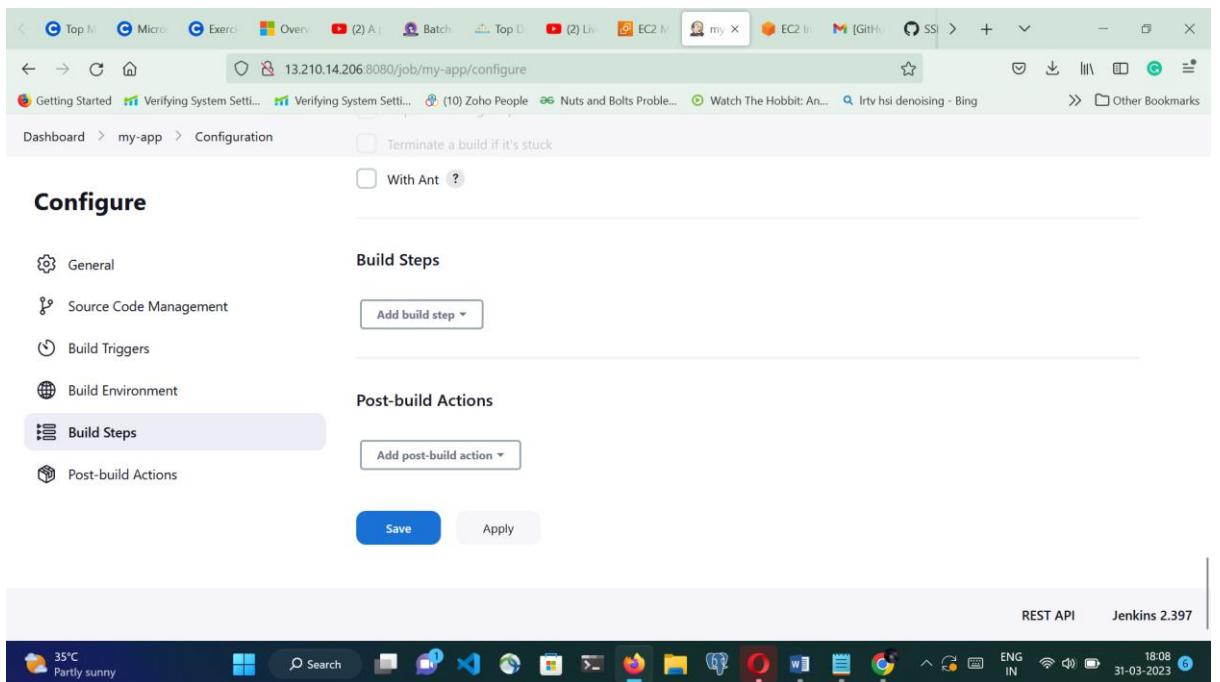
A red circle with the number "1" is positioned next to the text input field.

**Screenshot 2: Source Code Management Configuration**

The "Source Code Management" section is selected in the sidebar. The "Git" tab is active, indicated by a blue dot. The "Repositories" section is expanded, showing a "Repository URL" input field containing "https://github.com/shashiitp19/node-todo-cicd.git" and a "Credentials" dropdown menu set to "ubuntu (integrating github with jenkins)".

A red circle with the number "1" is positioned next to the Repository URL input field.

Both screenshots show a standard Windows taskbar at the bottom with various application icons and system status indicators.



- Click on build now to create the build and if build is successful then jenkins has verified the credentials and we can proceed.

The screenshot shows a Jenkins project page for 'my-app'. The top navigation bar includes links for 'Dashboard', 'my-app', 'Status', 'Changes', 'Workspace', 'Build Now', 'Configure', 'Delete Project', 'GitHub', and 'Rename'. A 'Permalinks' section lists recent builds: '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', and 'Last completed build (#1), 1 min 4 sec ago'. The bottom of the page features a 'Build History' section with a table for build #1, showing details like 'Mar 31, 2023, 12:38 PM', 'Atom feed for all', and 'Atom feed for failures'. The status bar at the bottom indicates '35°C Partly sunny' and the date '31-03-2023'.

Started by user **Shashi Kumar**  
Running as **SYSTEM**  
The recommended git tool is: **NONE**  
using credential **github-jenkins**  
Cloning the remote Git repository  
Cloning repository <https://github.com/shashiiotp19/node-todo-cicd.git>  
> git init /var/lib/jenkins/workspace/my-app # timeout=10  
Fetching upstream changes from <https://github.com/shashiiotp19/node-todo-cicd.git>  
> git --version # timeout=10  
> git --version # 'git version 2.34.1'  
using **GIT\_SSH** to set credentials integrating github with jenkins  
Verifying host key using known hosts file  
You're using 'Known hosts file' strategy to verify ssh host keys, but your `known_hosts` file does not exist, please go to 'Manage Jenkins' -> 'Configure Global Security' -> 'Git Host Key Verification Configuration' and configure

## Build has been created.

```
Aw+TxIgyQHoeYyI1xVDRKR+UnjJ1aTiv8RI/4c16l01SLOk1vLf4b2Gt15zmgnPnMLAZAAAA
wA4+Z11Z1Vsjf2BTG0/4Cfxql147IMMoeoJy0LwsCKZE892o/fYpi4Ck1cydFyQHmRdnauU
zzcA7+kaFF5QnHB1oCg9XNNEV1dEz2UcyOh3PSC3+bysoMOpqwedNfIZ6E6Q1y0fV4Zq
fmoBkZKUP330n694w6KnMujPsfqlLvv0caffVogcjjox9A97BCE18Z1p3D1dZhblxkZU1B
yev3eighldj+/-sppqDBFS1x+sAbv2EVnVAxTL8k1otGAAAMAEazF+qDNBCZk8L5)EO
11wwFuds875jVGTeetVP063r1G6higGLJxHwnn1ak1J10ExxF/4v/QkQgsxu8cIo4ap2E
0qoub3BUW6SQqyE4A3MwajJpbINS52v1JRoY8EwKyEVSoAkW74sj)CdzzV8llRbhNTOP
HBdSdnbpjMnedY6H7GPxFEpvdvnnaYneqnoLq/90GceXvab09L1H01311pTjgmvOK8i
Nd4P1GhOZdpJvwWr3a6W31ax7voSjAAAeWQbjz35E185ibXAKz0@ccuqfKAMA8QL3x5X
hIpDc2NCvz140VzQGLiUNCBgksRmbgApqzKhOB1++/rg0koYyAfpXMj/+0oudkT9Pe2b
nmMF7mYsEMynF4/90UGt79Hwv3Hpvz24Yadcx94TBg0ayYni5wOmSy935Xx/YvQJ1Zrq
AU/ZeyQidpw1/tZHiySA27bERV+AnVs39GBhAM9DKZQLTxW8ifxjMoAvUmXlk/6Xgi1w0
eCu8RtisBwSN0AAAxdW1bn1QG1wLTEm102MS0xMy0xOTQBqME
-----END OPENSSH PRIVATE KEY-----
ubuntu@ip-172-31-13-194:~/.ssh$ cd /var/lib/jenkins/workspace/my-app
ubuntu@ip-172-31-13-194:/var/lib/jenkins/workspace/my-app$ ls
Dockerfile Jenkinsfile README.md app.js docker-compose.yaml package-lock.json package.json test.js views
ubuntu@ip-172-31-13-194:/var/lib/jenkins/workspace/my-app$
```

i-04576aaba24f7857f (jenkins-master)  
Public IPs: 13.210.14.206 Private IPs: 172.31.13.194

**Important – Now we want to run our app on server**

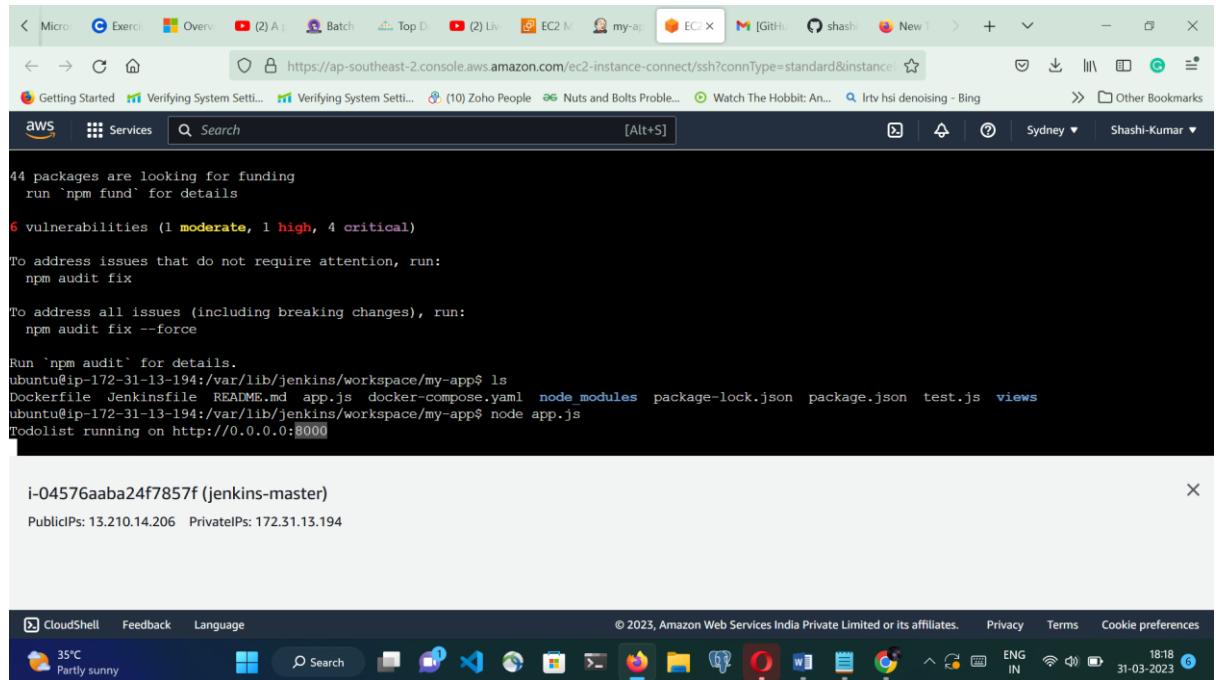
**We will explore 3 ways to run our app.**

### 1. Directly from our VM :

- So here basically we have to install all the dependencies as mentioned in `readme.md`

## Follow README.md

```
sudo apt install nodejs  
sudo apt install npm  
npm install (installs dependencies from package.json file)  
node app.js (now this will run the app on server)
```



```
44 packages are looking for funding
  run 'npm fund' for details

6 vulnerabilities (1 moderate, 1 high, 4 critical)

To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
ubuntu@ip-172-31-13-194:/var/lib/jenkins/workspace/my-app$ ls
Dockerfile Jenkinsfile README.md app.js docker-compose.yaml node_modules package-lock.json package.json test.js views
ubuntu@ip-172-31-13-194:/var/lib/jenkins/workspace/my-app$ node app.js
Todolist running on http://0.0.0.0:8000

i-04576aaba24f7857f (jenkins-master)
PublicIPs: 13.210.14.206 PrivateIPs: 172.31.13.194
```

- App is running on port 8000, so open this port from ec2 security groups.

The screenshot shows the AWS CloudWatch Metrics interface with several metrics listed:

- CloudWatch Metrics
- CloudWatch Metrics Insights
- CloudWatch Metrics Insights Analytics
- CloudWatch Metrics Insights Metrics
- CloudWatch Metrics Insights Metrics Insights
- CloudWatch Metrics Insights Metrics Insights Insights
- CloudWatch Metrics Insights Metrics Insights Insights Insights
- CloudWatch Metrics Insights Metrics Insights Insights Insights Insights
- CloudWatch Metrics Insights Metrics Insights Insights Insights Insights Insights
- CloudWatch Metrics Insights Metrics Insights Insights Insights Insights Insights Insights
- CloudWatch Metrics Insights Metrics Insights Insights Insights Insights Insights Insights Insights
- CloudWatch Metrics Insights Metrics Insights Insights Insights Insights Insights Insights Insights Insights
- CloudWatch Metrics Insights Metrics Insights Metrics Insights Metrics Metrics Metrics Metrics

See if our app is running -

The screenshot shows a browser window displaying the URL `13.210.14.206:8000/todo`. The page content is:

**GNA Students are Super Duper Awesome**

What should I do?  | [Add](#)

Note: Now if we terminate our terminal by Ctrl + C, the app stop running.

```

44 packages are looking for funding
  run `npm fund` for details

6 vulnerabilities (1 moderate, 1 high, 4 critical)

To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
ubuntu@ip-172-31-13-194:/var/lib/jenkins/workspace/my-app$ ls
Dockerfile Jenkinsfile README.md app.js docker-compose.yaml node_modules package-lock.json package.json test.js views
ubuntu@ip-172-31-13-194:/var/lib/jenkins/workspace/my-app$ node app.js
Todolist running on http://0.0.0.0:8000
^C
ubuntu@ip-172-31-13-194:/var/lib/jenkins/workspace/my-app$ 

i-04576aaba24f7857f (jenkins-master)
Public IPs: 13.210.14.206 Private IPs: 172.31.13.194

```

Firefox can't establish a connection to the server at 13.210.14.206:8000.

- The site could be temporarily unavailable or too busy. Try again in a few moments.
- If you are unable to load any pages, check your computer's network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Firefox is permitted to access the Web.

[Try Again](#)



To solve this issue we will use containers approach.

## 2. Create A Docker Container :

Automate all the steps through Docker -

“Remove the already provided docker”

### - Install docker

1. sudo apt install docker.io

2. sudo vi Dockerfile, press i,
3. Add below command lines in Dockerfile
4. For exit press 'Esc' key then press ':' key and wq and hit Enter.

```
FROM node:12.2.0-alpine
```

```
COPY package.json package.json
```

```
WORKDIR app
```

```
COPY ..
```

```
RUN npm install
```

```
EXPOSE 8000
```

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

```

< Overview (2) A Batch Top D (2) Liv EC2 M EC2 In my-ap EC2 In [GitHub] shashi Problem Images > + - ×
← → ⌂ ⌂ https://ap-southeast-2.console.aws.amazon.com/ec2-instance-connect/ssh?region=ap-southeast-2&conn=
Getting Started Verifying System Setti... Verifying System Setti... (10) Zoho People Nuts and Bolts Proble... Watch The Hobbit: An... lrtv hsi denoising - Bing > Other Bookmarks
aws Services Search [Alt+S]
No containers need to be restarted.
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-13-194:/var/lib/jenkins/workspace/my-app$ sudo vim Dockerfile
[1]+ Stopped sudo vim Dockerfile
ubuntu@ip-172-31-13-194:/var/lib/jenkins/workspace/my-app$ sudo vim Dockerfile
ubuntu@ip-172-31-13-194:/var/lib/jenkins/workspace/my-app$ sudo vim Dockerfile
ubuntu@ip-172-31-13-194:/var/lib/jenkins/workspace/my-app$ cat Dockerfile
FROM node:12.2.0-alpine
WORKDIR app
COPY . .
RUN npm install
CMD ["node","app.js"]
ubuntu@ip-172-31-13-194:/var/lib/jenkins/workspace/my-app$ i-04576aaba24f7857f (jenkins-master)
PublicIPs: 13.210.14.206 PrivateIPs: 172.31.13.194

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences
35°C Partly sunny 31-03-2023 19:28 ENG IN

```

## Now give permission to the docker user

- sudo usermod -aG docker \$USER

The screenshot shows an AWS CloudShell interface. At the top, there's a navigation bar with links like 'Exercises', 'Overview', 'Batch', 'Top D...', 'Live', 'Connect', 'EC2 X', 'EC2 In', 'my-app', 'GitHub', 'node', 'My sha...', and a search bar. Below the navigation bar is a toolbar with icons for 'Getting Started', 'Verifying System Setti...', '(10) Zoho People', 'Nuts and Bolts Proble...', 'Watch The Hobbit: An...', 'Irvin Hsi denoising - Bing', and 'Other Bookmarks'. The main area is a terminal window titled 'clean room' with the AWS logo. It displays a Dockerfile content:

```
FROM node:12.2.0-alpine
COPY package.json package.json
WORKDIR app
COPY . .
RUN npm install
EXPOSE 8000
CMD ["node", "app.js"]
```

Below the Dockerfile, the terminal shows the build process for a container named 'i-04576aaba24f7857f (jenkins-master)'. It lists public and private IP addresses: 'PublicIPs: 13.210.14.206 PrivateIPs: 172.31.13.194'. The bottom of the terminal shows system status and a timestamp: '14:38 01-04-2023'.

## Now Build docker

- sudo docker build . -t new-app

The screenshot shows an AWS CloudShell interface. The terminal window at the top has the title 'clean room' and shows the command 'sudo docker build . -t my-app' being run. The output of the build process is displayed, showing the steps taken by the Docker daemon to build the image, including copying files, setting the working directory, installing dependencies, and exposing port 8000. The build is successful, resulting in a container named 'i-04576aaba24f7857f (jenkins-master)' with the same public and private IP addresses as before. The bottom of the terminal shows system status and a timestamp: '14:41 01-04-2023'.

## Now run docker

- docker system prune – “to delete unused containers”
- docker run -it --rm -d -p 8000:8000 --name new-app new-app

```

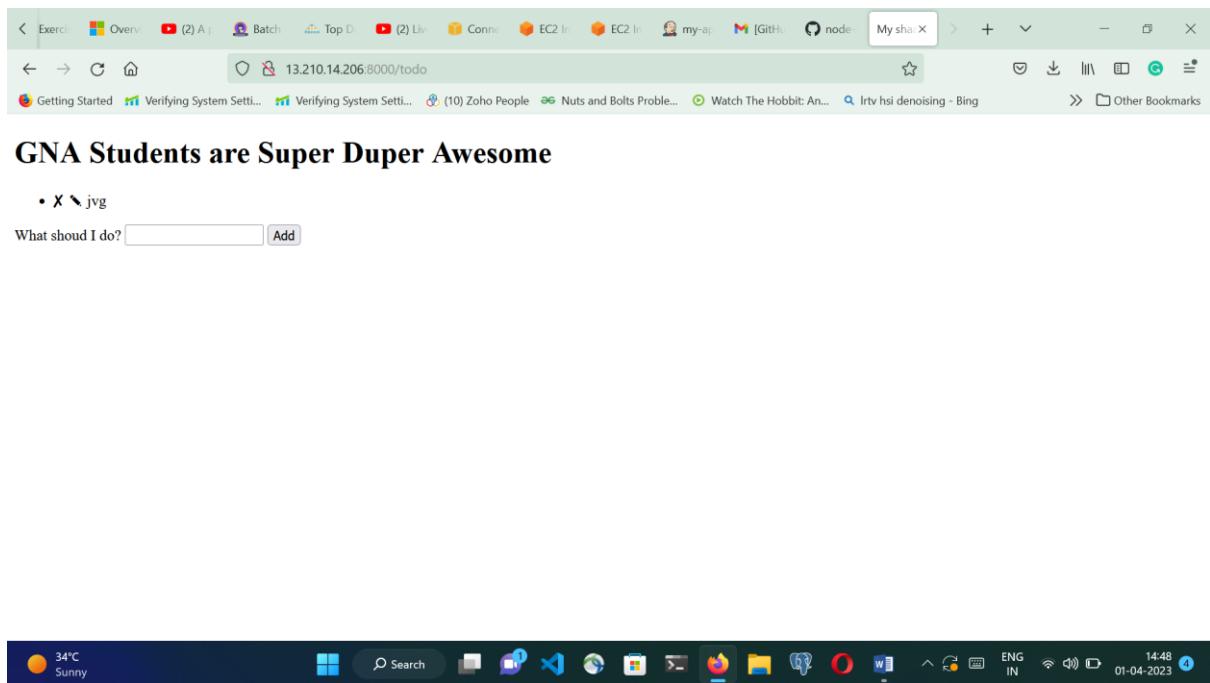
ubuntu@ip-172-31-13-194:/var/lib/jenkins/workspace/my-app$ docker run -it --rm -d -p 8000:8000 --name todo-app my-app
ecd025044d9a1f9e12324ac11997b35a7d7ef24eaff16f00d32dd3442abda72
ubuntu@ip-172-31-13-194:/var/lib/jenkins/workspace/my-app$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
ecd025044d9a my-app "node app.js" 18 seconds ago Up 18 seconds 0.0.0.0:8000->8000/tcp, :::8000->8000/tcp todo-app
ubuntu@ip-172-31-13-194:/var/lib/jenkins/workspace/my-app$ 

```

i-04576aaba24f7857f (jenkins-master)  
PublicIPs: 13.210.14.206 PrivateIPs: 172.31.13.194

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences 34°C Sunny 14:43 01-04-2023

App is running on 8000



Now kill the docker that we have used and app stops running.

### 3. Run the app through jenkins

**Now give permission to the jenkins**

- sudo chmod 777 /var/lib/jenkins/workspace/my-app
- sudo usermod -a -G docker Jenkins
- sudo service jenkins restart

## Build steps in configuration on Jenkins page-

The screenshot shows the Jenkins configuration interface for a job named "my-app". On the left, a sidebar lists "Configure" options: General, Source Code Management, Build Triggers, Build Environment, **Build Steps**, and Post-build Actions. The "Build Steps" section is currently selected. A detailed view of the "Execute shell" step is shown, containing the command:

```
docker build . -t new-app-2
docker run -it --rm -d -p 8000:8000 --name new-app new-app-2
```

Below the command, there is an "Advanced" dropdown menu. At the bottom of the configuration panel are "Save" and "Apply" buttons.

The screenshot shows the Jenkins project page for "my-app". The top navigation bar includes links for Overview, Batch, Top D, Live, Connect, EC2, my, GitHub, node, My shaan, How, Getting Started, Verifying System Setti..., Zoho People, Nuts and Bolts Proble..., Watch The Hobbit: An..., Irtv hsi denoising - Bing, and Other Bookmarks. The user "Shashi Kumar" is logged in.

The main content area displays the "Project my-app" page. On the left, a sidebar provides links for Status, Changes, Workspace, Build Now, Configure, Delete Project, GitHub, and Rename. The "Build Now" button is highlighted. To the right, the "Permalinks" section lists recent builds:

- Last build (#8), 3 min 46 sec ago
- Last stable build (#8), 3 min 46 sec ago
- Last successful build (#8), 3 min 46 sec ago
- Last failed build (#7), 7 min 39 sec ago
- Last unsuccessful build (#7), 7 min 39 sec ago
- Last completed build (#8), 3 min 46 sec ago

At the bottom of the page, there is a "Build History" section with a "trend" dropdown set to "delay=0sec". The footer of the browser window shows the same navigation bar and system status as the top.

The screenshot shows the Jenkins Build History page for the job 'my-app'. The top navigation bar includes links for Overview, Batch, Top, Live, Connect, EC2, GitHub, node-webkit, My sha..., How To, and Other Bookmarks. The main content displays a summary of recent builds:

- Last build (#8), 3 min 46 sec ago
- Last stable build (#8), 3 min 46 sec ago
- Last successful build (#8), 3 min 46 sec ago
- Last failed build (#7), 7 min 39 sec ago
- Last unsuccessful build (#7), 7 min 39 sec ago
- Last completed build (#8), 3 min 46 sec ago

The 'Build History' section lists individual build logs with their numbers and timestamps:

- #8 Apr 1, 2023, 12:05 PM
- #7 Apr 1, 2023, 12:01 PM
- #6 Apr 1, 2023, 12:01 PM
- #5 Apr 1, 2023, 11:59 AM
- #4 Apr 1, 2023, 11:57 AM
- #3 Apr 1, 2023, 11:56 AM
- #1 Mar 31, 2023, 12:38 PM

Links for Atom feed for all and Atom feed for failures are provided at the bottom.

At the bottom right, there are links for REST API and Jenkins 2.397.

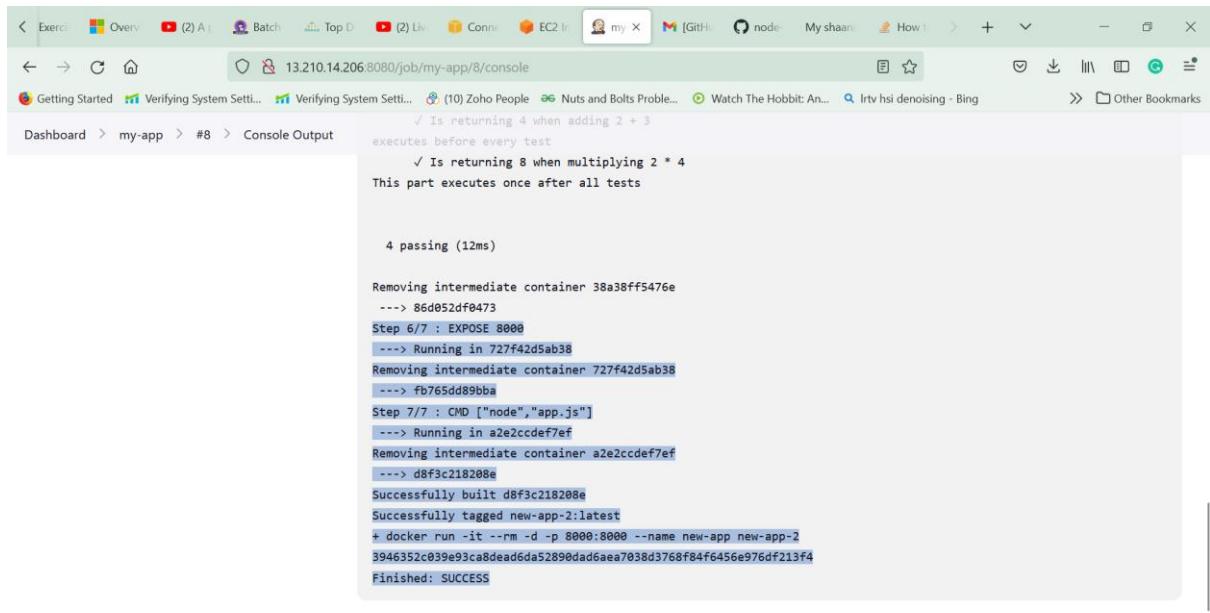
The screenshot shows the Jenkins Console Output page for build #8 of the 'my-app' job. The top navigation bar is identical to the previous screenshot. The main content area is titled 'Console Output' and shows the build log:

```
Started by user Shashi Kumar
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/my-app
The recommended git tool is: NONE
using credential github-jenkins
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/my-app/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/shashiiitp19/node-todo-cicd.git # timeout=10
Fetching upstream changes from https://github.com/shashiiitp19/node-todo-cicd.git
> git --version # timeout=10
> git --version # 'git version 2.34.1'
using GIT_SSH to set credentials integrating github with jenkins
Verifying host key using known hosts file
You're using 'Known hosts file' strategy to verify ssh host keys, but your known_hosts file does not exist, please
go to 'Manage Jenkins' -> 'Configure Global Security' -> 'Git Host Key Verification Configuration' and configure
```

The left sidebar provides navigation options:

- Status
- </> Changes
- Console Output (selected)
- View as plain text
- Edit Build Information
- Delete build '#8'
- Git Build Data
- ← Previous Build

At the bottom right, there are links for ENG IN, 17:40, and 01-04-2023.



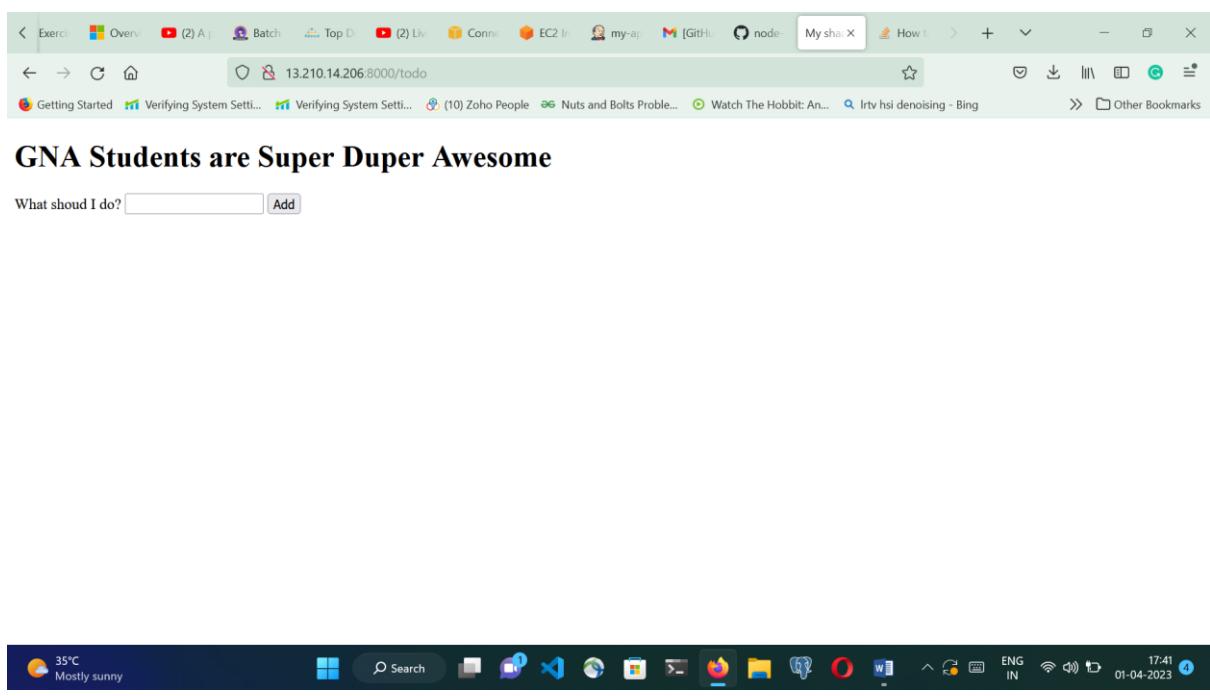
```
✓ Is returning 4 when adding 2 + 3
executes before every test
✓ Is returning 8 when multiplying 2 * 4
This part executes once after all tests

4 passing (12ms)

Removing intermediate container 38a38ff5476e
--> 86d052df0473
Step 6/7 : EXPOSE 8000
--> Running in 727f42d5ab38
Removing intermediate container 727f42d5ab38
--> fb765dd89bba
Step 7/7 : CMD ["node","app.js"]
--> Running in a2e2ccdef7ef
Removing intermediate container a2e2ccdef7ef
--> d8f3c218208e
Successfully built d8f3c218208e
Successfully tagged new-app-2:latest
+ docker run -it --rm -p 8000:8000 --name new-app new-app-2
3946352c039e93ca8dead6da52890dad6aea7038d3768f84f6456e976df213f4
Finished: SUCCESS
```

Build successfully created by Jenkins.

Now our app will be running again.



GNA Students are Super Duper Awesome

**Important: Till now we were manually doing Build Now in Jenkins**

**But how to automate this to achieve Continuous Deployment**

## - using webhooks

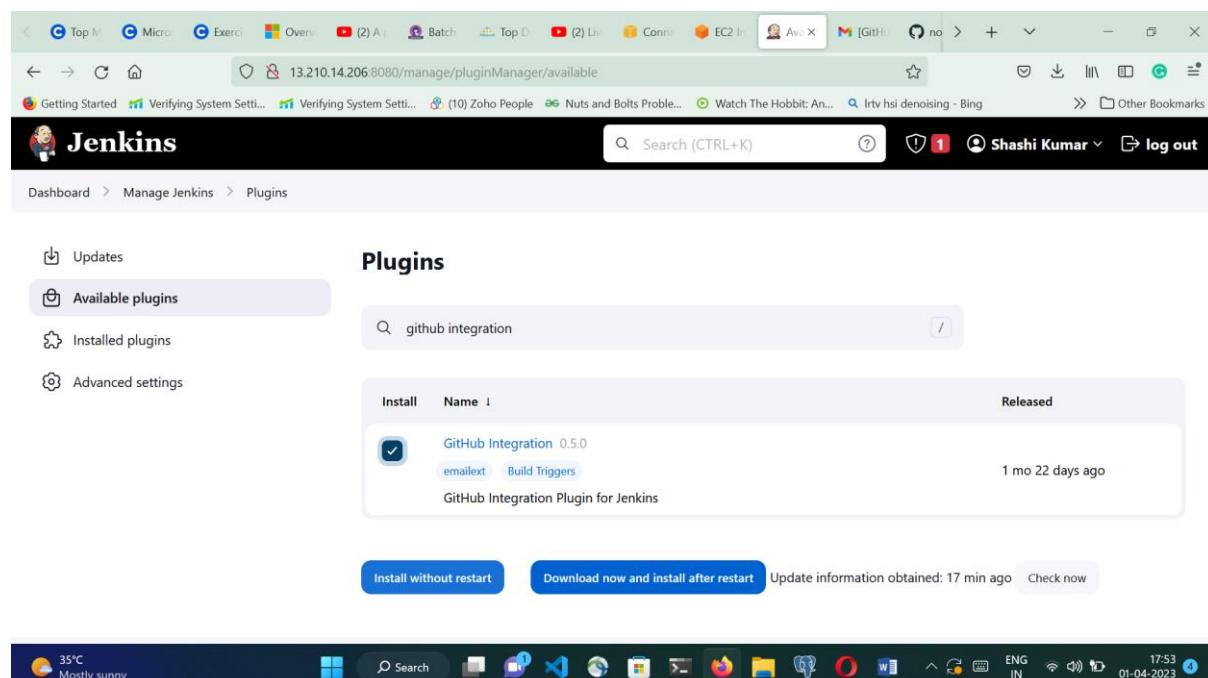
Webhooks - A webhook is an HTTP-based callback function that allows lightweight, event-driven communication between 2 [application programming interfaces \(APIs\)](#). Webhooks are used by a wide variety of web apps to receive small amounts of data from other apps, but webhooks can also be used to trigger automation workflows in [GitOps](#) environments.

Or

A webhook can be thought of as a type of API that is driven by events rather than requests. Instead of one application making a request to another to receive a response, a webhook is a service that allows one program to send data to another as soon as a particular event takes place. Webhooks are sometimes referred to as “reverse APIs,” because communication is initiated by the application sending the data rather than the one receiving it. With web services becoming increasingly interconnected, webhooks are seeing more action as a lightweight solution for enabling real-time notifications and data updates without the need to develop a full-scale API.

## **Step-1. Install GitHub Integration from Jenkins page plugins:**

This integrates GitHub with Jenkins.



The screenshot shows the Jenkins plugin manager interface. The URL in the browser is 13.210.14.206:8080/manage/pluginManager/available. The Jenkins logo is at the top left. The user is logged in as Shashi Kumar. The navigation bar shows 'Dashboard' > 'Manage Jenkins' > 'Plugins'. On the left, there are tabs for 'Updates', 'Available plugins' (which is selected), 'Installed plugins', and 'Advanced settings'. In the center, a search bar contains 'github integration'. Below it, a table lists the plugin details:

Install	Name	Released
<input checked="" type="checkbox"/>	GitHub Integration 0.5.0 emailtext   Build Triggers	1 mo 22 days ago
GitHub Integration Plugin for Jenkins		

At the bottom, there are two buttons: 'Install without restart' and 'Download now and install after restart'. A status message says 'Update information obtained: 17 min ago' and a 'Check now' link. The system tray at the bottom shows a weather icon (35°C, Mostly sunny), a search icon, and various application icons like File Explorer, Task Manager, and Edge. Language and date/time settings are also visible.



## Step-2. Add webhooks on GitHub project repo

Create a Webhook:

- Copy the url of Jenkins till port number (8080).
- Add /github-webhooks
- Looks like this <http://13.210.14.206:8080/github-webhook>
- Add this to github repo settings under create webhook



The screenshot shows a Microsoft Edge browser window with the GitHub URL <https://github.com/shashiipt19/node-todo-cicd/settings/hooks/new>. The page title is "shashiipt19 / node-todo-cicd". The "Settings" tab is active. On the left, a sidebar lists "General", "Access", "Collaborators", "Moderation options", "Code and automation" (with "Branches", "Tags", and "Actions" listed), and "Webhooks" (which is currently selected). The main content area is titled "Webhooks / Add webhook" and contains instructions about sending POST requests to the specified URL. The "Payload URL" input field contains `http://13.210.14.206:8080/github-webhook/`. The "Content type" dropdown menu is open, showing "application/json" as the selected option, along with "application/x-www-form-urlencoded". The status bar at the bottom shows the date as 01-04-2023.

This screenshot is identical to the one above, showing the same GitHub page and configuration. The difference is in the "Content type" dropdown, where "application/x-www-form-urlencoded" is now selected instead of "application/json". The rest of the interface and status bar remain the same.

## Edit port 8080 to the world from ec2-instance security groups:

- So that anybody can access our app

The screenshot shows the AWS EC2 Inbound Security Rules configuration page. A 'Custom' security group is selected. Five rules are listed:

- Rule 1: SSH (TCP port 22) from 0.0.0.0/0
- Rule 2: Custom TCP (TCP port 8080) from Anywhere-IPv4 (0.0.0.0/0)
- Rule 3: HTTPS (TCP port 443) from Anywhere-IPv4 (0.0.0.0/0)
- Rule 4: Custom TCP (TCP port 8000) from My IP (0.0.0.0/0)
- Rule 5: Custom TCP (TCP port 8000) from 0.0.0.0/0

An 'Add rule' button is at the bottom left.

Verify if Webhook is successfully created.

The screenshot shows the GitHub repository settings page for 'node-todo-cicd'. The 'Webhooks' tab is selected. A single webhook is listed:

- URL: http://13.210.14.206:8080/github-w... (push)
- Buttons: Edit, Delete

The sidebar shows other repository settings like General, Access, Collaborators, and Moderation options.

Now go to Jenkins app configuration and Build Triggers:

Dashboard > my-app > Configuration

## Configure

### 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 ?
- Poll SCM ?

### Build Environment

Delete workspace before build starts

Save Apply

33°C Mostly sunny

Dashboard > my-app >

## Project my-app

This is cicd project

Changes  Workspace  Build Now  Configure  Delete Project  GitHub Hook Log  GitHub  Rename

Edit description  Disable Project

### Permalinks

- Last build (#8), 1 hr 29 min ago
- Last stable build (#8), 1 hr 29 min ago
- Last successful build (#8), 1 hr 29 min ago
- Last failed build (#7), 1 hr 32 min ago
- Last unsuccessful build (#7), 1 hr 32 min ago
- Last completed build (#8), 1 hr 29 min ago

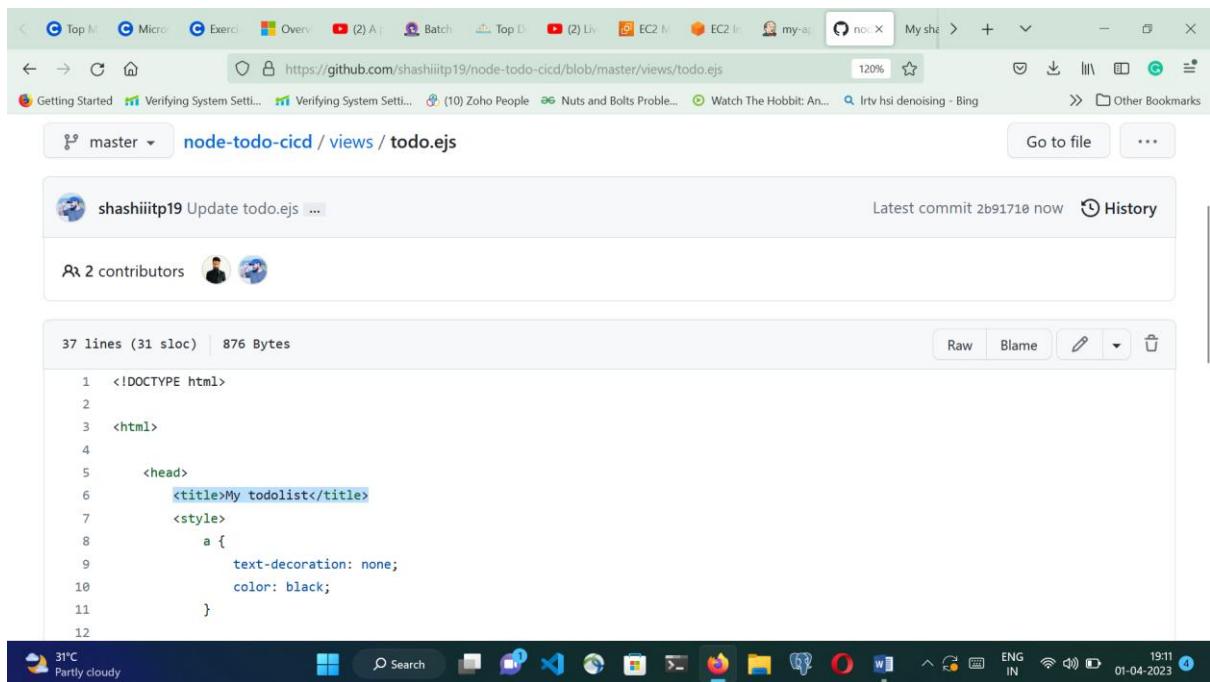
33°C Mostly sunny

Search

ENG IN 19:04 01-04-2023

**Now if any change has been made to the repository, Jenkins will automatically create a new build instantly.**

Jenkins automatically creates a new build through the trigger we build now.

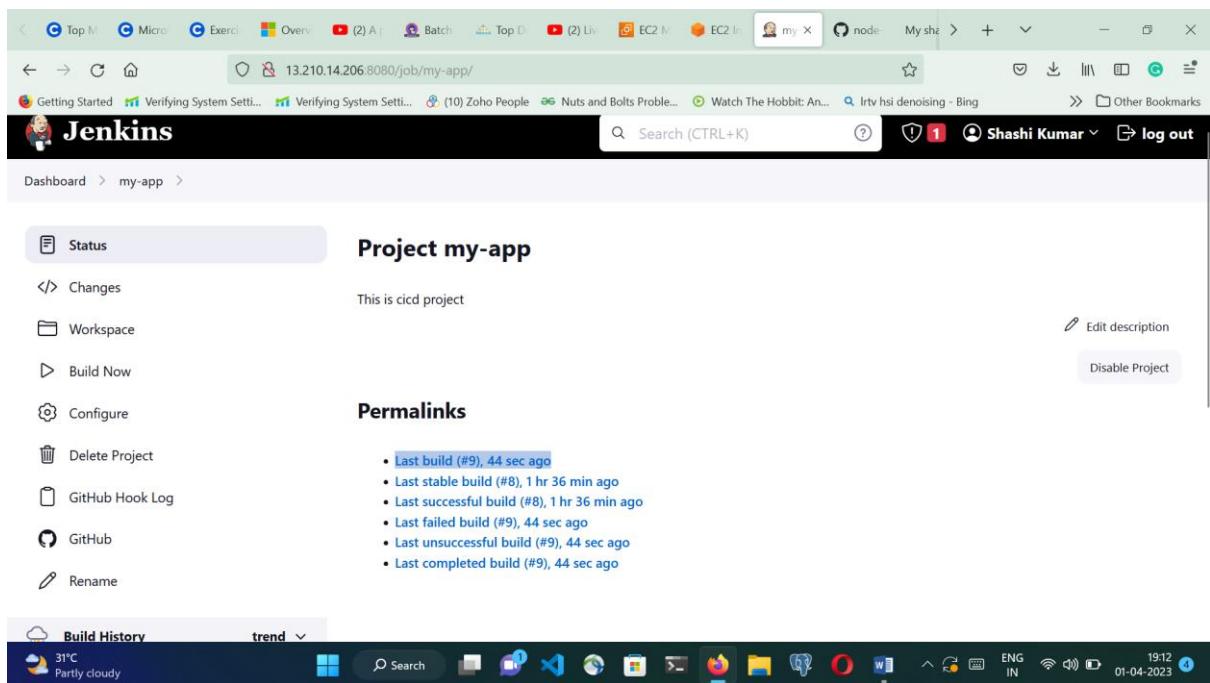


The screenshot shows a GitHub repository page for 'node-todo-cicd'. The file 'views/todo.ejs' is displayed. The code is as follows:

```
1  <!DOCTYPE html>
2
3  <html>
4
5      <head>
6          <title>My toolist</title>
7          <style>
8              a {
9                  text-decoration: none;
10                 color: black;
11             }
12         </style>

```

Automatically build has been created.



The screenshot shows the Jenkins dashboard for the project 'my-app'. The build history is listed as follows:

- Last build (#9), 44 sec ago
- Last stable build (#8), 1 hr 36 min ago
- Last successful build (#8), 1 hr 36 min ago
- Last failed build (#9), 44 sec ago
- Last unsuccessful build (#9), 44 sec ago
- Last completed build (#9), 44 sec ago

```

4 passing (11ms)

Removing intermediate container 88ec8310c6b2
--> 9012b4086414
Step 6/7 : EXPOSE 8000
---- Running in 5e108d722f50
Removing intermediate container 5e108d722f50
--> 4386c4leaf10a
Step 7/7 : CMD ["node","app.js"]
---- Running in 4f0c15a8da2c
Removing intermediate container 4f0c15a8da2c
--> 9601f705e239
Successfully built 9601f705e239
Successfully tagged new-app_2:latest
+ docker run -it --rm -d -p 8000:8000 --name new-app new-app-2
docker: Error response from daemon: Conflict. The container name "/new-app" is already in use by container "3946352c039e93ca8dead6da52890dad6aea7038d3768f84f6456e976df213f4". You have to remove (or rename) that container to be able to reuse that name.
See 'docker run --help'.
Build step 'Execute shell' marked build as failure
Finished: FAILURE

```

Build has failed because already there is one container with the same name exist.

Changes the name of container-

**Configure**

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

**Build Steps**

**Execute shell**

Command

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

```

docker build . -t new-app-6
docker run -it --rm -d -p 8000:8000 --name new-app_2 new-app-2

```

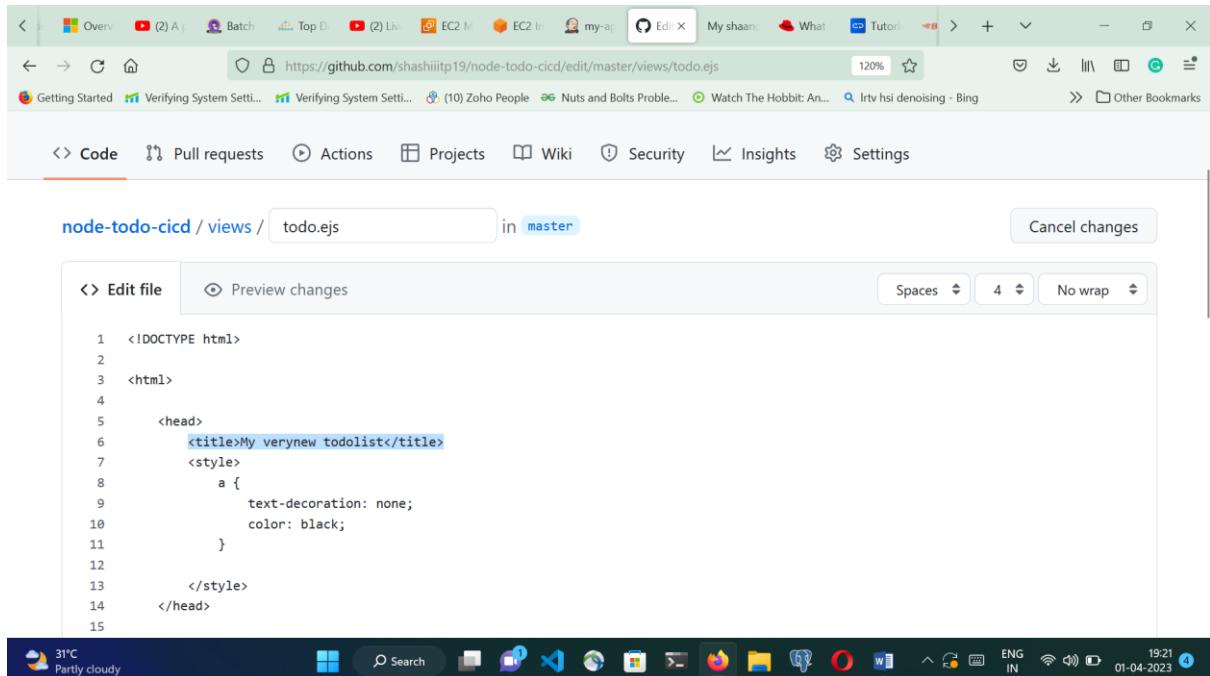
Add build step ▾

Save Apply

Before changing code on GitHub, make sure no container is running on 8000 port.

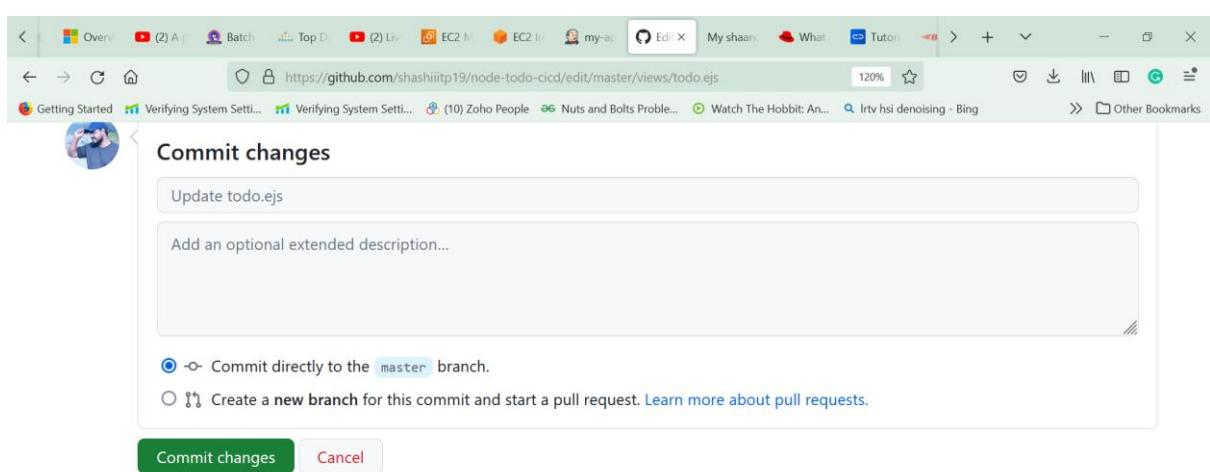
If running kill it using – “docker kill container-id”

**Now do some change in the GitHub repo and commit changes to verify if Jenkins is automatically creates a new build.**



The screenshot shows the GitHub code editor for the file `todo.ejs`. The code is as follows:

```
1 <!DOCTYPE html>
2
3 <html>
4
5   <head>
6     <title>My verynew todolist</title>
7     <style>
8       a {
9         text-decoration: none;
10        color: black;
11      }
12
13     </style>
14   </head>
15 
```



The screenshot shows the GitHub commit changes dialog. The user has entered "Update todo.ejs" in the title field. Below it, there is a text area for an optional extended description. At the bottom, there are two radio button options:

- o- Commit directly to the `master` branch.
- ↗ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

At the bottom of the dialog are two buttons: "Commit changes" (green) and "Cancel".



© 2023 GitHub, Inc.

[Terms](#)

[Privacy](#)

[Security](#)

[Status](#)

[Docs](#)

[Contact GitHub](#)

[Pricing](#)

[API](#)

[Training](#)

[Blog](#)

[About](#)



Dashboard > my-app > Permalinks

Delete Project GitHub Hook Log GitHub Rename

Build History trend

Filter builds... /

- Last build (#11), 5.8 sec ago
- Last stable build (#8), 1 hr 46 min ago
- Last successful build (#8), 1 hr 46 min ago
- Last failed build (#10), 5 min 35 sec ago
- Last unsuccessful build (#10), 5 min 35 sec ago
- Last completed build (#10), 5 min 35 sec ago

Apr 1, 2023, 1:51 PM

Apr 1, 2023, 1:46 PM

Apr 1, 2023, 1:41 PM

Apr 1, 2023, 12:05 PM

Apr 1, 2023, 12:01 PM

Apr 1, 2023, 12:00 PM

13.210.14.206:8080/job/my-app/11/

31°C Partly cloudy

Search

ENG IN 19:22 01-04-2023

Build successfully-

Dashboard > my-app > GitHub Hook Log

GitHub Rename

Build History trend

Filter builds... /

- Last successful build (#8), 1 hr 52 min ago
- Last failed build (#11), 6 min 2 sec ago
- Last unsuccessful build (#11), 6 min 2 sec ago
- Last completed build (#11), 6 min 2 sec ago

Apr 1, 2023, 1:57 PM

Apr 1, 2023, 1:51 PM

Apr 1, 2023, 1:46 PM

Apr 1, 2023, 1:41 PM

Apr 1, 2023, 12:05 PM

Apr 1, 2023, 12:01 PM

Apr 1, 2023, 12:01 PM

Apr 1, 2023, 11:59 AM

Apr 1, 2023, 11:57 AM

Apr 1, 2023, 11:56 AM

Mar 31, 2023, 12:38 PM

31°C Partly cloudy

Search

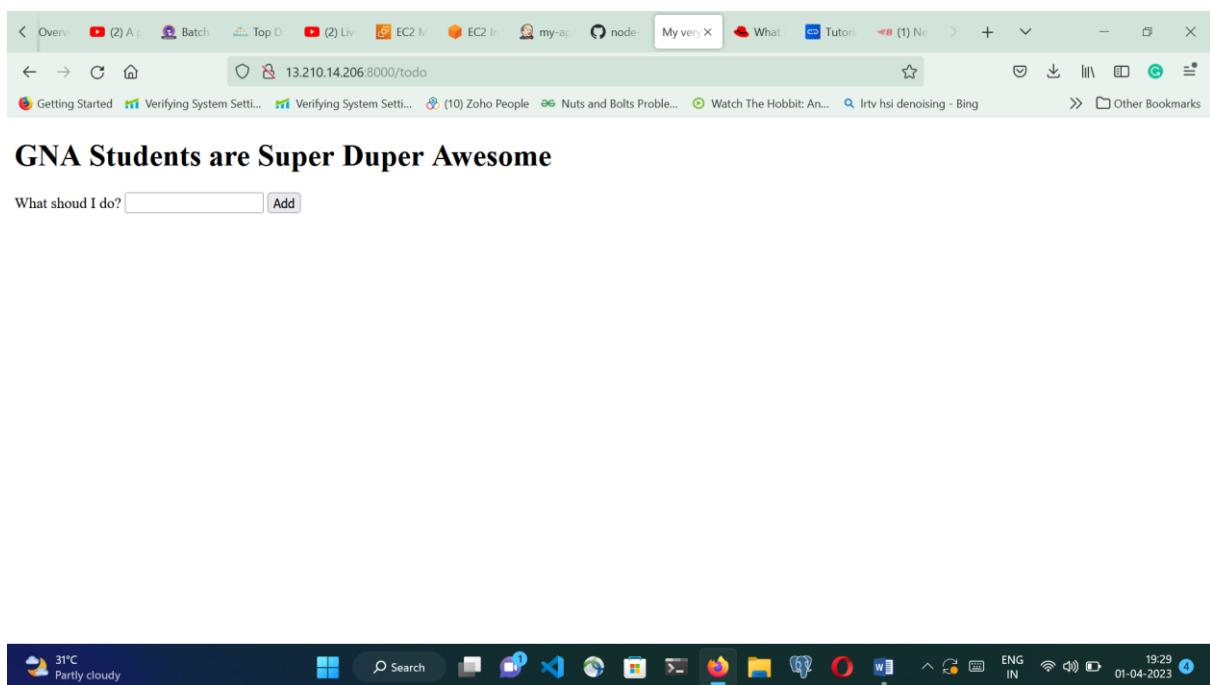
ENG IN 19:28 01-04-2023

The screenshot shows a Jenkins console output page. The title bar says "Console Output". The main content area displays the build log:

```
Started by GitHub push by shashiiitp19
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/my-app
The recommended git tool is: NONE
using credential github-jenkins
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/my-app/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/shashiiitp19/node-todo-cicd.git # timeout=10
Fetching upstream changes from https://github.com/shashiiitp19/node-todo-cicd.git
> git --version # timeout=10
> git --version # 'git' version 2.34.1'
using GIT_SSH to set credentials integrating github with jenkins
Verifying host key using known hosts file
You're using 'Known hosts file' strategy to verify ssh host keys, but your known_hosts file does not exist, please go to 'Manage Jenkins' -> 'Configure Global Security' -> 'Git Host Key Verification Configuration' and configure
```

The sidebar on the left includes links for Status, Changes, Console Output (which is selected), View as plain text, Edit Build Information, Delete build '#12', Polling Log, Git Build Data, and Previous Build.

**Build has been created successfully and app is running on port 80.**



# CONGRATULATIONS

- If you reached here I am sure you have successfully deployed it.

- Support me by clicking like and writing your opinions in the chat if you find it useful.

**THANK YOU...**