

CI-CD JENKINS PROJECT DOCUMENTATION

Table of Contents:

1. Fork repository from GitHub
2. Generate SSH key
3. Configuring GitHub
4. Configuring GitHub-Webhook
5. Installing GitHub integration plugin
6. Configuring jenkins

1. Fork this repository:

The screenshot shows the GitHub interface for creating a new fork of the repository 'node-todo-cicd' by user 'LondheShubham153'. The page title is 'Create a new fork'. Below the title, there is a brief explanation: 'A fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. [View existing forks.](#)'

The form fields are as follows:

- Owner:** A dropdown menu showing 'aghatikar82'.
- Repository name:** A text input field containing 'node-todo-cicd' with a green checkmark icon to its right.
- Description (optional):** A text input field.
- Copy the `master` branch only:** A checkbox that is checked, with a link to 'Learn more.'
- Information:** A note stating 'You are creating a fork in your personal account.'

A green 'Create fork' button is located at the bottom of the form.

The footer of the page includes the GitHub logo, copyright information '© 2023 GitHub, Inc.', and links to 'Terms', 'Privacy', 'Security', 'Status', 'Docs', 'Contact GitHub', 'Pricing', 'API', 'Training', 'Blog', and 'About'.

2. Create a connection to your Jenkins job and your GitHub Repository via GitHub Integration.

Generate the SSH keys for integrating your Jenkins project with your git repository.

Use ssh-keygen command to create public and private key. Here public key used in Github side and private key used in jenkins side.

```
ubuntu@ip-172-31-0-71: ~/ssh
~/ssh/known_hosts: [hashed name]
~/ssh/known_hosts: [hashed name]
~/ssh/known_hosts: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-18-181-77-43.ap-northeast-1.compute.amazonaws.com' (E025519) to the list of known hosts.
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-1028-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Wed Feb  8 19:38:00 UTC 2023

System load:  0.1357421875   Processes:            127
Usage of /:    89.5% of 7.57GB   Users logged in:      1
Memory usage:  76%           IPv4 address for docker0: 172.17.0.1
Swap usage:    0%             IPv4 address for eth0:  172.31.0.71

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.
   https://ubuntu.com/aws/pro

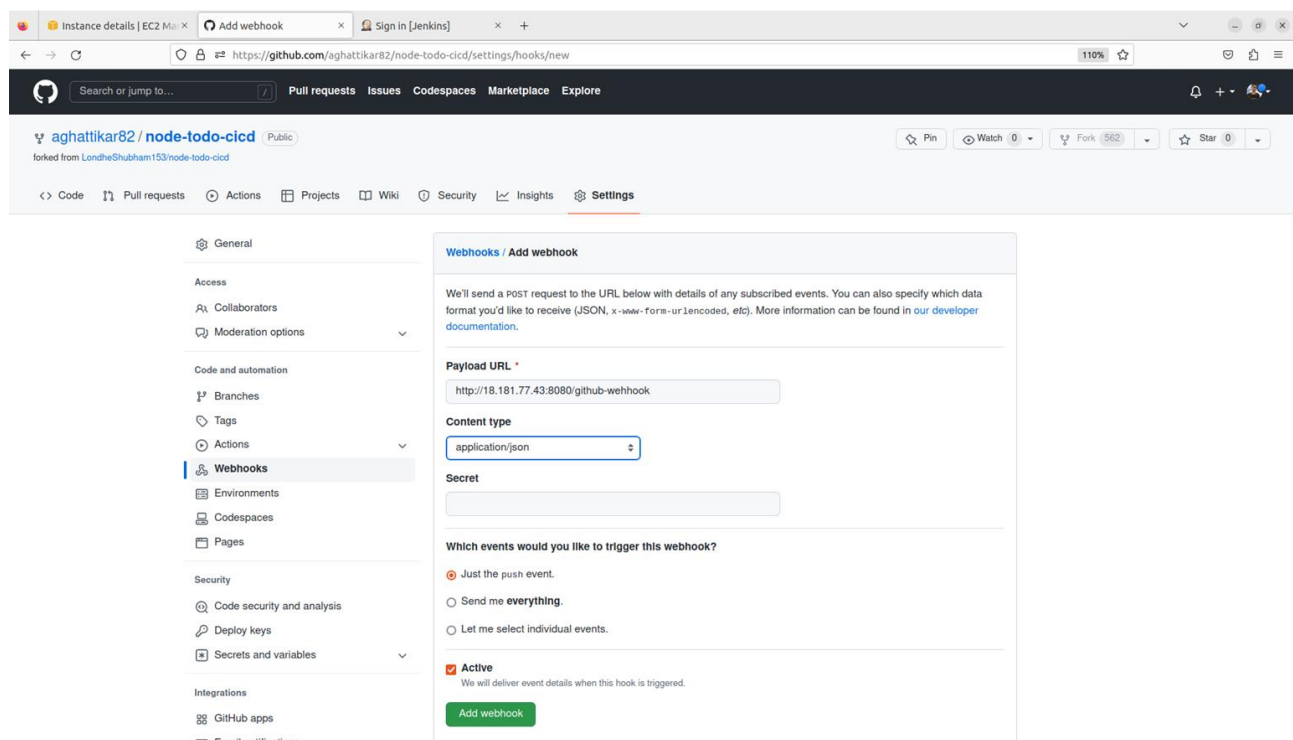
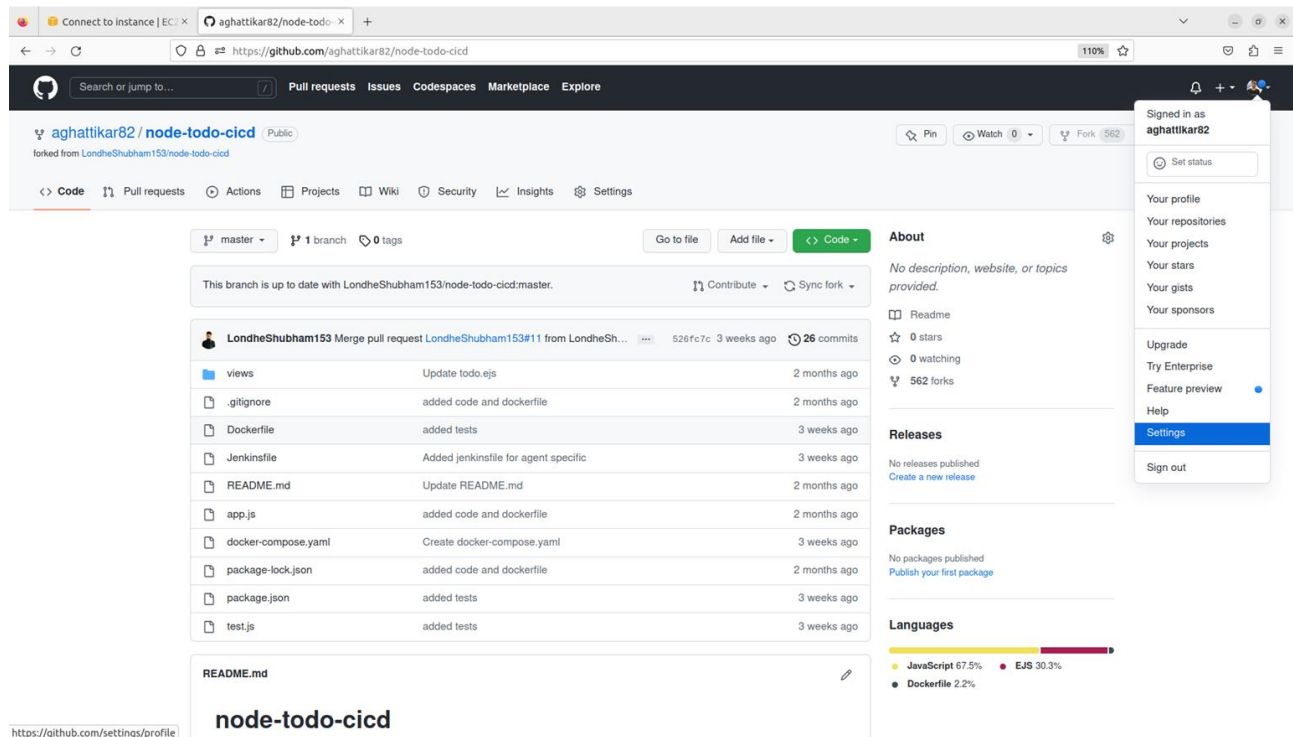
19 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Last login: Wed Feb  8 19:38:02 2023 from 157.45.255.164
ubuntu@ip-172-31-0-71:~$ ssh-keygen
Bad escape character 'ygen'.
ubuntu@ip-172-31-0-71:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_rsa
Your public key has been saved in /home/ubuntu/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:CAF2zxyGzqTaav7XzsmwJG2JdoLCoarnsDFLHIOXTE ubuntu@ip-172-31-0-71
The key's randomart image is:
+---[RSA 3072]---+
|  o.E.o          |
| . .oO .         |
|  =O +           |
|...+.O. .        |
|+*oo . S         |
|*+ =             |
|O+ = O           |
|oo* *. =..       |
|B=.B+. =         |
+---[SHA256]-----+
ubuntu@ip-172-31-0-71:~$ cd .ssh/
ubuntu@ip-172-31-0-71:~/.ssh$ ls
authorized_keys  id_rsa  id_rsa.pub
ubuntu@ip-172-31-0-71:~/.ssh$
```

3. Configuring GitHub

1. Go to your GitHub account settings.

2. Go to SSH and GPG keys, Add the public key that we created using ssh-keygen and select the key-type Authentication key.

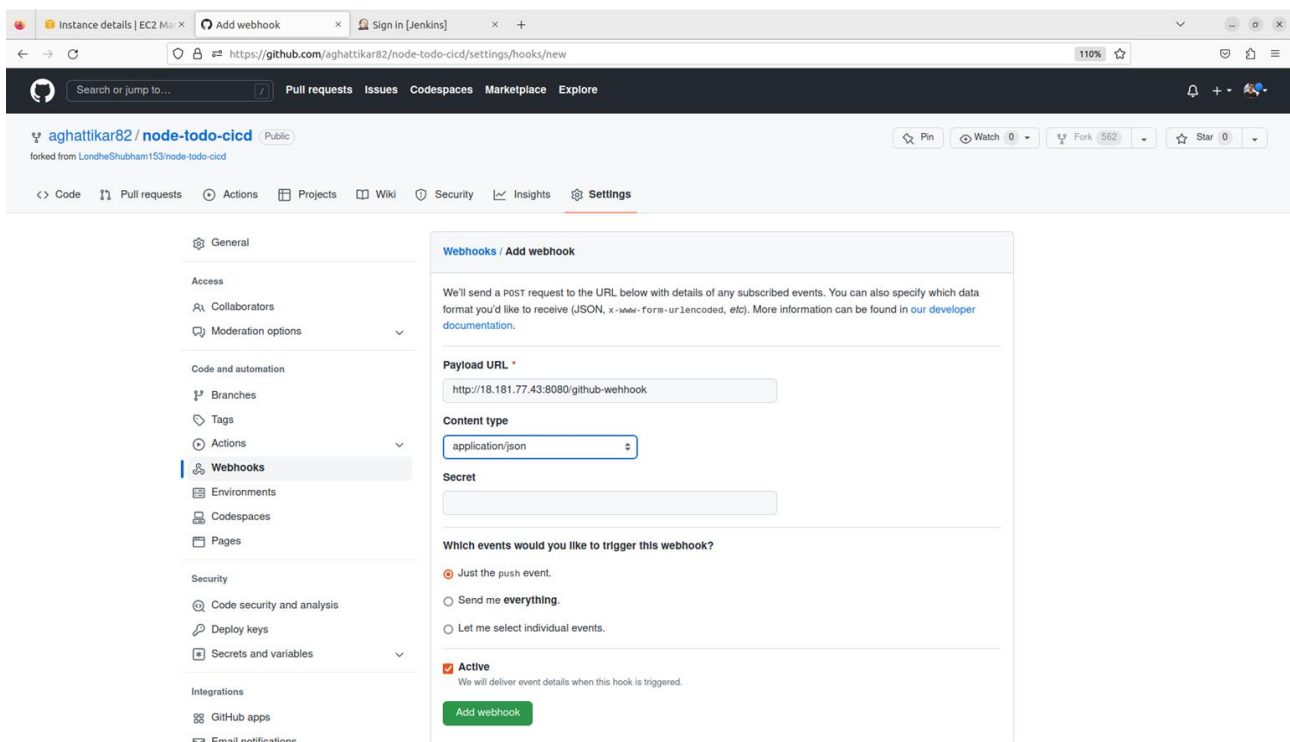


4. Configuring GitHub-Webhook:
GitHub-webhook:

Jenkins GitHub Webhook provides an environment for Continuous Integration (CI) and is widely used in the DevOps process. A few benefits of Jenkins GitHub Webhook are

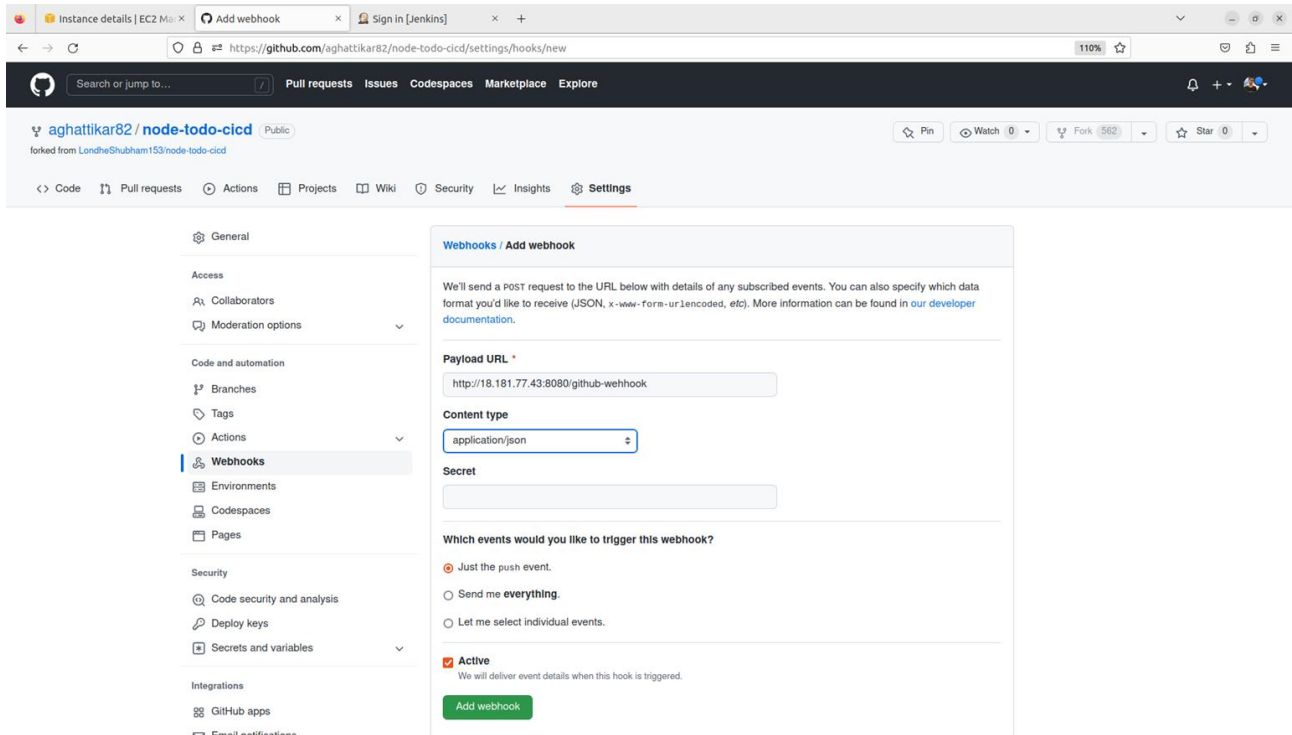
listed below:

- Jenkins GitHub Webhook makes the development process faster as it allows Developers to automatically perform a specified action when any commit is done to the source code.
- It helps Developers detect issues and resolve them almost right away which keeps the application in a state to deploy safely.
- Jenkins GitHub Webhook automates the build, test, and deployment of projects on any platform



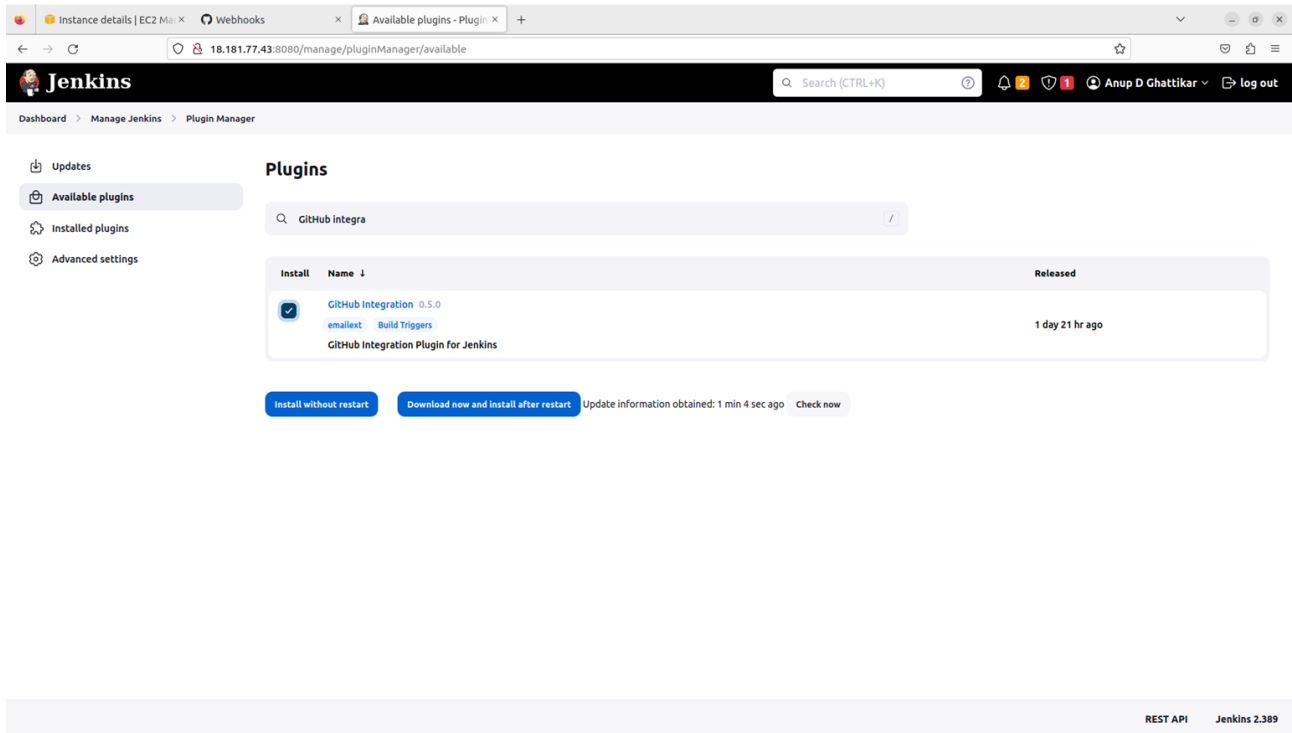
Steps:

1. Go to your GitHub repository and click on Settings.
2. Click on Webhooks and then click on Add webhook.
3. In the 'Payload URL' field, paste your Jenkins environment URL. At the end of this URL add /github-webhook/. In the 'Content type' select: 'application/json' and leave the 'Secret' field empty.



5. For Installing GitHub Integration plugin in Jenkins

1. Open your Jenkins dashboard.
2. Click on the Manage Jenkins button on your Jenkins dashboard
3. Click on Manage Plugins
4. Install GitHub Integration plugin



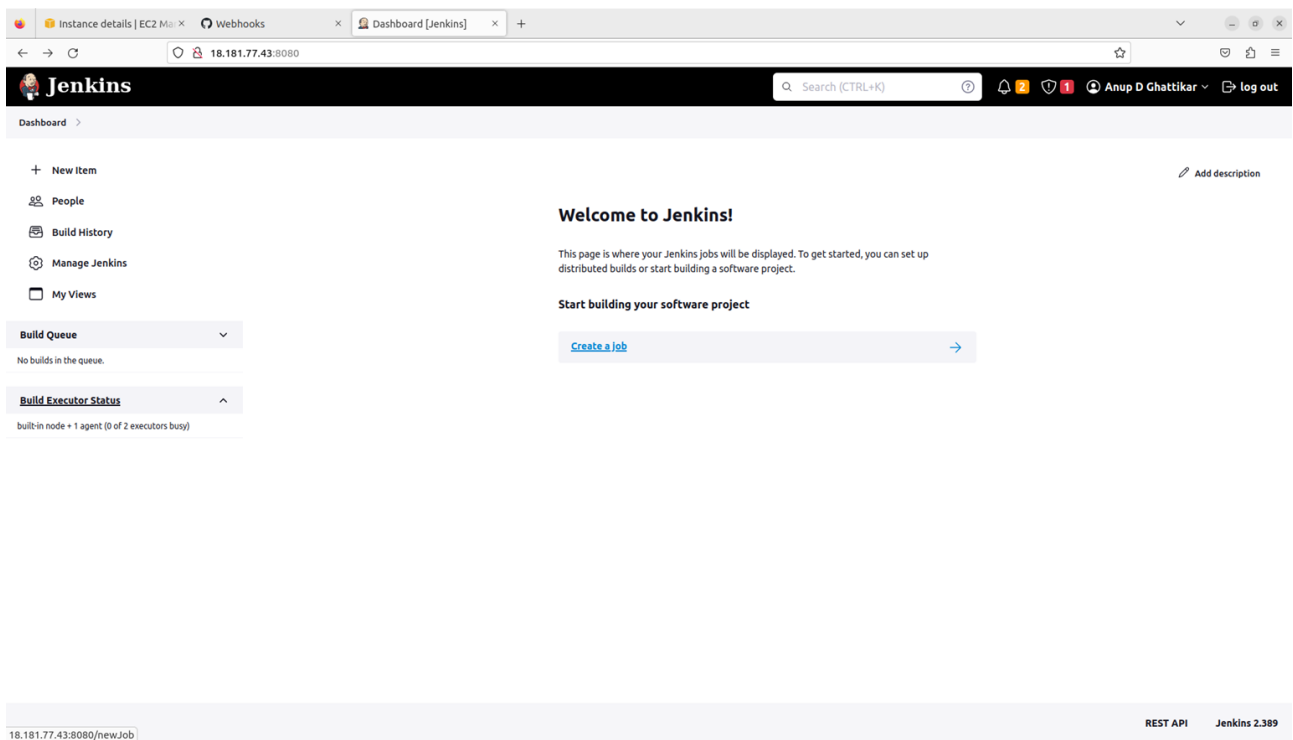
The screenshot shows the Jenkins web interface. The top navigation bar includes the Jenkins logo, a search bar, and user information. The left sidebar contains links to Updates, Available plugins, Installed plugins, and Advanced settings. The main content area is titled 'Plugins' and shows a search for 'GitHub Integra'. A table lists the 'GitHub Integration' plugin (version 0.5.0) as installed, with a 'Released' date of '1 day 21 hr ago'. Below the table are buttons for 'Install without restart' and 'Download now and install after restart', along with a 'Check now' button. The bottom right corner of the interface displays 'REST API' and 'Jenkins 2.389'.

Install	Name	Released
<input checked="" type="checkbox"/>	GitHub Integration 0.5.0 emailtext Build Triggers GitHub Integration Plugin for Jenkins	1 day 21 hr ago

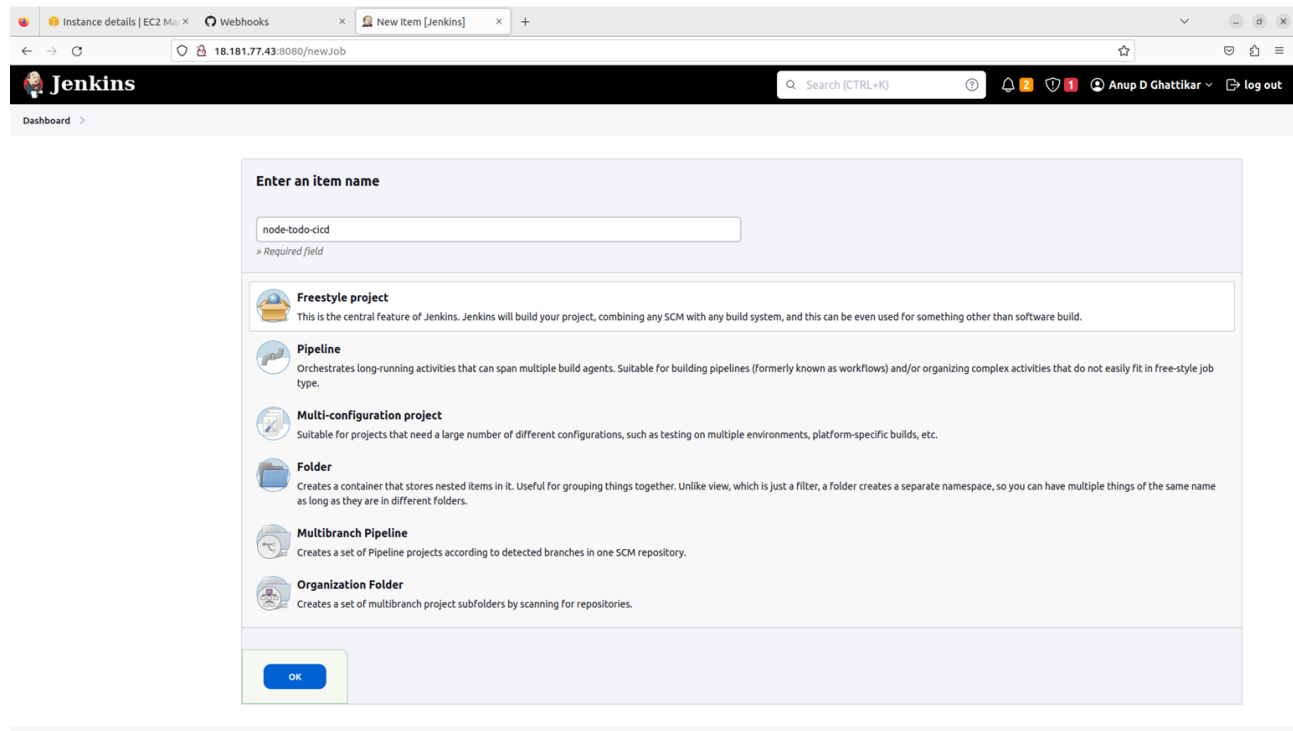
Install without restart Download now and install after restart Update information obtained: 1 min 4 sec ago Check now

REST API Jenkins 2.389

6. Configuring Jenkins:
 1. Create a Jenkins job



2. Create node-todo-app freestyle project



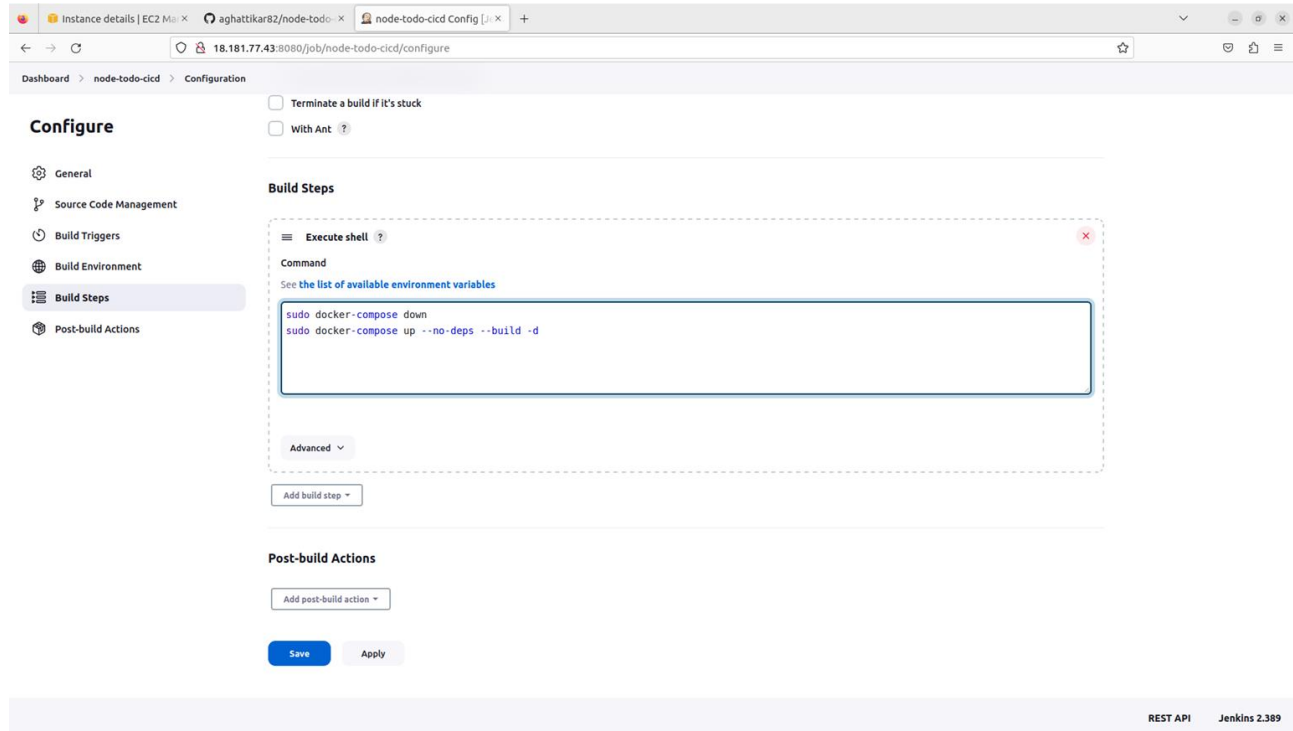
3. In Configure, GitHub project URL write your project GitHub URL

4. In Git, add credentials for Jenkins

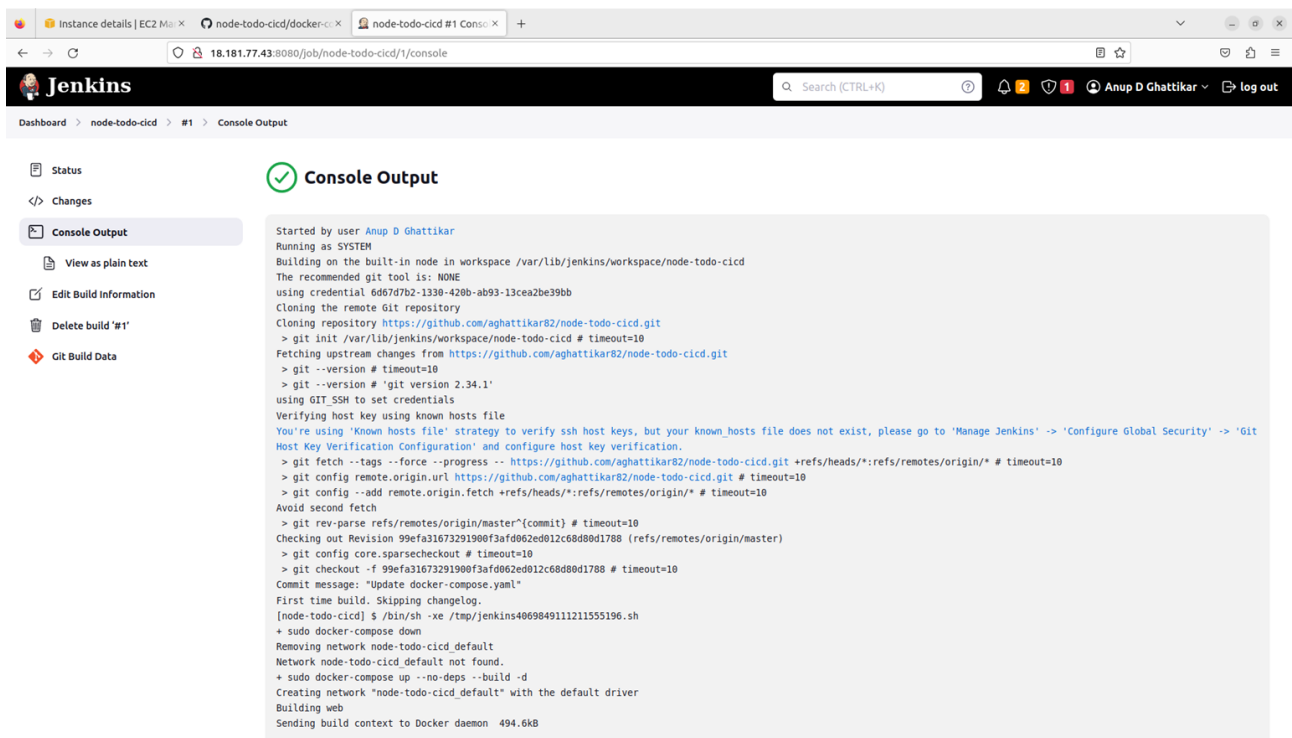
5. Add private key that we created using the `ssh-keygen` command.

6. Click on the 'Build Triggers' tab and then on the 'GitHub hook trigger for GITScm polling'.

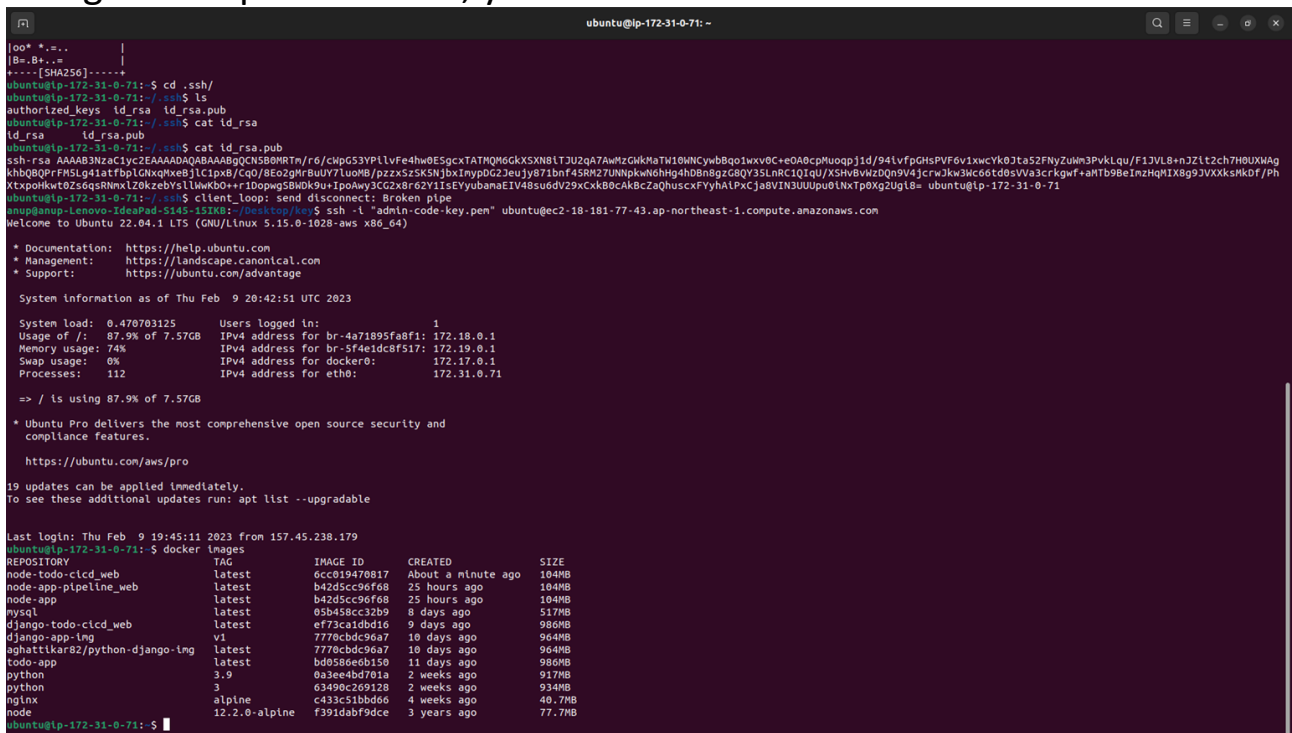
In the Execute shell run the application using Docker compose



You will have to make a Docker Compose file for this Project
After build you can check console output.



Using docker ps command, you can see the container is created.



Browse public IP address with port no.8000

Thank you