You have been hired as a Sr. DevOps Engineer in Abode Software. They want to implement DevOps Lifecycle in their company. You have been asked to implement this lifecycle as fast as possible. Abode Software is a product-based company and their product is available on this GitHub link.

<https://github.com/hshar/website.git>

Following are the specifications of the lifecycle:

1. Install the necessary software on the machines using a configuration management tool

2. Git workflow must be implemented

3. CodeBuild should automatically be triggered once a commit is made to master branch or develop branch.

a. If a commit is made to master branch, test, and push to prod

b. If a commit is made to develop branch, just test the product, do not push to prod

4. The code should be containerized with the help of a Dockerfile. The Dockerfile should be built every time there is a push to GitHub. Use the following pre-built container for your application: hshar/webapp the code should reside in '/var/www/html'

5. The above tasks should be defined in a Jenkins Pipeline with the following jobs:

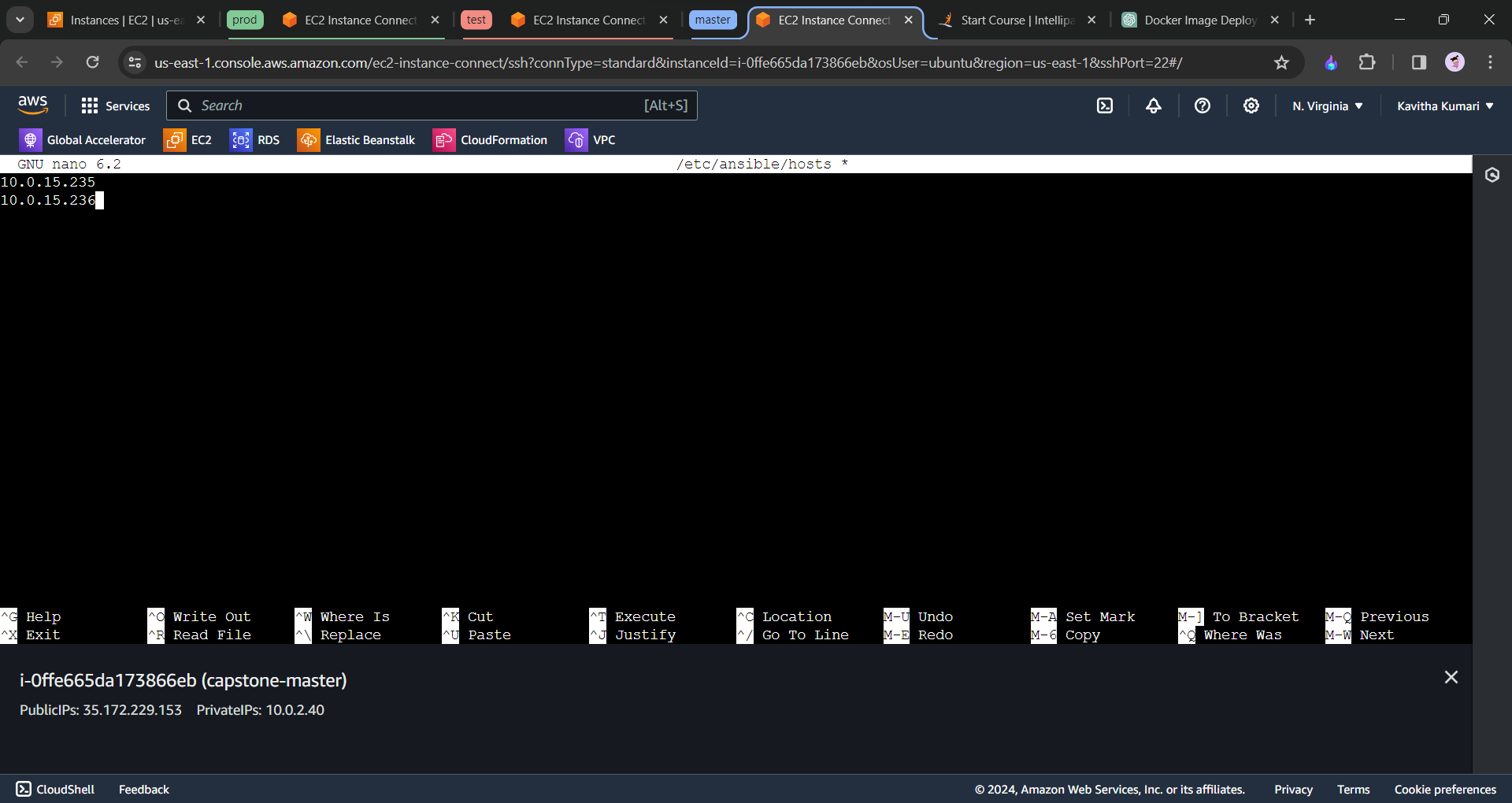
a. Job1: build

b. Job2:test

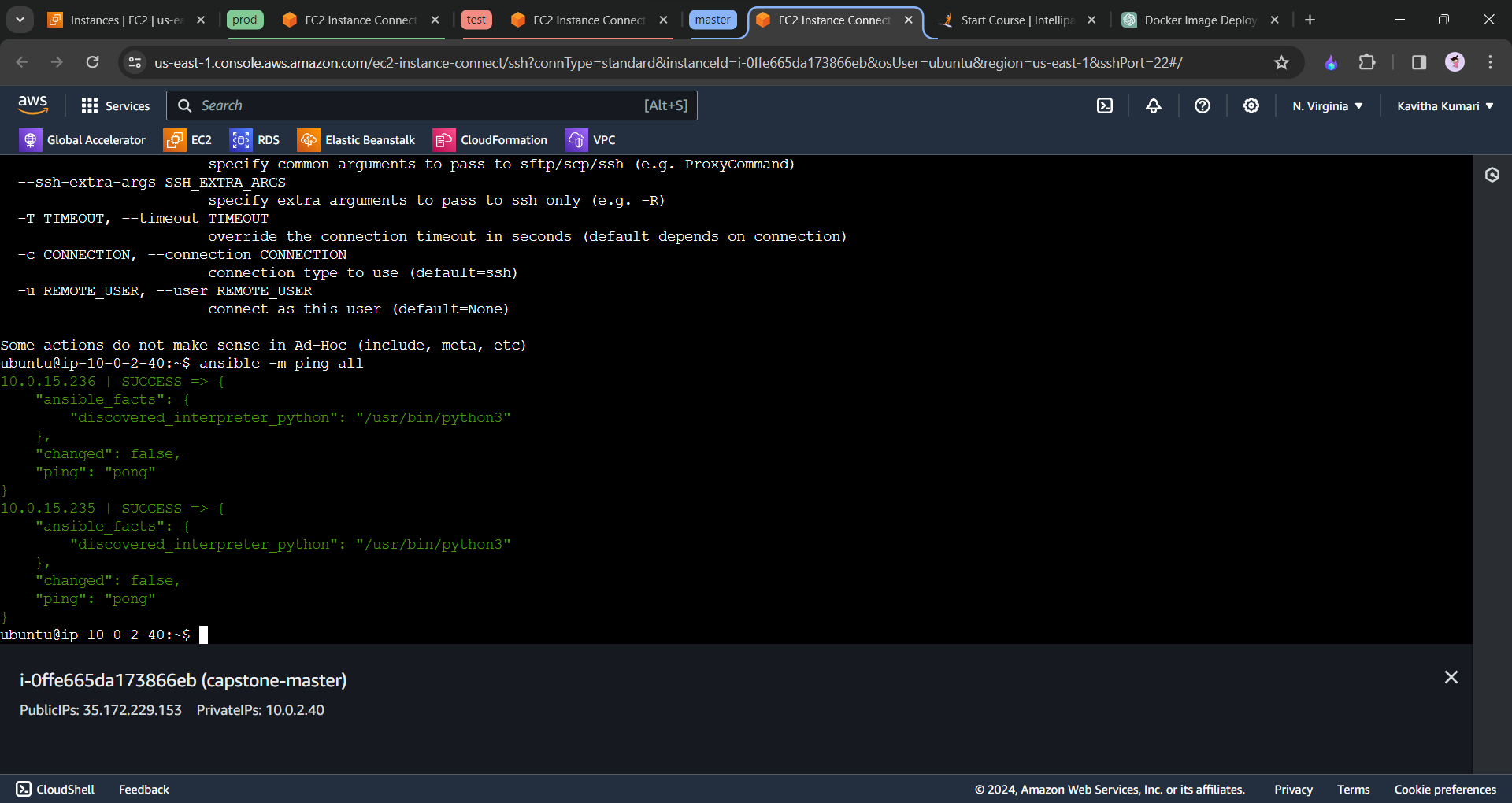
c.Job3: prod

Procedure: -

* Ansible master= jenkins and java with the help of ansibles
* java, docker on the slaves
* one = master, two and three= slaves. Total three instances
* install ansible on the master
* keygen to connect all the slave with the master
* sudo nano /etc/ansible/hosts and paste the private ips of both of the slaves



* save and exit from the file.
* Check whether we are sucessfully ping the test and prod “ansible -m ping all”



* Now we will create playbook to install jenkins and java on the slave machine
* Sudo nano play.yaml

---  
- name: tasks for the master  
 hosts: localhost //master machine  
 become: true  
 tasks:  
 - name: executing script on master  
 script: master.sh  
  
- name: tasks for slave  
 hosts: all //refers to all the slave machine  
 become: true  
 tasks:  
 - name: executing script on slave  
 script: slave.sh

* sudo nano master.sh

sudo apt update

sudo apt install openjdk-11-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

--------------------------------or-------------------

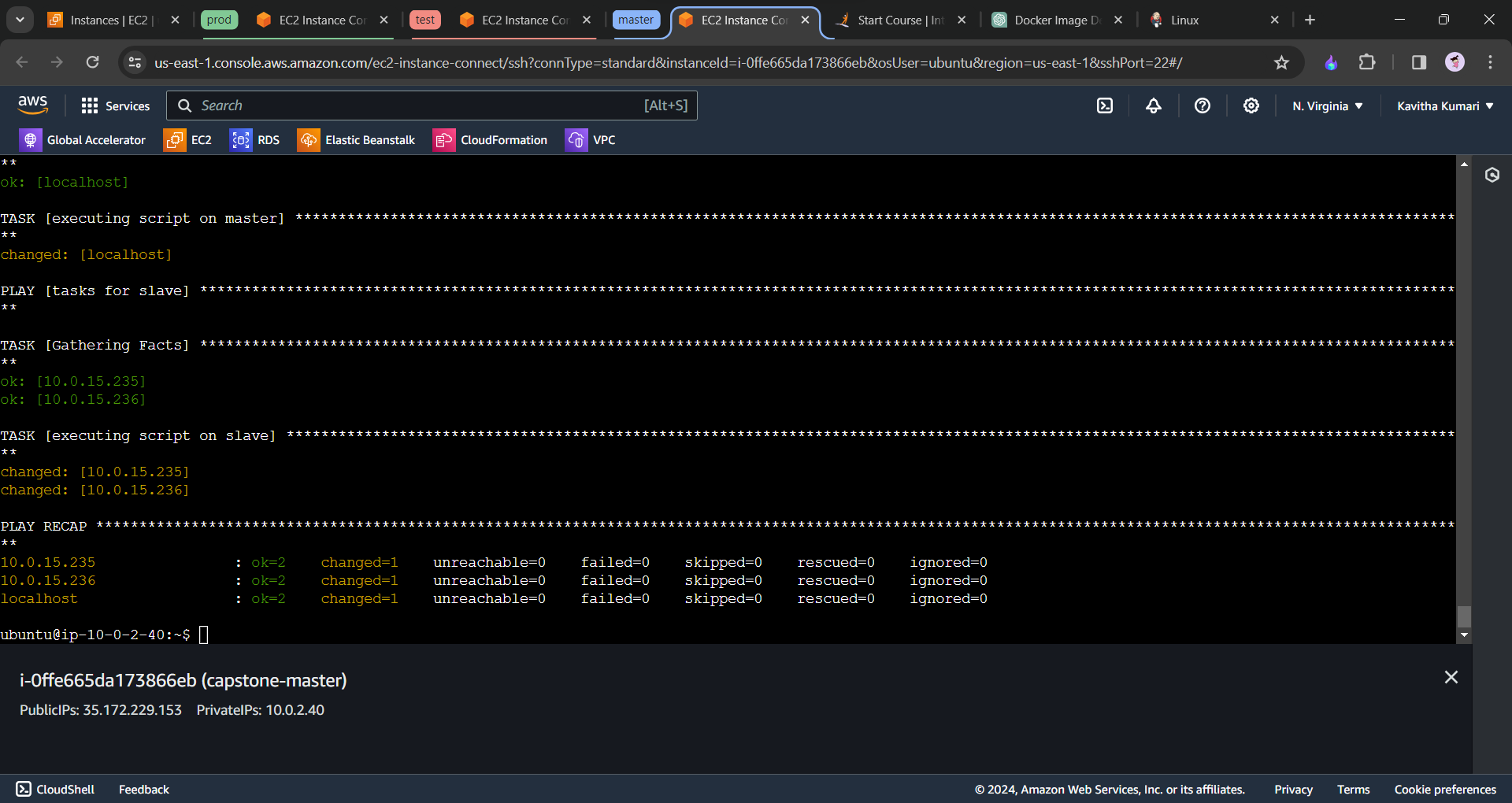
* slave.sh

sudo apt update

sudo apt install openjdk-11-jdk -y

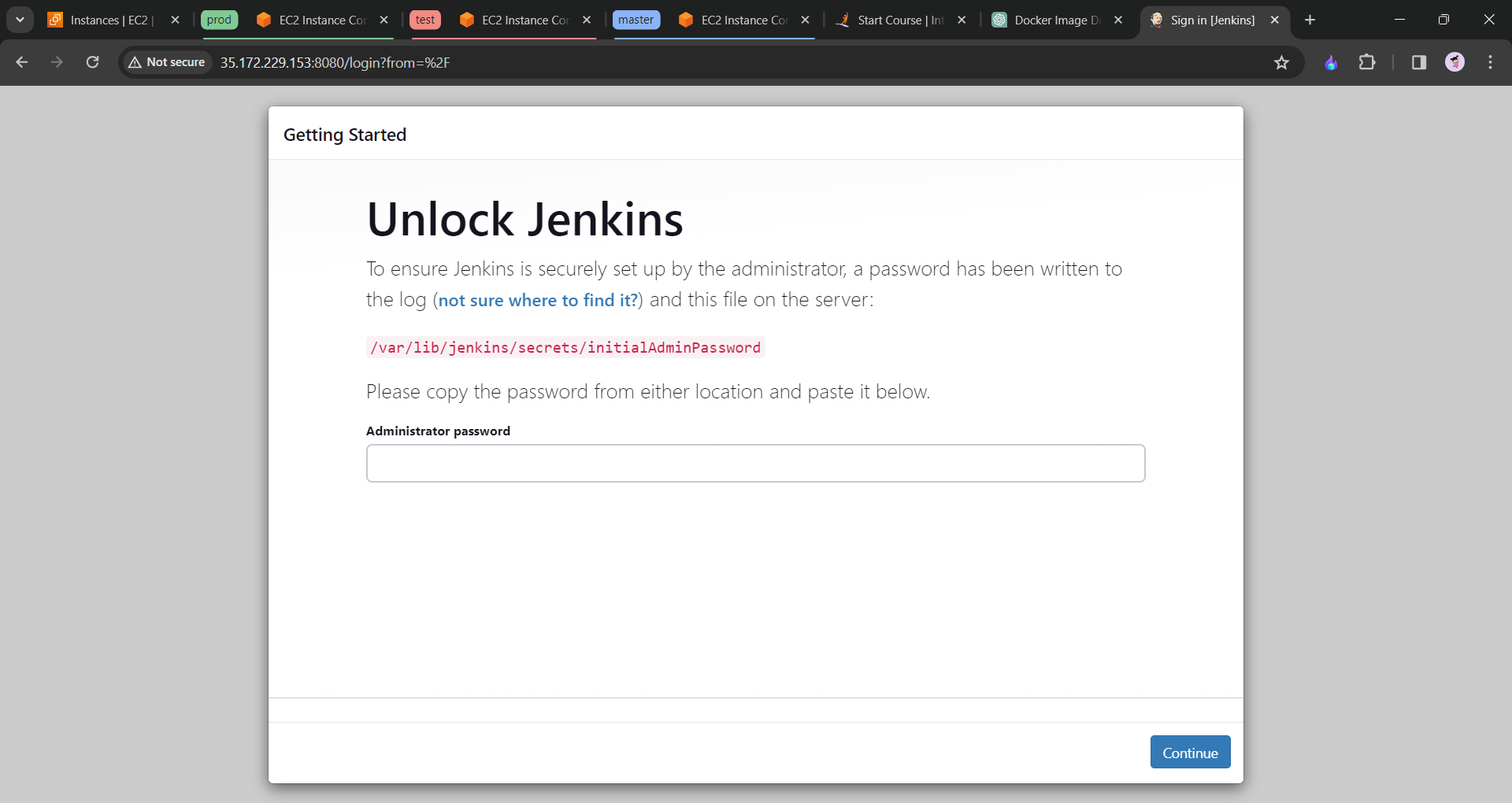
sudo apt install docker.io -y

* ansible-playbook play.yaml –-syntax-check
* ansible-playbook play.yaml –-check //for the dry run
* ansible-playbook play.yaml

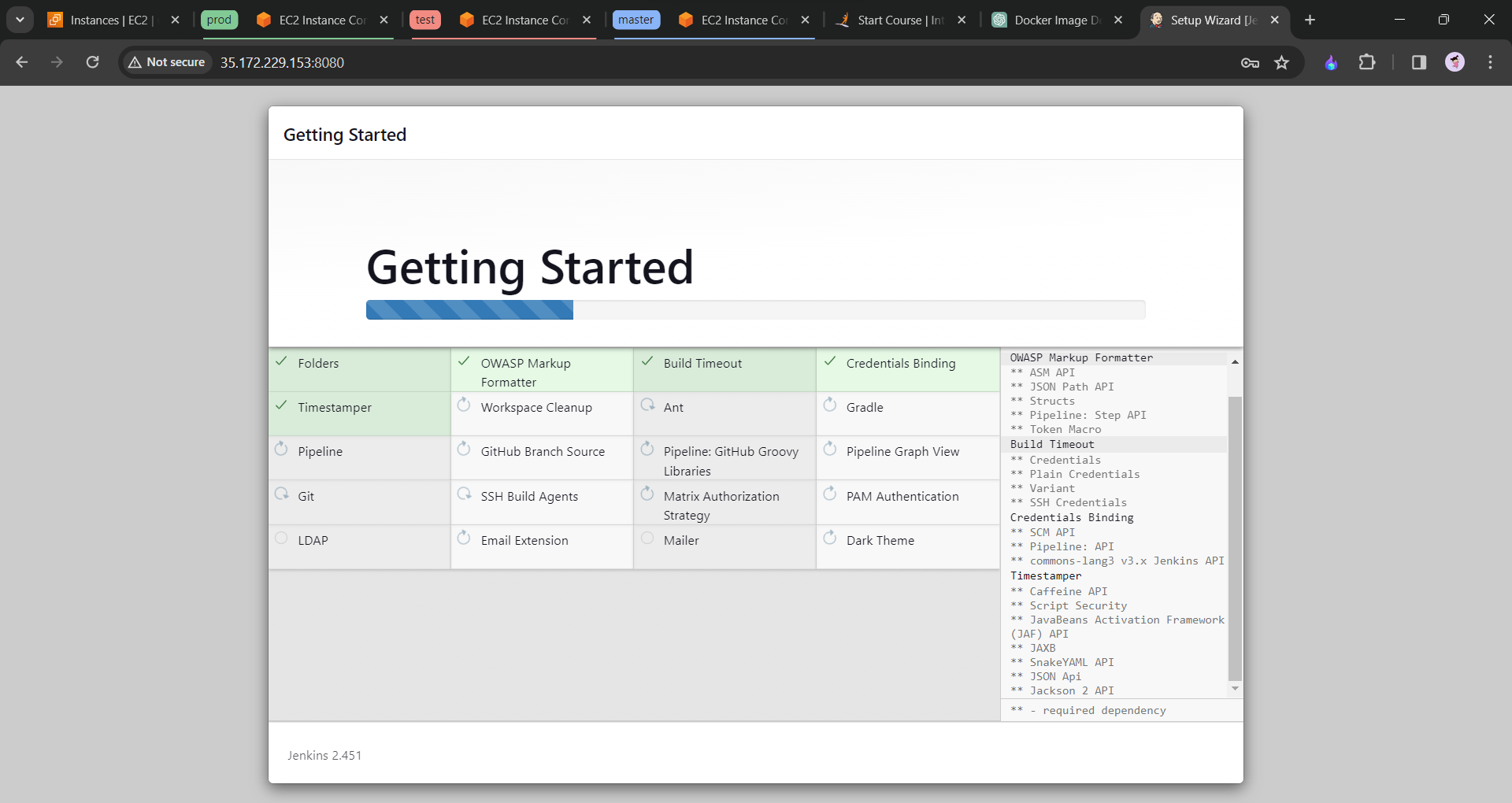


* We need to set the jenkins dashborad and to add the nodes over jenkins

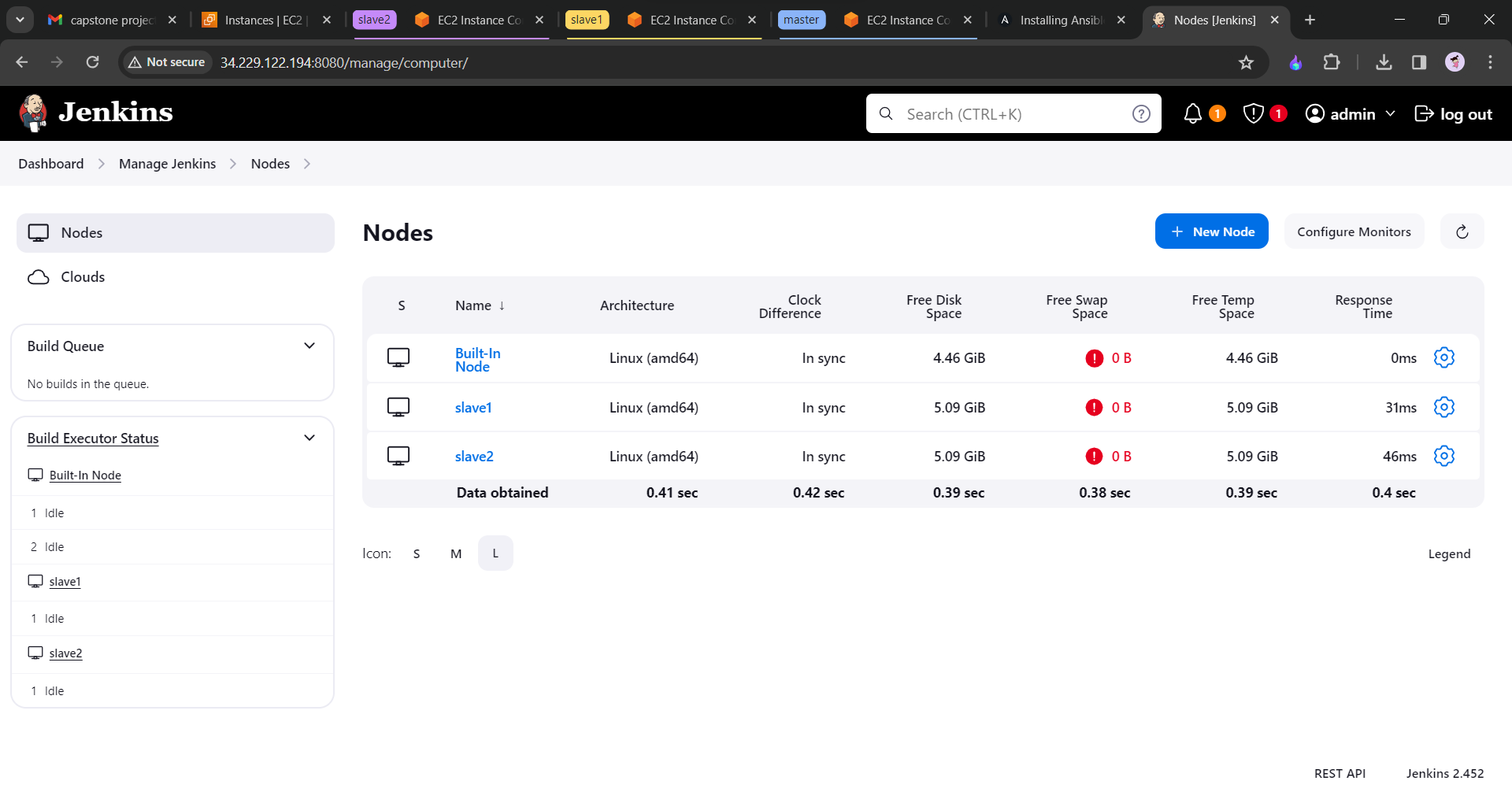
Dashboard setting of jenkins



* Unlock jenkins with the password
* Install the suggested pluggins



* We need to create the admin user(username,password and emailID)
* Please install the plugin called “SSH Agent” to connect the nodes.
* We are ready to see the dashboard.
* We will add the nodes now.
* Goto to manage jenkins>add nodes>newnode>
* Give “name” permanent agent and create
* Add description “remote root directory as :**home/ubuntu/jenkins**
* Launch method. Launch agent via ssh
* Hosts : add ip address of the slave private ip
* Credentials: click on add and click on jenkis “select kind ssh username with private key”
* Below scroll username as ubuntu private key copy its content and paste it
* Host key verification strategy to non verifying strategy
* Click on save



We will create the repository in github that we are going to use for the project.

* Fork the repository that is give by the project.
* Then we will create the docker file to containarize
* On the github create a new file with the name “Dockerfile”

FROM ubuntu

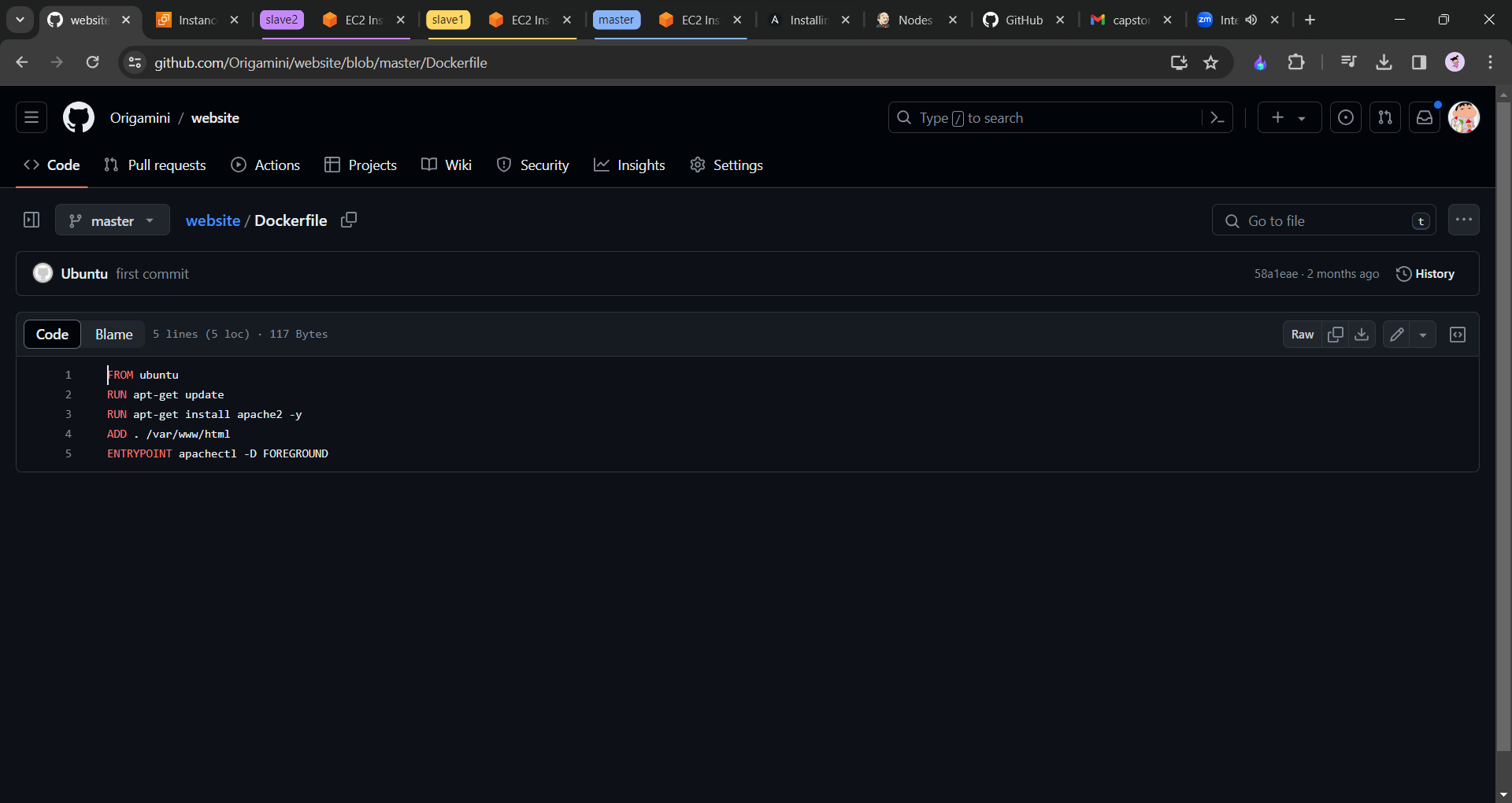
RUN apt update

RUN apt install apache2 -y

ADD . /var/www/html

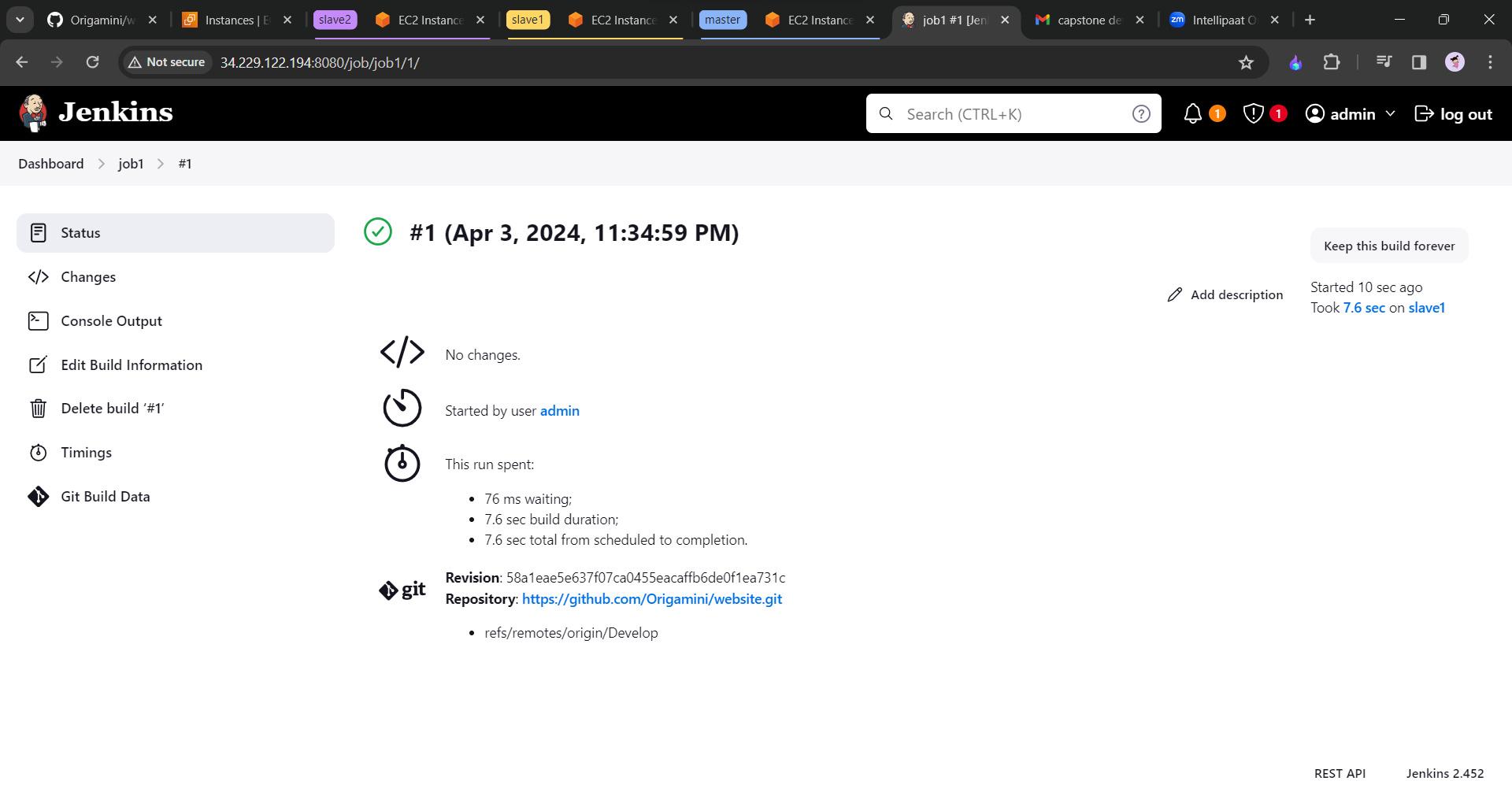
ENTRYPOINT apachectl -D FOREGROUND

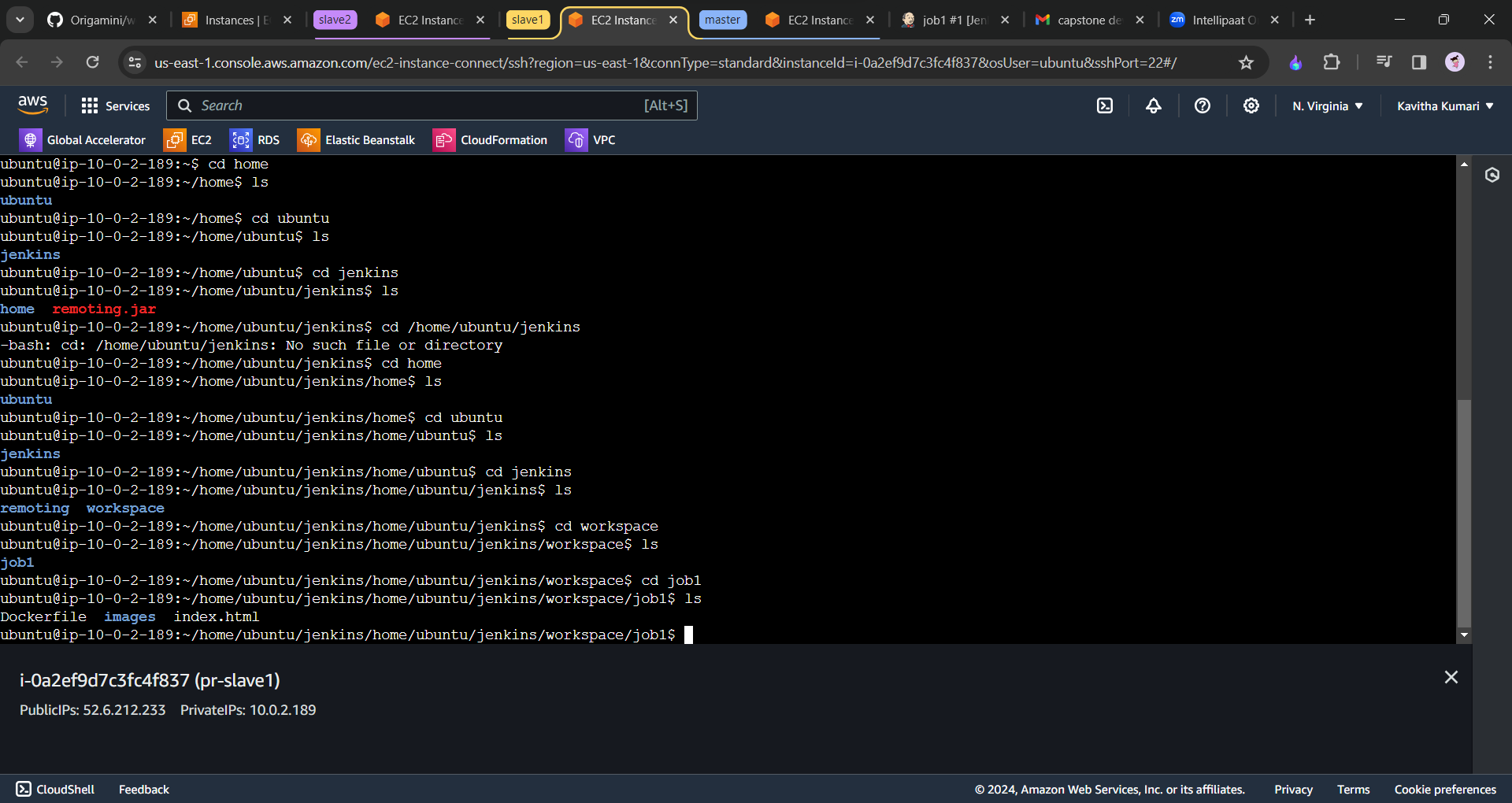
* Commit the changes
* We need to also create a new branch with the name develop.



Now we will create the jobs on the jenkins dashboard.

* Job1 = slave1-develop,job2=slave1-main,job3=slave2-main
* Click on newitem name it as job1 and select freesyle project click on next.
* Restrict where this project can be run checkbox
* Slave1 click on that
* Source code management select “git”
* Repo url :https://
* Branches to build : name of the branch “\*/develop”
* Goto buildstep and select execute shell



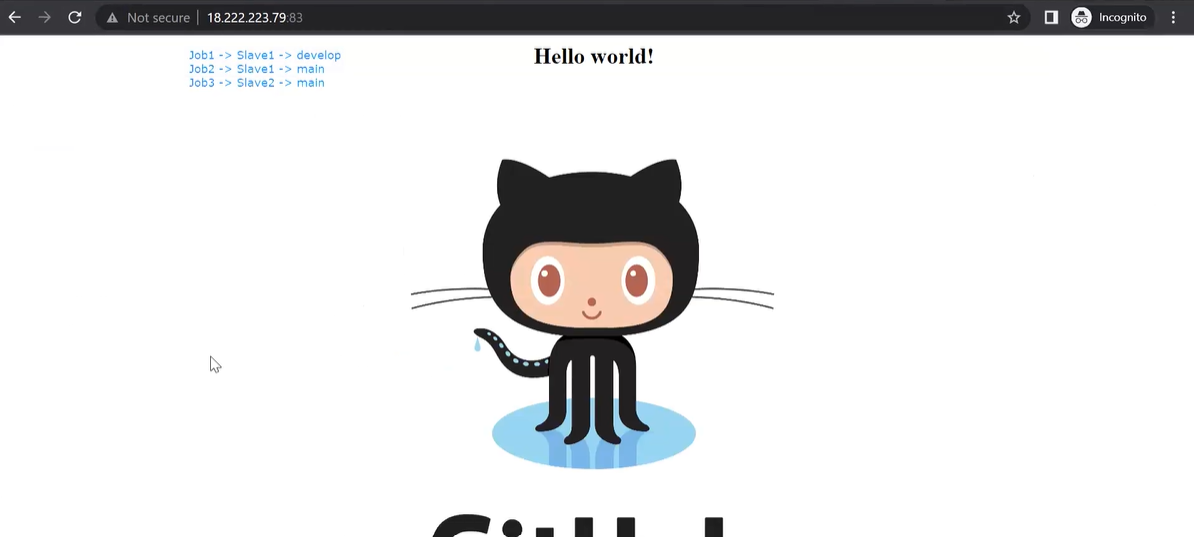


After this again configure the file with the following command.

sudo docker rm -f c1

sudo docker build /home/ubuntu/home/ubuntu/jenkins/home/ubuntu/jenkins/workspace/job1/ -t job1 //name of the image

sudo docker run -itd -p 83:80 --name=c1 job1



Goto to build trigger and click on github hook trigger for GItSCM pollling

Now we will create webhook

* Goto to settings and webhooks click on add webhook put the url of the jenkins
* <https://address/github-webhook/>
* Refresh it
* Now if we make any changes to the repository the job1 will be automatically executed.
* Now we will not be able to add the sudo docker rm -f c1 command,

Now we will create the job2( slave1 machine, select the master branch) and job3(select the slave2 machine , select the master branch)