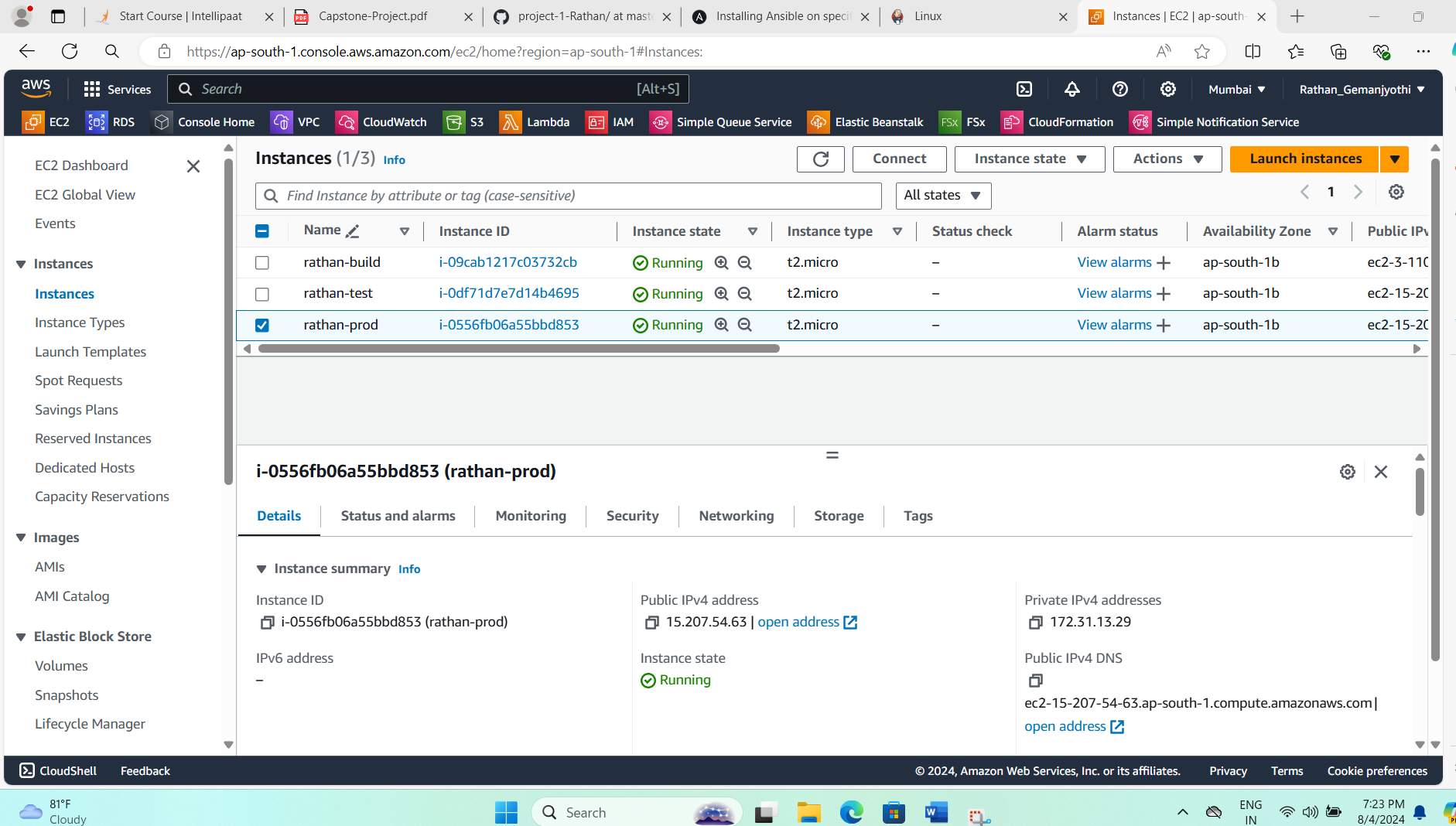


Create 3 ec2 instances or nodes Build, test and Prod



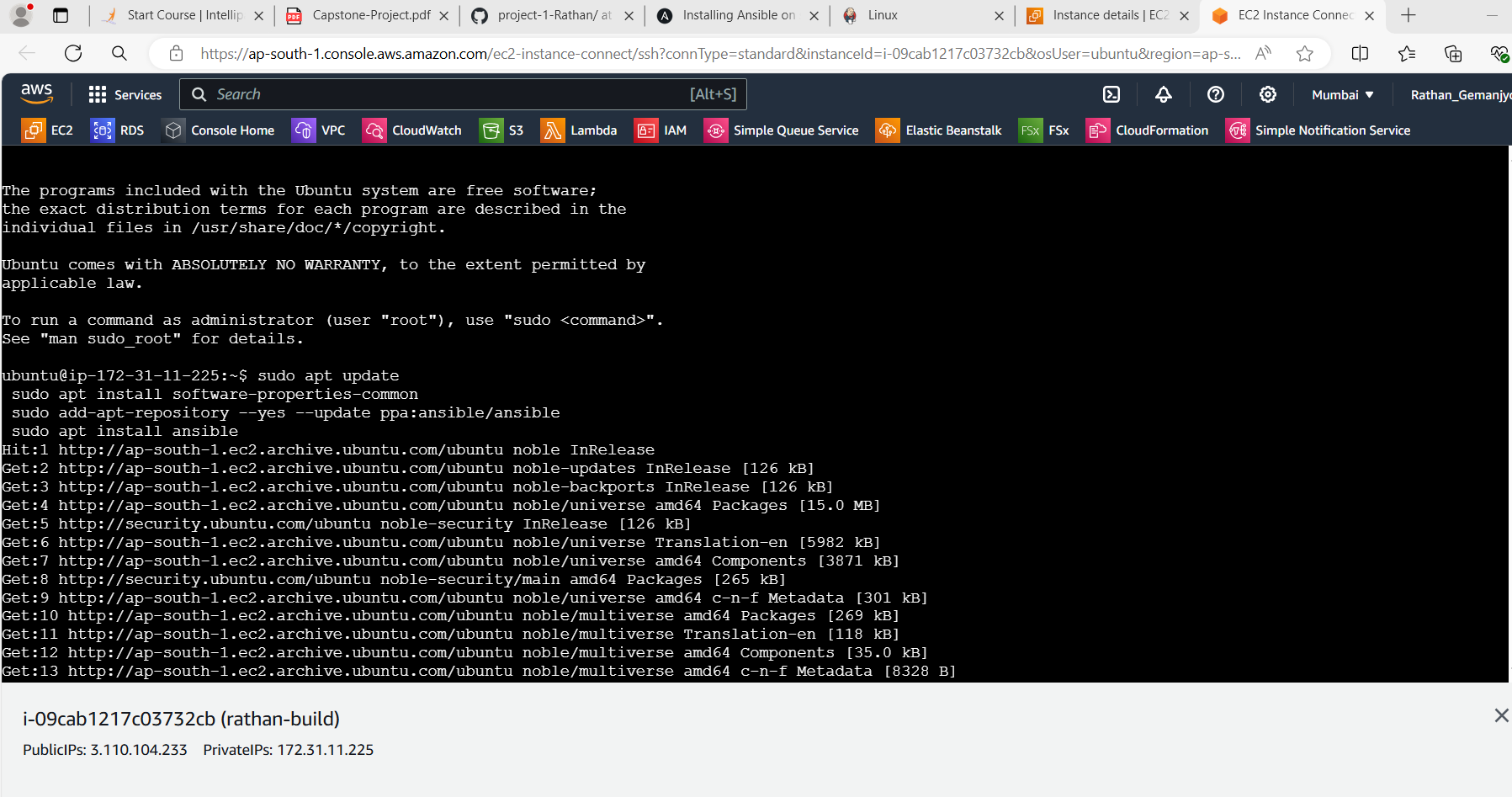
Install ansible on build node

**sudo apt update**

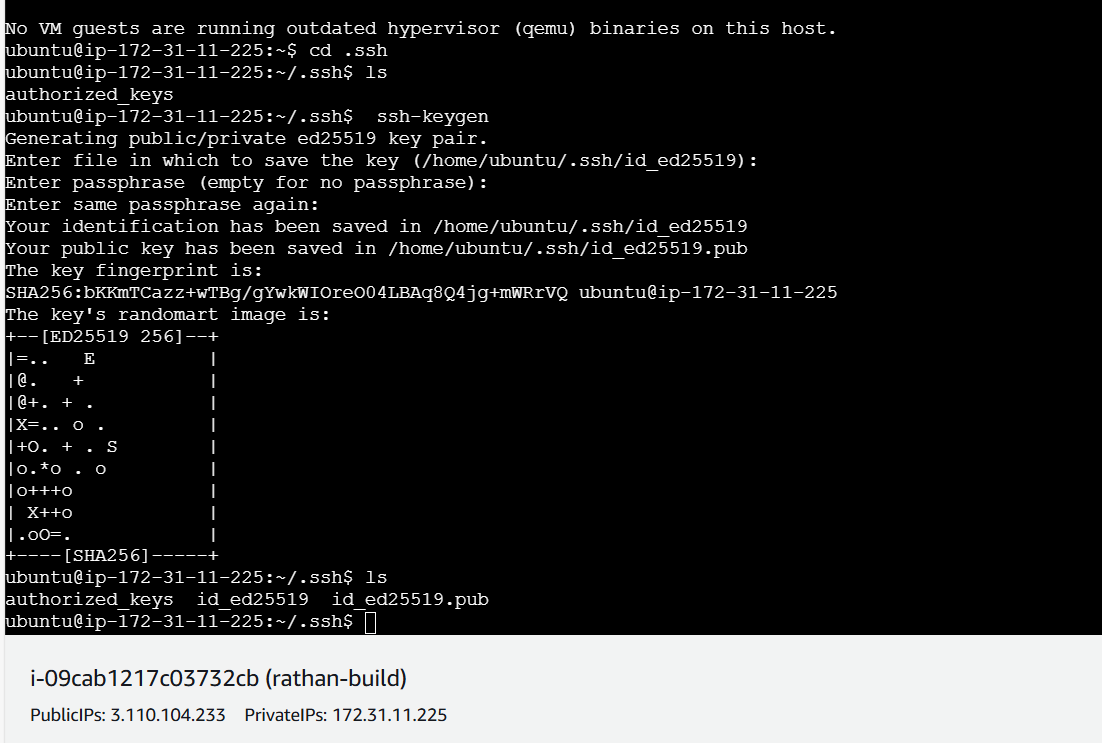
**sudo apt install software-properties-common**

**sudo add-apt-repository --yes --update ppa:ansible/ansible**

**sudo apt install ansible**

****

Configure the build node with test node and prod node by creating the keys by **ssh-keygen** at **.ssh** directory



Now paste the public key data in authorized\_keys of test node and prod node.

Now cd /etc/ansible

Ls

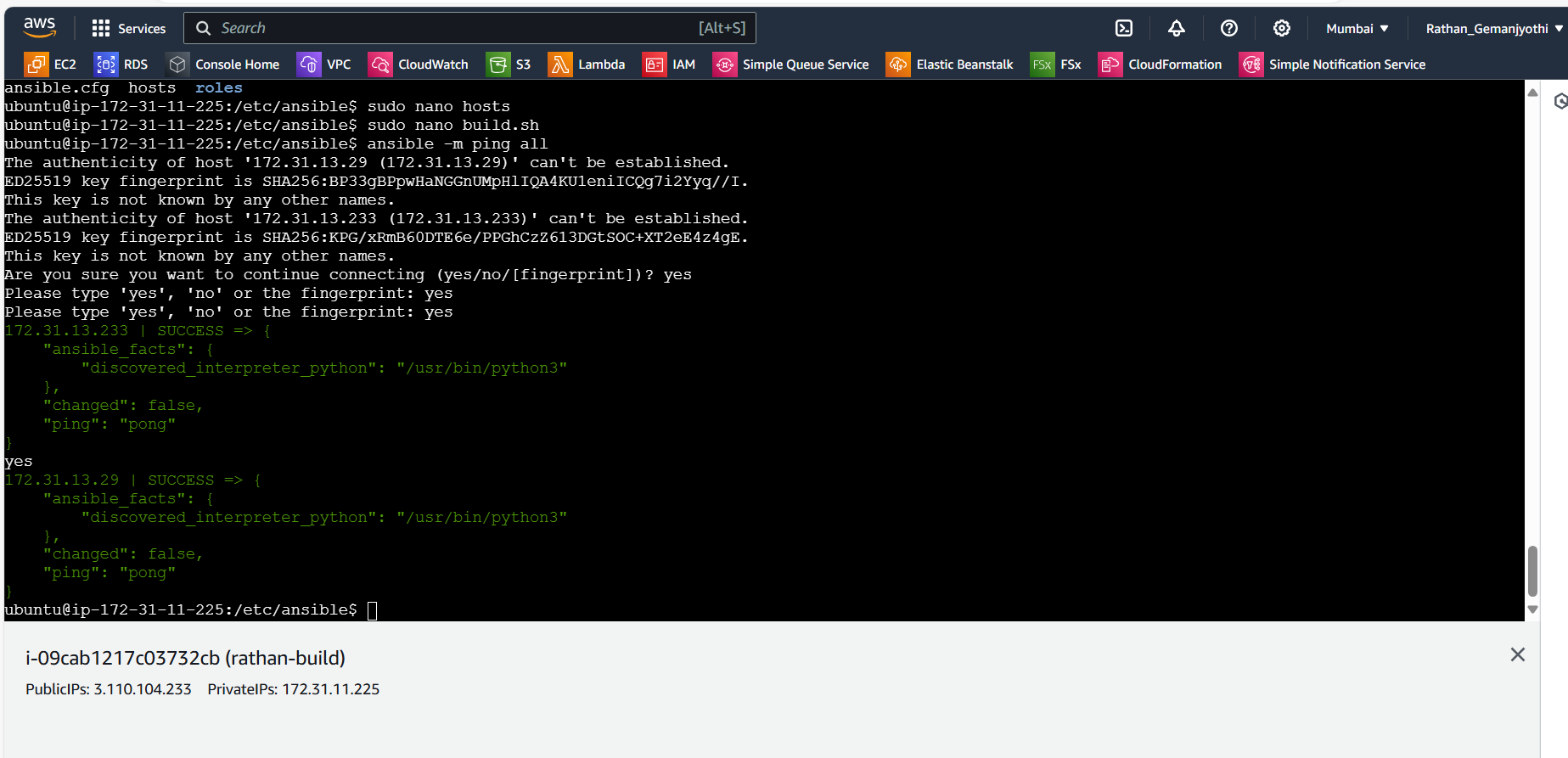
Sudo nano hosts

[prod]

Private ip

Private ip

Now do **ansible -m ping all**



If ping=pong connection is successful

Now create 3 files, need to install Jenkins in build node, need to install java and docker in test and prod nodes

commands to install java and Jenkins for build node in build.sh

commands to install java and docker for test and prod nodes in test.sh and prod.sh

**Build.sh**

**Sudo apt install openjdk-17-jdk -y**

**sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \**

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

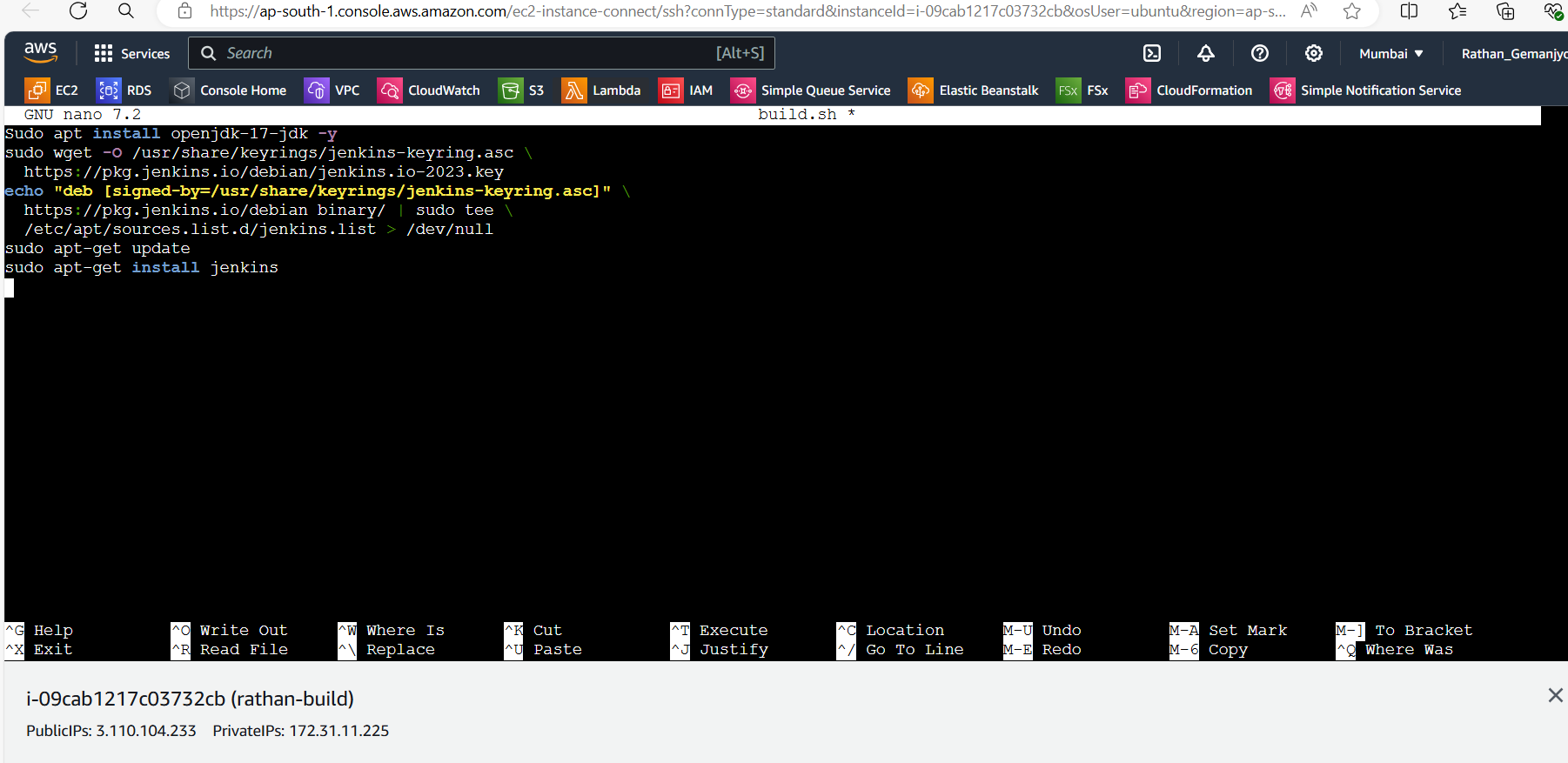
**echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]" \**

**https://pkg.jenkins.io/debian binary/ | sudo tee \**

**/etc/apt/sources.list.d/jenkins.list > /dev/null**

**sudo apt-get update**

**sudo apt-get install Jenkins**

****

Prod.sh

**Sudo apt update**

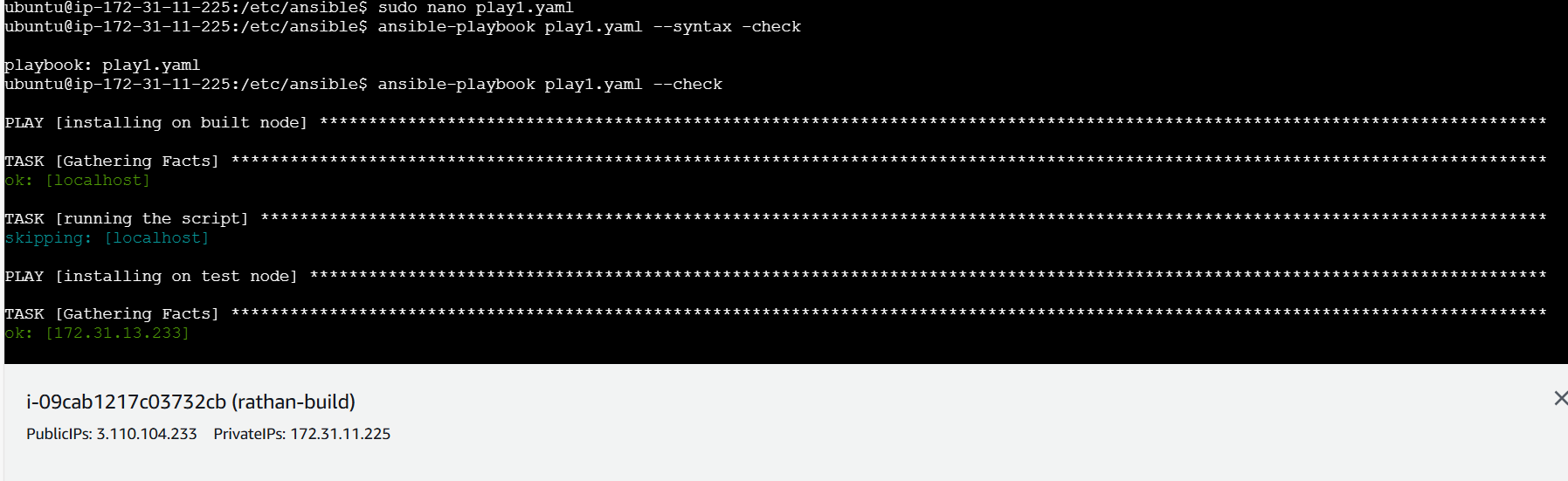
**Sudo apt install openjdk-17-jdk -y**

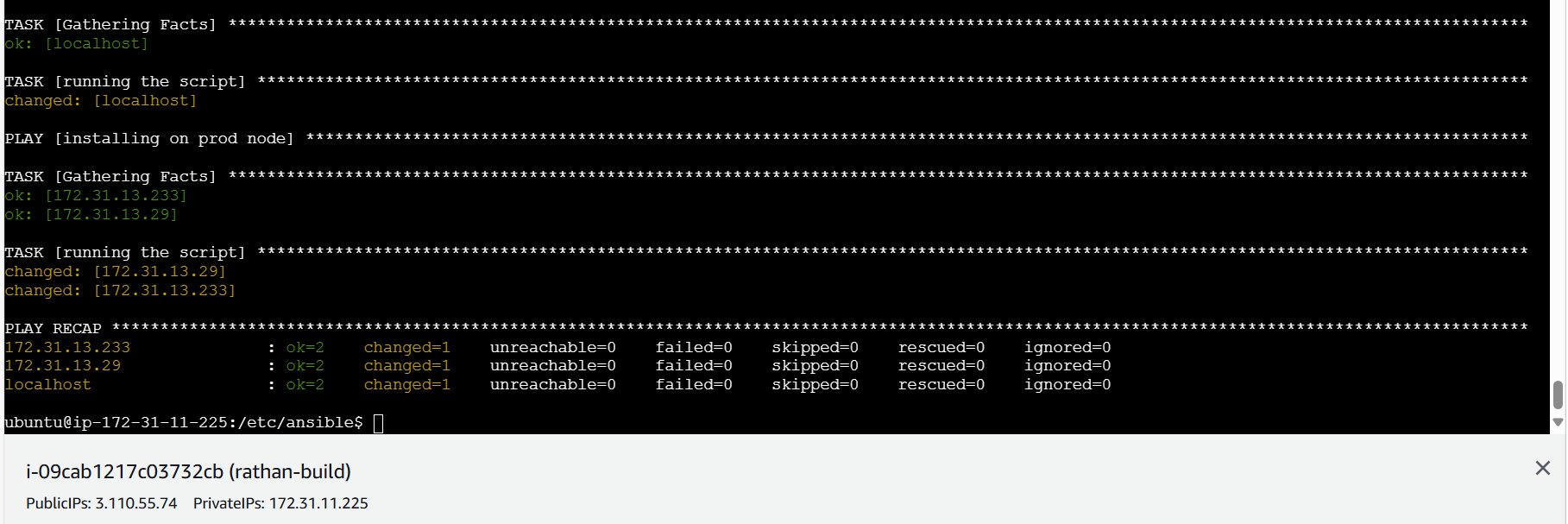
**Sudo apt install docker.io**

Sudo nano play1.yaml

Add the yaml data to install the java Jenkins and docker in the build test and prod







---

- name: installing on built node

hosts: localhost

become: true

tasks:

- name: running the script

script: build.sh

---

- name: installing on nodes

hosts: prod

become: true

tasks:

- name: running the script

script: prod.sh

Ansible-playbook play1.yaml – syntax -check

Anisble-playbook play1.yaml – check

Anisble-playbook play1.yaml

--------------------------------------------------------------------------------------------------------------------------------------- Now fork the url provided and create a **Dockerfile** and add commands to create the file to run the container with apache2 on container

Now

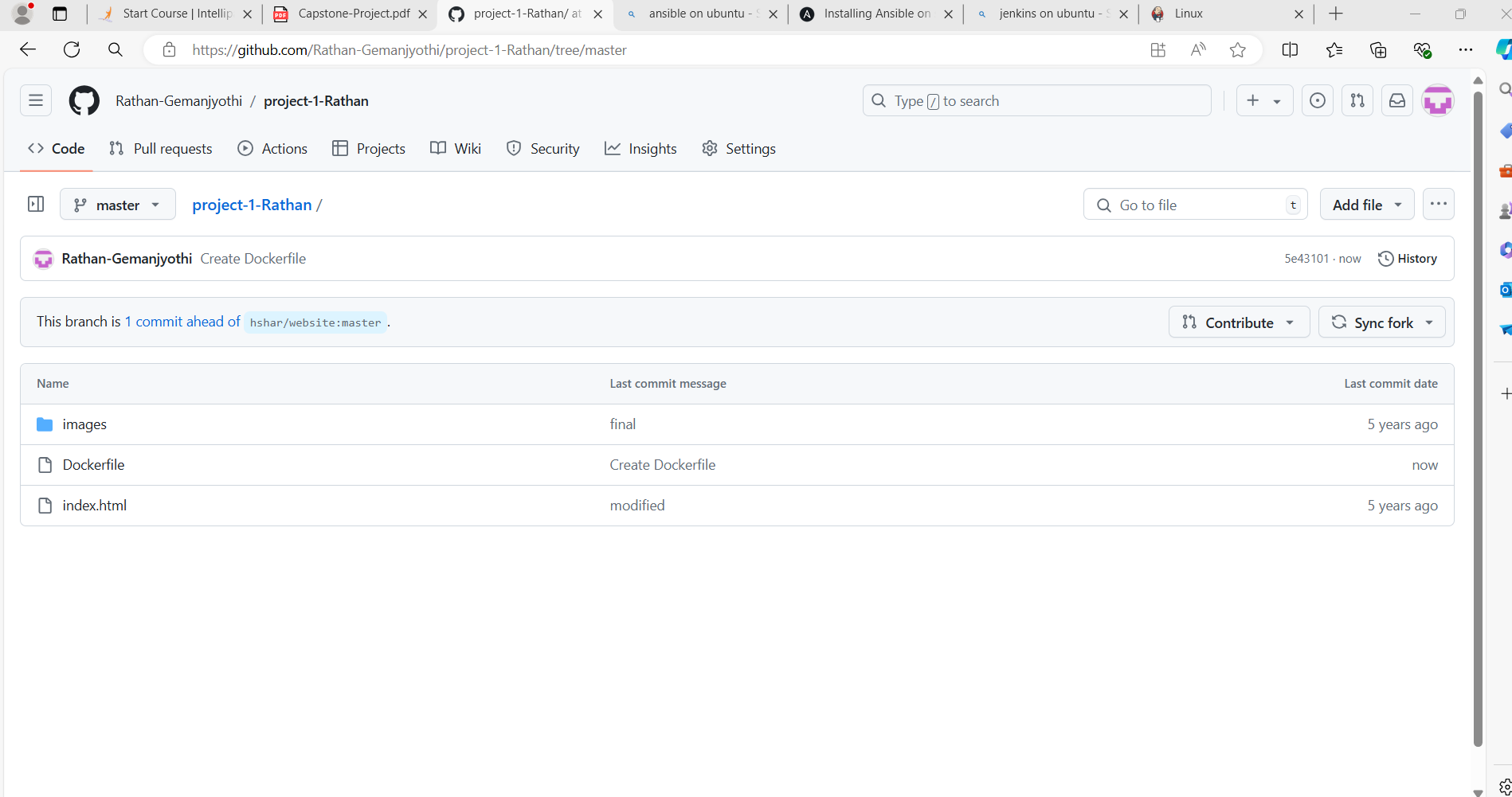
**FROM ubuntu**

**RUN apt update**

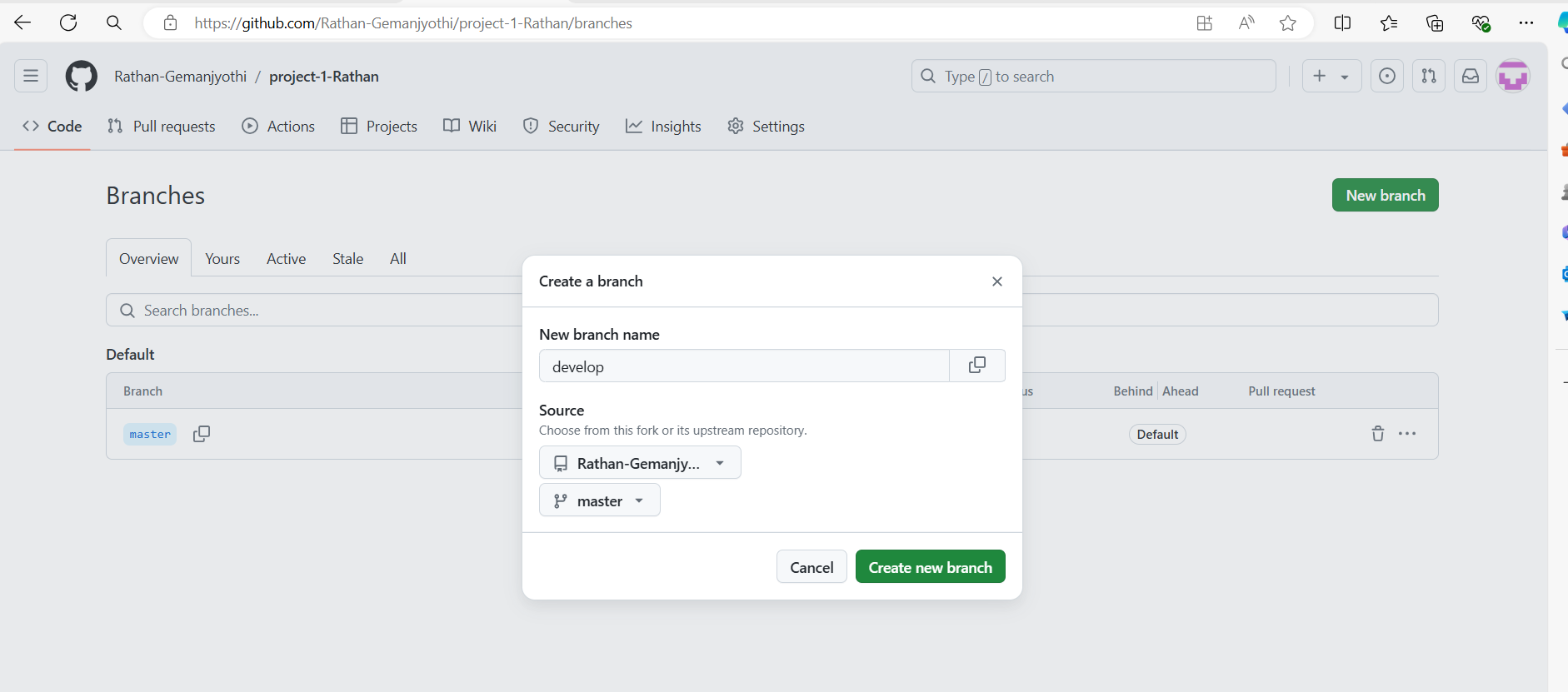
**RUN apt install apache2 -y**

**ADD . /var/www/html/**

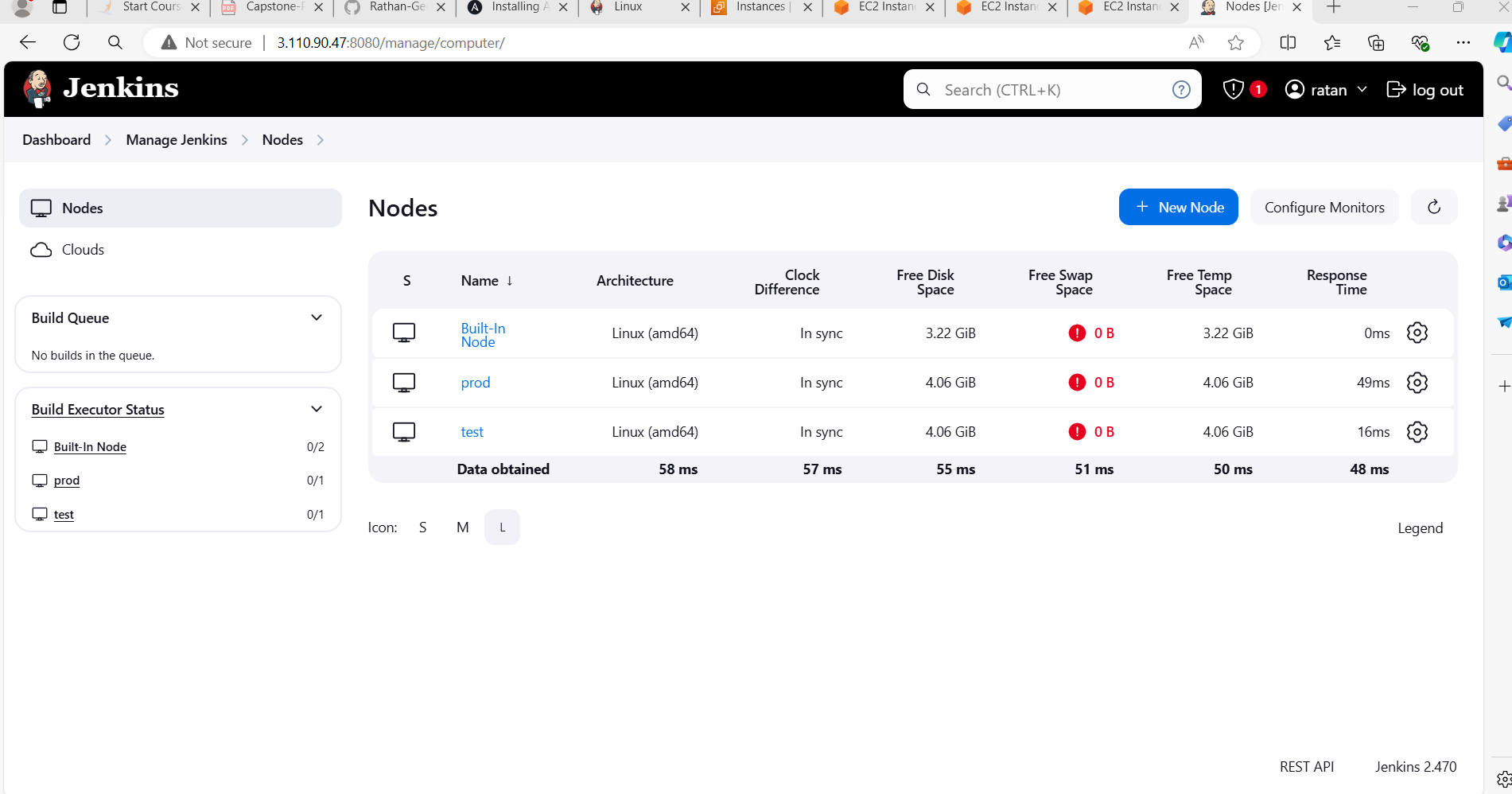
**ENDPOINT apachectl -D FOREGROUND**

****

And also creating the branch in that as develop



Now setup Jenkins and adding ssh agent plugin and connecting build node with test and prod node



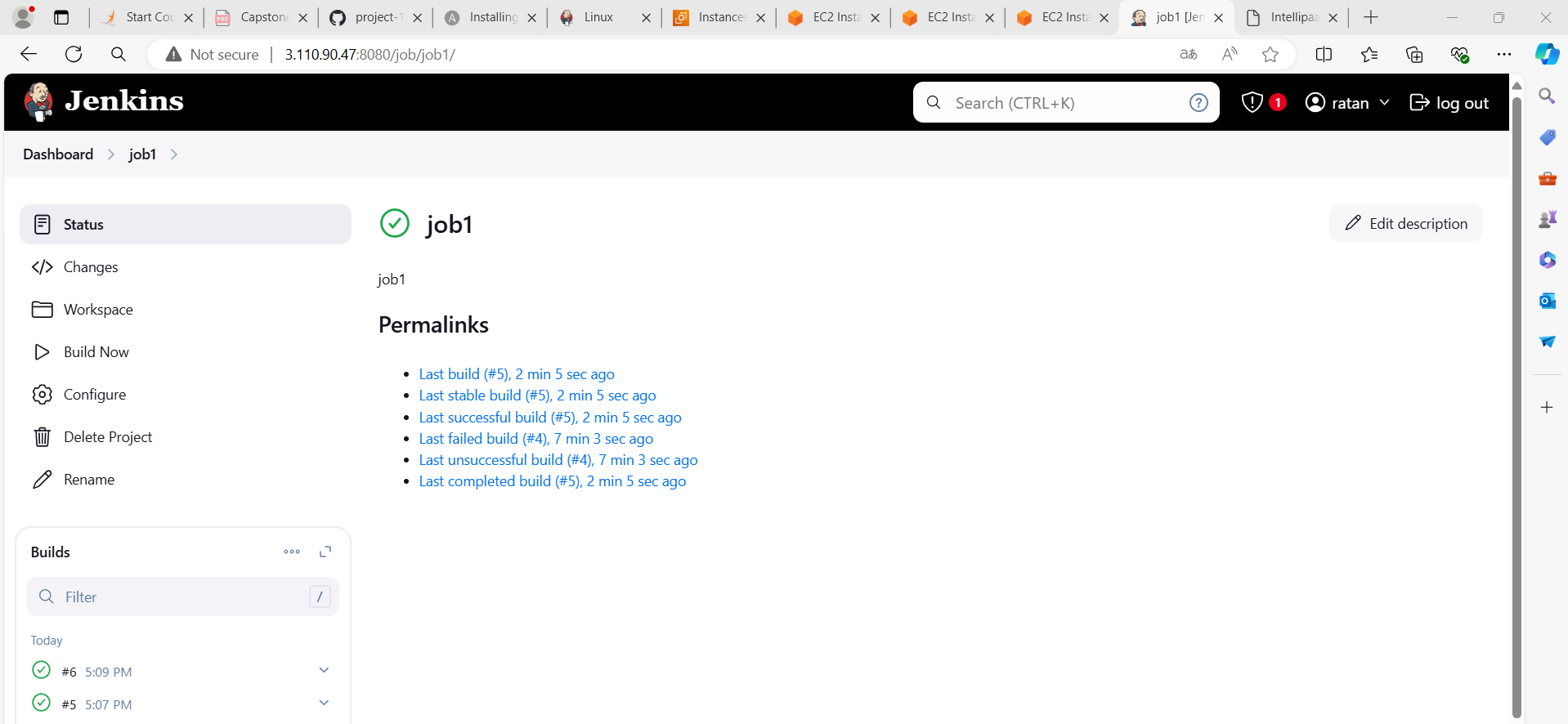
Now creating job1 and giving git URL on develop branch and giving it as ristrict job on test node and under build triggers select the execute shell

Here am building the comtainer from the dockerfile

**Sudo docker rm -f c1**

**Sudo docker build /home/ubuntu/jenkins/workspace/job1/ -t job1**

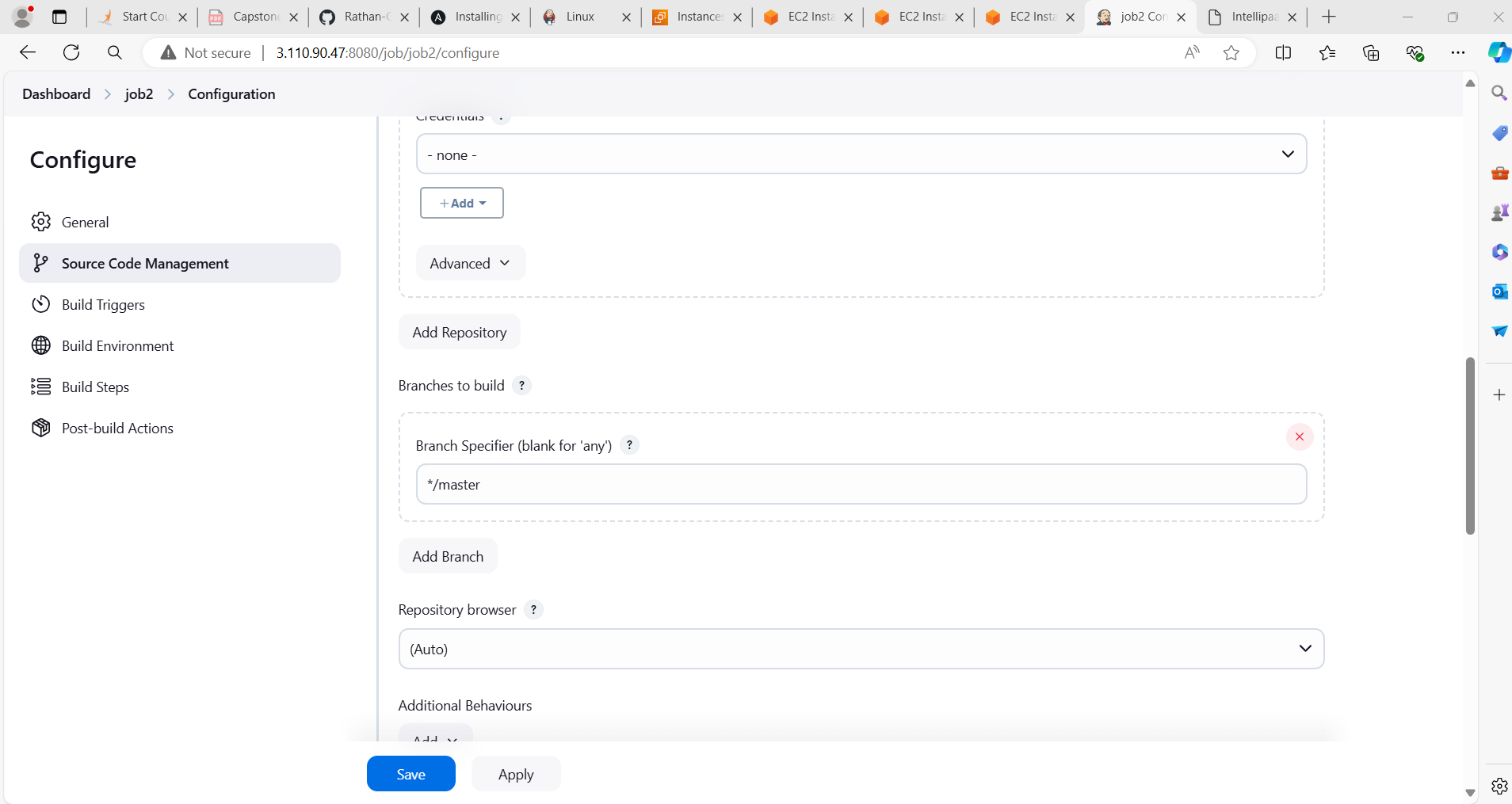
**Sudo docker run -itd -p 80:80 –-name=c1 job1**

****

****

This will delete previous build and create image everytime we click build.

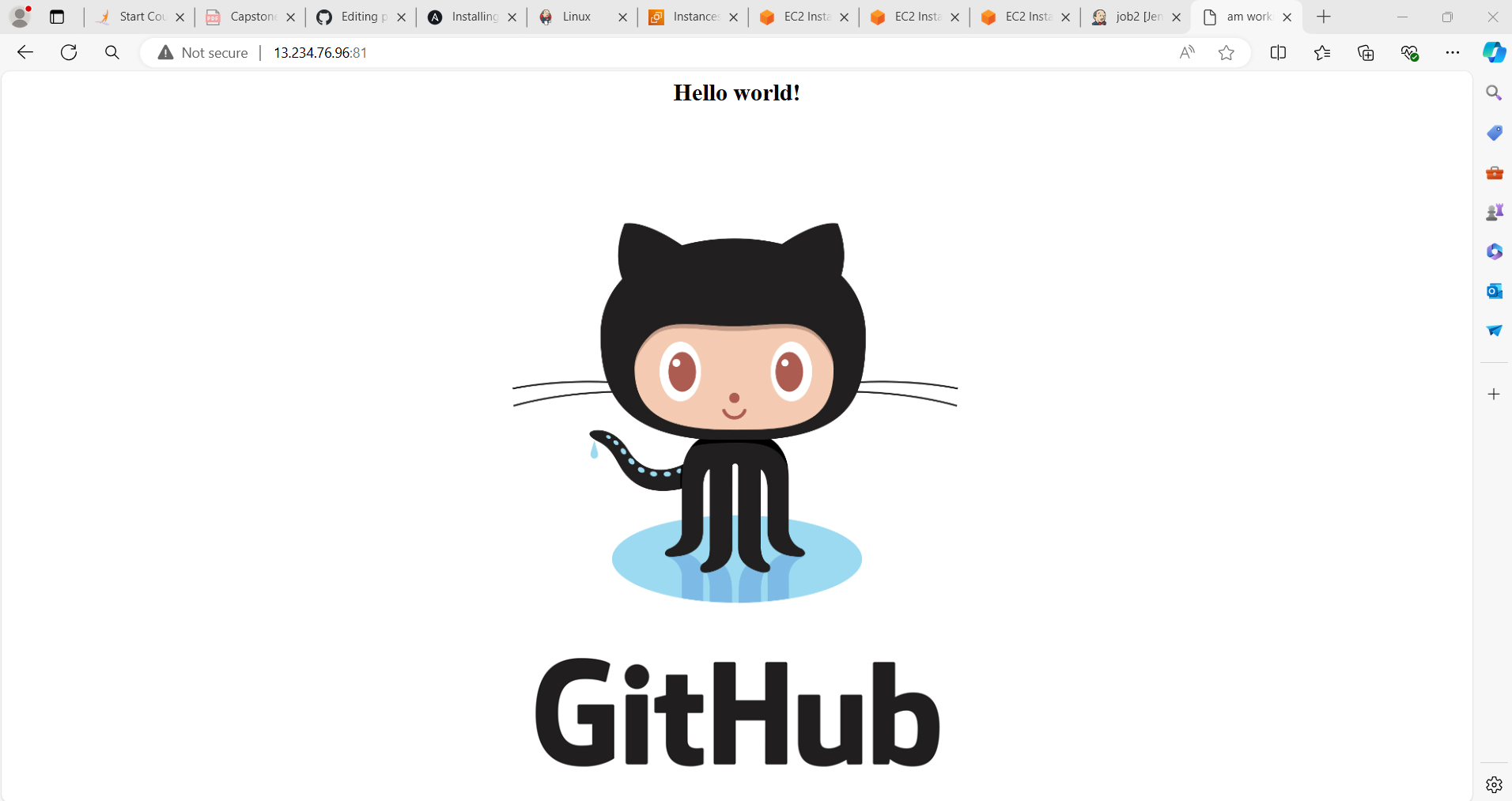
Now similarly create job2 givning url and build it on master node and ristrict job on test node and whenever there is code change build should trigger to do that we use github-webhook with ip in git hub and under build triggers select the execute shell and add commands

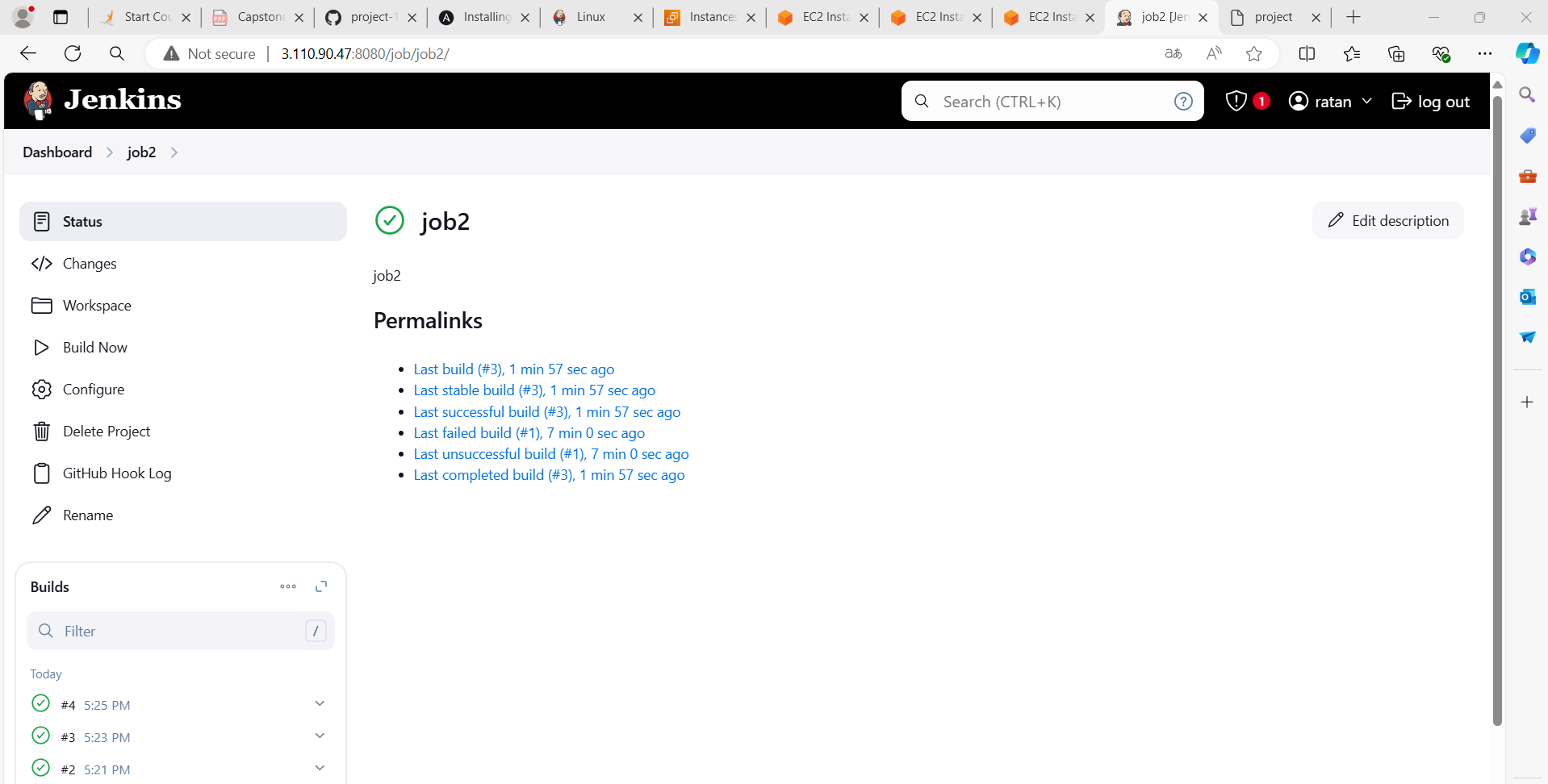


**Sudo docker rm -f c2**

**Sudo docker build /home/ubuntu/jenkins/workspace/job2/ -t job2**

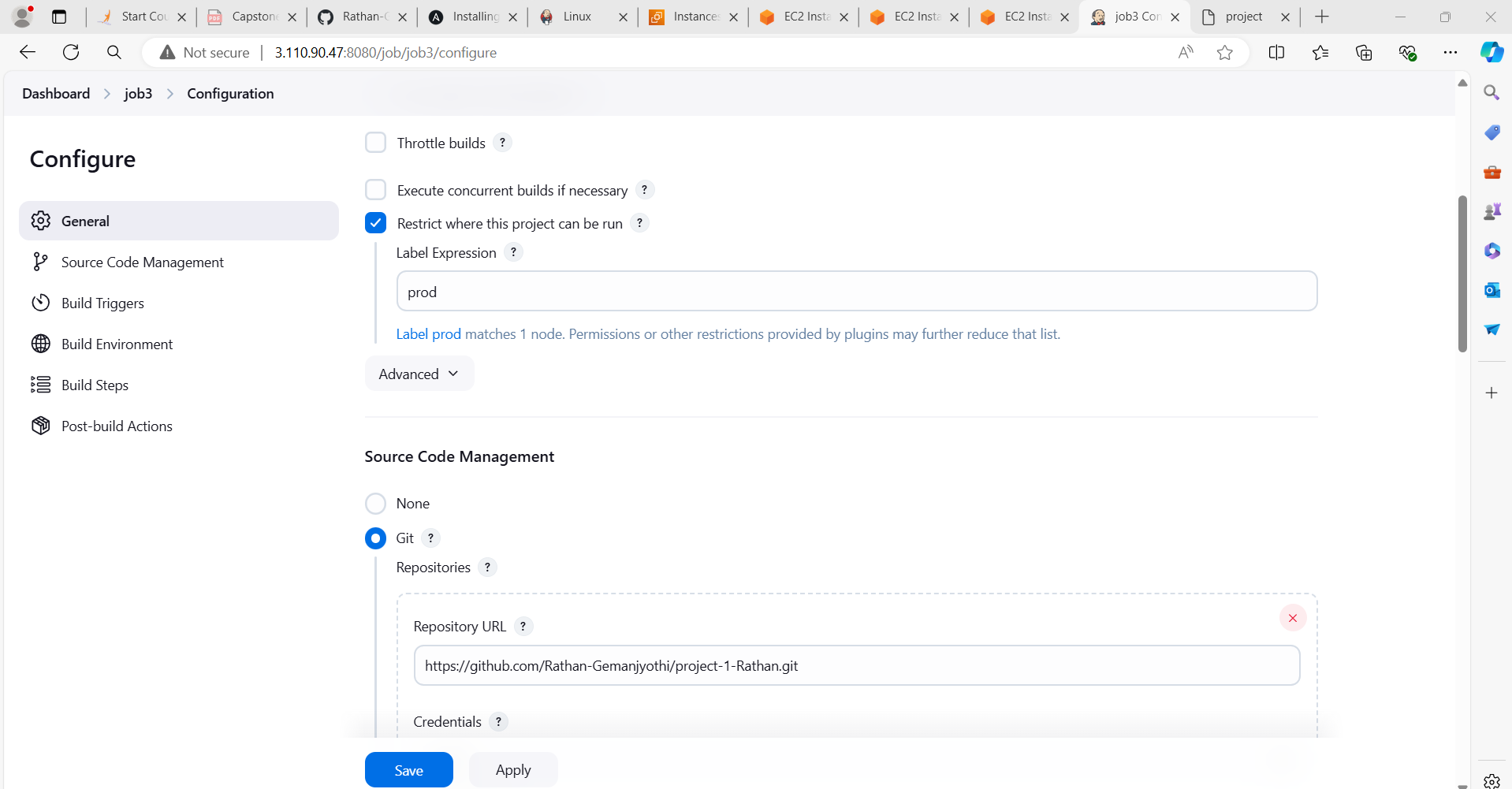
**Sudo docker run -itd -p 81:80 –-name=c2 job2**

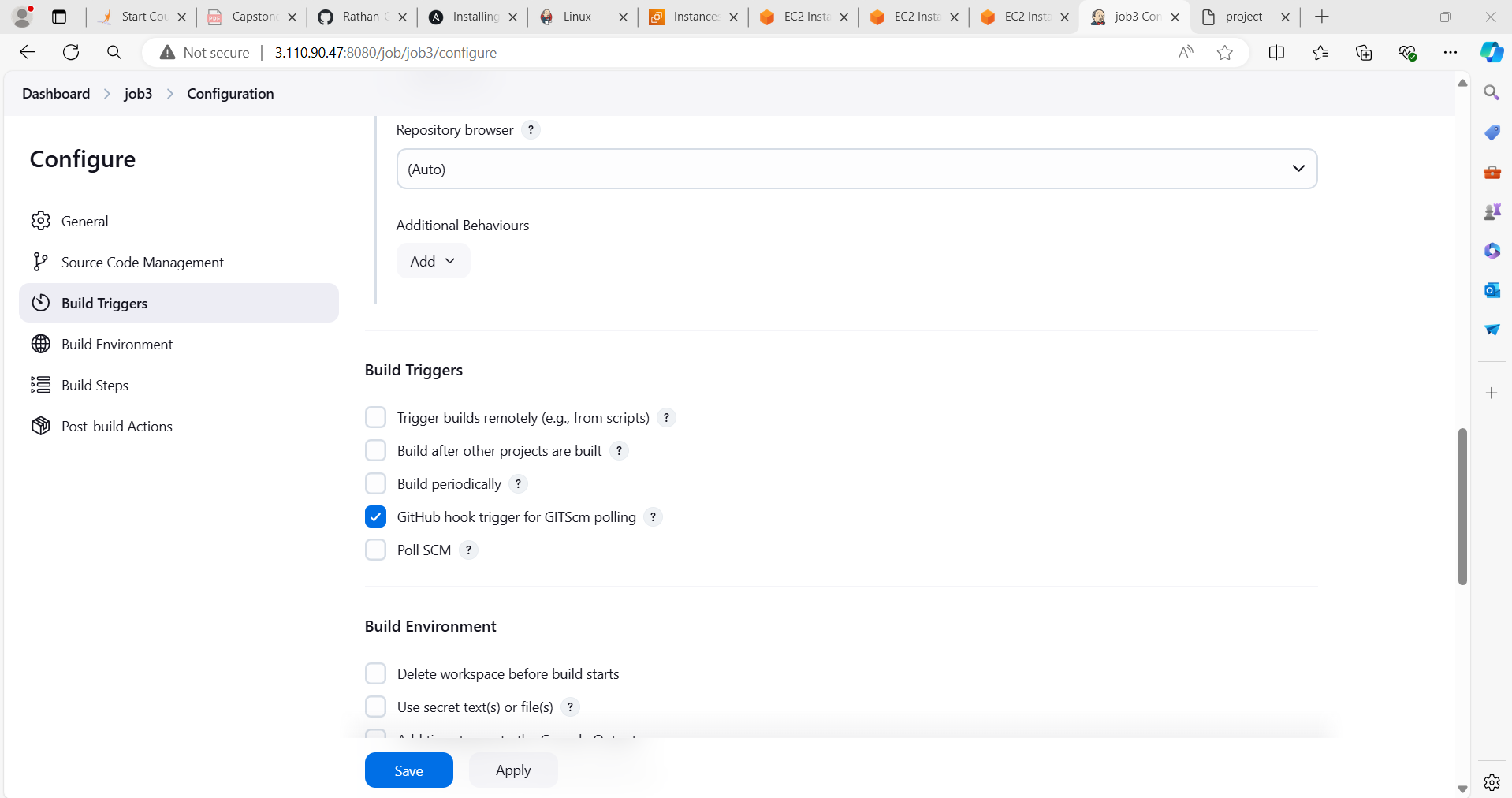
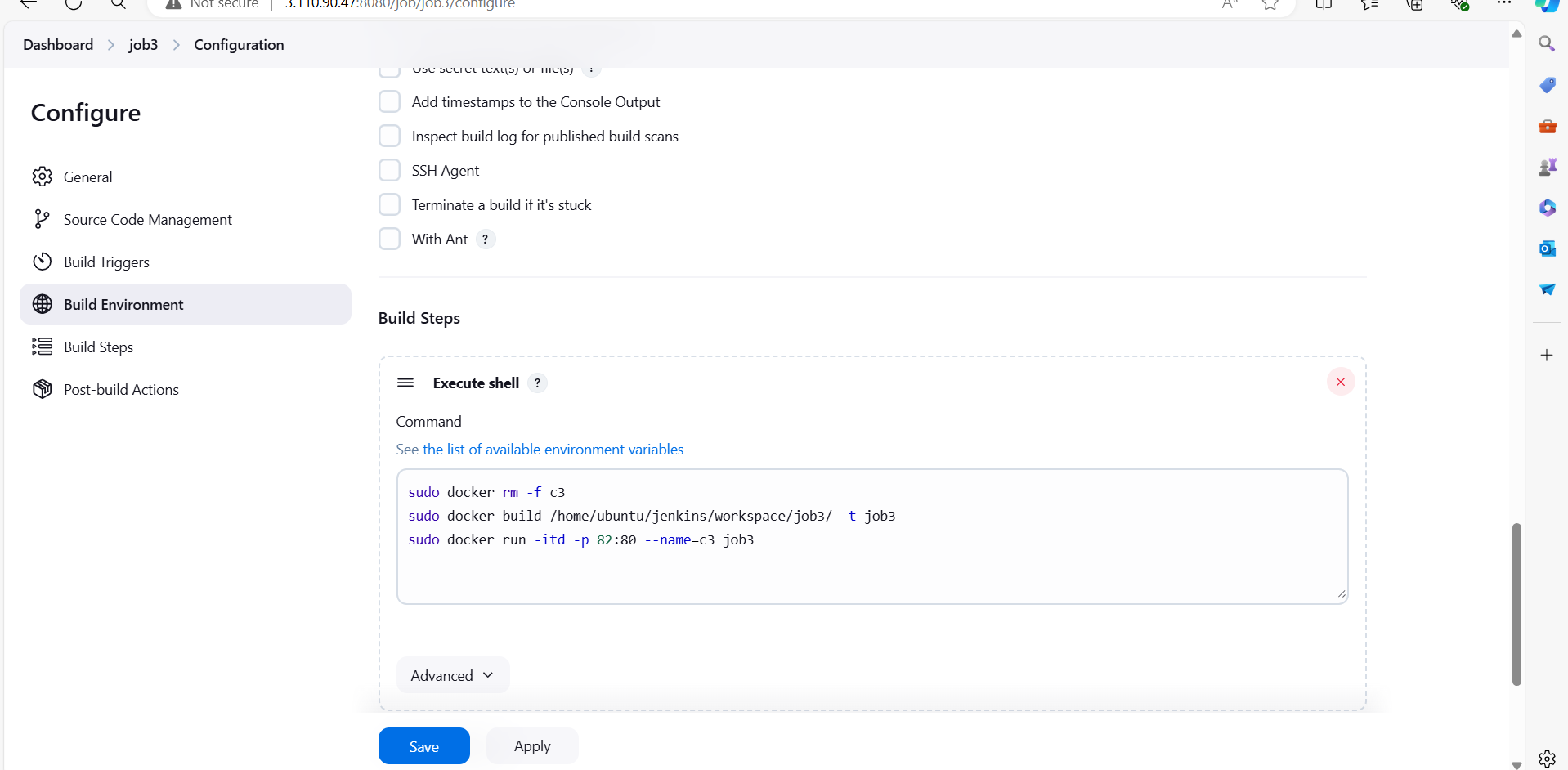
****

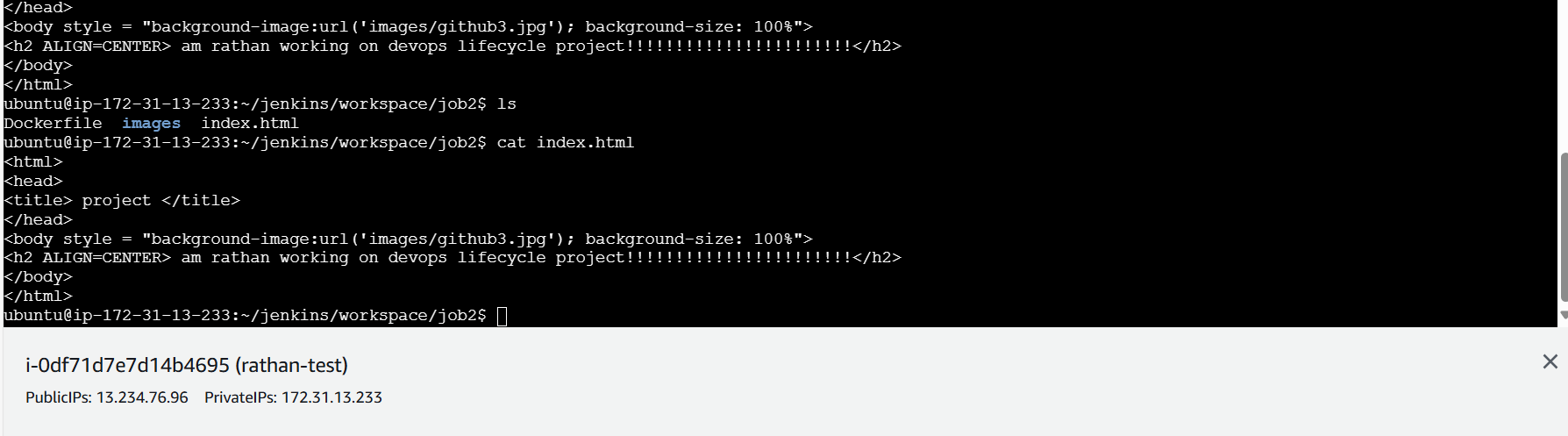
****

****

**Similarly**

****

**** ****

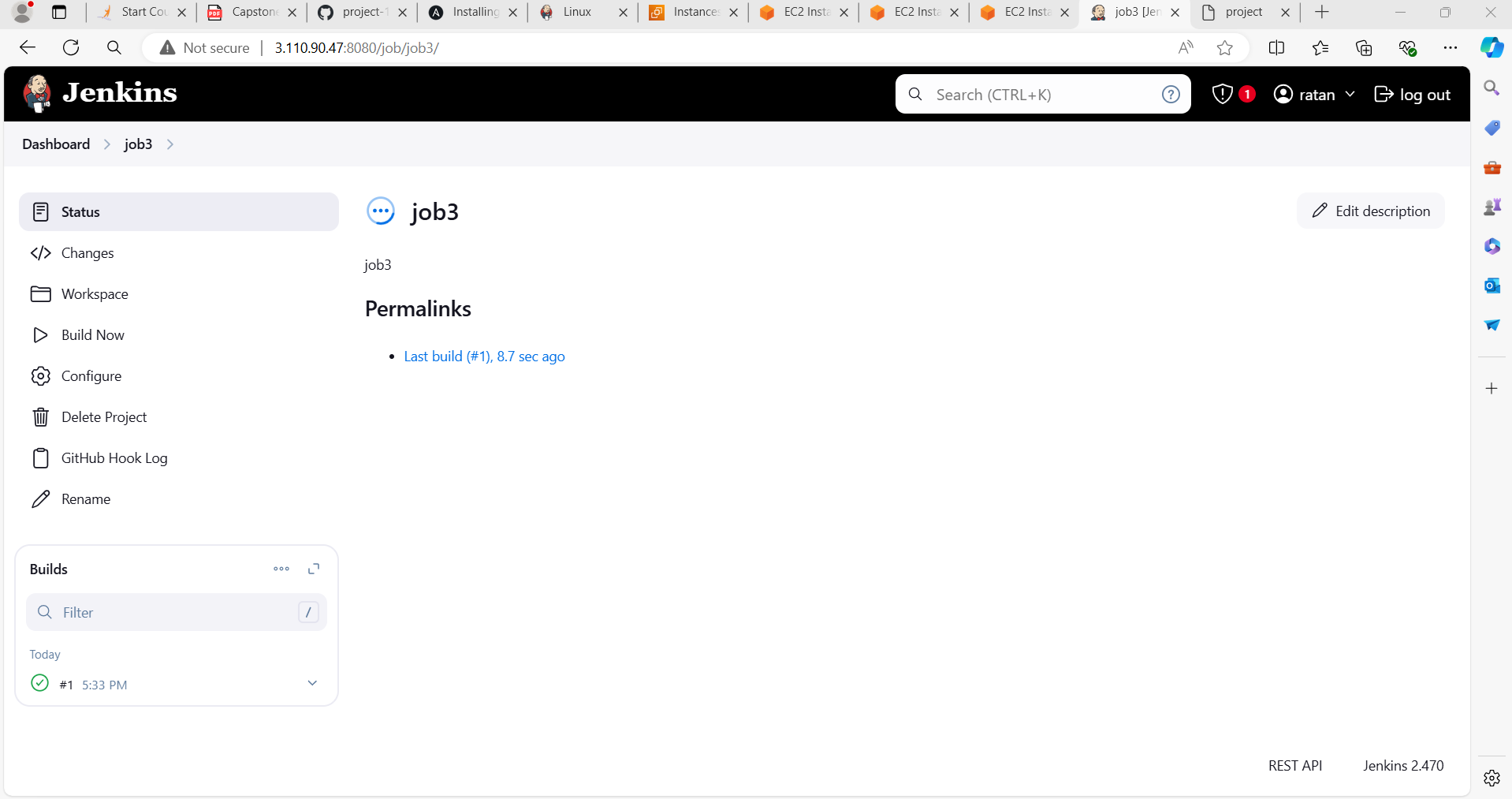
****

Now similarly create job3 givning url and build it on master node and ristrict job on prod node and whenever there is code change build should trigger to do that we use github-webhook with ip in git hub and under build triggers select the execute shell and add commands

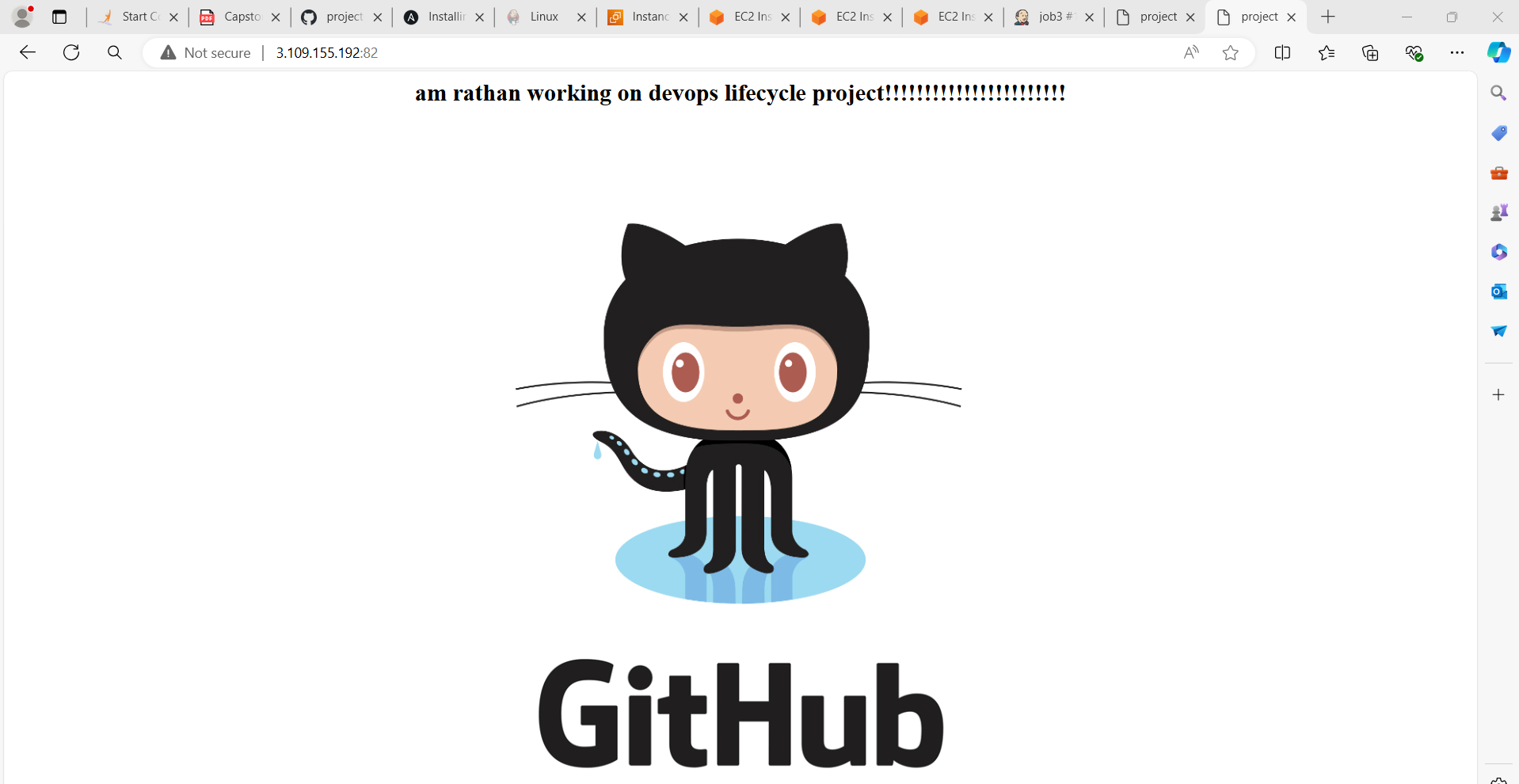
**Sudo docker rm -f c3**

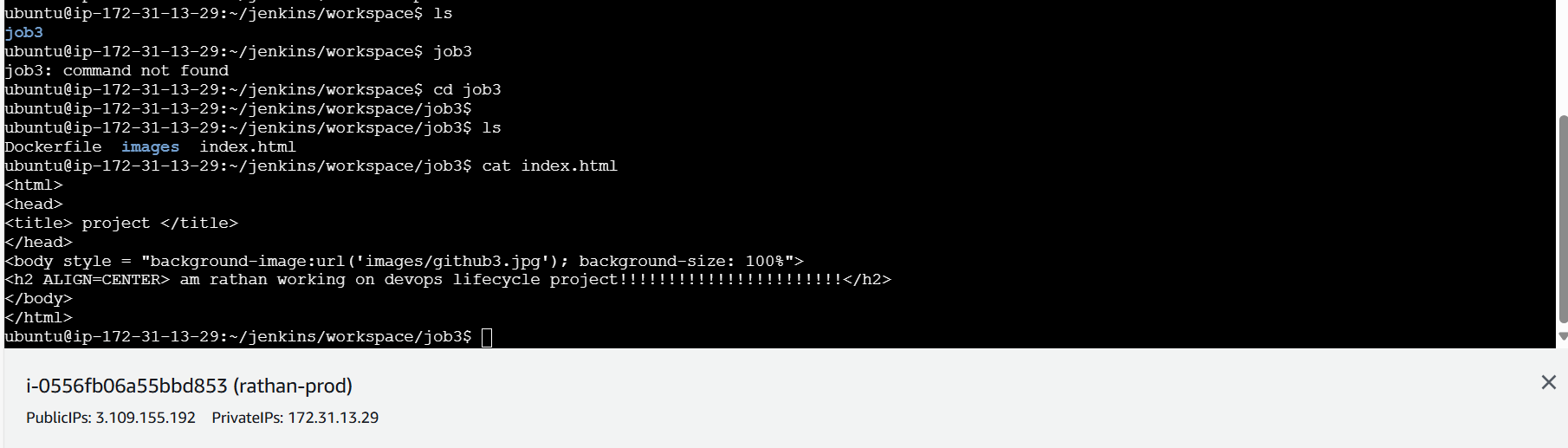
**Sudo docker build /home/ubuntu/jenkins/workspace/job3/ -t job3**

**Sudo docker run -itd -p 82:80 –-name=c3 job3**

****

**And whenever we do any changes on master a new docker file get created and with execute shell of build trigger a new container will be created with updated change**

****

****