CI-CD INTEGRATION

AIM : Jenkins CICD with GitHub Integration. CICD -Continuous Integration and Continuous Deployment. We are going to do CICD pipeline in this assessment.

We are going to implement a pipeline where the developer code run effectively.

Tools used :

1. AWS
2. JENKINS
3. Github
4. Docker

Resources implemented :

5. EC2
6.

Process /Implementation:

>First we take the code from the developer from the GitHub, where it is usually stored in the repository .The DevOps Engineer takes the code from it and performs the required actions.

> The DevOps Engineer role is to check that the code is continuously Integrated in the required environment or not.

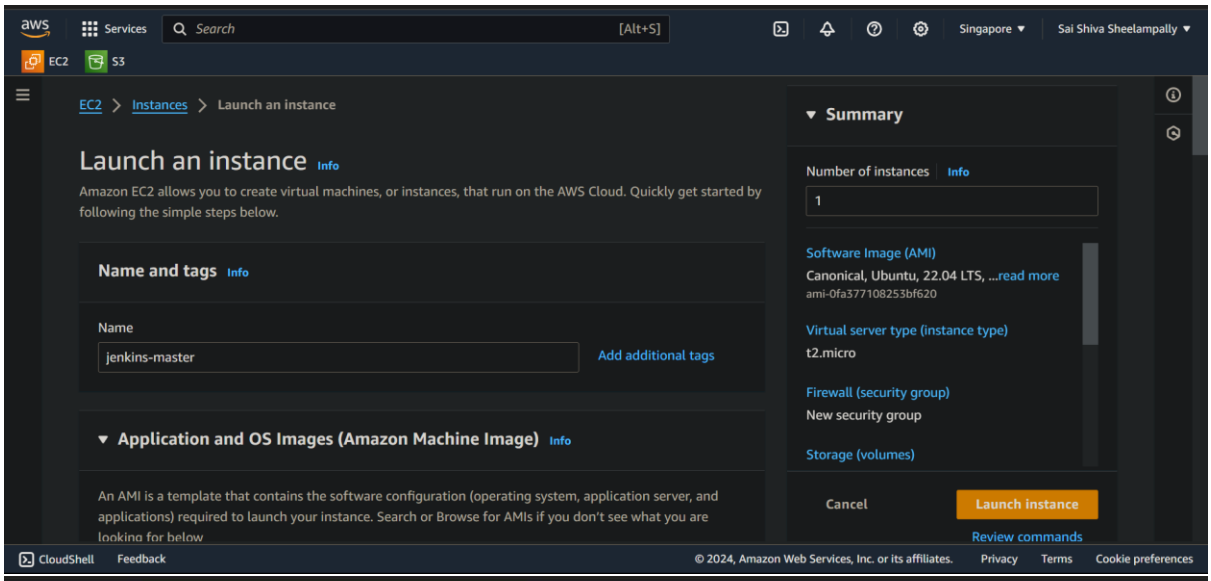
> The DevOps Engineer needs to make sure that the code integrated in every type of environment(MacOS, windows, linux).

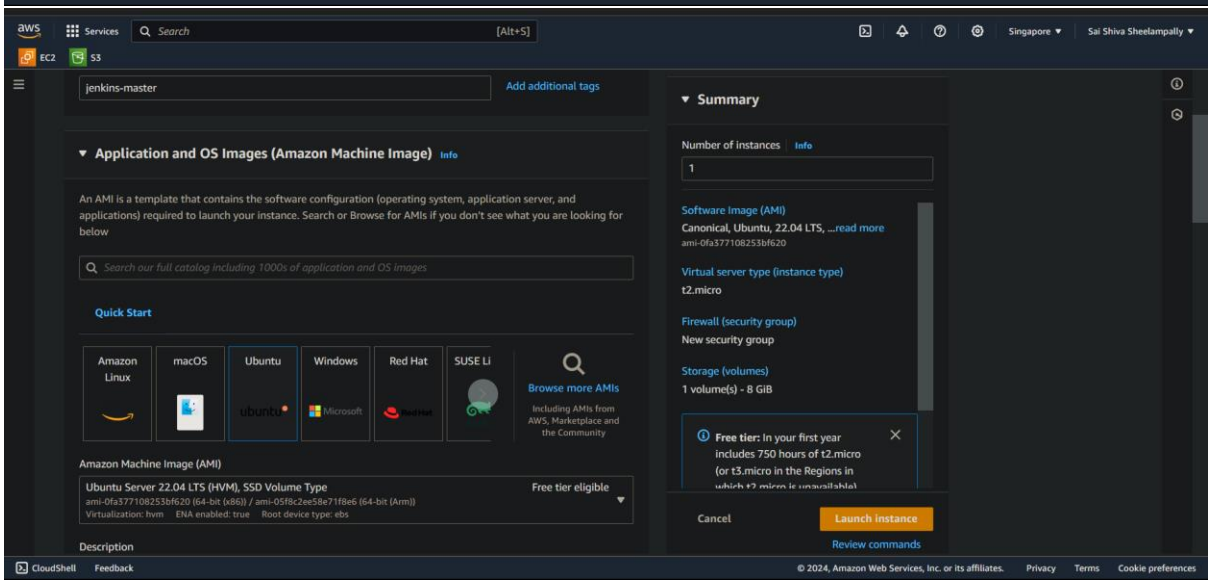
>The DevOps Engineer uses a tool called as Docker to make a virtualised container so that the code runs in any environment.

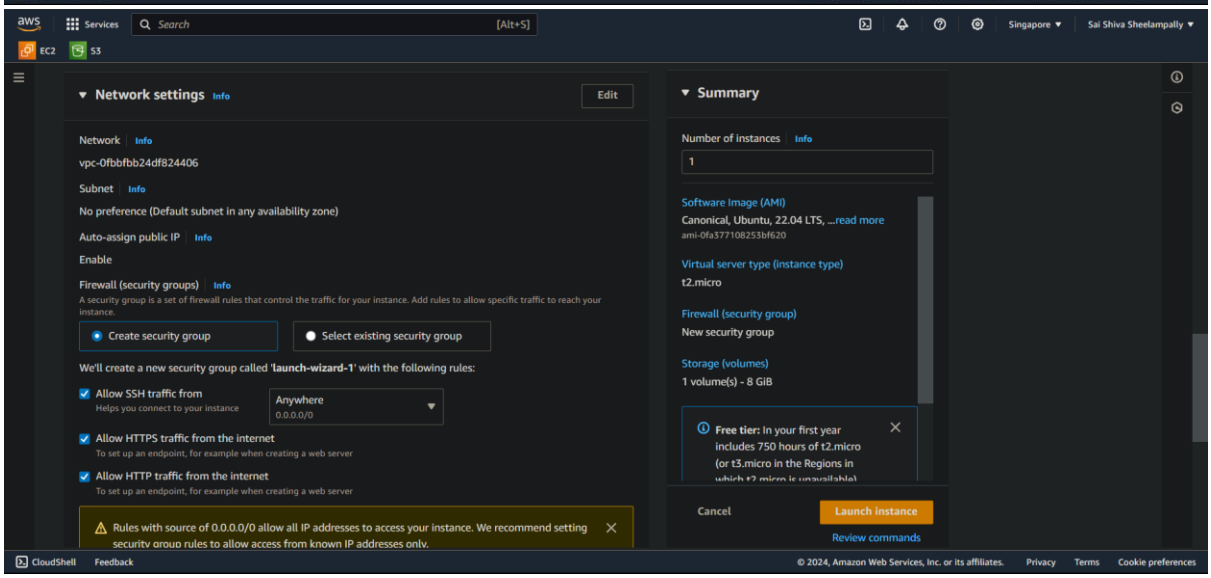
>We use Amazon EC2 instance to deliver the containerized product.

>We use the pipeline, for making a flow for taking the code, containerising it and then delivering it . Jenkins does this work .

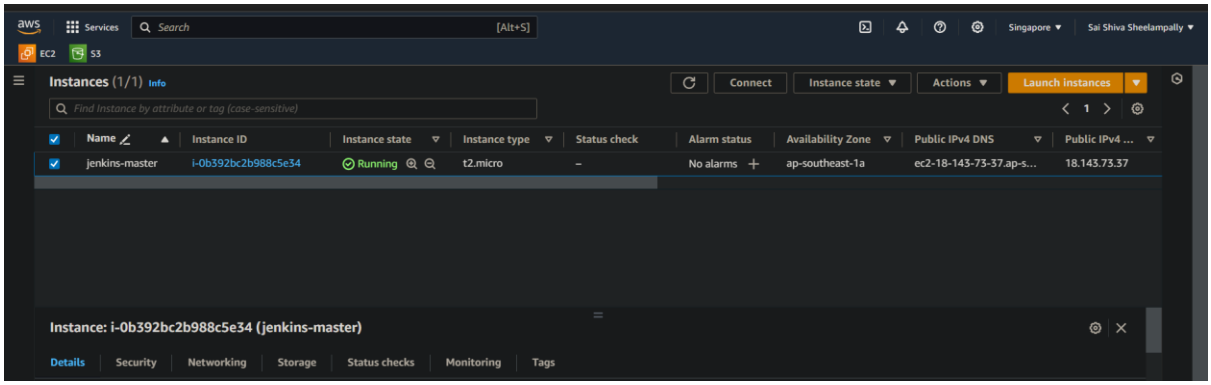
1. Create an Instance







Changing the network settings so that anyone form internet can access this



2. Now connect the instance and install Jenkins on aws

For that run some commands on the server. They are

- sudo apt update
- sudo apt install openjdk-11-jre
- sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \

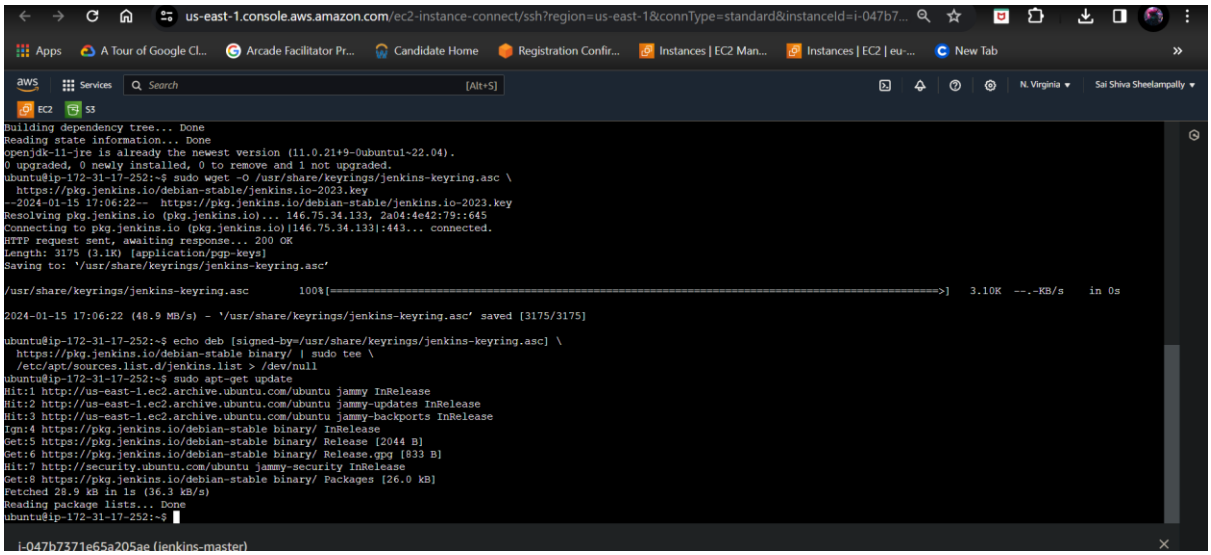
  https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key

- 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

- sudo apt-get update
- sudo apt-get install jenkins



Jenkins successfully installed

```
(Reading database ... 66425 files and directories currently installed.)
Preparing to unpack .../net-tools_1.60+git20181103.0eebece-1ubuntu5_amd64.deb ...
Unpacking net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Selecting previously unselected package jenkins.
Preparing to unpack .../jenkins_2.426.2_all.deb ...
Unpacking jenkins (2.426.2) ...
Setting up net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Setting up jenkins (2.426.2) ...
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /lib/system
d/system/jenkins.service.
Processing triggers for man-db (2.10.2-1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

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-172-31-94-111:~$
```

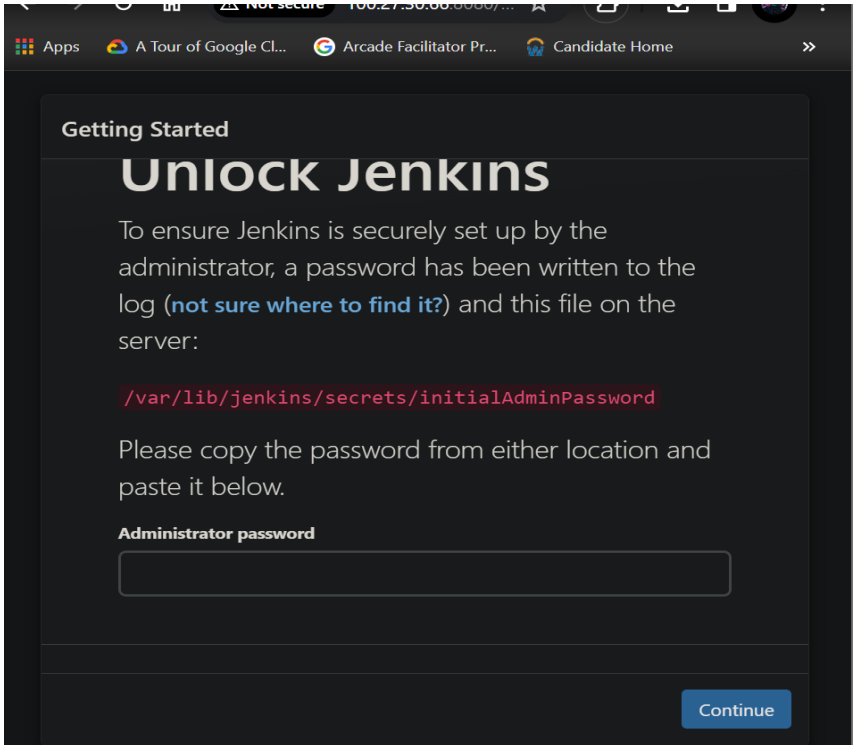
i-039353b36bc1593a4 (jenkins-master)

PublicIPs: 100.27.30.66 PrivateIPs: 172.31.94.111

Start Jenkins by running the following commands;

- sudo systemctl enable jenkins
- sudo systemctl start jenkins
- sudo systemctl status Jenkins

Open the Jenkins application in your web browser by entering the public IP address of your instance, followed by port 8080.

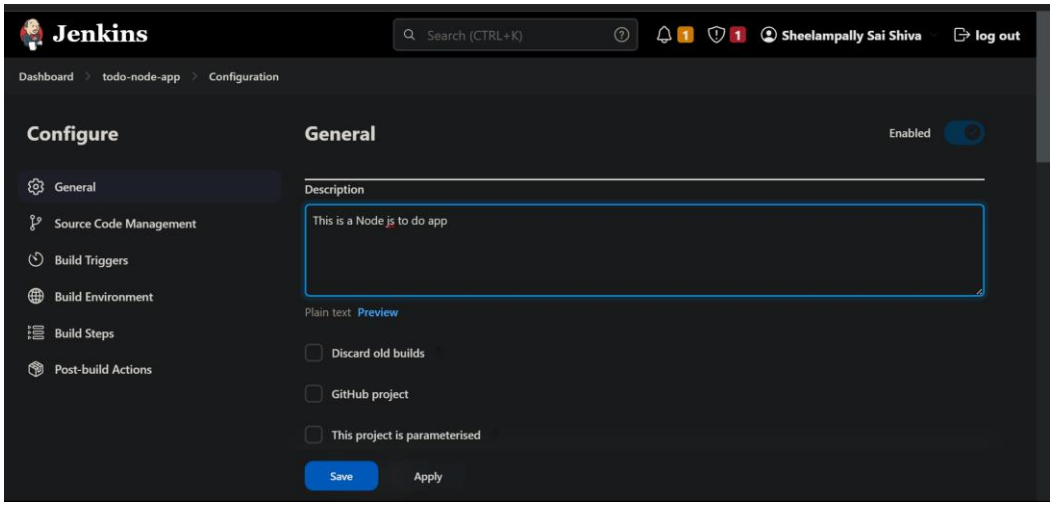


```
ubuntu@ip-172-31-94-111:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
915eff847fb94122a6a7740134179faa
ubuntu@ip-172-31-94-111:~$
```

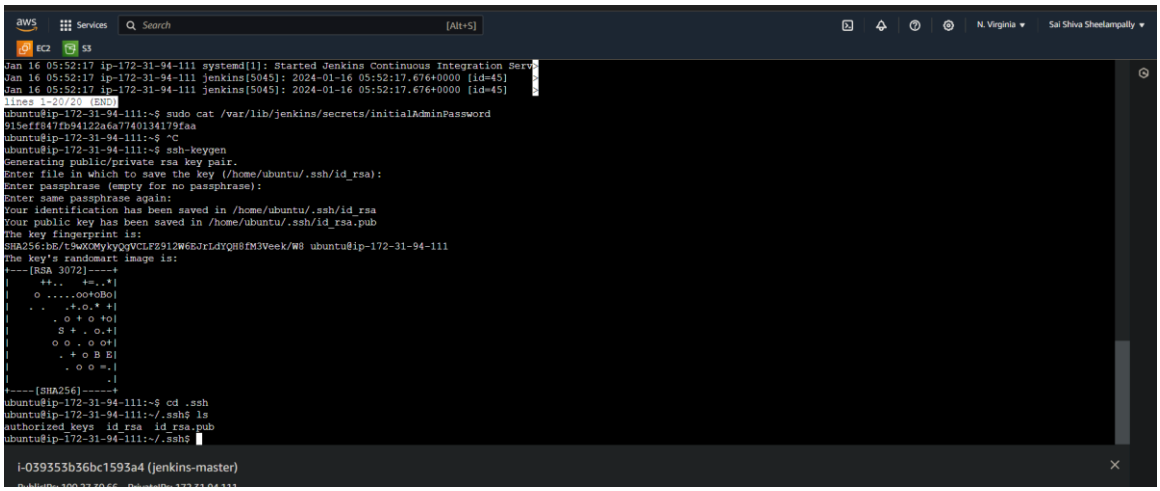
i-039353b36bc1593a4 (jenkins-master)

PublicIPs: 100.27.30.66 PrivateIPs: 172.31.94.111

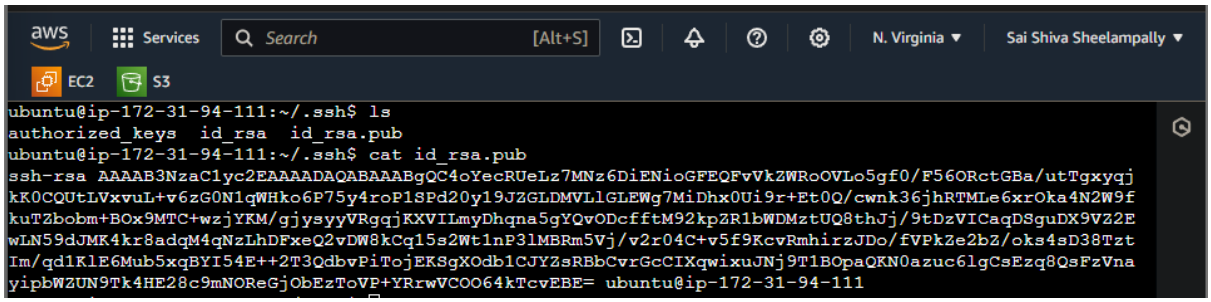
Paste the password then create a new account and then do the freestyle app of Node js Todo app



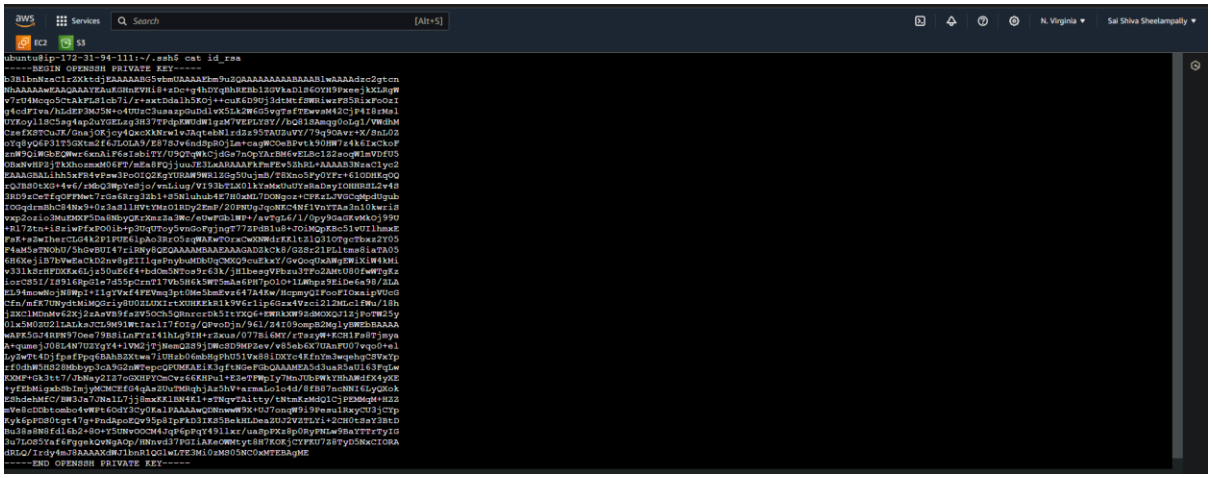
Key-generation



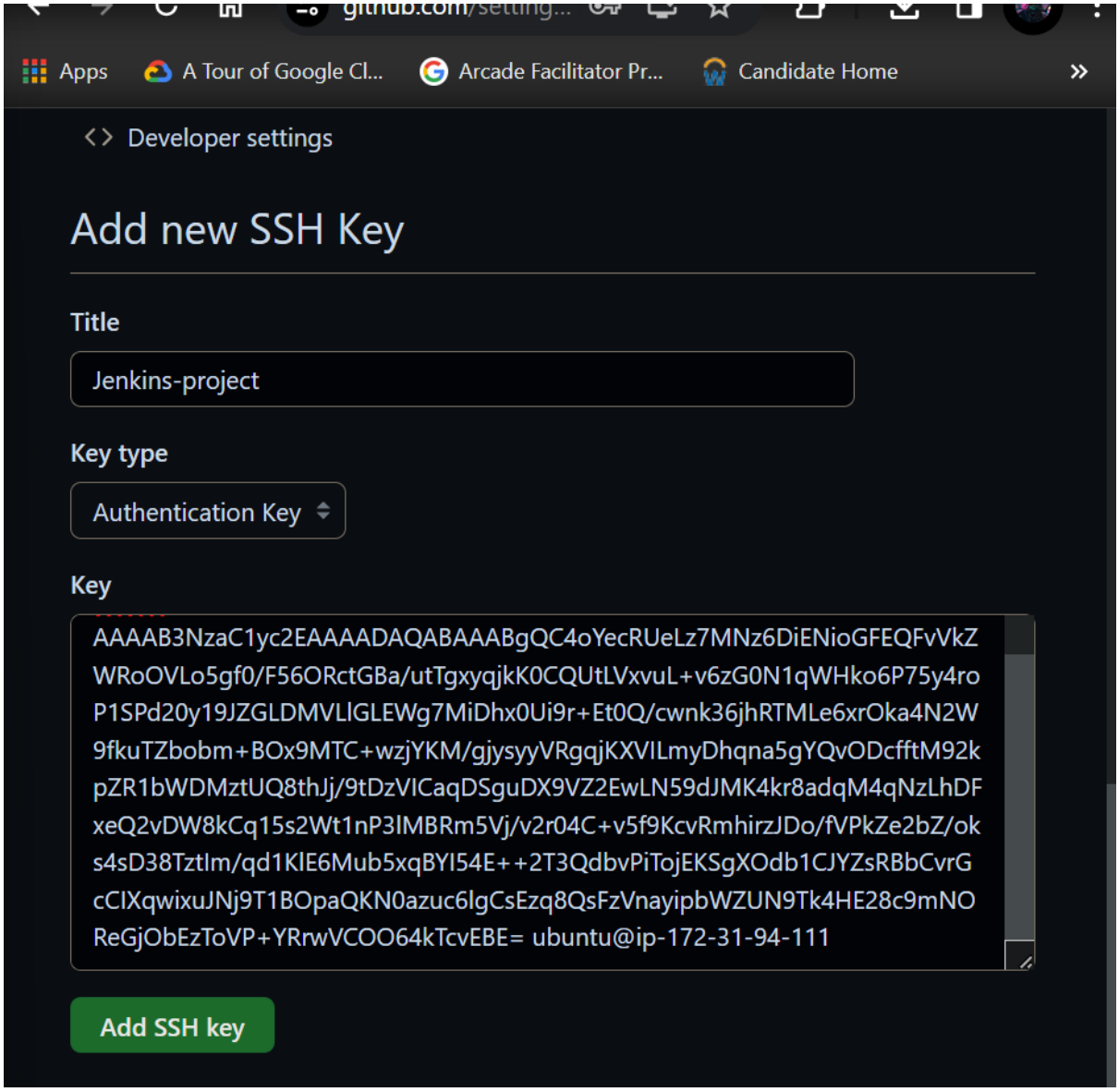
Public key

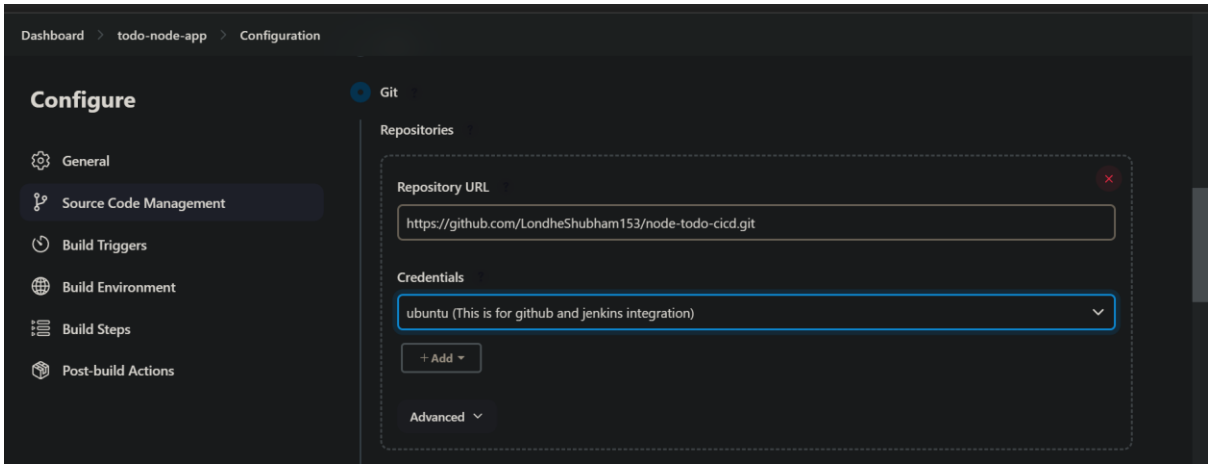


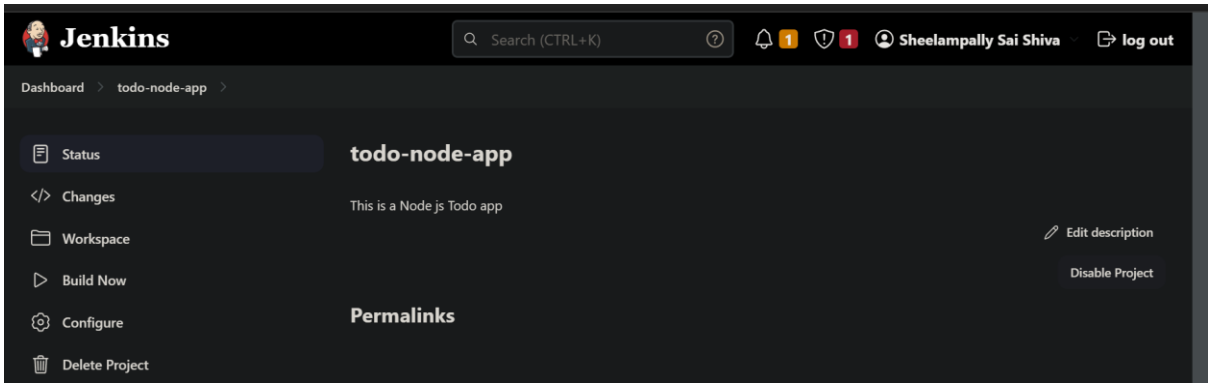
Private key



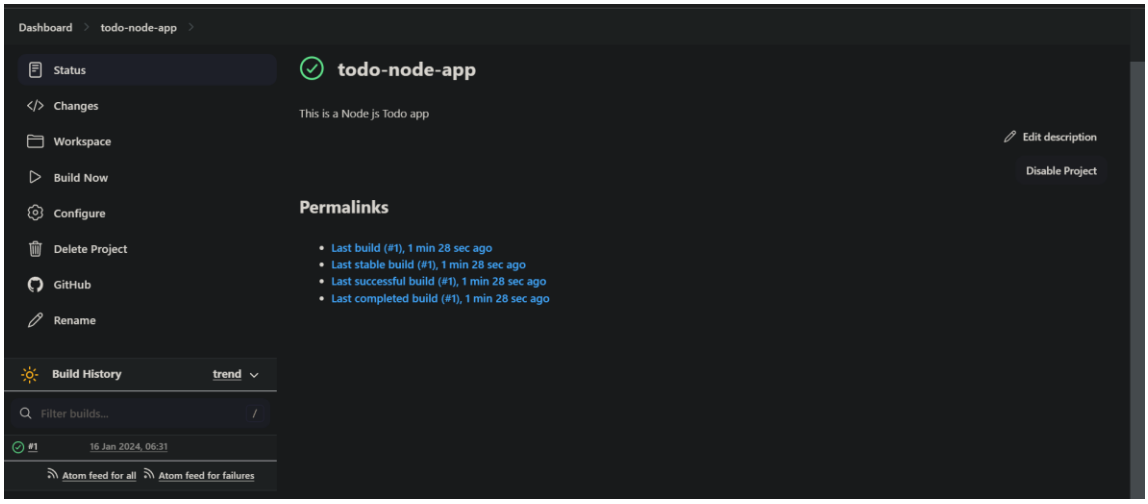
If we want the github to connect to our server we need to give it a public key.

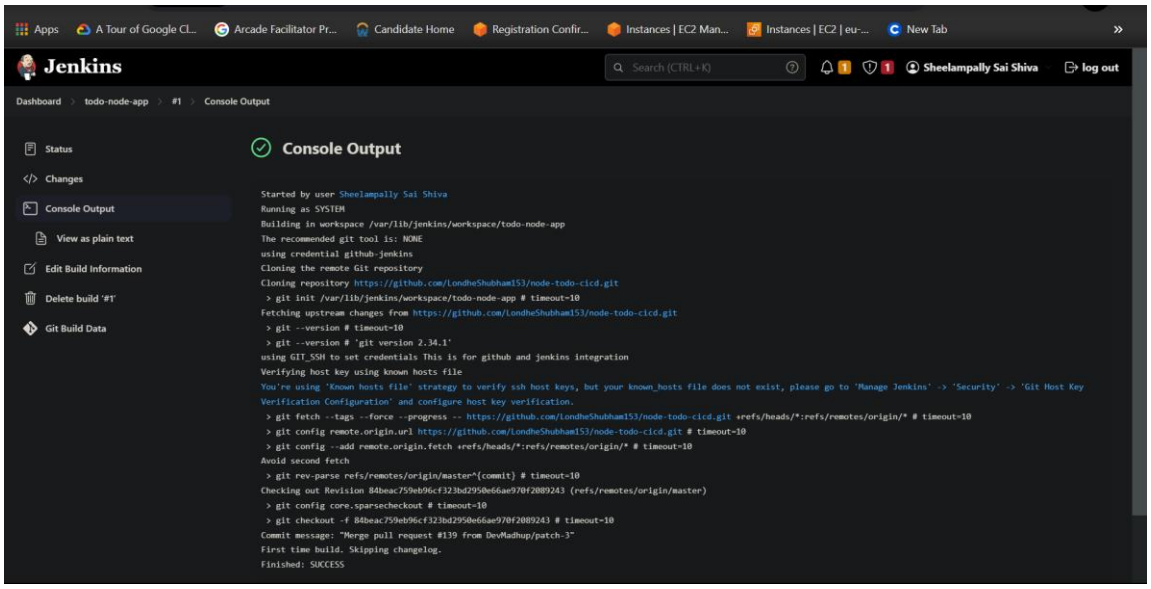


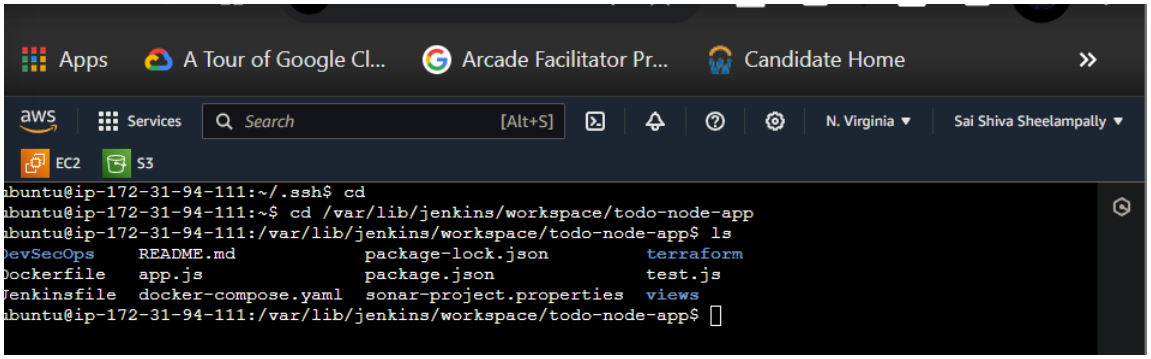




Let us now enter the build now for the piple*********







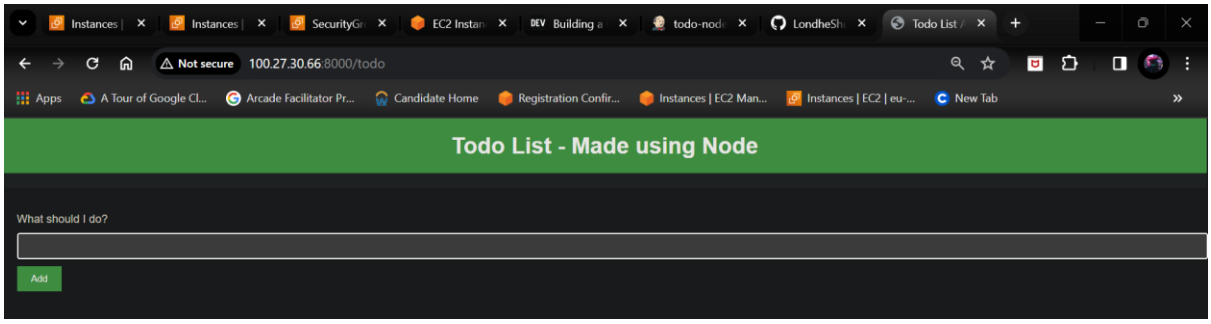
Now we can confirm that the Jenkins and github are integrated

Now for running the code run the following commands in the EC2 instance connect terminal

- sudo apt install nodejs
- sudo apt install npm
- sudo npm install
- To run the app run this command :: node app.js

```
npm WARN deprecated mkdirp@0.5.4: Legacy versions of mkdirp are no longer supported. Please update t
o mkdirp 1.x. (Note that the API surface has changed to use Promises in 1.x.)
npm WARN deprecated uuid@3.4.0: Please upgrade  to version 7 or higher.  Older versions may use Math
.random() in certain circumstances, which is known to be problematic.  See https://v8.dev/blog/math-
random for details.
npm WARN deprecated formidable@1.2.6: Please upgrade to latest, formidable@v2 or formidable@v3! Chec
k these notes: https://bit.ly/2ZEqIau
npm WARN deprecated debug@3.2.6: Debug versions >=3.2.0 <3.2.7 || >=4 <4.3.1 have a low-severity ReD
os regression when used in a Node.js environment. It is recommended you upgrade to 3.2.7 or 4.3.1. (
https://github.com/visionmedia/debug/issues/797)
npm WARN deprecated superagent@3.8.3: Please upgrade to v7.0.2+ of superagent.  We have fixed numero
us issues with streams, form-data, attach(), filesystem errors not bubbling up (ENOENT on attach()),
 and all tests are now passing.  See the releases tab for more information at <https://github.com/vi
sionmedia/superagent/releases>.

added 291 packages, and audited 292 packages in 21s

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

10 vulnerabilities (3 moderate, 2 high, 5 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-172-31-94-111:/var/lib/jenkins/workspace/todo-node-app$ node app.js
Todolist running on http://0.0.0.0:8000
```

Change inbound rules and paste the public IP along with the port 8000



Now we need to make docker to make the app as the virtualised container

For that we need to install Docker. So now by using this command we can install docker

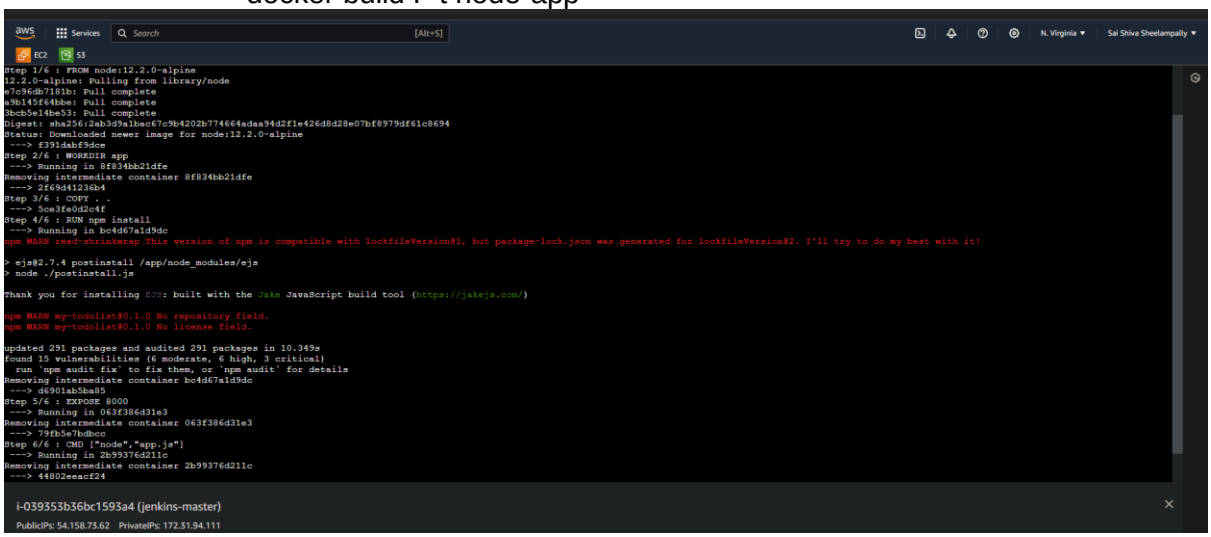
- sudo apt install docker.io
- for making docker file we use this command :: vi Dockerfile

Now Enter these commands in the file

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

Now for building the docker file use this command

- docker build . -t node-app
- sudo usermod -a -G docker $USER
- sudo reboot
- docker build . -t node-app

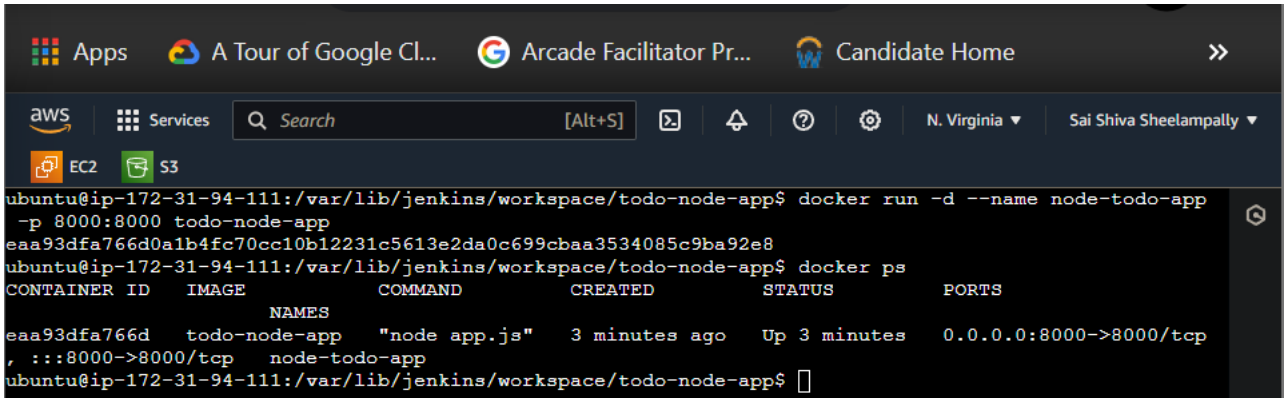


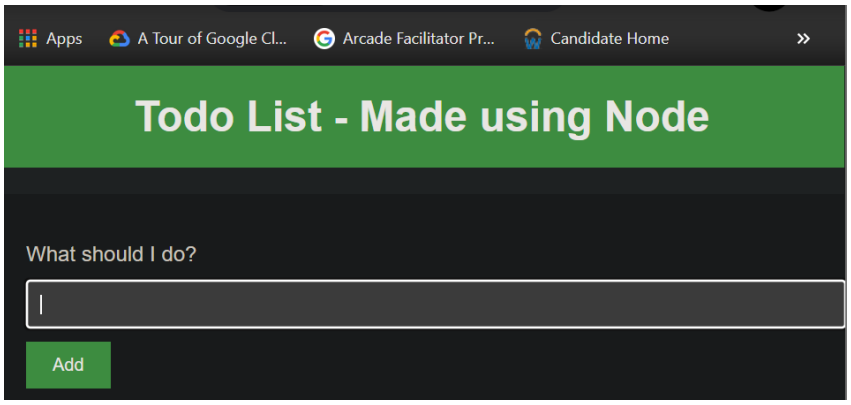
We have successfully built docker container

To run the container run the following command

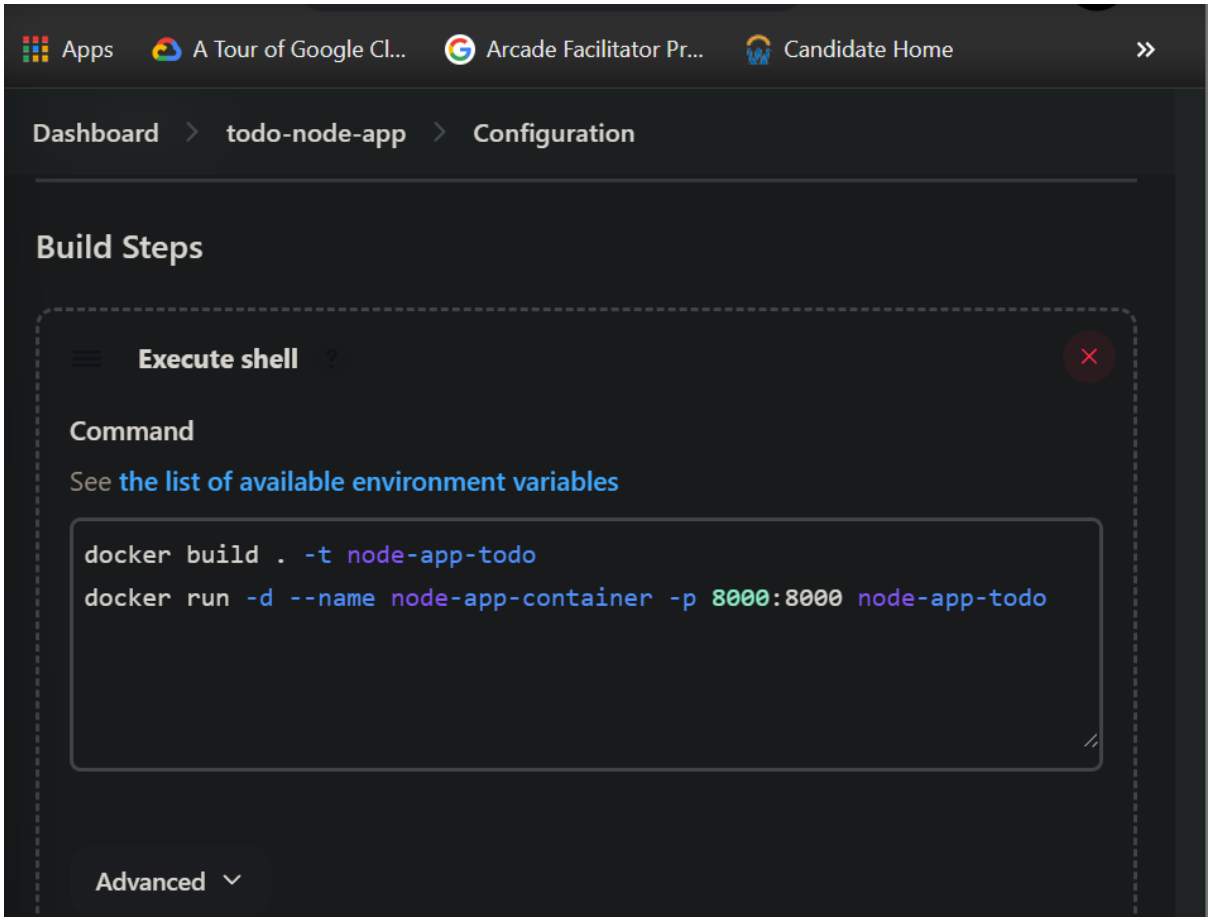
- docker run -d --name node-todo-app -p 8000:8000 todo-node-app

Now the app is running



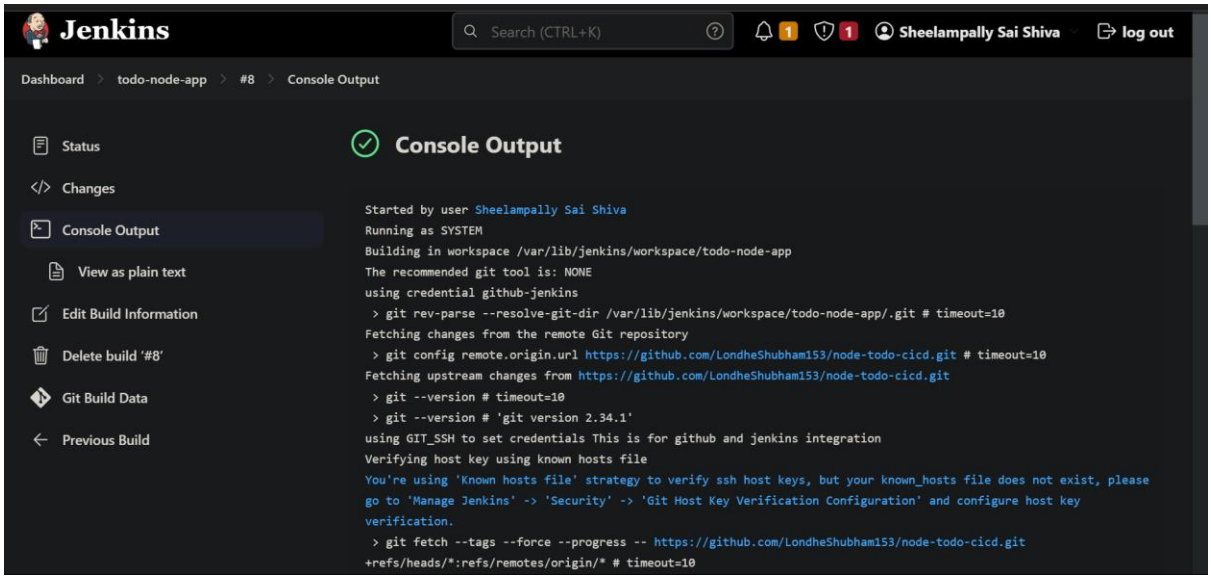


Now go to jenkins job and then toExecute shell and paste the following commands

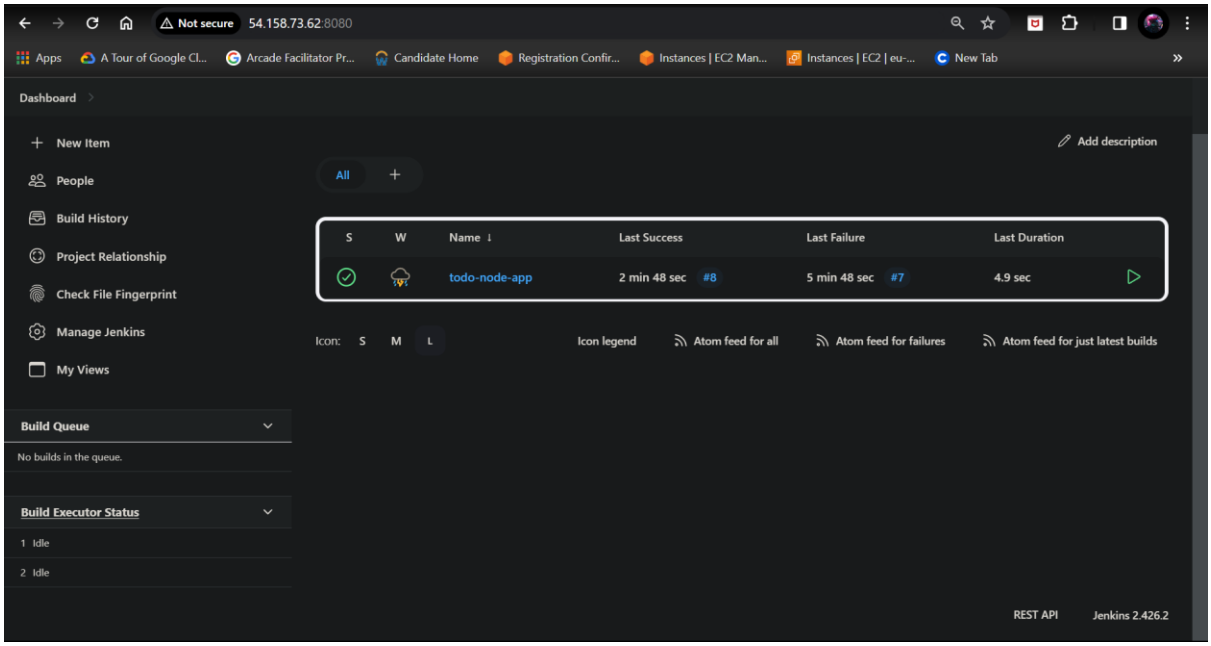


sudo usermod -a -G docker Jenkins

sudo systemctl restart Jenkins

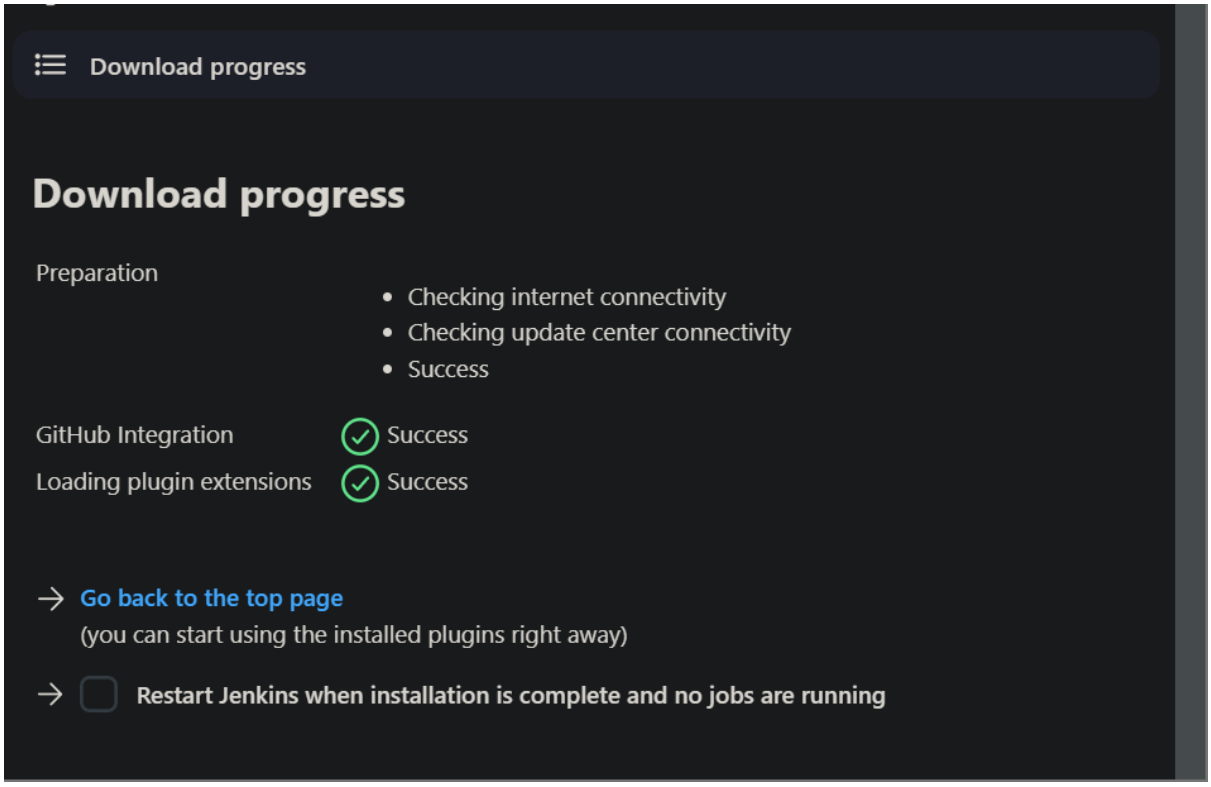


Now the app is running through Jenkins

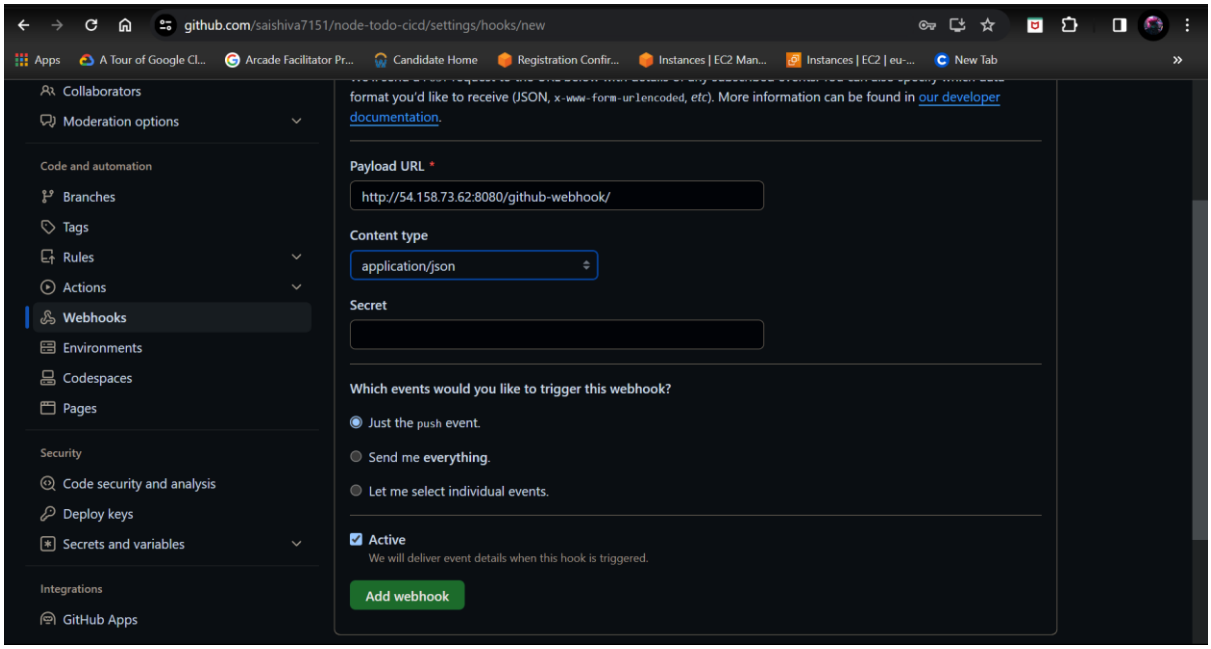


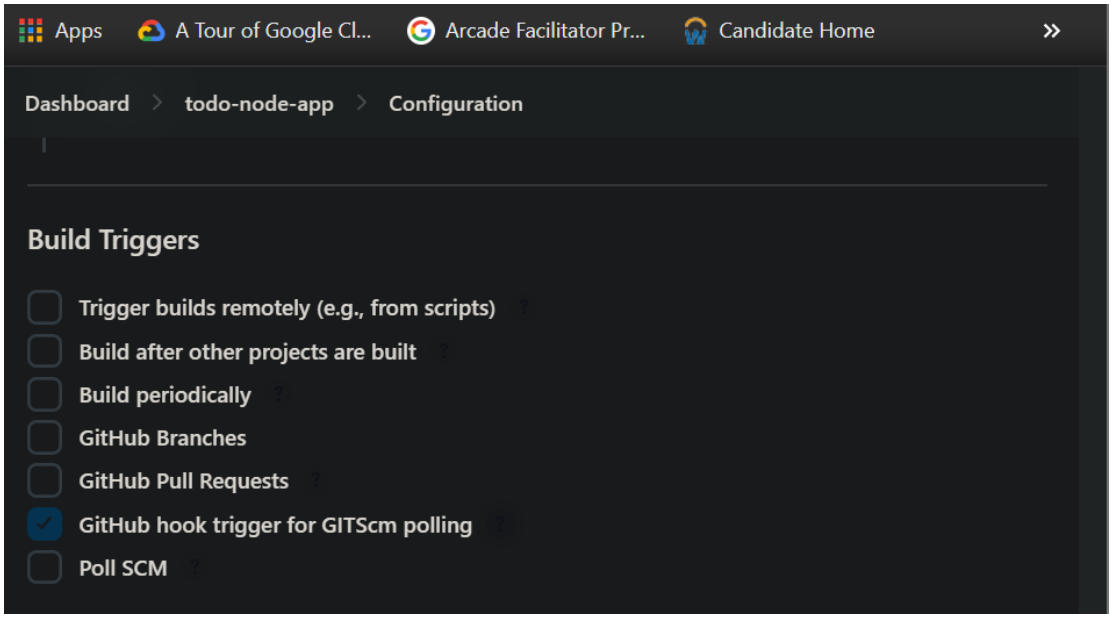
Thus we have implemented CICD pipeline and now let us do this by using webhooks : Automated Jenkins

>>For this first we have install some plugins, check for github integration plugin

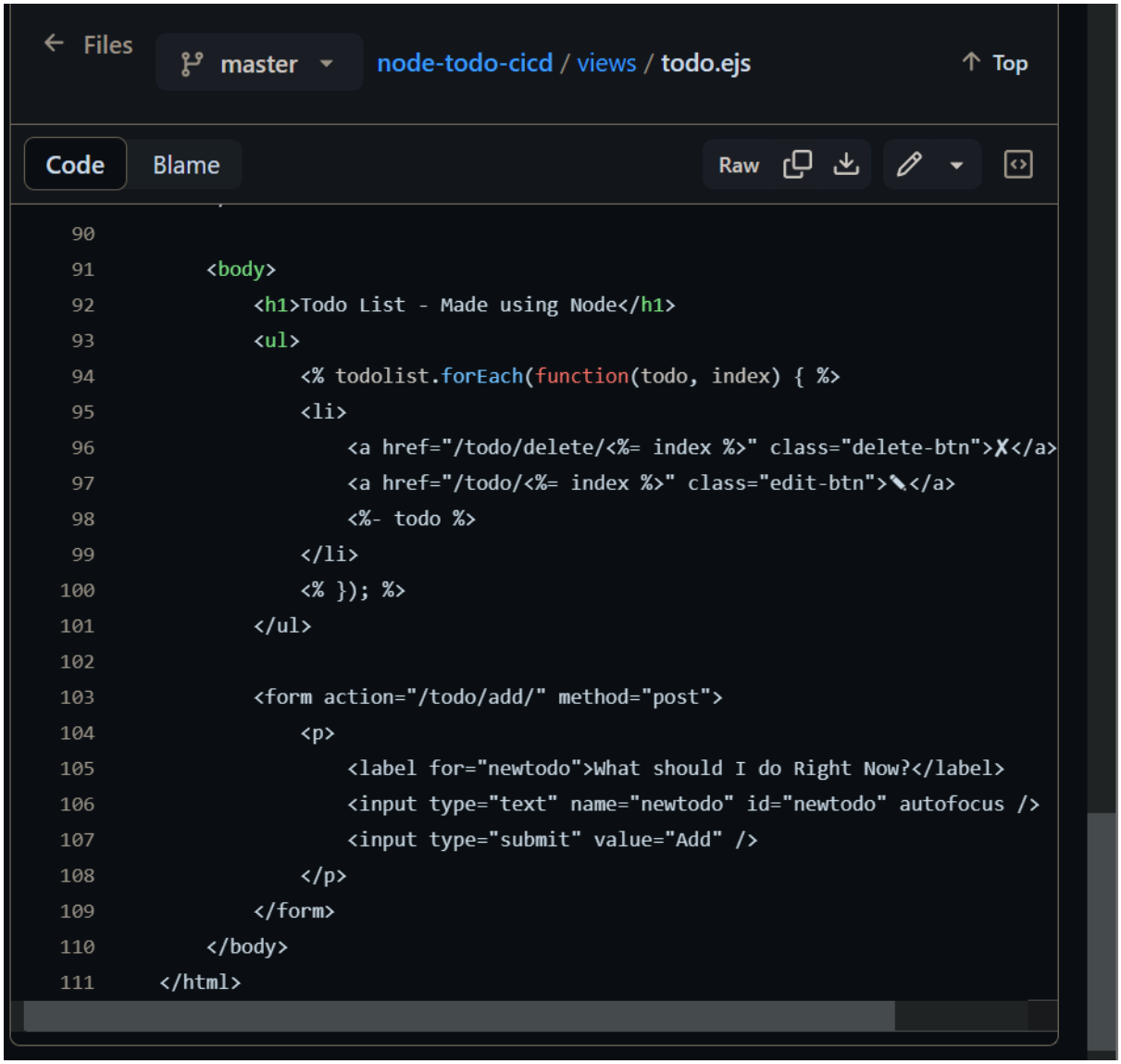


Add webhooks



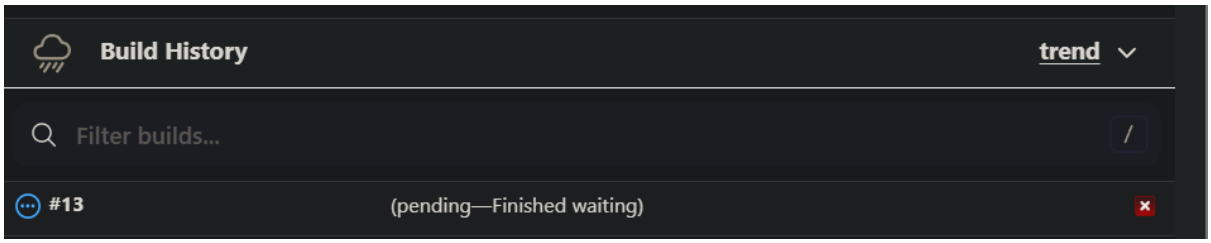


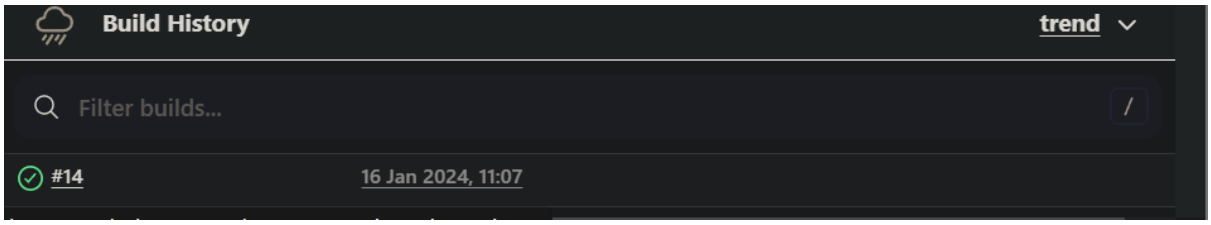
After chaning the code int the github account code




```
90
91      <body>
92          <h1>Todo List - Made using Node</h1>
93          <ul>
94              <% todolist.forEach(function(todo, index) { %>
95              <li>
96                  <a href="/todo/delete/<%= index %>" class="delete-btn">✗</a>
97                  <a href="/todo/<%= index %>" class="edit-btn">✎</a>
98                  <%- todo %>
99              </li>
100             <% }); %>
101         </ul>
102
103         <form action="/todo/add/" method="post">
104             <p>
105                 <label for="newtodo">What should I do Right Now?</label>
106                 <input type="text" name="newtodo" id="newtodo" autofocus />
107                 <input type="submit" value="Add" />
108             </p>
109         </form>
110     </body>
111 </html>
```

New build history generated like this pattern





If success it is shown like this

Dashboard > todo-node-app > #14 > Console Output

```
Sending build context to Docker daemon  25.32MB

Step 1/7 : FROM node:12.2.0-alpine
 ---> f391dabf9dce
Step 2/7 : WORKDIR app
 ---> Using cache
 ---> 2f69d41236b4
Step 3/7 : COPY . .
 ---> 2d4cacf7d315
Step 4/7 : RUN npm install
 ---> Running in b5047dd31aa8
npm[0m[91m [0mWARN[0m[91m [0mread-shrinkwrap[0m[91m This version of npm is compatible with lockfileVersion@1, but package-lock.json was generated
for lockfileVersion@2. I'll try to do my best with it!
[0m
> ejs@2.7.4 postinstall /app/node_modules/ejs
> node ./postinstall.js

Thank you for installing [35mEJS[0m: built with the [32mJake[0m JavaScript build tool ([32mhttps://jakejs.com/[0m)

npm[0m[91m [0mWARN[0m[91m my-todolist@0.1.0 No repository field.
npm[0m[91m [0mWARN[0m[91m my-todolist@0.1.0 No license field.
[0m[91m
[0mupdated 291 packages and audited 291 packages in 10.387s
found 15 vulnerabilities (6 moderate, 6 high, 3 critical)
  run `npm audit fix` to fix them, or `npm audit` for details
Removing intermediate container b5047dd31aa8
 ---> d81275158fe5
Step 5/7 : RUN npm run test
 ---> Running in 57bfc24a99e5

> my-todolist@0.1.0 test /app
> mocha --recursive --exit
```

Dashboard > todo-node-app > #14 > Console Output

```
      ✓ Is returning 5 when adding 2 + 3
executes before every test
      ✓ Is returning 6 when multiplying 2 * 3
    Test2
executes before every test
      ✓ 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 57bfc24a99e5
 ---> 4c351919927f
Step 6/7 : EXPOSE 8000
 ---> Running in 642e8aa7af64
Removing intermediate container 642e8aa7af64
 ---> fe68f49b6722
Step 7/7 : CMD ["node","app.js"]
 ---> Running in 0d583d0d25be
Removing intermediate container 0d583d0d25be
 ---> 48330b6f2df9
Successfully built 48330b6f2df9
Successfully tagged node-app-todo:latest
+ docker run -d --name node-app-container5 -p 8000:8000 node-app-todo
d030e9b8e103a5314b2425efa2295217ada3443bbd8e21b165f071d2b84c9d3a
Finished: SUCCESS
```

REST API Jenkins 2.426.2