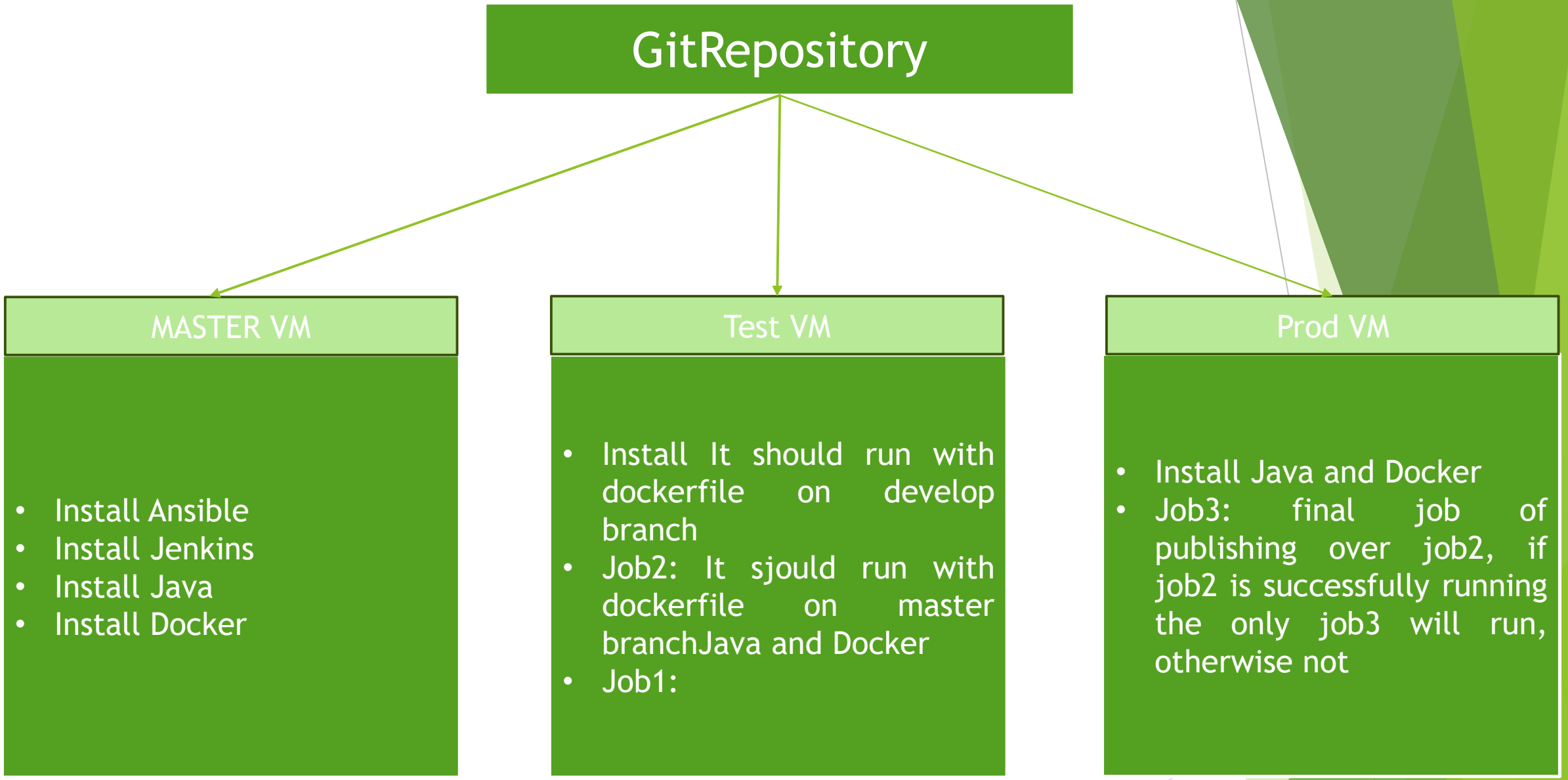
DEVOPS CAPSTONE : PROJECT I

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.

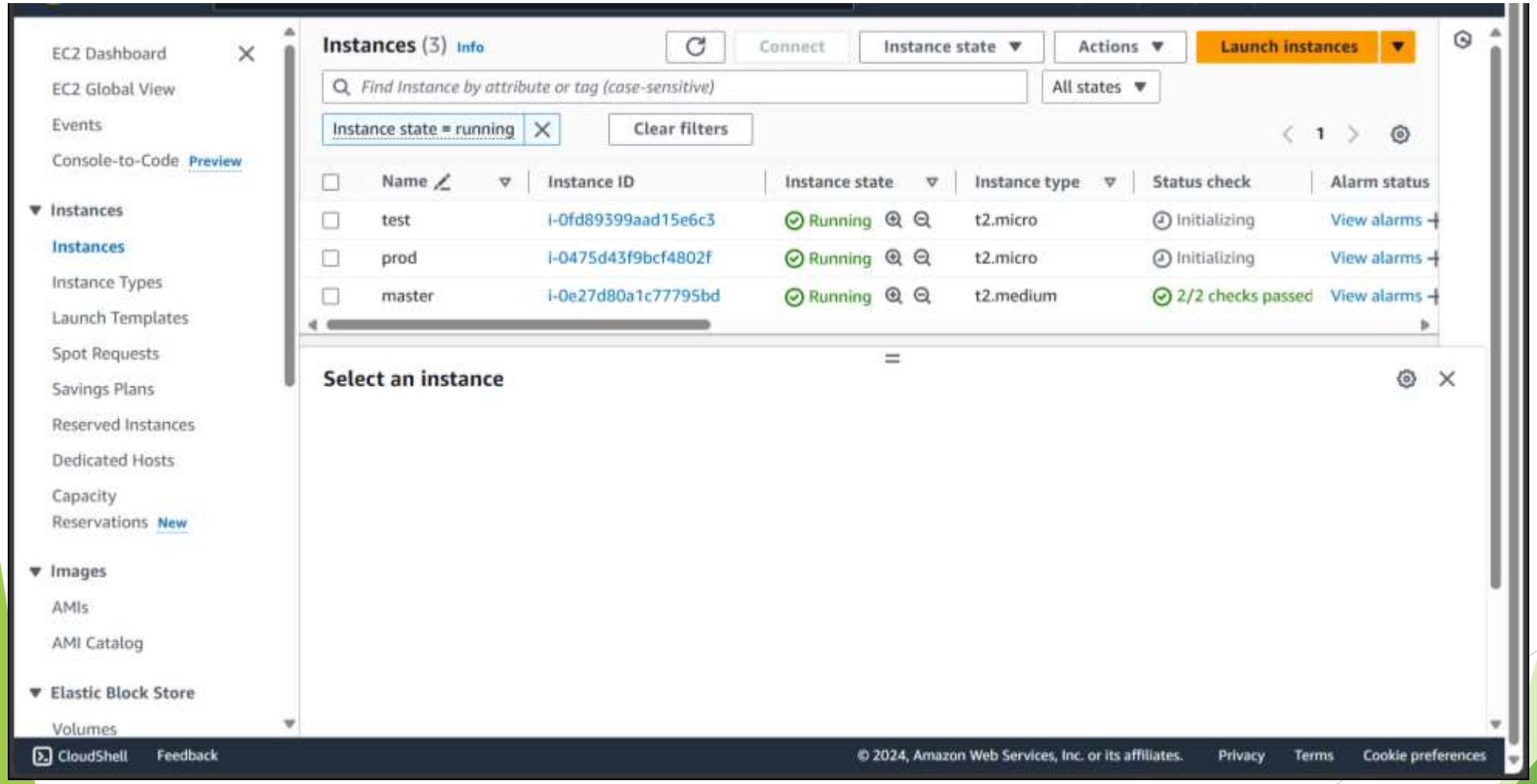
https://github.com/hshar/website.git

Following are the specifications of the lifecycle:

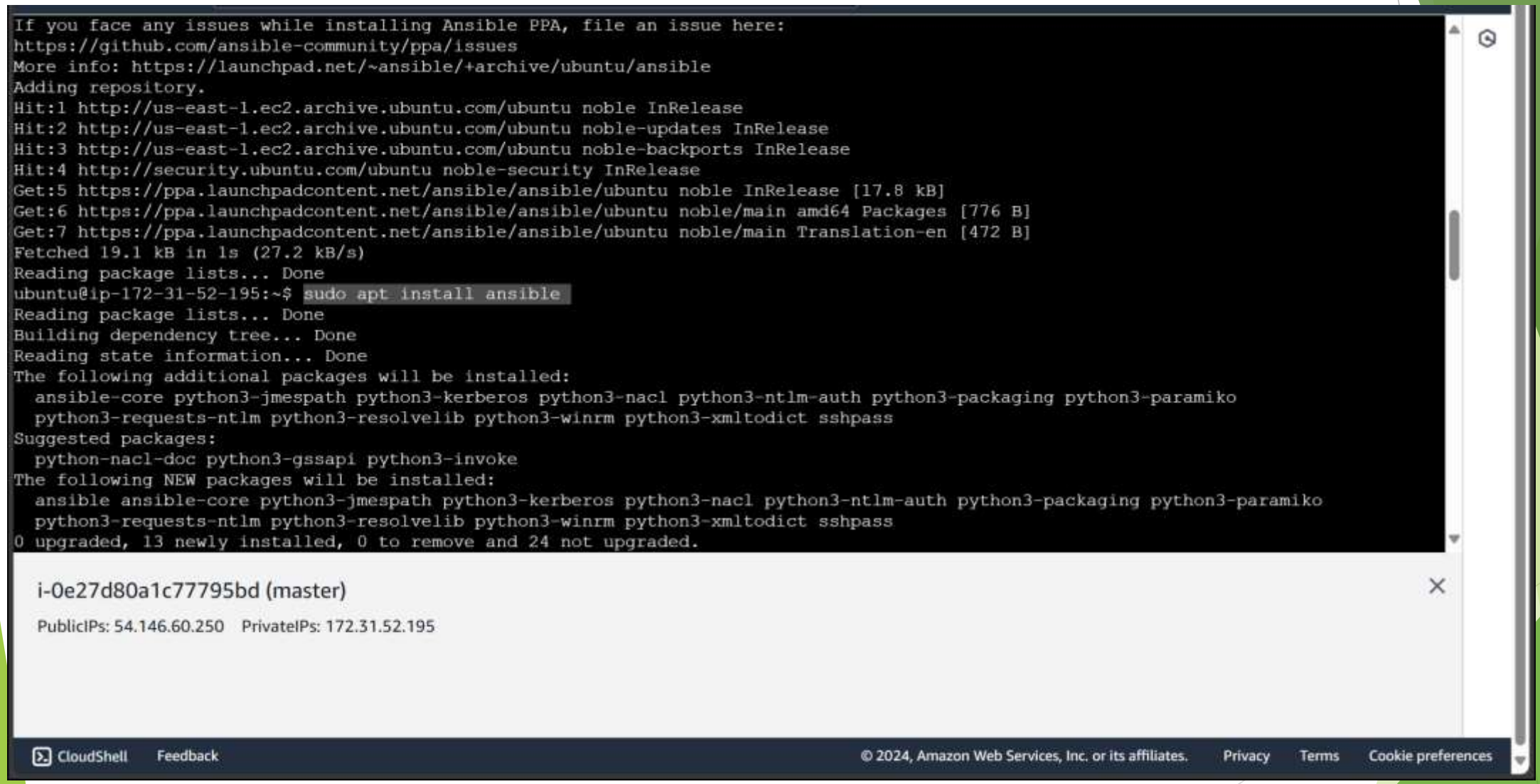
1. Install the necessary software on the machines using a configuration management tool
2. Git workflow has to be implemented
3. CodeBuild should automatically be triggered once a commit is made to master branch or develop branch.
    a. If a commit is made to master branch, test and push to prod
    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


GitRepository
MASTER VM
Install Ansible
Install Jenkins
Install Java
Install Docker
Test VM
Install It should run with dockerfile on develop branch
Job2: It sjould run with dockerfile on master branchJava and Docker
Job1:
Prod VM
Install Java and Docker
Job3: final job of publishing over job2, if job2 is successfully running the only job3 will run, otherwise not

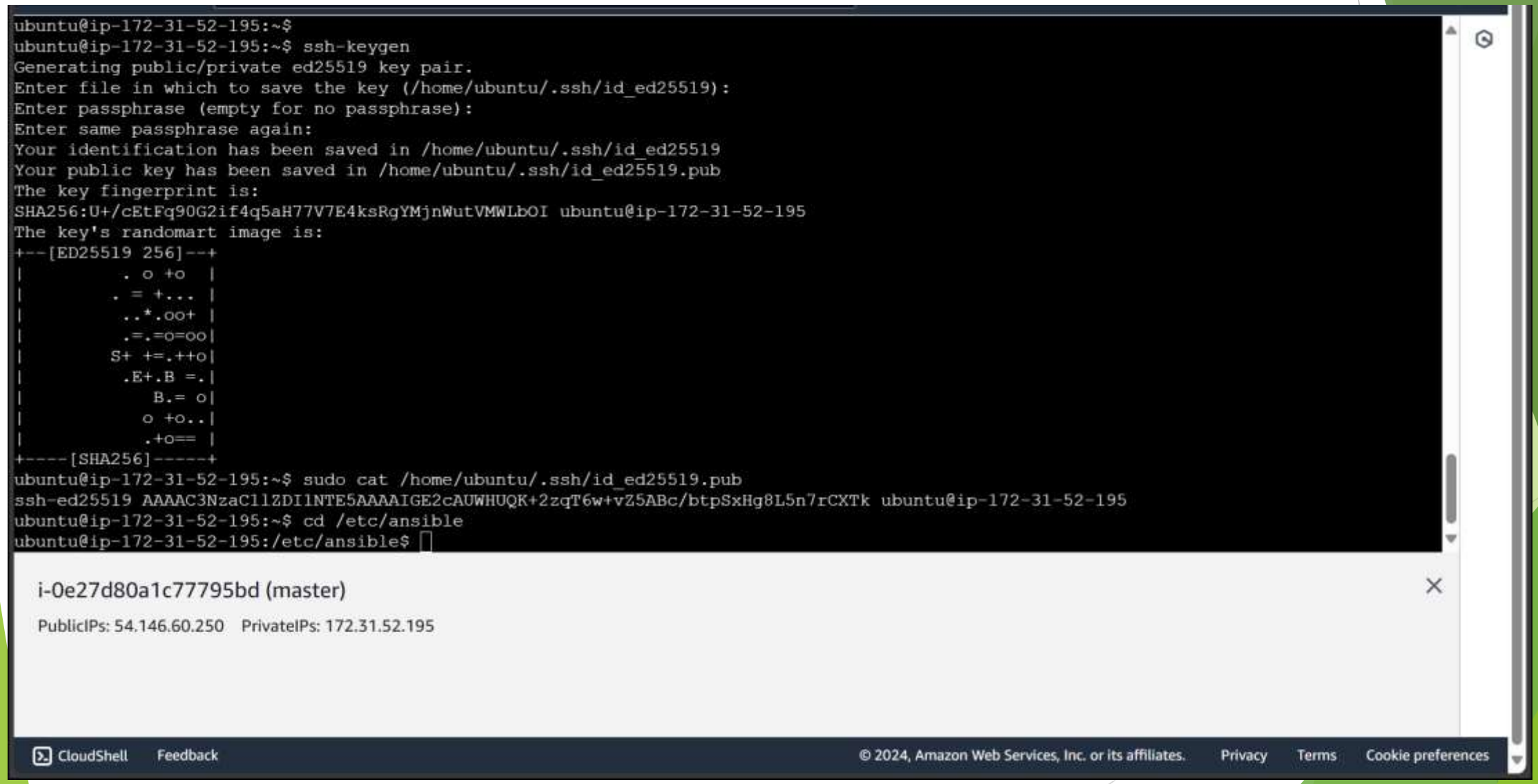
Creating Instance Master , test and prod



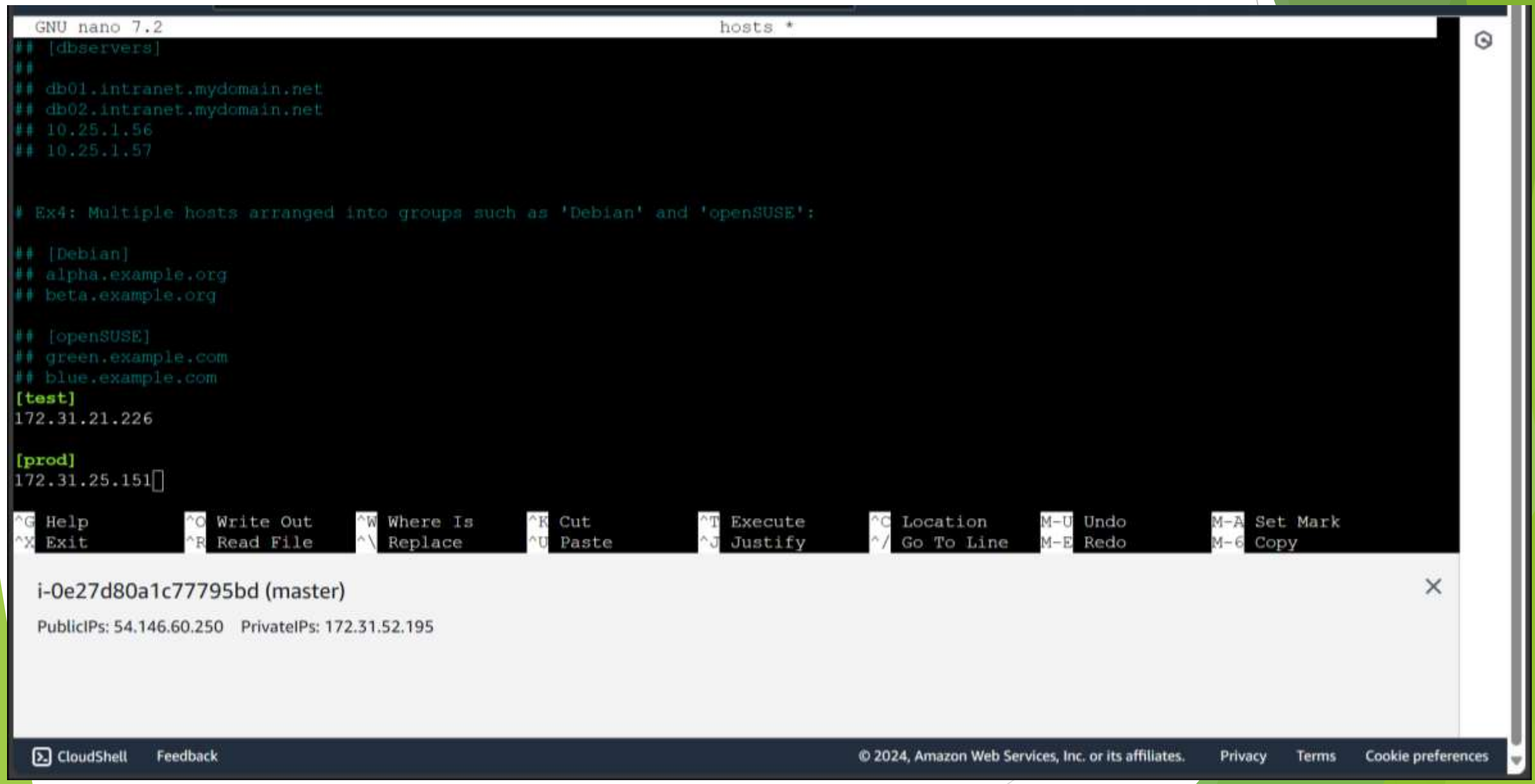
Installing ansible in master instance



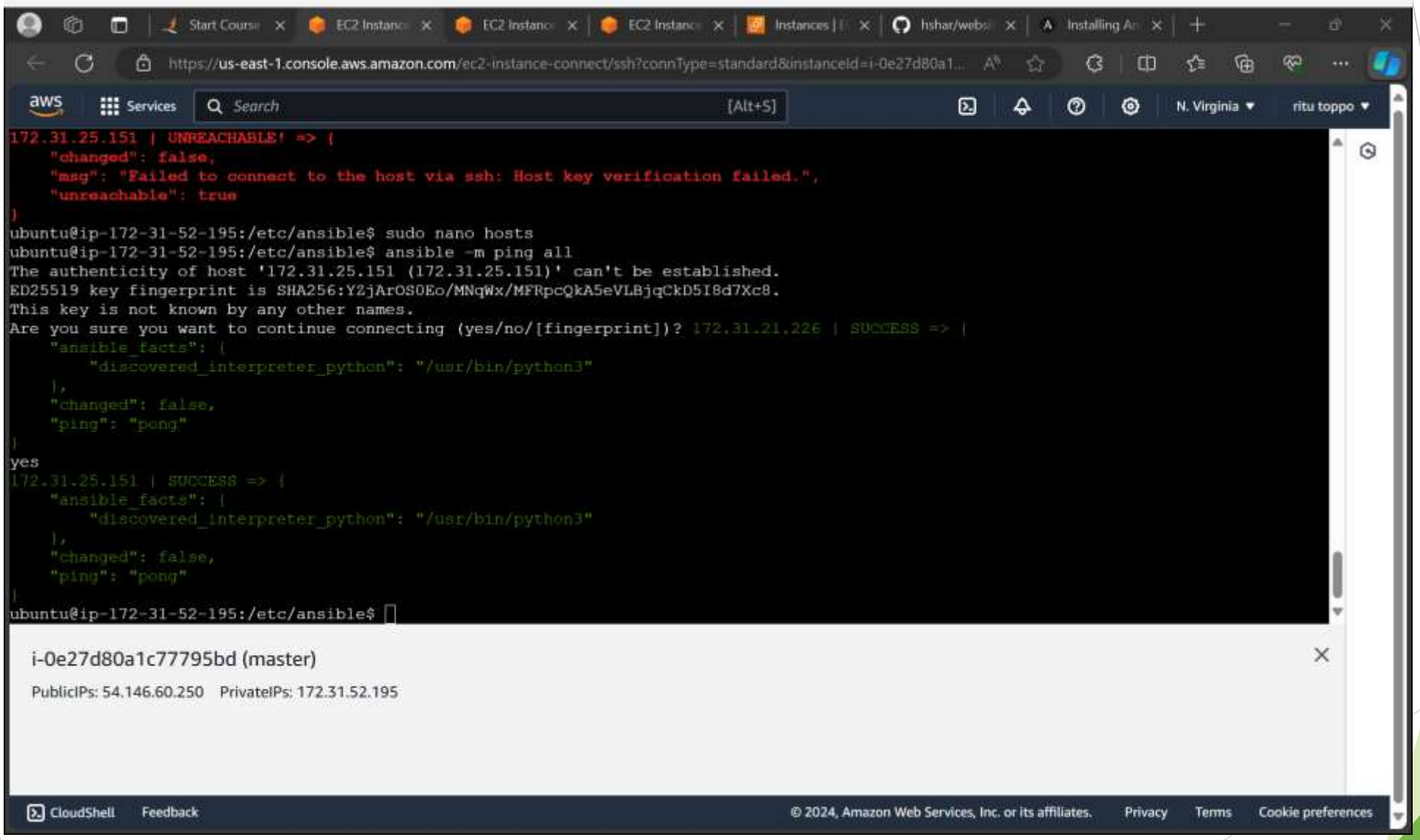
Generating Ssh-keygen to get public key to get connect with test and prod instance



Open hosts file to add the hosts i.e. test and prod

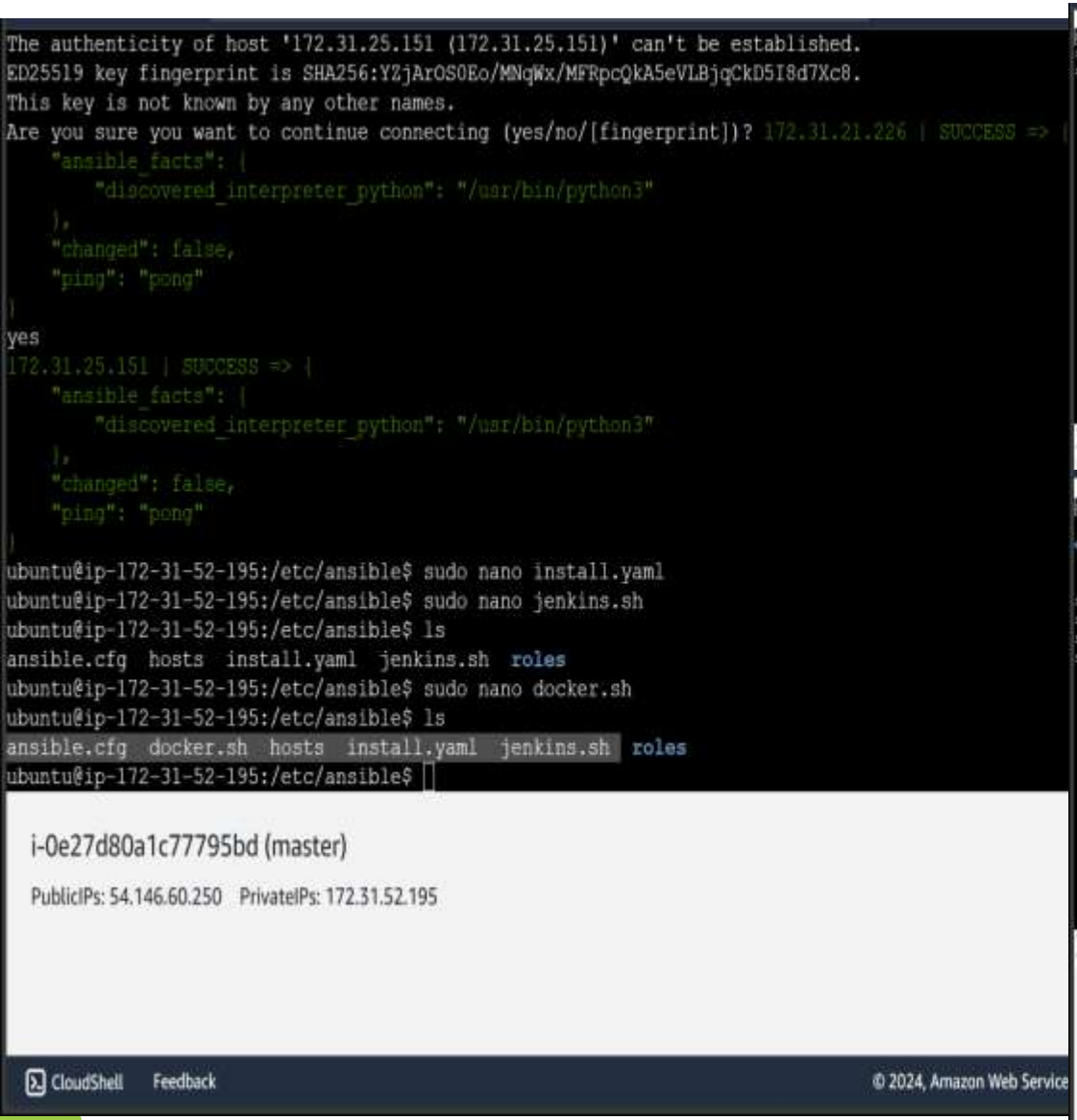


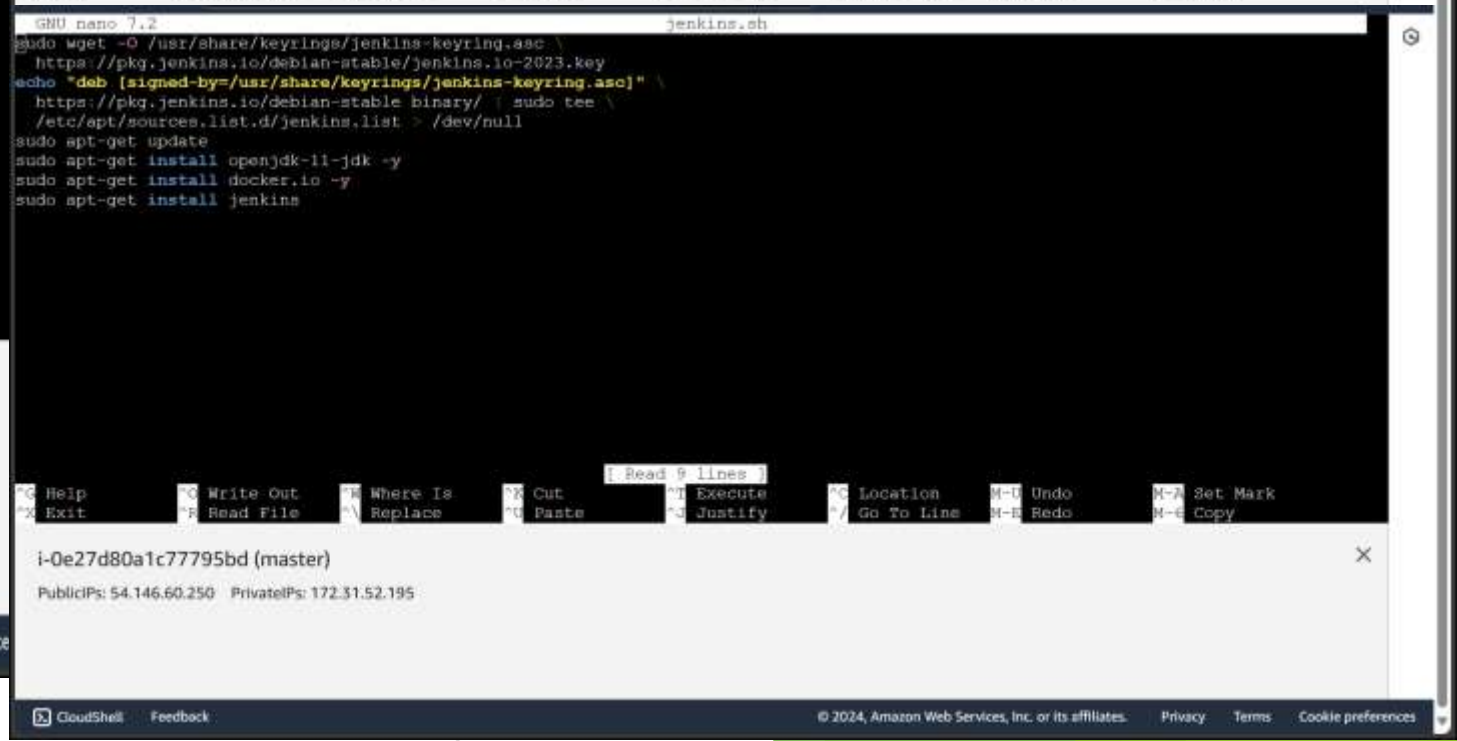
Connection to master, test and prod has been done successfully



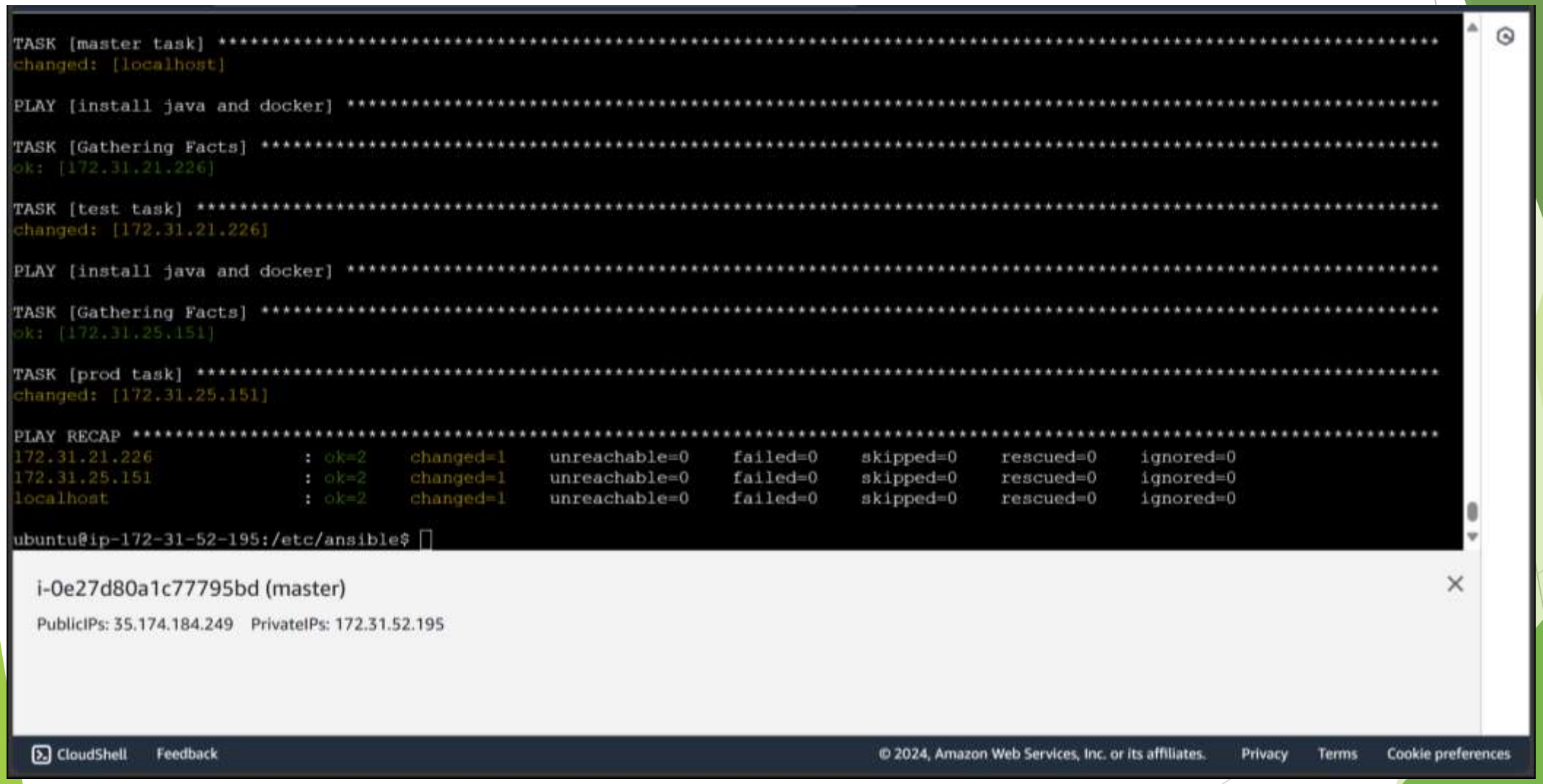
Installing the Jenkins java docker in master and java docker in test and prod instance using playbook

Created a docker.sh and Jenkins.sh file

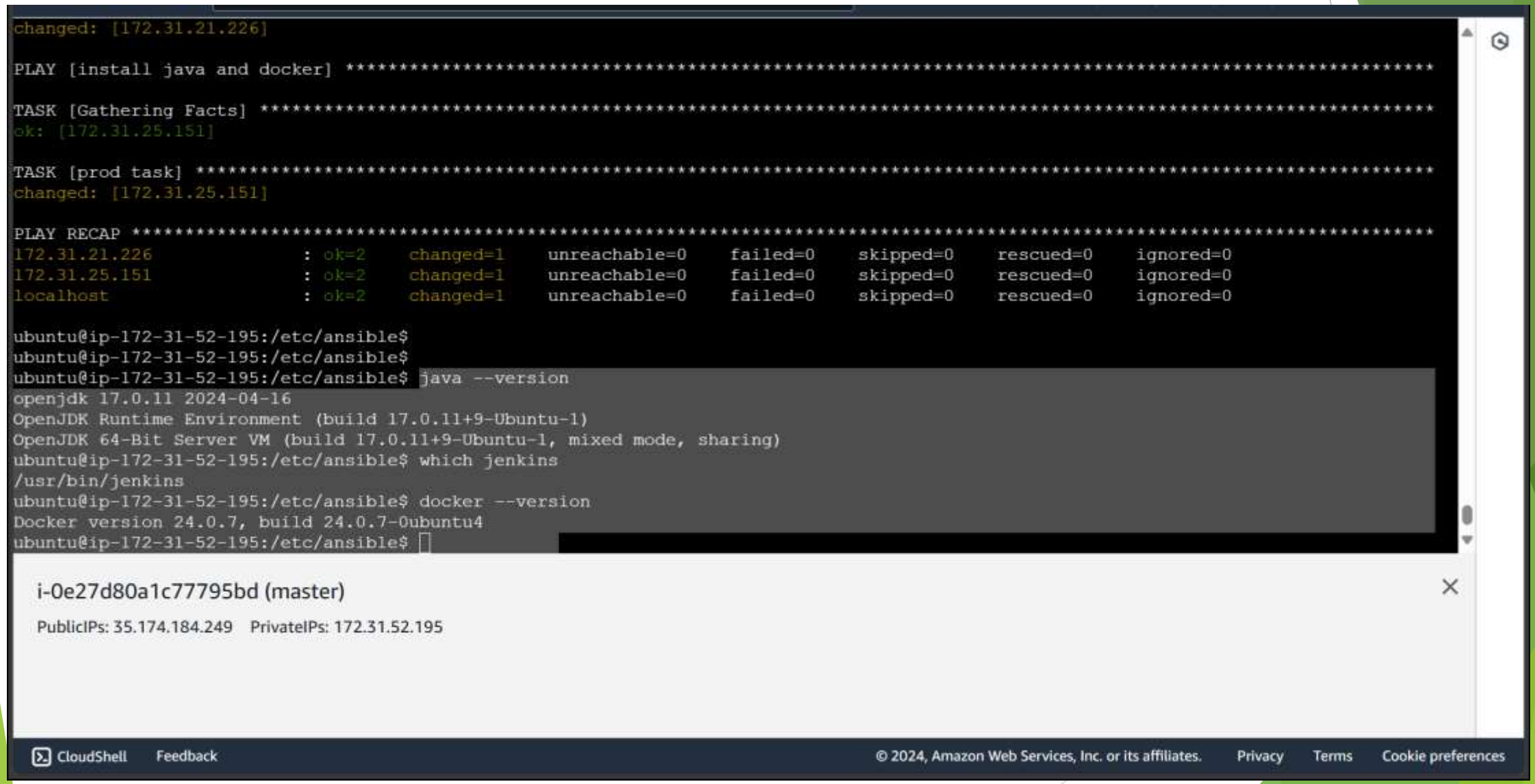




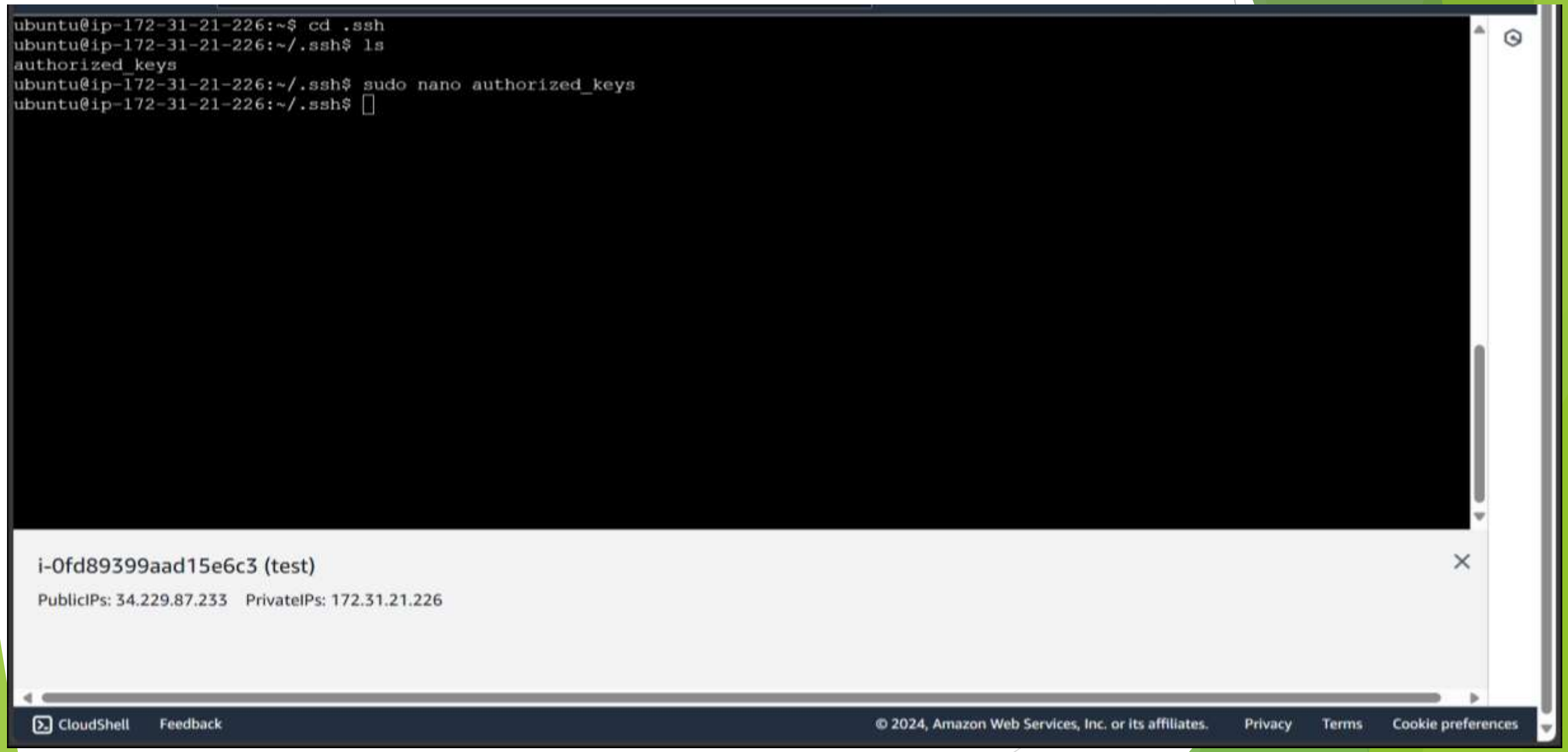
Syntax Checking and executing the playbook install.yaml successfully installed



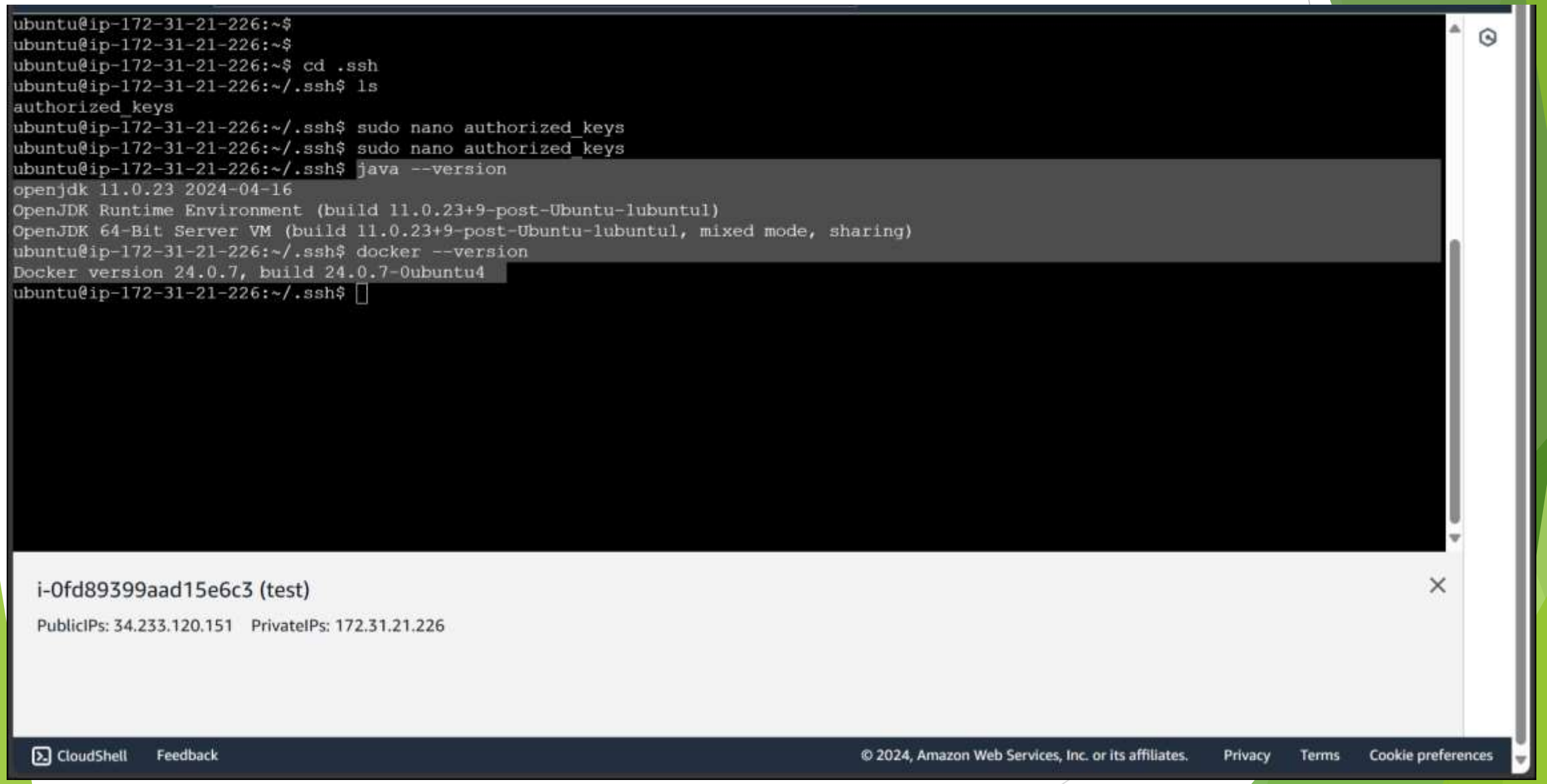
Checking the installation of Java, Jenkins and docker in master successfully installed



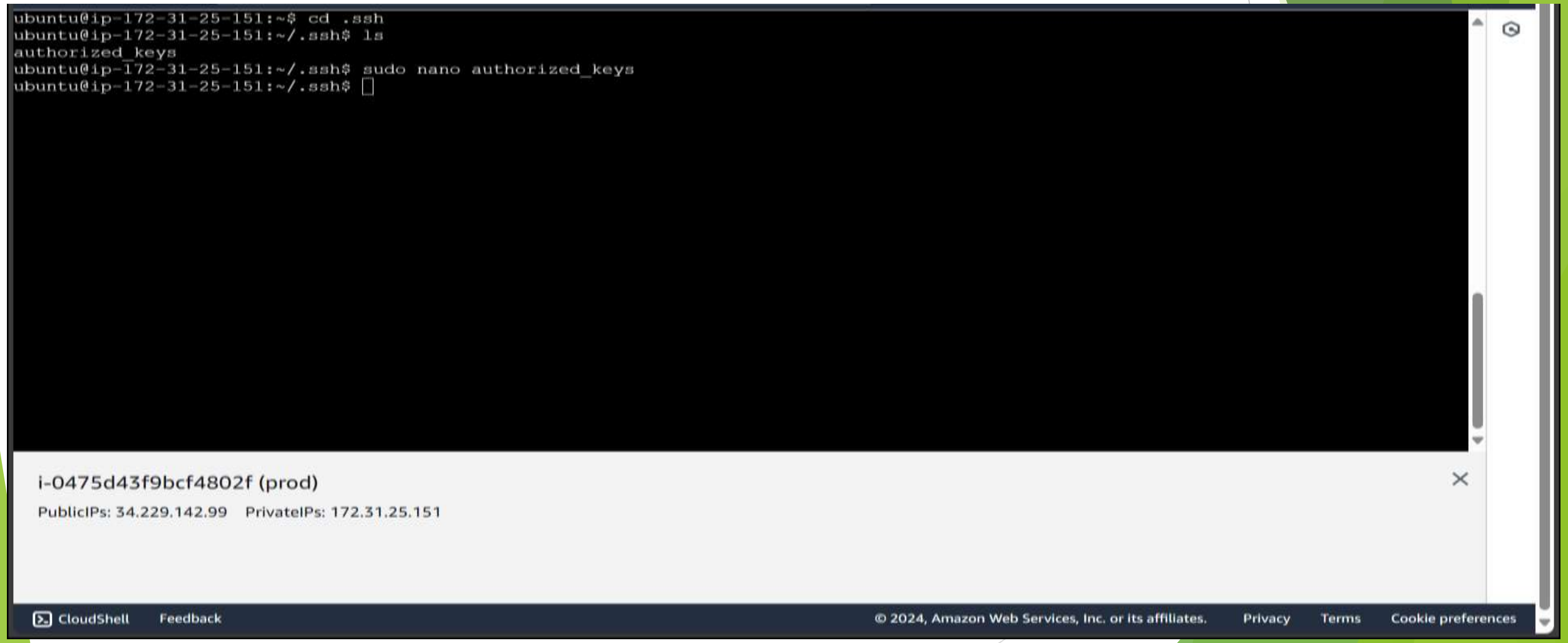
Adding public key ssh connection in test instance



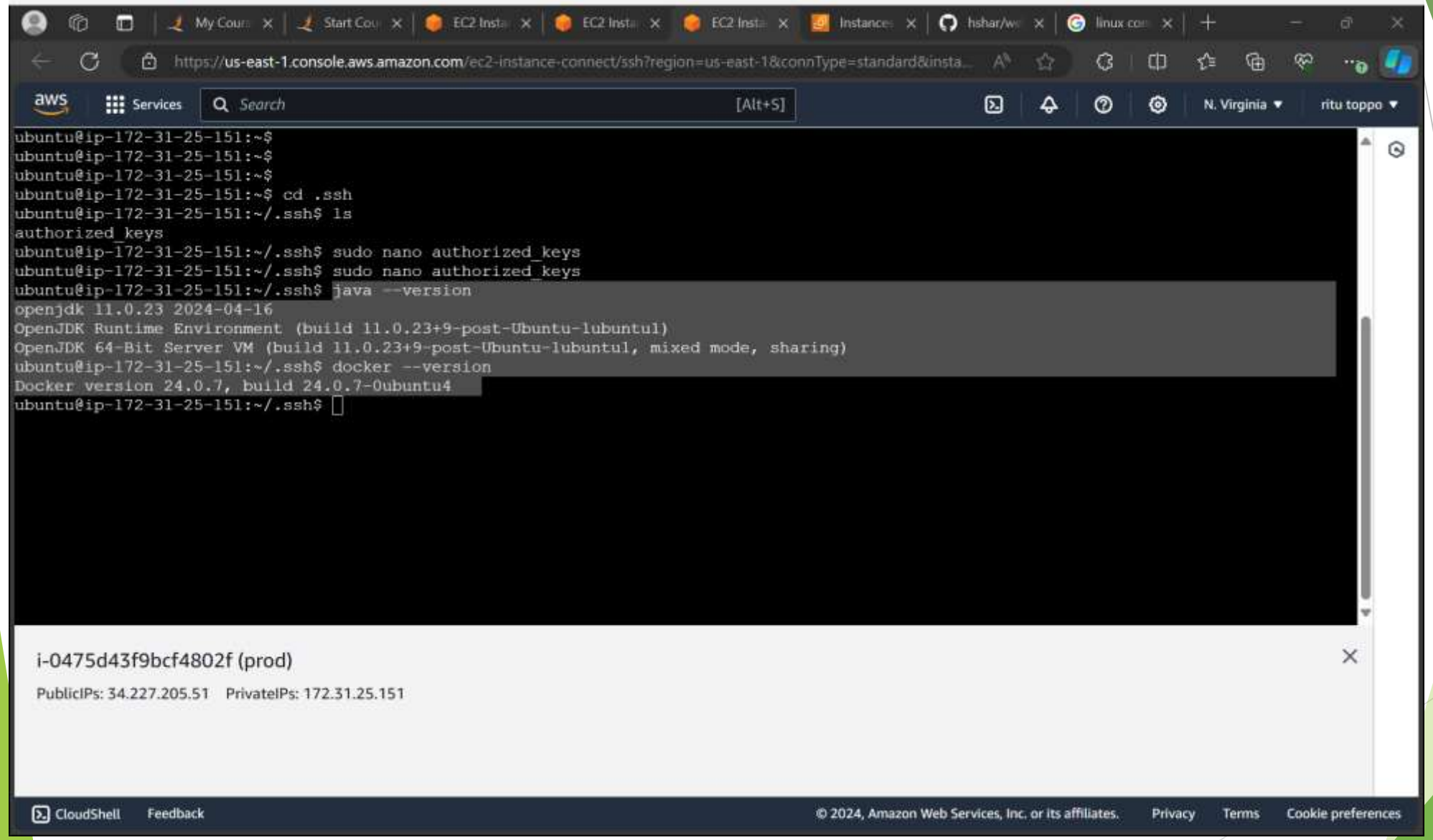
Checking the installation of Java, Jenkins and docker in test



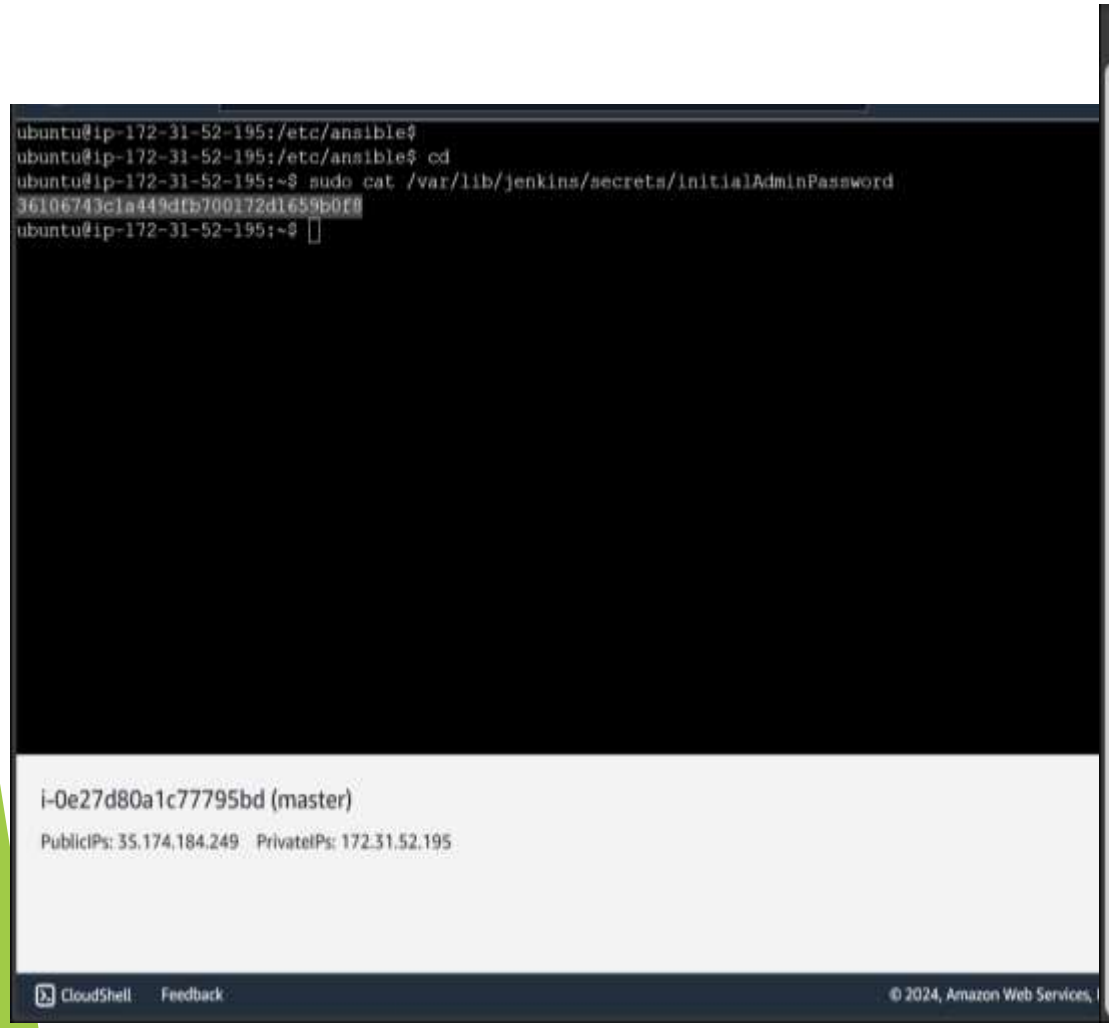
prod instance
ssh connection with master

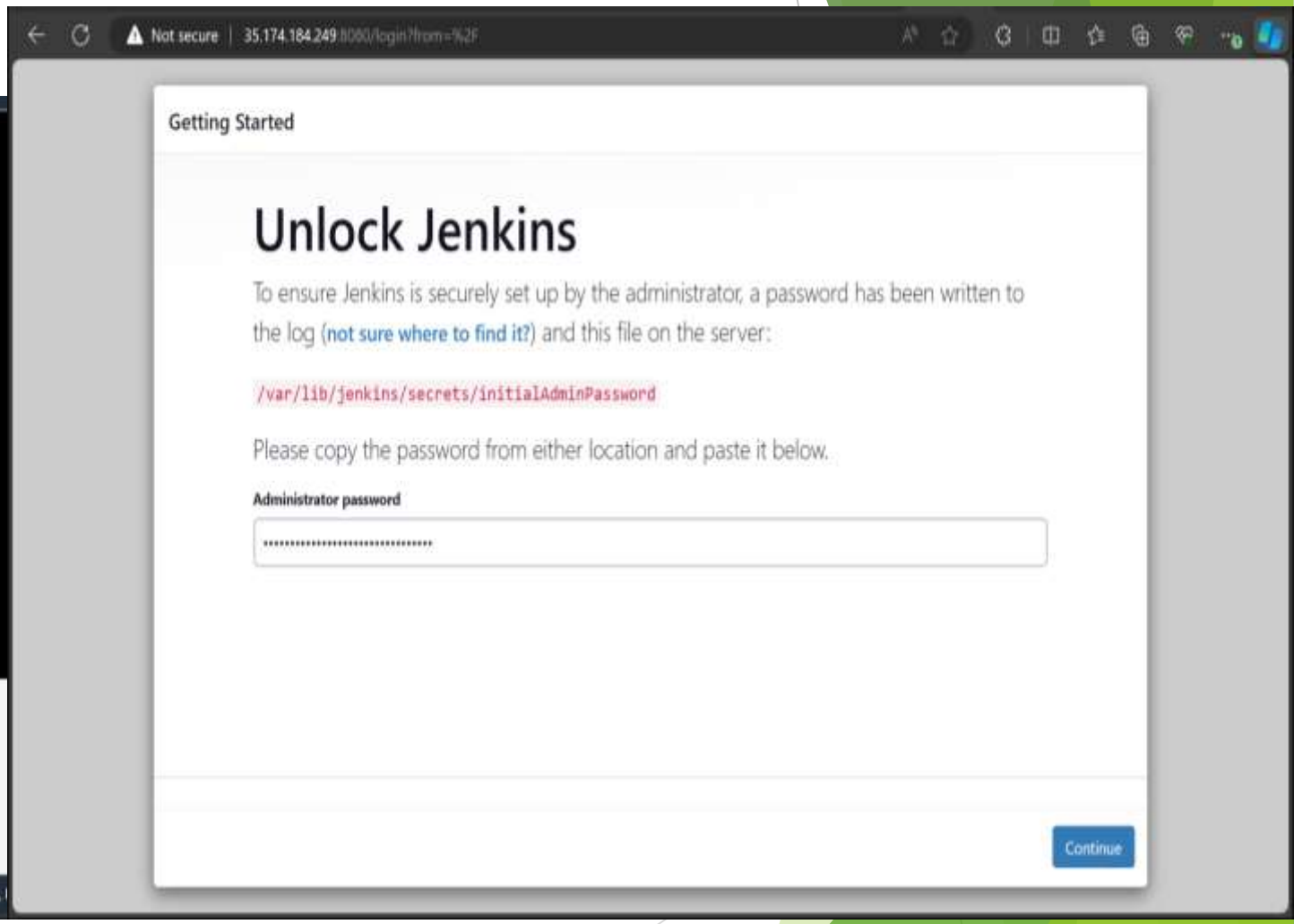


Checking the installation of Java and docker in prod

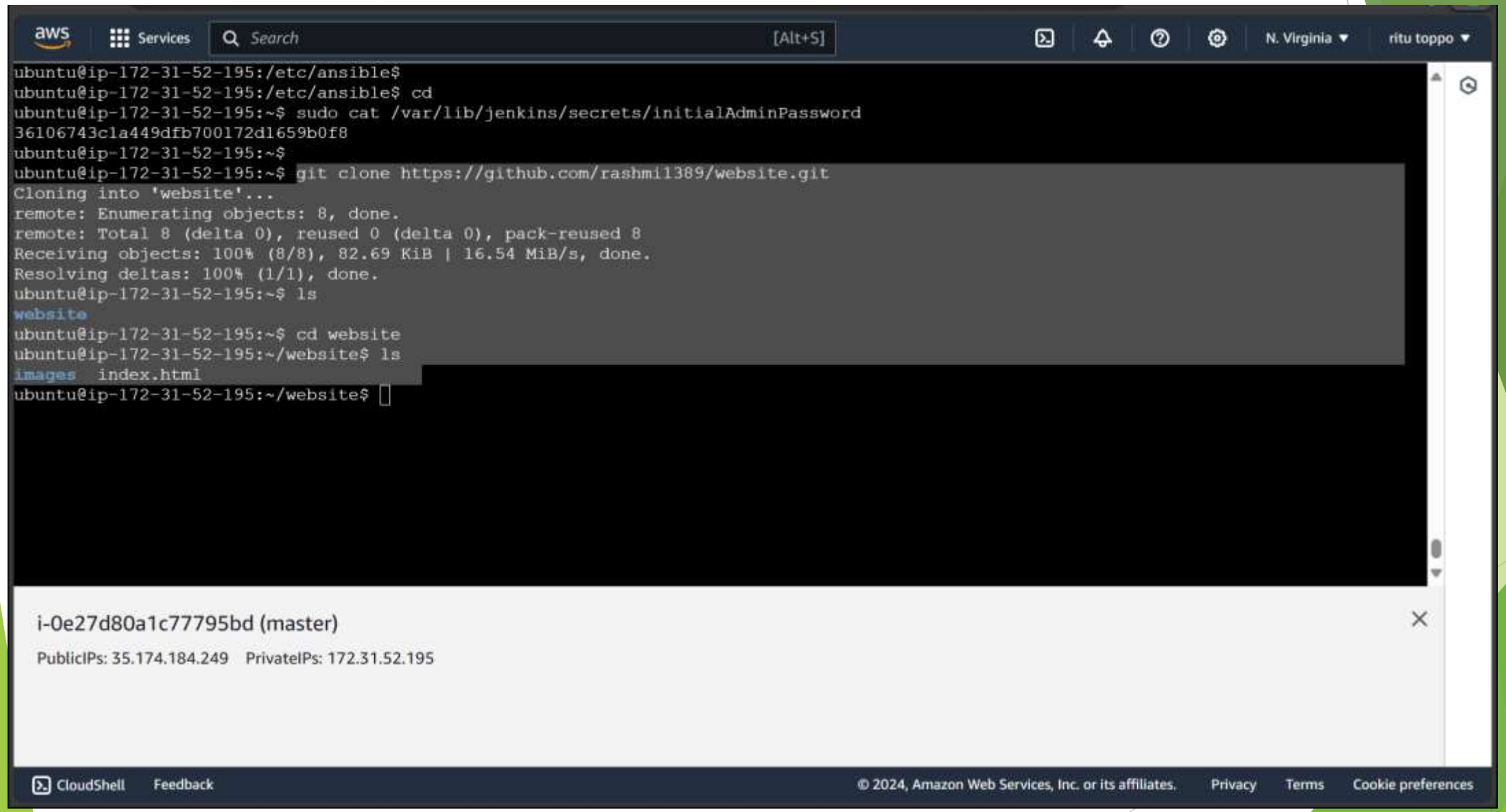


Login to Jenkins

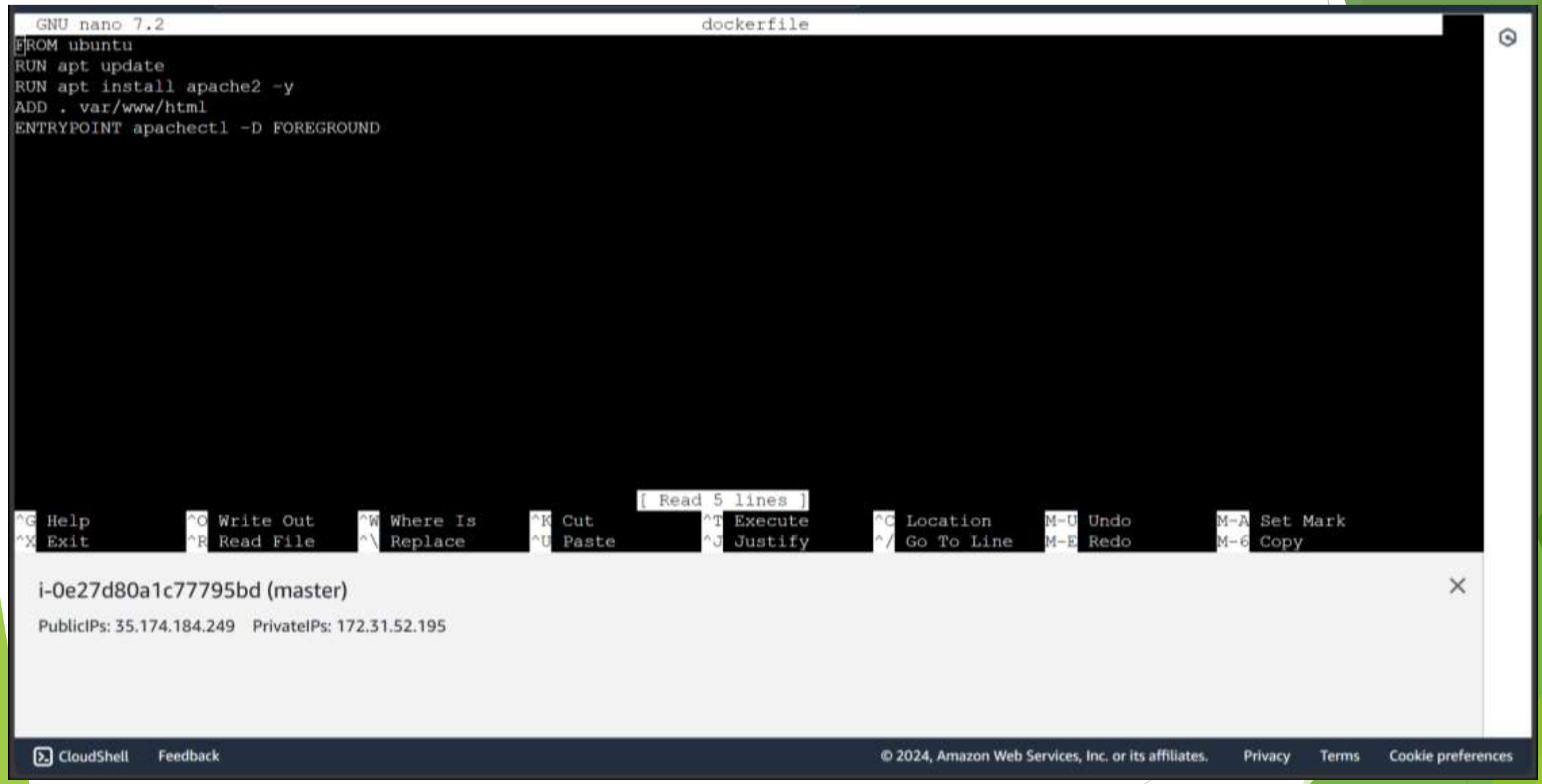




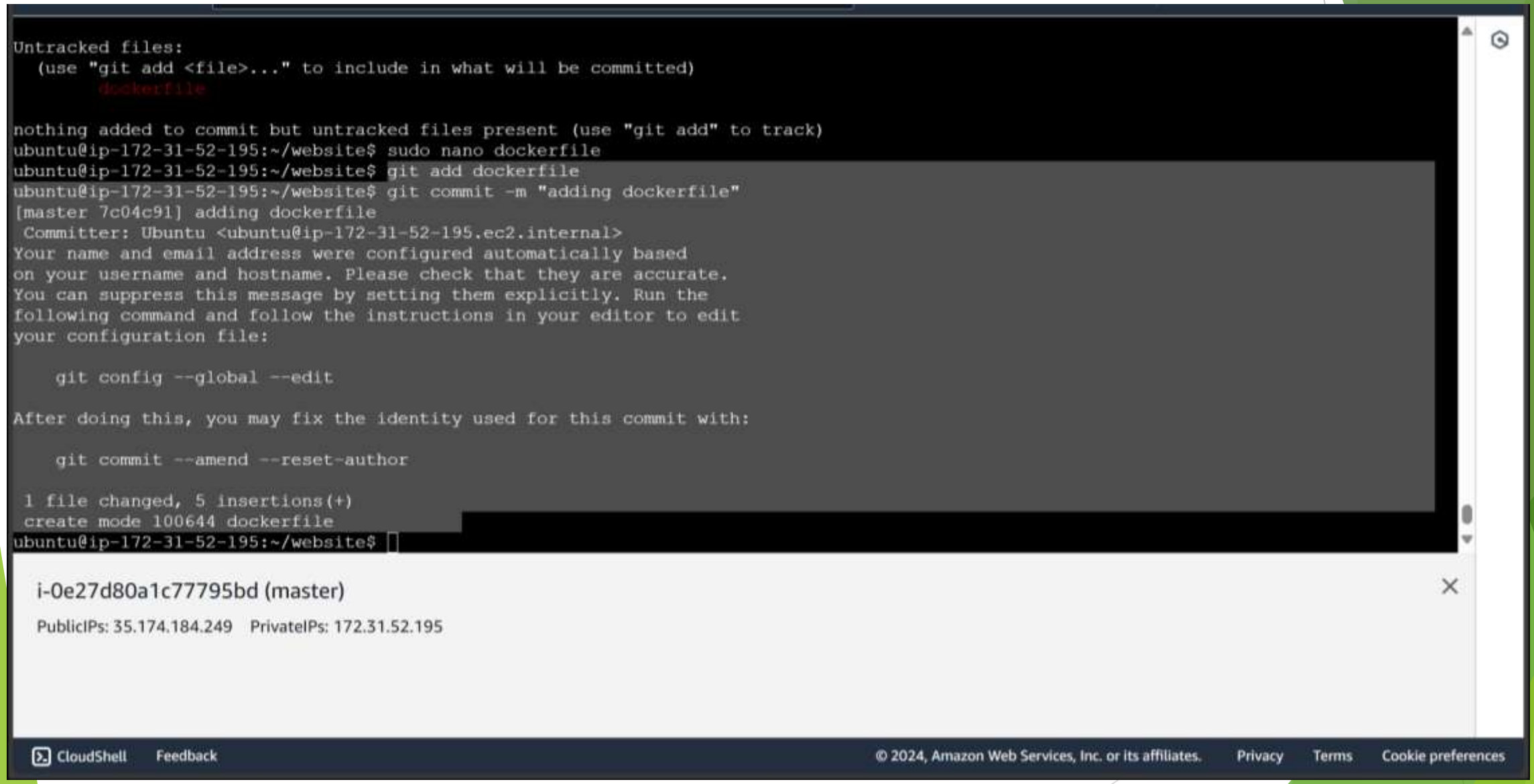
Clone the git repository in master with the given url in project



Create a docker file in master



Add the file in git



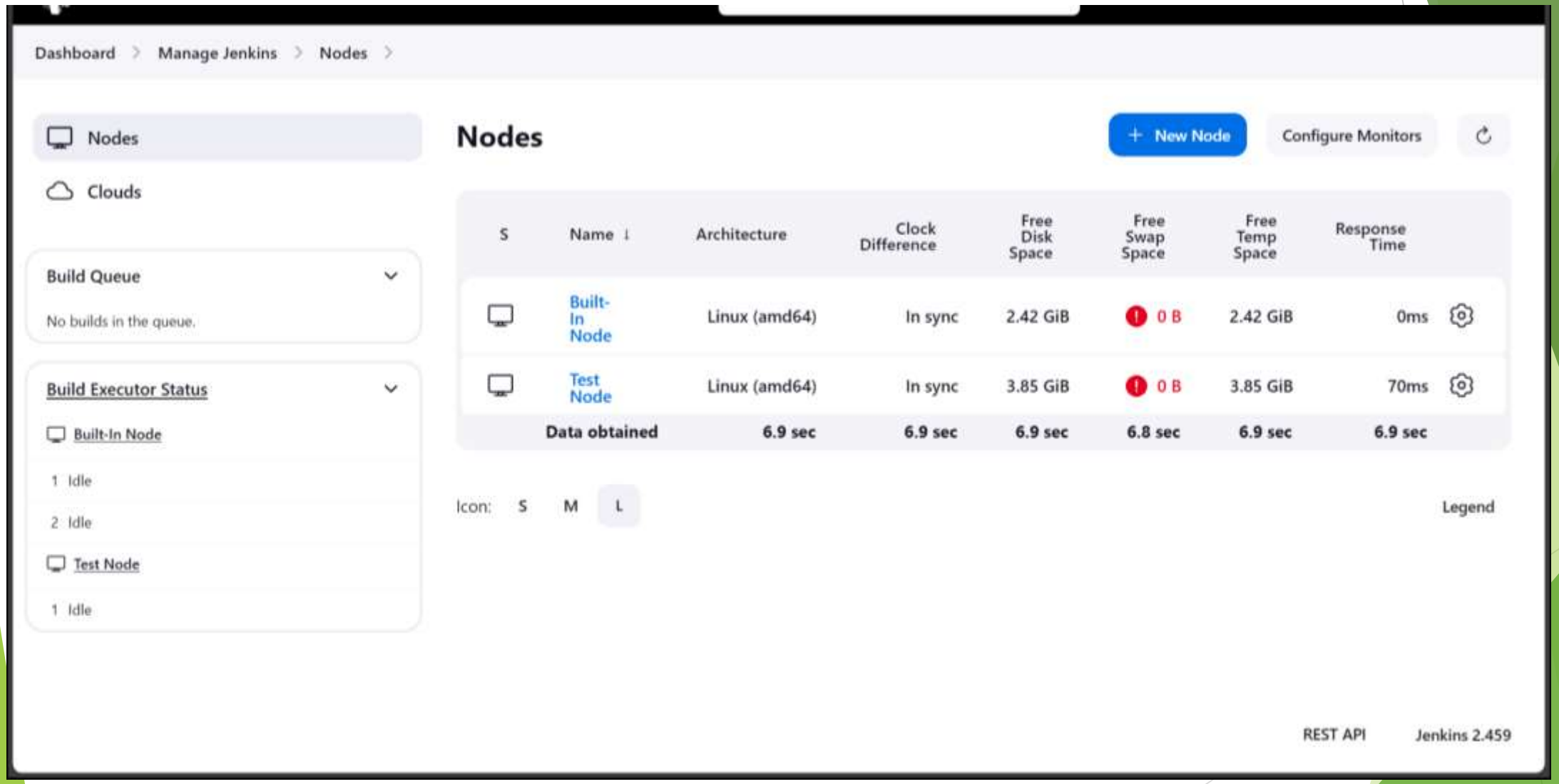
Create a develop branch

```
ubuntu@ip-172-31-52-195:~/website$ sudo nano dockerfile
ubuntu@ip-172-31-52-195:~/website$ git add dockerfile
ubuntu@ip-172-31-52-195:~/website$ git commit -m "adding dockerfile"
[master 7c04c91] adding dockerfile
 Committer: Ubuntu <ubuntu@ip-172-31-52-195.ec2.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 1 file changed, 5 insertions(+)
 create mode 100644 dockerfile
ubuntu@ip-172-31-52-195:~/website$ git branch
* master
ubuntu@ip-172-31-52-195:~/website$ git branch develop
ubuntu@ip-172-31-52-195:~/website$ git branch
  develop
* master
ubuntu@ip-172-31-52-195:~/website$
```

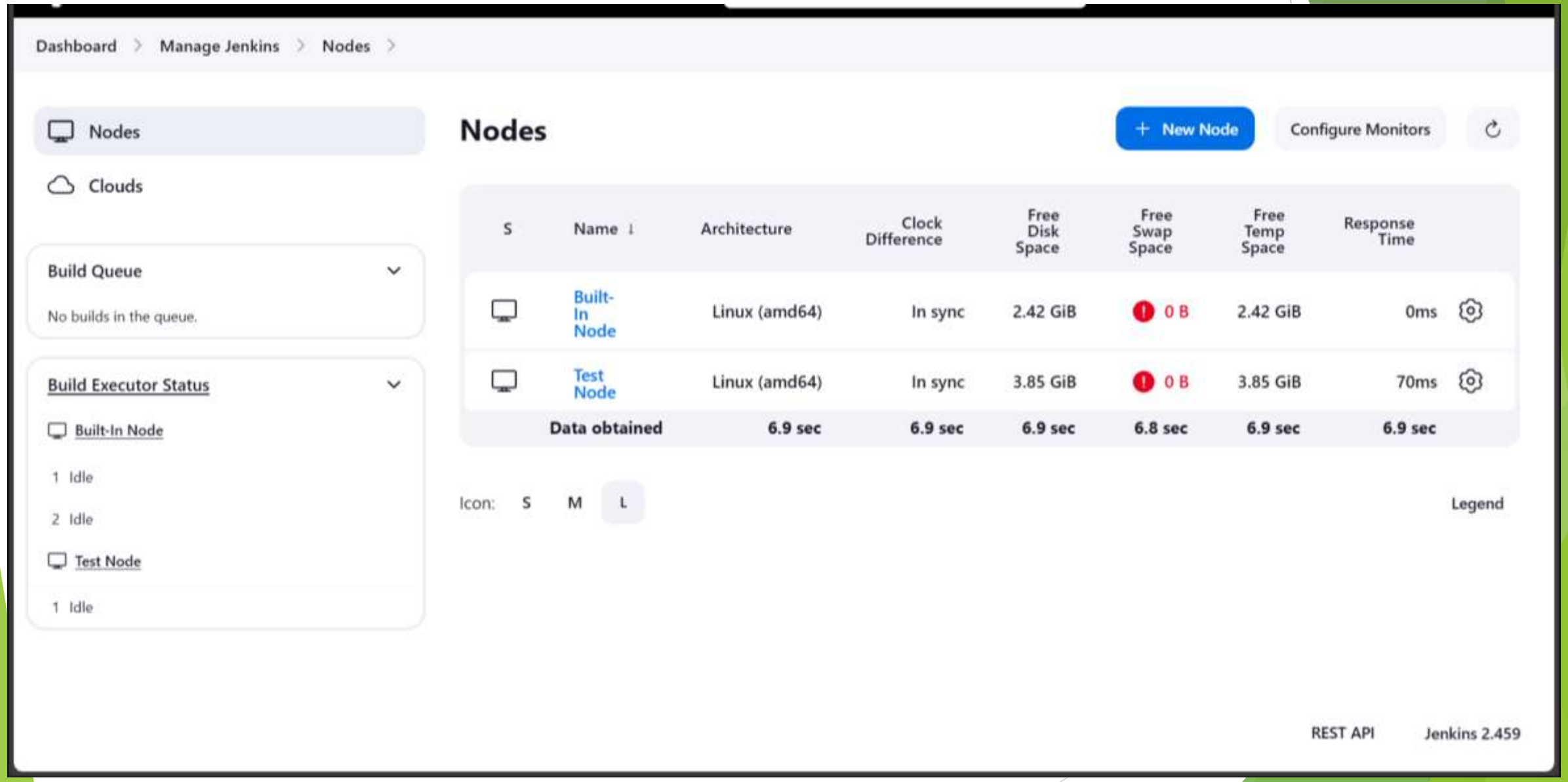
i-0e27d80a1c77795bd (master)

PublicIPs: 35.174.184.249 PrivateIPs: 172.31.52.195

Connect the test instance to Jenkins as test node



Connect the prod instance to Jenkins as prod node



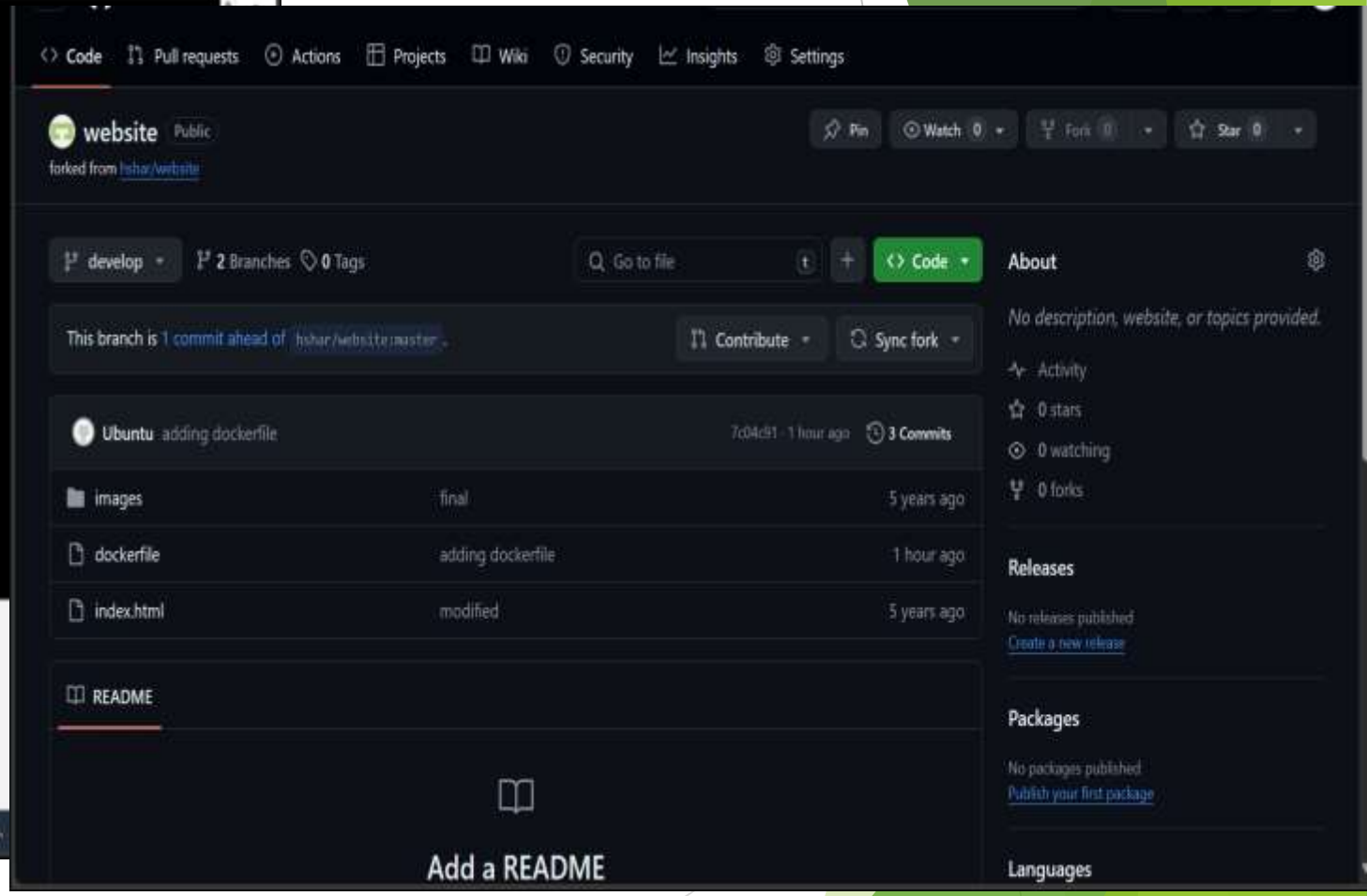
Push the dockerfile to develop branch

```
ubuntu@ip-172-31-52-195:~/website$ git branch
  develop
* master
ubuntu@ip-172-31-52-195:~/website$ git checkout develop
Switched to branch 'develop'
ubuntu@ip-172-31-52-195:~/website$ git push origin develop
Username for 'https://github.com': rashmi1389
Password for 'https://rashmi1389@github.com':
remote: Invalid username or password.
fatal: Authentication failed for 'https://github.com/rashmi1389/website.git/'
ubuntu@ip-172-31-52-195:~/website$ git push origin develop
Username for 'https://github.com': rashmi1389
Password for 'https://rashmi1389@github.com':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 2 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 410 bytes | 410.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:      https://github.com/rashmi1389/website/pull/new/develop
remote:
To https://github.com/rashmi1389/website.git
 * [new branch]      develop -> develop
ubuntu@ip-172-31-52-195:~/website$
```

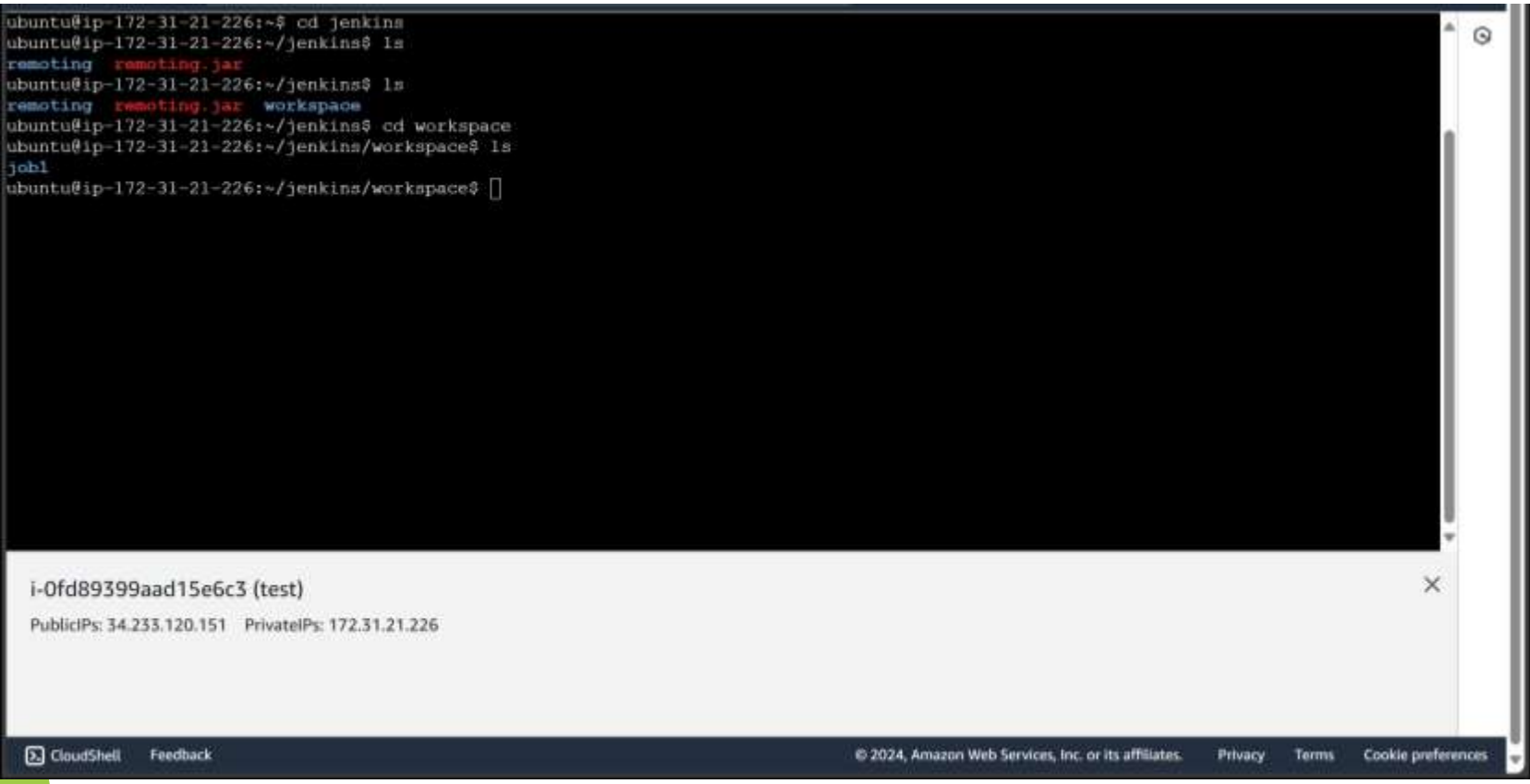
i-0e27d80a1c77795bd (master)

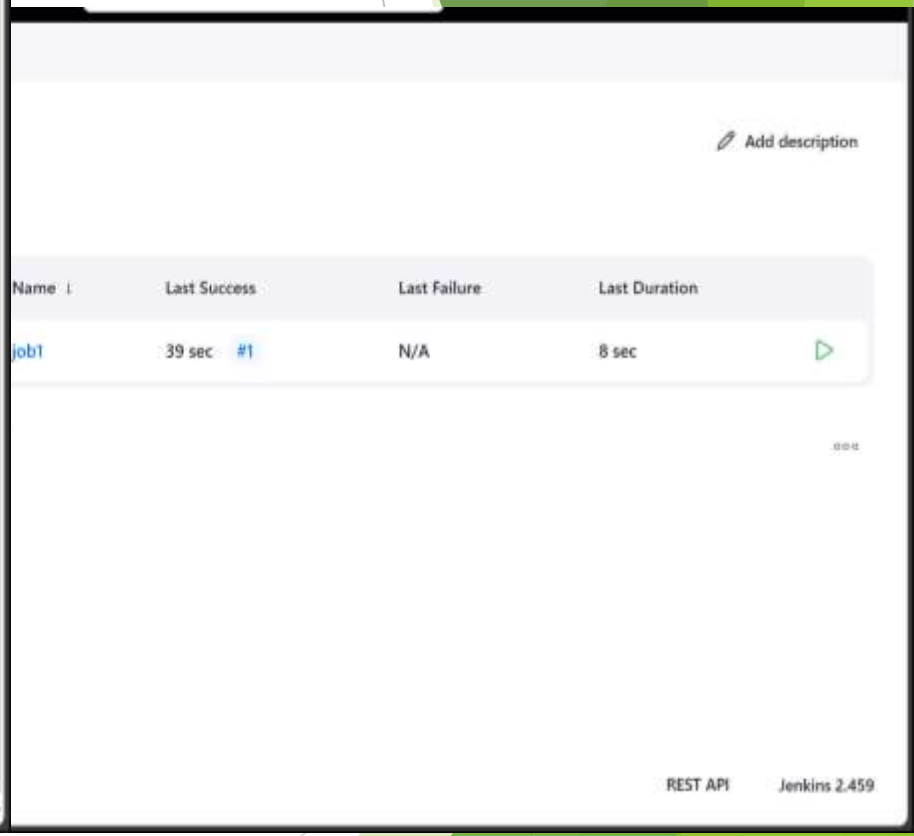
PublicIPs: 35.174.184.249 PrivateIPs: 172.31.52.195

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates.

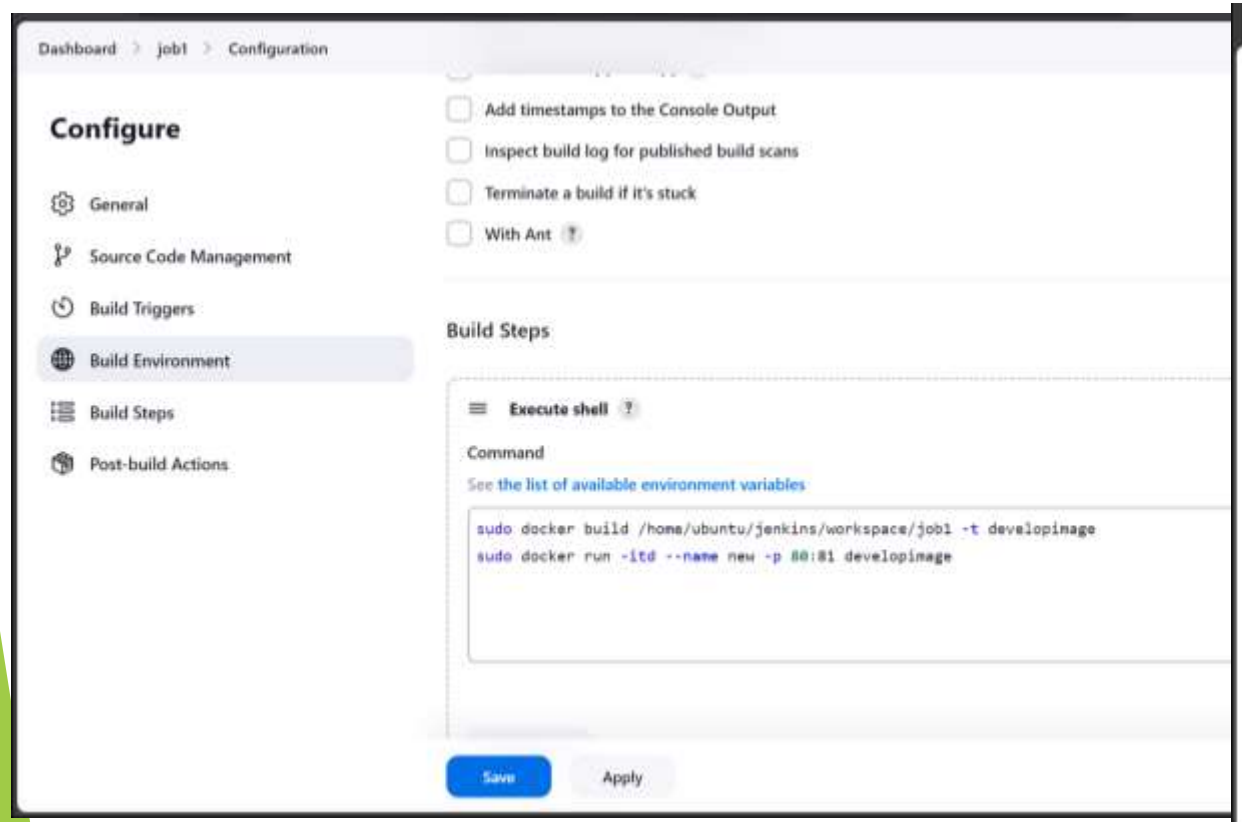


Now create a new job1 and build is successful checking in test instance

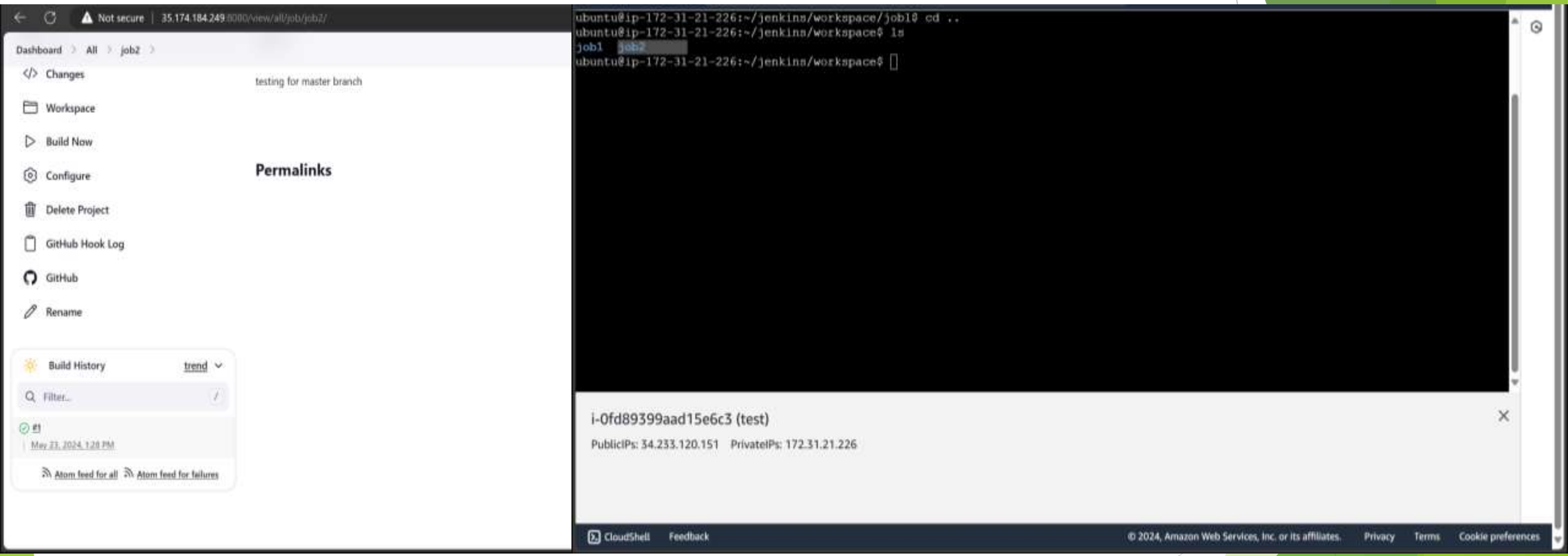




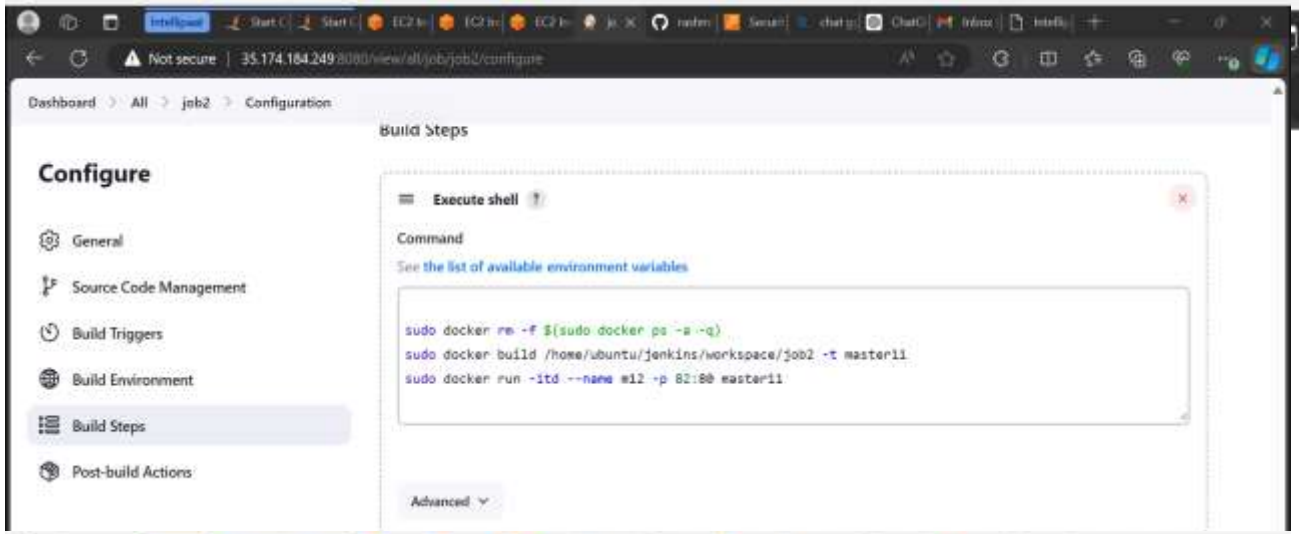
Build a container so that our job is running on local host or not

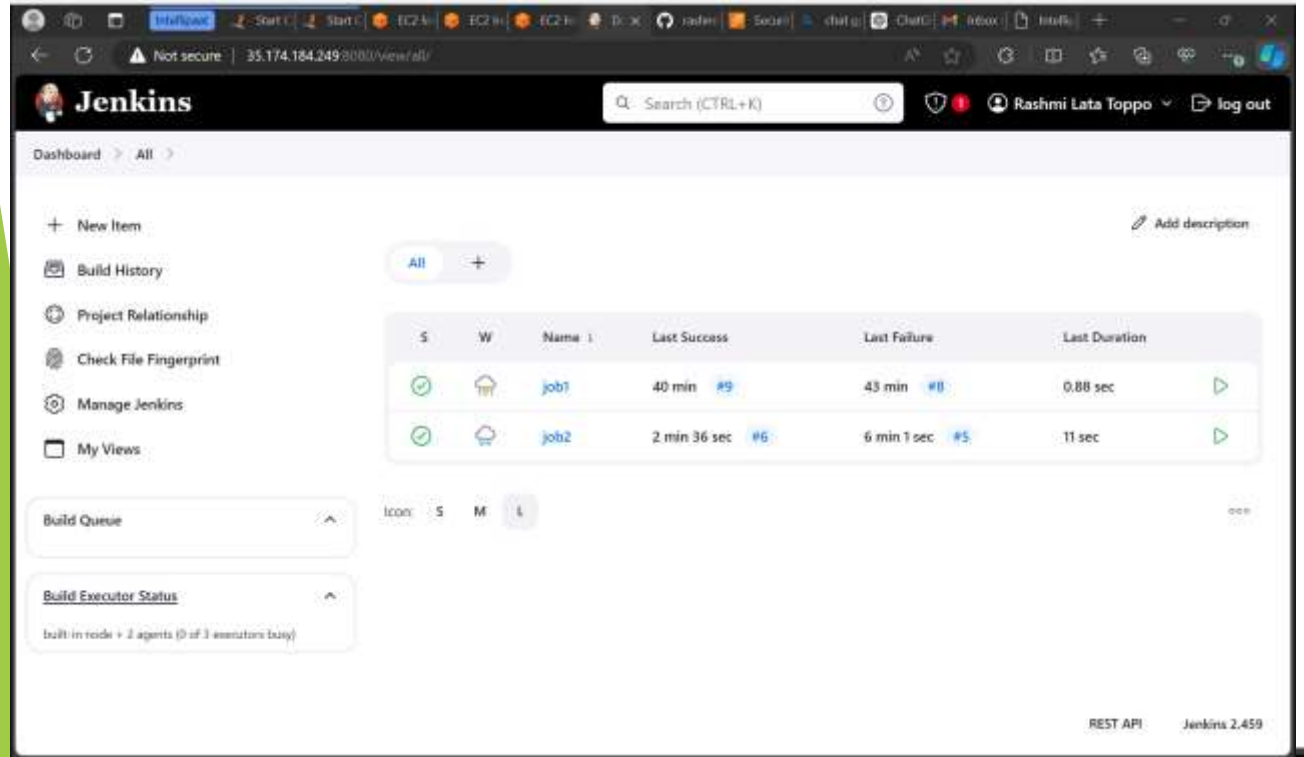


Now create a new job2



Build a container so that our job2 is running on local host or not





Now create a new job3 and build is successful

Build a container so that our job3 is running on local host or not

Not secure | 3.94.100.20

Dashboard > job2 > Configuration

Configure

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

Post-build Actions

Build other projects ?

Projects to build

job3

- (•) Trigger only if build is stable
- ( ) Trigger even if the build is unstable
- ( ) Trigger even if the build fails

Add post-build action

Save Apply

REST API Jenkins 2.459

Hello world!

GitHub