

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

```
graph TD; GR[GitRepository] --> MV[MASTER VM]; GR --> TV[Test VM]; GR --> PV[Prod VM];
```

MASTER VM

- Install Ansible
- Install Jenkins
- Install Java
- Install Docker

Test VM

- Install It should run with dockerfile on develop branch
- Job2: It should 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

The screenshot displays the AWS Management Console's EC2 dashboard. On the left, a navigation sidebar lists various services including EC2 Dashboard, EC2 Global View, Events, Console-to-Code, and categories for Instances, Images, and Elastic Block Store. The main content area is titled 'Instances (3)' and includes a search bar, a filter for 'Instance state = running', and a 'Clear filters' button. Below this, a table lists three instances: 'test' (t2.micro), 'prod' (t2.micro), and 'master' (t2.medium). All instances are in the 'Running' state, with status checks indicating they are initialized or have passed checks. A 'Select an instance' dialog box is open at the bottom of the table.

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
<input type="checkbox"/>	test	i-0fd89399aad15e6c3	Running	t2.micro	Initializing	View alarms
<input type="checkbox"/>	prod	i-0475d43f9bcf4802f	Running	t2.micro	Initializing	View alarms
<input type="checkbox"/>	master	i-0e27d80a1c77795bd	Running	t2.medium	2/2 checks passed	View alarms

Select an instance

Installing ansible in master instance

```
If you face any issues while installing Ansible PPA, file an issue here:
https://github.com/ansible-community/ppa/issues
More info: https://launchpad.net/~ansible/+archive/ubuntu/ansible
Adding repository.
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:5 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu noble InRelease [17.8 kB]
Get:6 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu noble/main amd64 Packages [776 B]
Get:7 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu noble/main Translation-en [472 B]
Fetched 19.1 kB in 1s (27.2 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-52-195:~$ sudo apt install ansible
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ansible-core python3-jmespath python3-kerberos python3-nacl python3-ntlm-auth python3-packaging python3-paramiko
  python3-requests-ntlm python3-resolvelib python3-winrm python3-xlrd python3-xmldict sshpass
Suggested packages:
  python-nacl-doc python3-gssapi python3-invoke
The following NEW packages will be installed:
  ansible ansible-core python3-jmespath python3-kerberos python3-nacl python3-ntlm-auth python3-packaging python3-paramiko
  python3-requests-ntlm python3-resolvelib python3-winrm python3-xlrd python3-xmldict sshpass
0 upgraded, 13 newly installed, 0 to remove and 24 not upgraded.
```

i-0e27d80a1c77795bd (master)

PublicIPs: 54.146.60.250 PrivateIPs: 172.31.52.195

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

```
ubuntu@ip-172-31-52-195:~$  
ubuntu@ip-172-31-52-195:~$ ssh-keygen  
Generating public/private ed25519 key pair.  
Enter file in which to save the key (/home/ubuntu/.ssh/id_ed25519):  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/ubuntu/.ssh/id_ed25519  
Your public key has been saved in /home/ubuntu/.ssh/id_ed25519.pub  
The key fingerprint is:  
SHA256:U+/cEtFq90G2if4q5aH77V7E4ksRgYMjnWutVMWLbOI ubuntu@ip-172-31-52-195  
The key's randomart image is:  
+--[ED25519 256]--+  
|      . o +o |  
|      . = +... |  
|      ..*.oo+ |  
|      .=.o=oo |  
|      S+ +=.++o |  
|      .E+.B =. |  
|      B.= o |  
|      o +o.. |  
|      .+o== |  
+-----[SHA256]-----+  
ubuntu@ip-172-31-52-195:~$ sudo cat /home/ubuntu/.ssh/id_ed25519.pub  
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIGE2cAUWHUQK+2zqT6w+vZ5ABc/btpSxHg8L5n7rCXTk ubuntu@ip-172-31-52-195  
ubuntu@ip-172-31-52-195:~$ cd /etc/ansible  
ubuntu@ip-172-31-52-195:/etc/ansible$
```

i-0e27d80a1c77795bd (master)

PublicIPs: 54.146.60.250 PrivateIPs: 172.31.52.195

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

```
GNU nano 7.2                                hosts *
## [dbservers]
##
## db01.intranet.mydomain.net
## db02.intranet.mydomain.net
## 10.25.1.56
## 10.25.1.57

# Ex4: Multiple hosts arranged into groups such as 'Debian' and 'openSUSE':

## [Debian]
## alpha.example.org
## beta.example.org

## [openSUSE]
## green.example.com
## blue.example.com

[test]
172.31.21.226

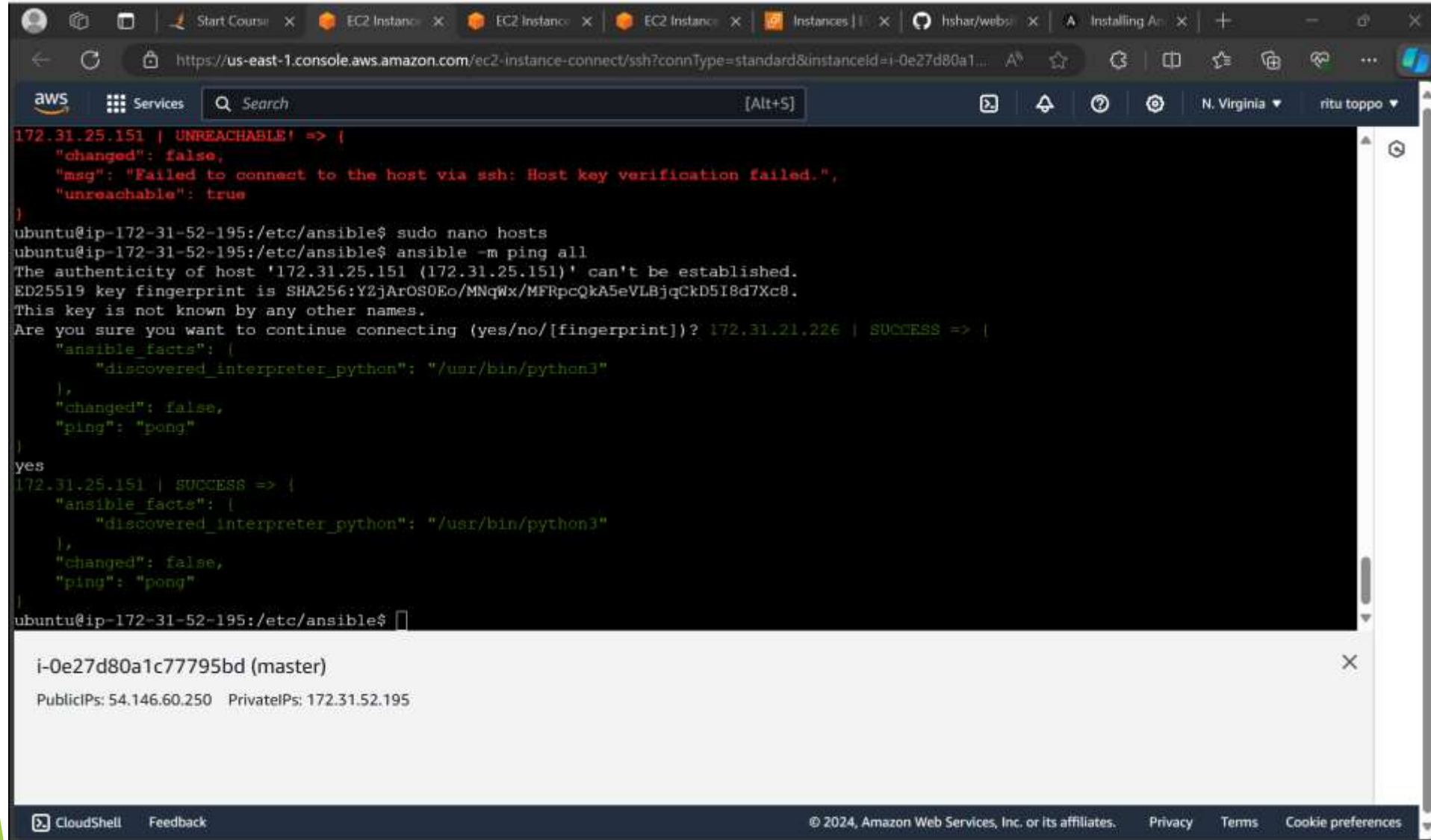
[prod]
172.31.25.151

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line M-E Redo      M-6 Copy

i-0e27d80a1c77795bd (master)
PublicIPs: 54.146.60.250 PrivateIPs: 172.31.52.195

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
```

Connection to master, test and prod has been done successfully



The screenshot shows the AWS CloudShell interface with a terminal window displaying Ansible command output. The terminal shows a failed connection to 172.31.25.151, followed by editing the hosts file and a successful connection to 172.31.21.226. Below the terminal, a summary box for instance i-0e27d80a1c77795bd (master) is visible, showing its public and private IP addresses.

```
172.31.25.151 | UNREACHABLE! => {
  "changed": false,
  "msg": "Failed to connect to the host via ssh: Host key verification failed.",
  "unreachable": true
}
ubuntu@ip-172-31-52-195:/etc/ansible$ sudo nano hosts
ubuntu@ip-172-31-52-195:/etc/ansible$ ansible -m ping all
The authenticity of host '172.31.25.151 (172.31.25.151)' can't be established.
ED25519 key fingerprint is SHA256:YZjArOS0Eo/MNqWx/MFRpcQkA5eVLBjqCkD5I8d7Xc8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? 172.31.21.226 | SUCCESS => |
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
yes
172.31.25.151 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
ubuntu@ip-172-31-52-195:/etc/ansible$
```

i-0e27d80a1c77795bd (master)
PublicIPs: 54.146.60.250 PrivateIPs: 172.31.52.195

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

```
GNU nano 7.2                                install.yaml *
```

```
- hosts: localhost
  become: true
  name: install jenkins, java and docker
  tasks:
    - name: master task
      script: jenkins.sh
- hosts: test
  become: true
  name: install java and docker
  tasks:
    - name: test task
      script: docker.sh
- hosts: prod
  become: true
  name: install java and docker
  tasks:
    - name: prod task
      script: docker.sh
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo M-A Set Mark ^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/_ Go To Line M-E Redo M-6 Copy

i-0e27d80a1c77795bd (master) ×

PublicIPs: 54.146.60.250 PrivateIPs: 172.31.52.195

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Created a docker.sh and Jenkins.sh file

```
The authenticity of host '172.31.25.151 (172.31.25.151)' can't be established.  
ED25519 key fingerprint is SHA256:YZjArOS0Eo/MNqWx/MFRpcQka5eVLbjqCkD5I8d7Xc8.  
This key is not known by any other names.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? 172.31.21.226 | SUCCESS => |  
  "ansible_facts": |  
    "discovered_interpreter_python": "/usr/bin/python3"  
  },  
  "changed": false,  
  "ping": "pong"  
}  
yes  
172.31.25.151 | SUCCESS => |  
  "ansible_facts": |  
    "discovered_interpreter_python": "/usr/bin/python3"  
  },  
  "changed": false,  
  "ping": "pong"  
}  
ubuntu@ip-172-31-52-195:/etc/ansible$ sudo nano install.yaml  
ubuntu@ip-172-31-52-195:/etc/ansible$ sudo nano jenkins.sh  
ubuntu@ip-172-31-52-195:/etc/ansible$ ls  
ansible.cfg  hosts  install.yaml  jenkins.sh  roles  
ubuntu@ip-172-31-52-195:/etc/ansible$ sudo nano docker.sh  
ubuntu@ip-172-31-52-195:/etc/ansible$ ls  
ansible.cfg  docker.sh  hosts  install.yaml  jenkins.sh  roles  
ubuntu@ip-172-31-52-195:/etc/ansible$
```

```
GNU nano 7.2 docker.sh  
sudo apt-get update  
sudo apt-get install openjdk-11-jdk -y  
sudo apt-get install docker.io -y
```

```
GNU nano 7.2 jenkins.sh  
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 openjdk-11-jdk -y  
sudo apt-get install docker.io -y  
sudo apt-get install jenkins
```

i-Oe27d80a1c77795bd (master)

PublicIPs: 54.146.60.250 PrivateIPs: 172.31.52.195

i-Oe27d80a1c77795bd (master)

PublicIPs: 54.146.60.250 PrivateIPs: 172.31.52.195

Syntax Checking and executing the playbook install.yaml successfully installed

```
TASK [master task] *****
changed: [localhost]

PLAY [install java and docker] *****

TASK [Gathering Facts] *****
ok: [172.31.21.226]

TASK [test task] *****
changed: [172.31.21.226]

PLAY [install java and docker] *****

TASK [Gathering Facts] *****
ok: [172.31.25.151]

TASK [prod task] *****
changed: [172.31.25.151]

PLAY RECAP *****
172.31.21.226      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
172.31.25.151     : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
localhost         : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

ubuntu@ip-172-31-52-195:/etc/ansible$
```

i-0e27d80a1c77795bd (master)

PublicIPs: 35.174.184.249 PrivateIPs: 172.31.52.195

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

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

changed: [172.31.21.226]

PLAY [install java and docker] *****

TASK [Gathering Facts] *****

ok: [172.31.25.151]

TASK [prod task] *****

changed: [172.31.25.151]

PLAY RECAP *****

172.31.21.226	: ok=2	changed=1	unreachable=0	failed=0	skipped=0	rescued=0	ignored=0
172.31.25.151	: ok=2	changed=1	unreachable=0	failed=0	skipped=0	rescued=0	ignored=0
localhost	: ok=2	changed=1	unreachable=0	failed=0	skipped=0	rescued=0	ignored=0

ubuntu@ip-172-31-52-195:/etc/ansible\$

ubuntu@ip-172-31-52-195:/etc/ansible\$

ubuntu@ip-172-31-52-195:/etc/ansible\$ java --version

openjdk 17.0.11 2024-04-16

OpenJDK Runtime Environment (build 17.0.11+9-Ubuntu-1)

OpenJDK 64-Bit Server VM (build 17.0.11+9-Ubuntu-1, mixed mode, sharing)

ubuntu@ip-172-31-52-195:/etc/ansible\$ which jenkins

/usr/bin/jenkins

ubuntu@ip-172-31-52-195:/etc/ansible\$ docker --version

Docker version 24.0.7, build 24.0.7-0ubuntu4

ubuntu@ip-172-31-52-195:/etc/ansible\$

i-0e27d80a1c77795bd (master)

PublicIPs: 35.174.184.249 PrivateIPs: 172.31.52.195

Adding public key ssh connection in test instance

```
ubuntu@ip-172-31-21-226:~$ cd .ssh
ubuntu@ip-172-31-21-226:~/.ssh$ ls
authorized_keys
ubuntu@ip-172-31-21-226:~/.ssh$ sudo nano authorized_keys
ubuntu@ip-172-31-21-226:~/.ssh$
```

i-Ofd89399aad15e6c3 (test)

PublicIPs: 34.229.87.233 PrivateIPs: 172.31.21.226

Checking the installation of Java, Jenkins and docker in test

```
ubuntu@ip-172-31-21-226:~$  
ubuntu@ip-172-31-21-226:~$  
ubuntu@ip-172-31-21-226:~$ cd .ssh  
ubuntu@ip-172-31-21-226:~/.ssh$ ls  
authorized_keys  
ubuntu@ip-172-31-21-226:~/.ssh$ sudo nano authorized_keys  
ubuntu@ip-172-31-21-226:~/.ssh$ sudo nano authorized_keys  
ubuntu@ip-172-31-21-226:~/.ssh$ java --version  
openjdk 11.0.23 2024-04-16  
OpenJDK Runtime Environment (build 11.0.23+9-post-Ubuntu-lubuntul)  
OpenJDK 64-Bit Server VM (build 11.0.23+9-post-Ubuntu-lubuntul, mixed mode, sharing)  
ubuntu@ip-172-31-21-226:~/.ssh$ docker --version  
Docker version 24.0.7, build 24.0.7-0ubuntu4  
ubuntu@ip-172-31-21-226:~/.ssh$
```

i-Ofd89399aad15e6c3 (test)

PublicIPs: 34.233.120.151 PrivateIPs: 172.31.21.226

prod instance

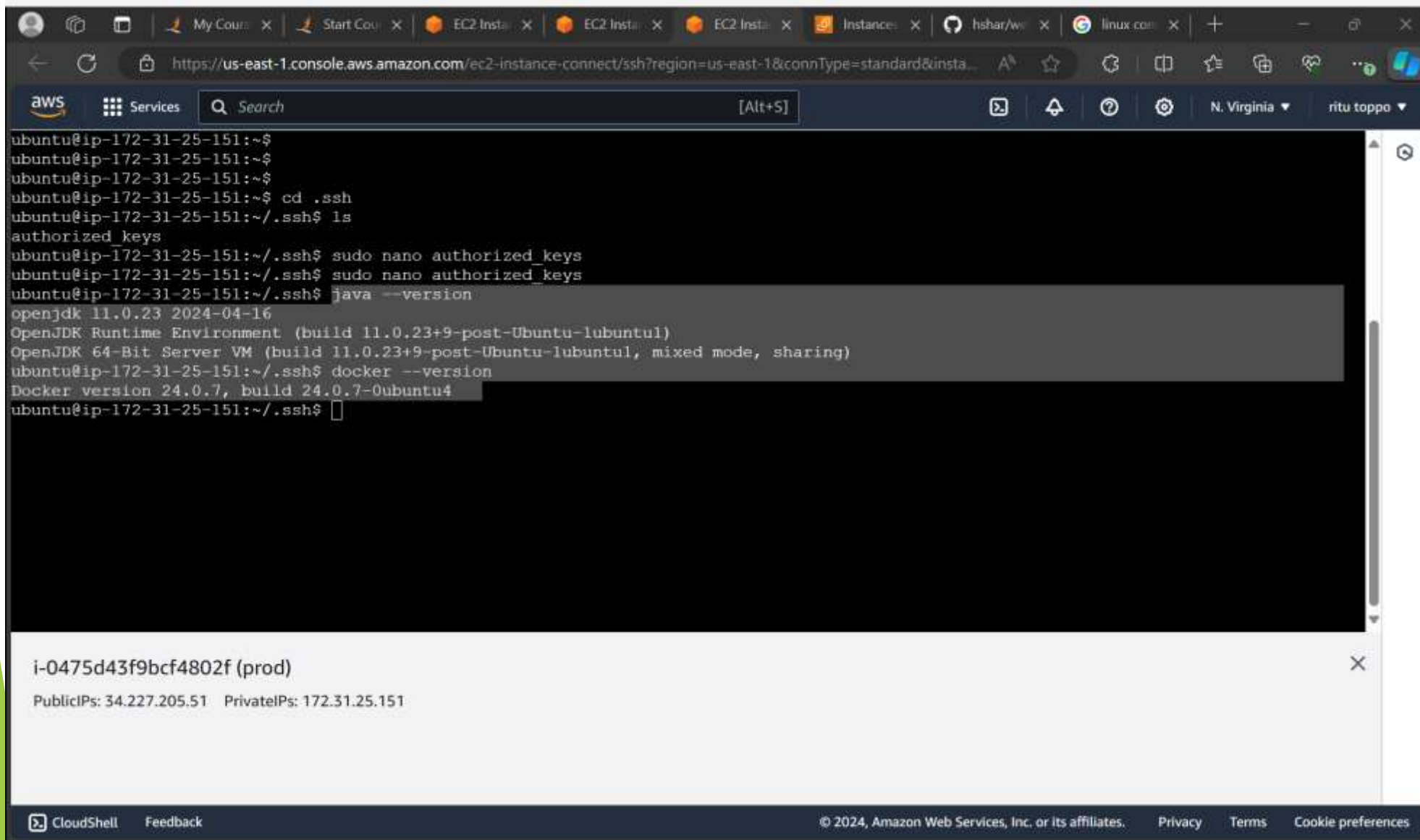
ssh connection with master

```
ubuntu@ip-172-31-25-151:~$ cd .ssh
ubuntu@ip-172-31-25-151:~/.ssh$ ls
authorized_keys
ubuntu@ip-172-31-25-151:~/.ssh$ sudo nano authorized_keys
ubuntu@ip-172-31-25-151:~/.ssh$
```

i-0475d43f9bcf4802f (prod)

PublicIPs: 34.229.142.99 PrivateIPs: 172.31.25.151

Checking the installation of Java and docker in prod



The screenshot shows an AWS CloudShell terminal window with the following content:

```
ubuntu@ip-172-31-25-151:~$  
ubuntu@ip-172-31-25-151:~$  
ubuntu@ip-172-31-25-151:~$  
ubuntu@ip-172-31-25-151:~$ cd .ssh  
ubuntu@ip-172-31-25-151:~/.ssh$ ls  
authorized_keys  
ubuntu@ip-172-31-25-151:~/.ssh$ sudo nano authorized_keys  
ubuntu@ip-172-31-25-151:~/.ssh$ sudo nano authorized_keys  
ubuntu@ip-172-31-25-151:~/.ssh$ java --version  
openjdk 11.0.23 2024-04-16  
OpenJDK Runtime Environment (build 11.0.23+9-post-Ubuntu-lubuntul)  
OpenJDK 64-Bit Server VM (build 11.0.23+9-post-Ubuntu-lubuntul, mixed mode, sharing)  
ubuntu@ip-172-31-25-151:~/.ssh$ docker --version  
Docker version 24.0.7, build 24.0.7-0ubuntu4  
ubuntu@ip-172-31-25-151:~/.ssh$
```

Below the terminal window, the instance details are shown:

i-0475d43f9bcf4802f (prod)
PublicIPs: 34.227.205.51 PrivateIPs: 172.31.25.151

The footer of the CloudShell window includes: CloudShell Feedback, © 2024, Amazon Web Services, Inc. or its affiliates, Privacy, Terms, and Cookie preferences.

Login to Jenkins

```
ubuntu@ip-172-31-52-195:/etc/ansible$  
ubuntu@ip-172-31-52-195:/etc/ansible$ cd  
ubuntu@ip-172-31-52-195:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword  
36106743c1a449dfb700172d1659b0f8  
ubuntu@ip-172-31-52-195:~$
```

i-Oe27d80a1c77795bd (master)

PublicIPs: 35.174.184.249 PrivateIPs: 172.31.52.195

CloudShell Feedback

© 2024, Amazon Web Services, Inc.

← ↻ ⚠ Not secure | 35.174.184.249:8080/login?from=%2F

Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

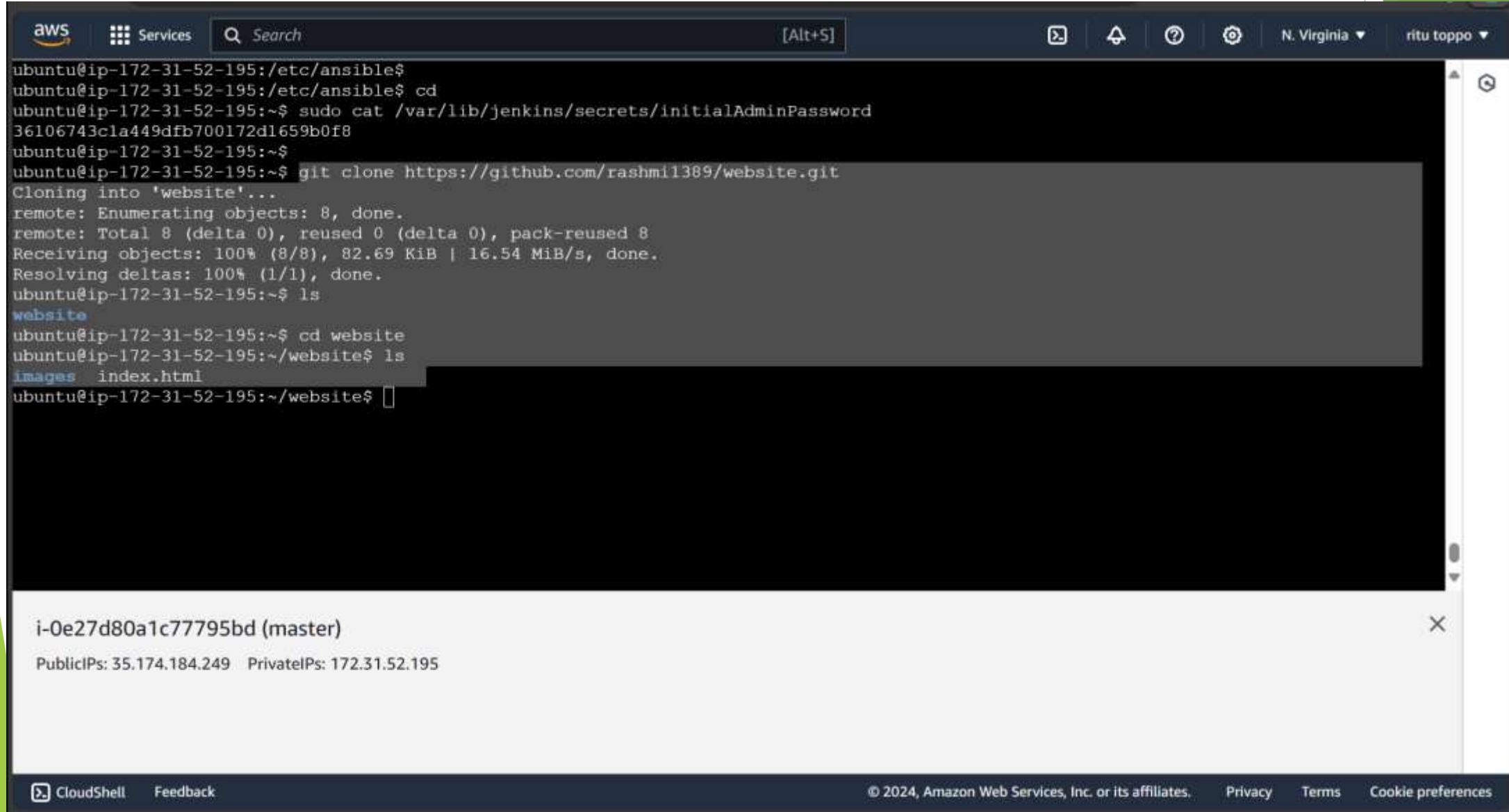
```
/var/lib/jenkins/secrets/initialAdminPassword
```

Please copy the password from either location and paste it below.

Administrator password

Continue

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

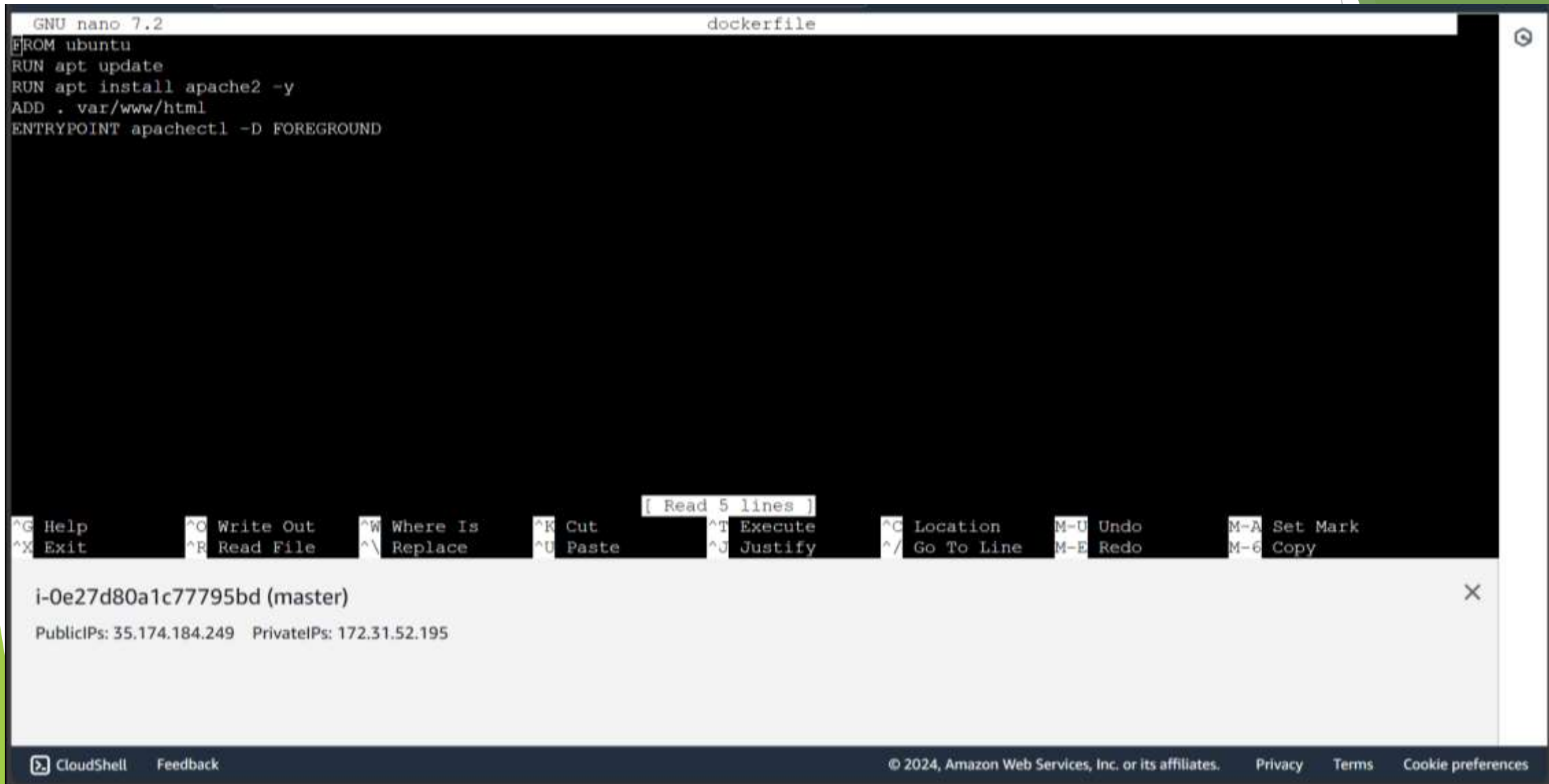


The screenshot shows an AWS CloudShell terminal window. The top bar includes the AWS logo, a 'Services' menu, a search bar, and a '[Alt+S]' shortcut. The right side of the bar shows the region 'N. Virginia' and the user 'ritu toppo'. The terminal output shows the following commands and results:

```
ubuntu@ip-172-31-52-195:/etc/ansible$  
ubuntu@ip-172-31-52-195:/etc/ansible$ cd  
ubuntu@ip-172-31-52-195:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword  
36106743cla449dfb700172d1659b0f8  
ubuntu@ip-172-31-52-195:~$  
ubuntu@ip-172-31-52-195:~$ git clone https://github.com/rashmi1389/website.git  
Cloning into 'website'...  
remote: Enumerating objects: 8, done.  
remote: Total 8 (delta 0), reused 0 (delta 0), pack-reused 8  
Receiving objects: 100% (8/8), 82.69 KiB | 16.54 MiB/s, done.  
Resolving deltas: 100% (1/1), done.  
ubuntu@ip-172-31-52-195:~$ ls  
website  
ubuntu@ip-172-31-52-195:~$ cd website  
ubuntu@ip-172-31-52-195:~/website$ ls  
images index.html  
ubuntu@ip-172-31-52-195:~/website$
```

Below the terminal window, a summary box displays the instance ID 'i-0e27d80a1c77795bd (master)' and IP addresses: 'PublicIPs: 35.174.184.249' and 'PrivateIPs: 172.31.52.195'. The bottom of the window features a 'CloudShell' logo, a 'Feedback' link, and copyright information: '© 2024, Amazon Web Services, Inc. or its affiliates.' along with links for 'Privacy', 'Terms', and 'Cookie preferences'.

Create a docker file in master



The screenshot shows a CloudShell terminal window with a nano 7.2 editor open to a file named 'dockerfile'. The editor contains the following text:

```
FROM ubuntu
RUN apt update
RUN apt install apache2 -y
ADD . var/www/html
ENTRYPOINT apachectl -D FOREGROUND
```

Below the editor, a status bar shows the file path 'i-0e27d80a1c77795bd (master)' and its IP addresses: 'PublicIPs: 35.174.184.249 PrivateIPs: 172.31.52.195'. A help menu is visible at the bottom of the editor, listing various commands and their shortcuts. The bottom of the terminal window features a footer with the CloudShell logo, a feedback link, and copyright information for Amazon Web Services, Inc. or its affiliates, along with links to privacy, terms, and cookie preferences.

```
GNU nano 7.2 dockerfile
FROM ubuntu
RUN apt update
RUN apt install apache2 -y
ADD . var/www/html
ENTRYPOINT apachectl -D FOREGROUND

[ Read 5 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location  M-U Undo     M-A Set Mark
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line M-E Redo     M-6 Copy

i-0e27d80a1c77795bd (master)
PublicIPs: 35.174.184.249 PrivateIPs: 172.31.52.195

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
```

Add the file in git

```
Untracked files:
  (use "git add <file>..." to include in what will be committed)
  dockerfile

nothing added to commit but untracked files present (use "git add" to track)
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$
```

i-0e27d80a1c77795bd (master)

PublicIPs: 35.174.184.249 PrivateIPs: 172.31.52.195

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

The screenshot shows the Jenkins 'Nodes' page. The left sidebar contains a 'Nodes' tab, a 'Clouds' tab, a 'Build Queue' section (showing 'No builds in the queue'), and a 'Build Executor Status' section. The 'Build Executor Status' section shows two nodes: 'Built-In Node' with 1 idle executor and 'Test Node' with 1 idle executor. The main area displays a table of nodes. The table has columns for 'S' (status icon), 'Name', 'Architecture', 'Clock Difference', 'Free Disk Space', 'Free Swap Space', 'Free Temp Space', and 'Response Time'. There are two rows: 'Built-In Node' and 'Test Node'. Both are Linux (amd64) and 'In sync'. The 'Test Node' has 3.85 GiB of free disk space, 0 B of free swap space, and 3.85 GiB of free temp space, with a response time of 70ms. Below the table, there is a 'Data obtained' row showing 6.9 sec for several metrics. At the bottom, there is a legend for the 'Icon' (S, M, L) and the version 'Jenkins 2.459'.

Dashboard > Manage Jenkins > Nodes >

Nodes

Clouds

Build Queue

No builds in the queue.

Build Executor Status

Built-In Node

1 Idle

2 Idle

Test Node

1 Idle

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	2.42 GiB	0 B	2.42 GiB	0ms
	Test Node	Linux (amd64)	In sync	3.85 GiB	0 B	3.85 GiB	70ms
Data obtained		6.9 sec	6.9 sec	6.9 sec	6.8 sec	6.9 sec	6.9 sec

Icon: S M L

Legend

REST API Jenkins 2.459

Connect the prod instance to Jenkins as prod node

Dashboard > Manage Jenkins > Nodes >

Nodes

Clouds

Build Queue

No builds in the queue.

Build Executor Status

Built-In Node

1 Idle

2 Idle

Test Node

1 Idle

Nodes

New Node

Configure Monitors

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	2.42 GiB	<div>! 0 B</div>	2.42 GiB	0ms
	Test Node	Linux (amd64)	In sync	3.85 GiB	<div>! 0 B</div>	3.85 GiB	70ms
Data obtained		6.9 sec	6.9 sec	6.9 sec	6.8 sec	6.9 sec	6.9 sec

Icon:

S

M

L

Legend

REST API

Jenkins 2.459

Push the dockerfile to develop branch

```
ubuntu@ip-172-31-52-195:~/website$ git branch
* 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': rashmil389
Password for 'https://rashmil389@github.com':
remote: Invalid username or password.
fatal: Authentication failed for 'https://github.com/rashmil389/website.git/'
ubuntu@ip-172-31-52-195:~/website$ git push origin develop
Username for 'https://github.com': rashmil389
Password for 'https://rashmil389@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), 418 bytes | 418.00 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/rashmil389/website/pull/new/develop
remote:
To https://github.com/rashmil389/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.

The screenshot shows the GitHub interface for a repository named 'website'. The repository is public and was forked from 'hsjar/website'. It has 2 branches and 0 tags. The 'develop' branch is selected, and it is 1 commit ahead of the 'hsjar/website:master' branch. A commit titled 'Ubuntu: adding dockerfile' is shown, made 1 hour ago. The commit details show three files: 'images' (final, 5 years ago), 'dockerfile' (adding dockerfile, 1 hour ago), and 'index.html' (modified, 5 years ago). The repository has no description, 0 stars, 0 watching, and 0 forks. There are no releases or packages published. A 'Add a README' button is visible at the bottom.

Code Pull requests Actions Projects Wiki Security Insights Settings

website Public

forked from [hsjar/website](#)

develop 2 Branches 0 Tags

Go to file + Code

About

No description, website, or topics provided.

Activity

0 stars

0 watching

0 forks

Releases

No releases published

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

Languages

Add a README

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

```
ubuntu@ip-172-31-21-226:~$ cd jenkins
ubuntu@ip-172-31-21-226:~/jenkins$ ls
remoting  remoting.jar
ubuntu@ip-172-31-21-226:~/jenkins$ ls
remoting  remoting.jar  workspace
ubuntu@ip-172-31-21-226:~/jenkins$ cd workspace
ubuntu@ip-172-31-21-226:~/jenkins/workspace$ ls
job1
ubuntu@ip-172-31-21-226:~/jenkins/workspace$
```

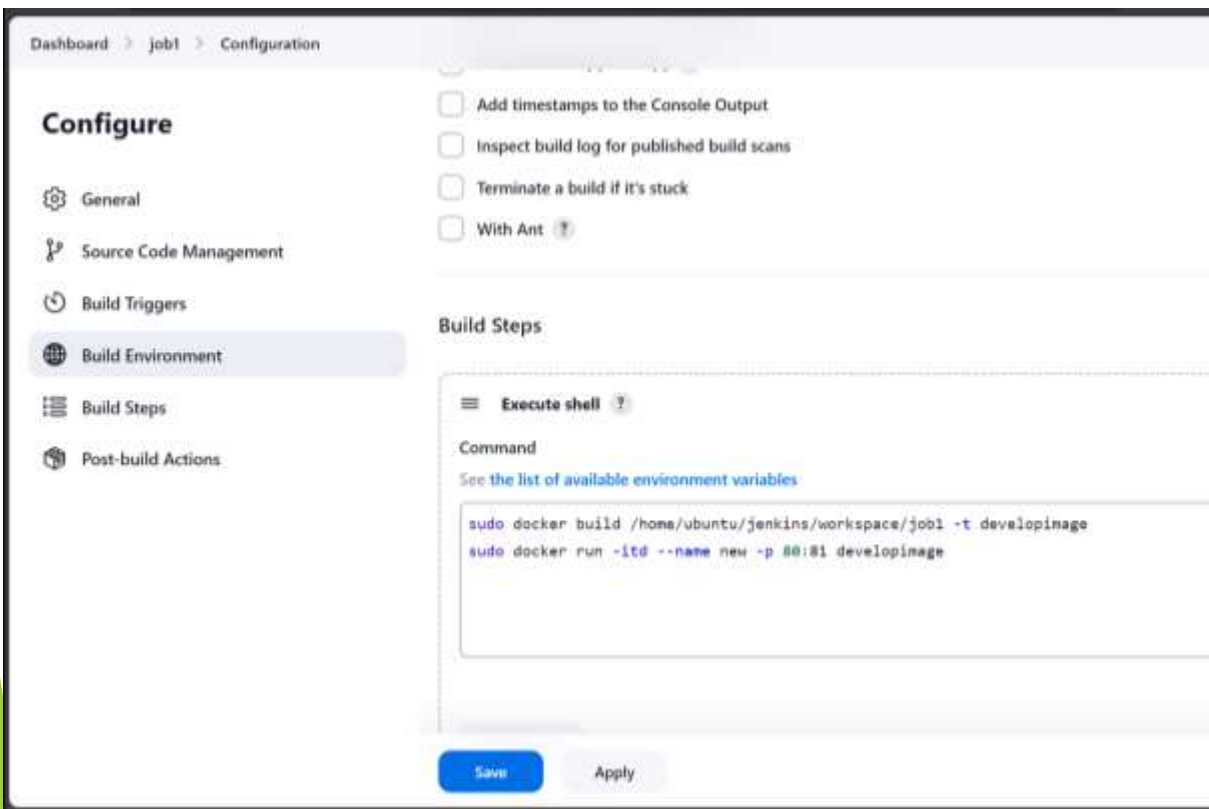
[Add description](#)

Name	Last Success	Last Failure	Last Duration
job1	39 sec #1	N/A	8 sec

...

REST API Jenkins 2.459

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



Now create a new job2

← ↻ ⚠ Not secure | 35.174.184.249:8000/view/all/job/job2/

Dashboard > All > job2

</> Changes

Workspace

▶ Build Now

⚙ Configure

🗑 Delete Project

📄 GitHub Hook Log

🔄 GitHub

✎ Rename

🔦 Build History trend

🔍 Filter...

✅ May 23, 2024, 1:20 PM

📡 Atom feed for all 📡 Atom feed for failures

testing for master branch

Permalinks

```
ubuntu@ip-172-31-21-226:~/jenkins/workspace/job1$ cd ..
ubuntu@ip-172-31-21-226:~/jenkins/workspace$ ls
job1  job2
ubuntu@ip-172-31-21-226:~/jenkins/workspace$
```

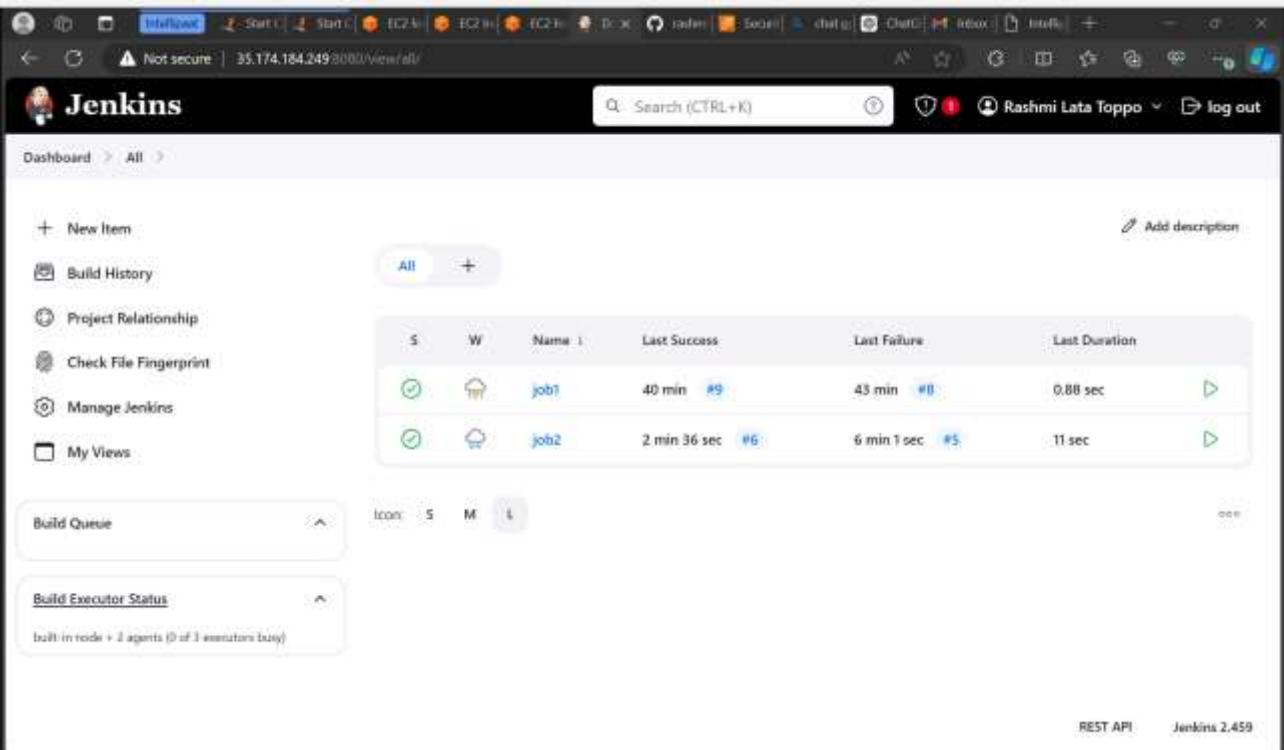
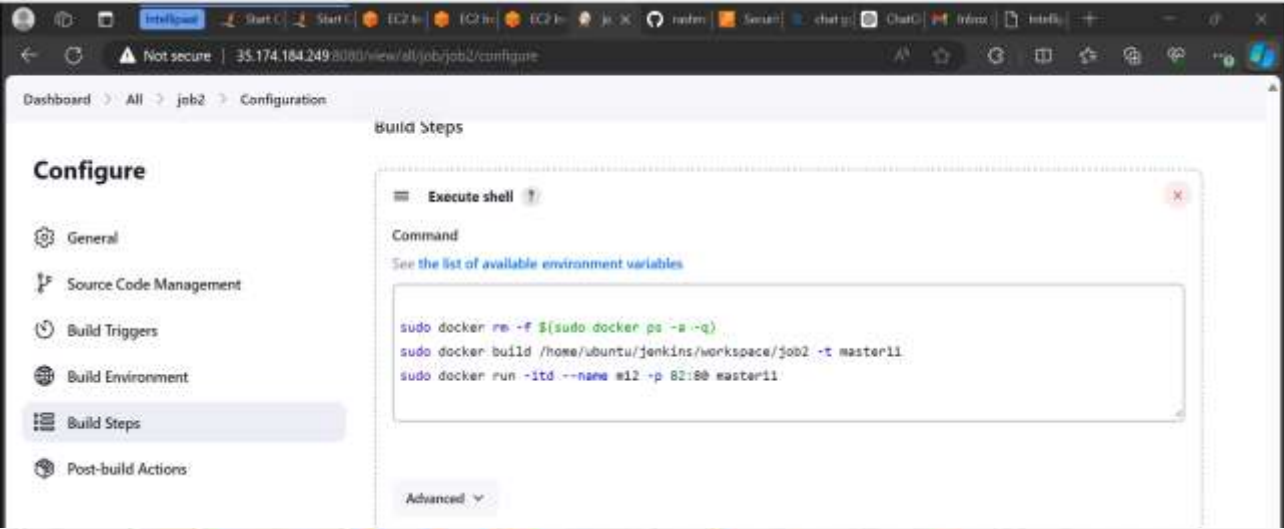
i-0fd89399aad15e6c3 (test)

PublicIPs: 34.233.120.151 PrivateIPs: 172.31.21.226

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

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



Now create a new job3 and build is successful

Dashboard > job3 >

Status

Changes

Workspace

Build Now

Configure

Delete Project

GitHub

Rename

job3

release product

Permalinks

Edit description

Disable Project

Build History

trend

Filter...

#1

May 23, 2024, 2:32 PM

Atom feed for all

Atom feed for failures

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

The image shows a Jenkins web interface. The left sidebar contains a 'Configure' section with a list of options: General, Source Code Management, Build Triggers, Build Environment, Build Steps, and Post-build Actions. The 'Post-build Actions' option is selected. The main content area is titled 'Post-build Actions' and shows a configuration for 'Build other projects'. The 'Projects to build' field contains 'job3'. There are three radio button options: 'Trigger only if build is stable' (selected), 'Trigger even if the build is unstable', and 'Trigger even if the build fails'. Below these options is a button 'Add post-build action'. At the bottom of the configuration area are 'Save' and 'Apply' buttons. The bottom right of the Jenkins interface shows 'REST API' and 'Jenkins 2.459'. To the right of the Jenkins interface is a separate window displaying 'Hello world!' at the top, a GitHub Octocat logo in the center, and the 'GitHub' logo at the bottom.

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

☒ 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