Problem Statement

AppleBite Co. is using Cloud for one of their products. The project uses modular components, multiple frameworks and want the components to be developed by different teams or by 3rd-party vendors.

The company's goal is to deliver the product updates frequently to production with High quality & Reliability. They also want to accelerate software delivery speed, quality and reduce feedback time between developers and testers.

As development progressed, they are facing multiple problems, because of various technologies involved in the project. Following are the problems:

- Building Complex builds is difficult
- Incremental builds are difficult to manage, and deploy

To solve these problems, they need to implement Continuous Integration & Continuous Deployment with DevOps using following tools:

Git – For version control for tracking changes in the code files

Jenkins – For continuous integration and continuous deployment

Docker – For deploying containerized applications

Ansible - Configuration management tools

This project will be about how to do deploy code to dev/stage/prod etc, just on a click of button.

Link for the sample PHP application: https://github.com/edureka-devops/projCert.git

Business challenge/requirement

As soon as the developer pushes the updated code on the GIT master branch, a new test server should be provisioned with all the required software. Post this, the code should be containerized and deployed on the test server.

The deployment should then be built and pushed to the prod server.

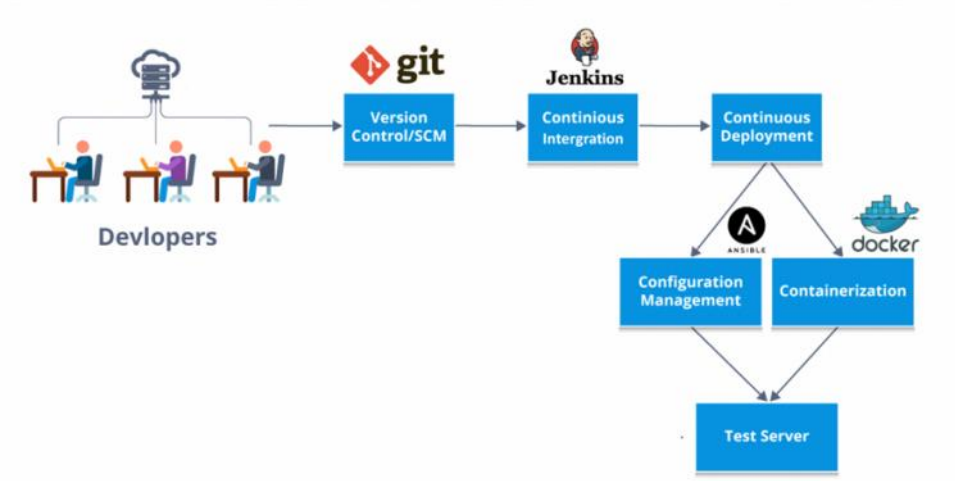
All this should happen automatically and should be triggered from a push to the GitHub master branch.

Steps for executing the solution:

- Use the Master VM for Jenkins, Ansible, GIT etc.
- Use the fresh instance for Jenkins Slave Node (Test Server)
- Change the IP address of the VMs accordingly
- Add Build Pipeline Plugin and Post-build task plugin to Jenkins on the master VM
- Install python, openssh-server and git on the slave node manually
- Use the image devopsedu/webapp and add your PHP website to it using a Dockerfile
- Push the PHP website, and the Dockerfile to a git repository

Below tasks should be automated through Jenkins by creating a pipeline:

1. Install and configure puppet agent on the slave node (Job 1)
2. Push an Ansible configuration on test server to install docker (Job 2)
3. Pull the PHP website, and the Dockerfile from the git repo and build and deploy your PHP docker container. After. (Job 3)
4. If Job 3 fails, delete the running container on Test Server



==========

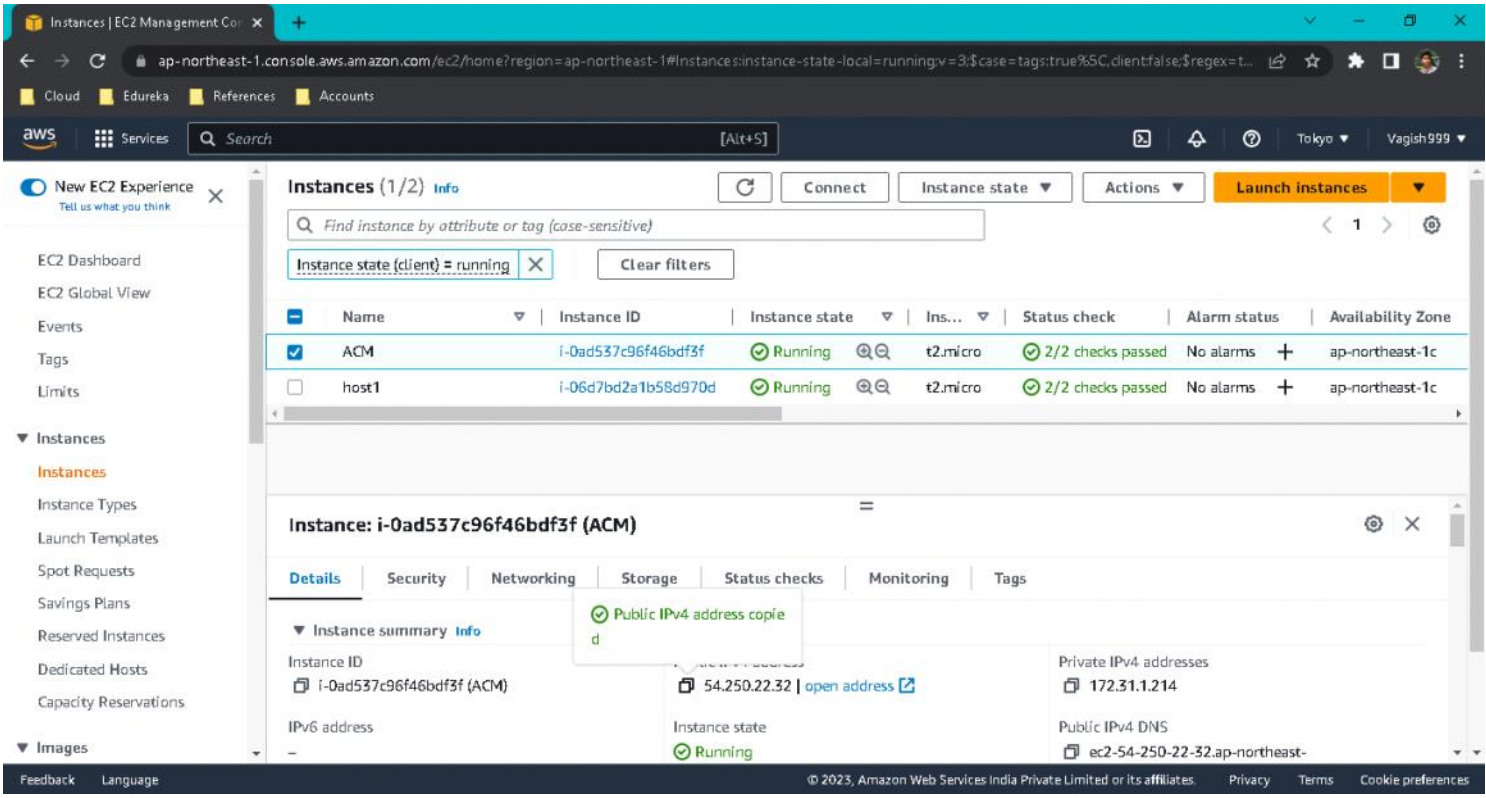
Solution:

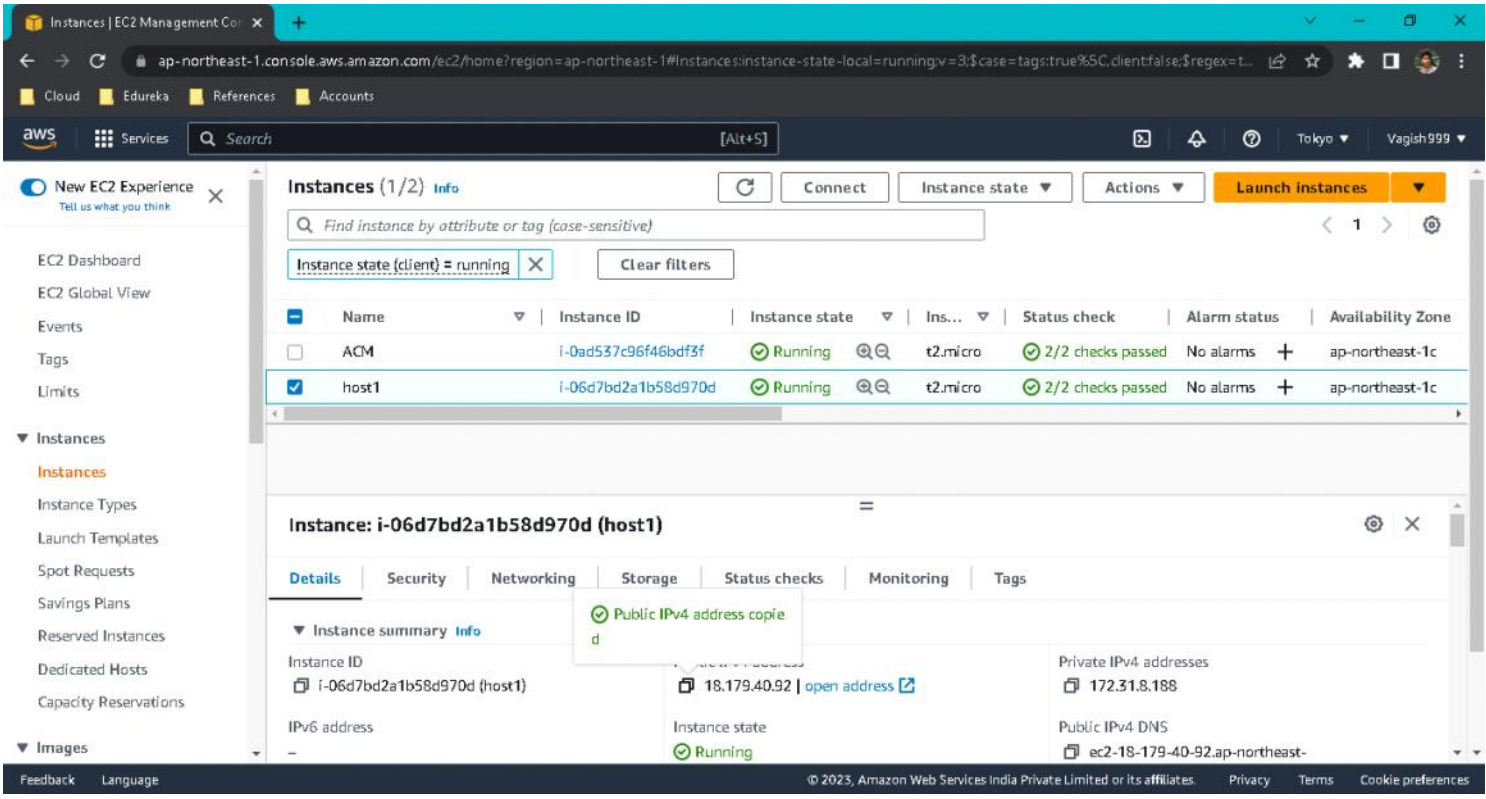
- Downloaded shared sample project and uploaded on a fresh GIT Repo
  a. GIT Repo URL: https://github.com/Vagish619/Edureka-Projects.git
  b. Branch Name: project_1

==========

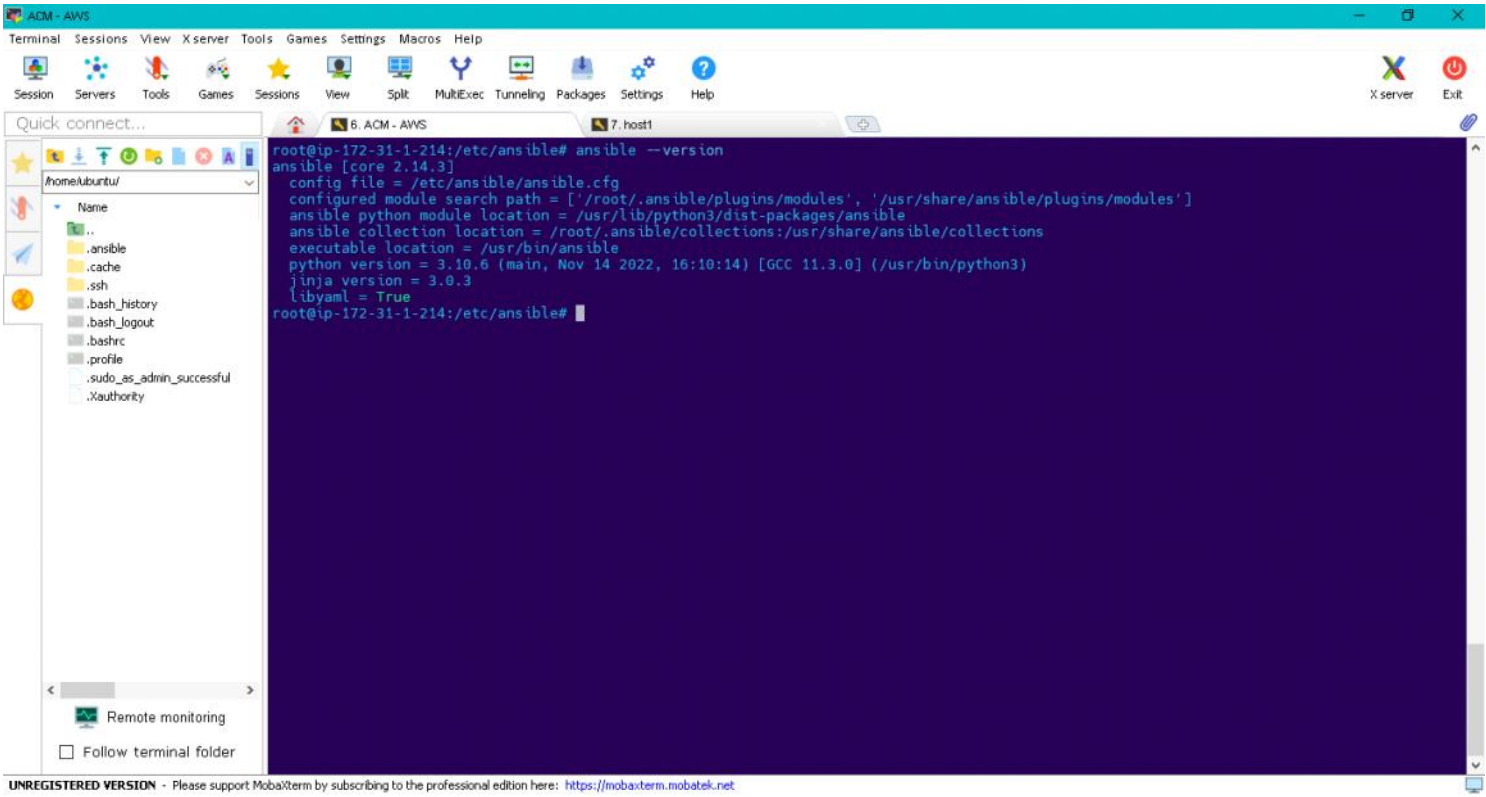
- Created Two VM Instances on AWS to be served as Master & Slav e for this project.

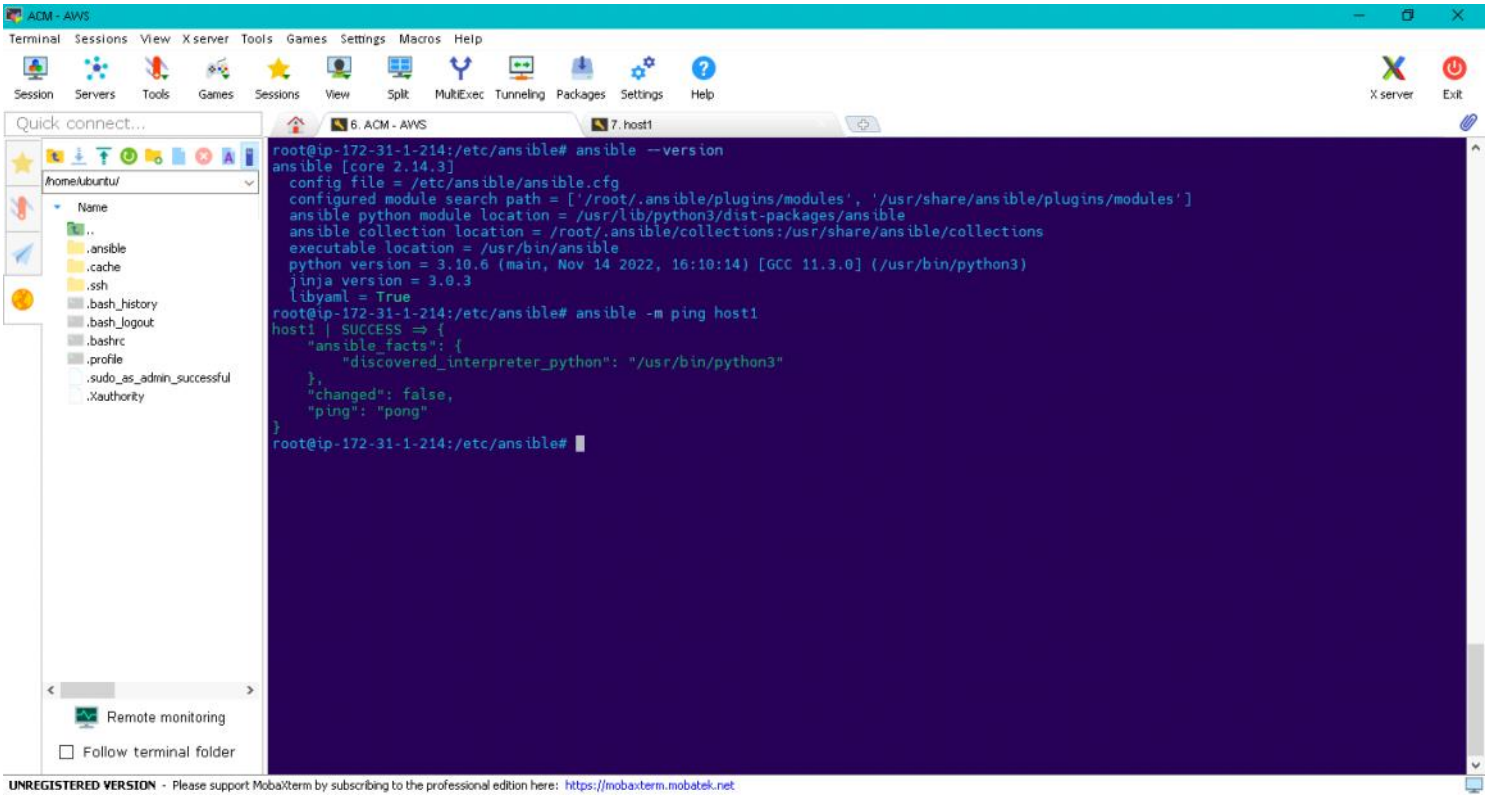
a. Master VM Screenshot details:
   i. Name: ACM
   ii. IP : 54.250.22.32
b. Slave VM Screenshot details:
   i. Name: host1
   ii. IP: 18.179.40.92

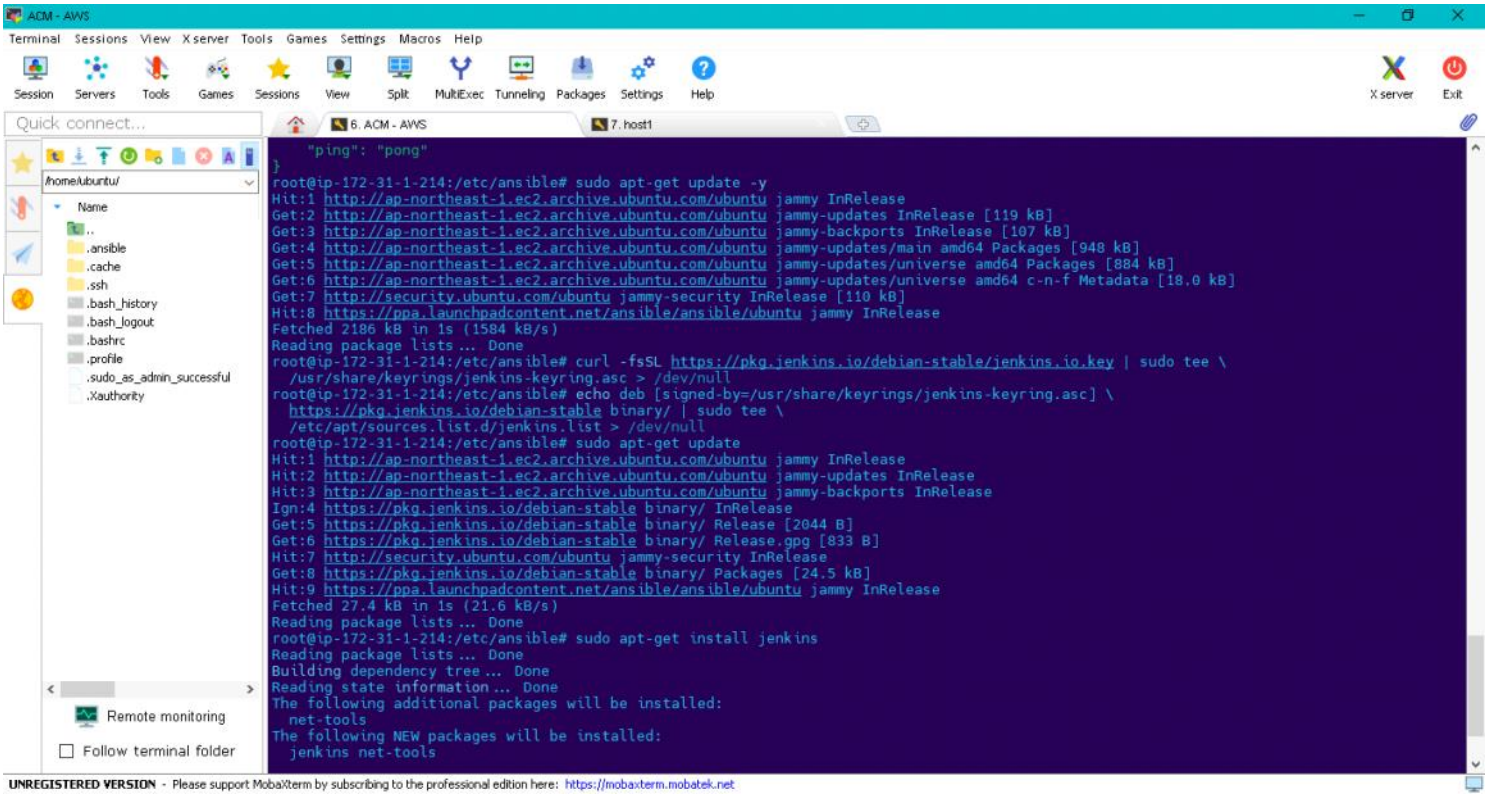


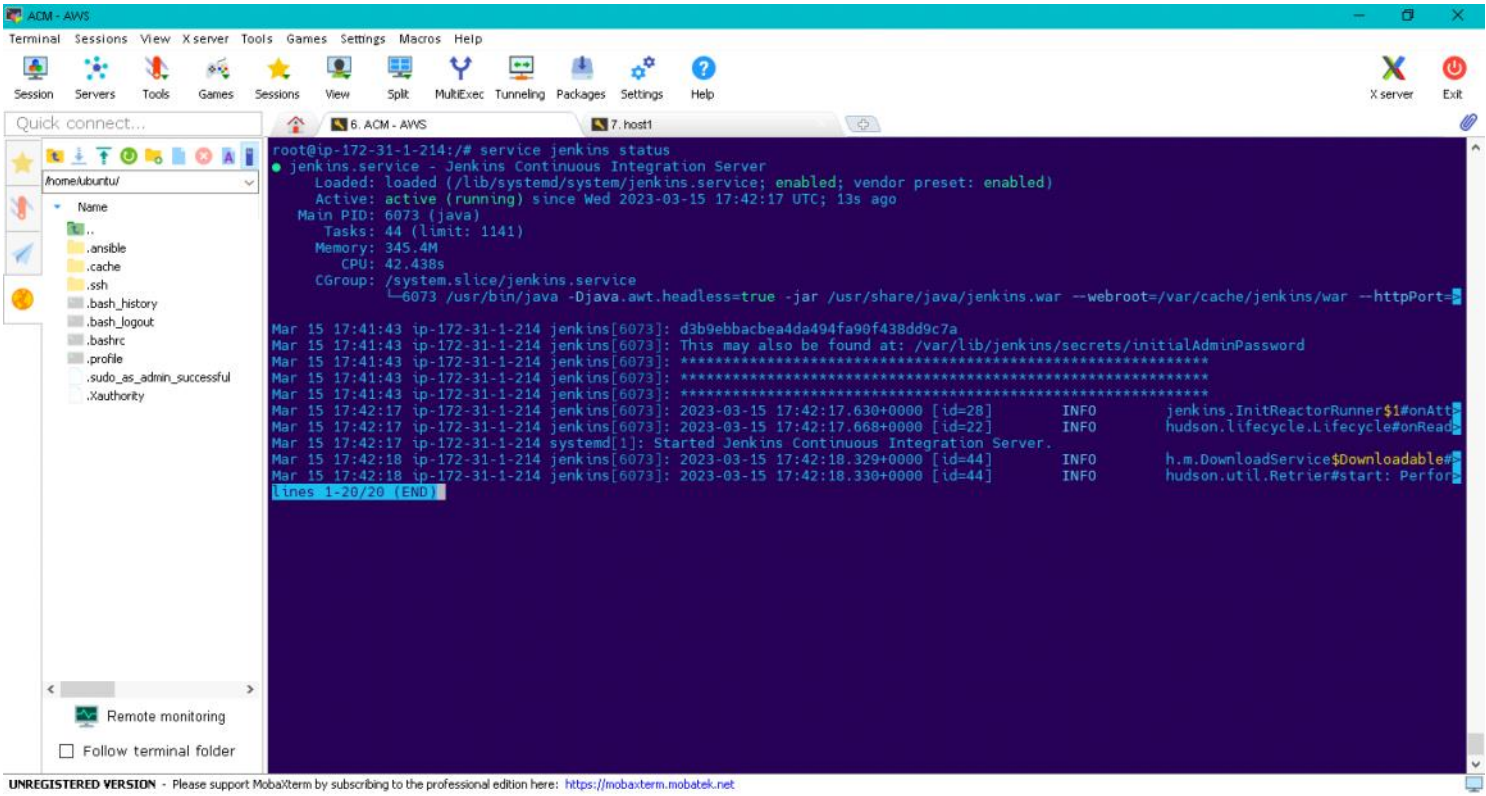


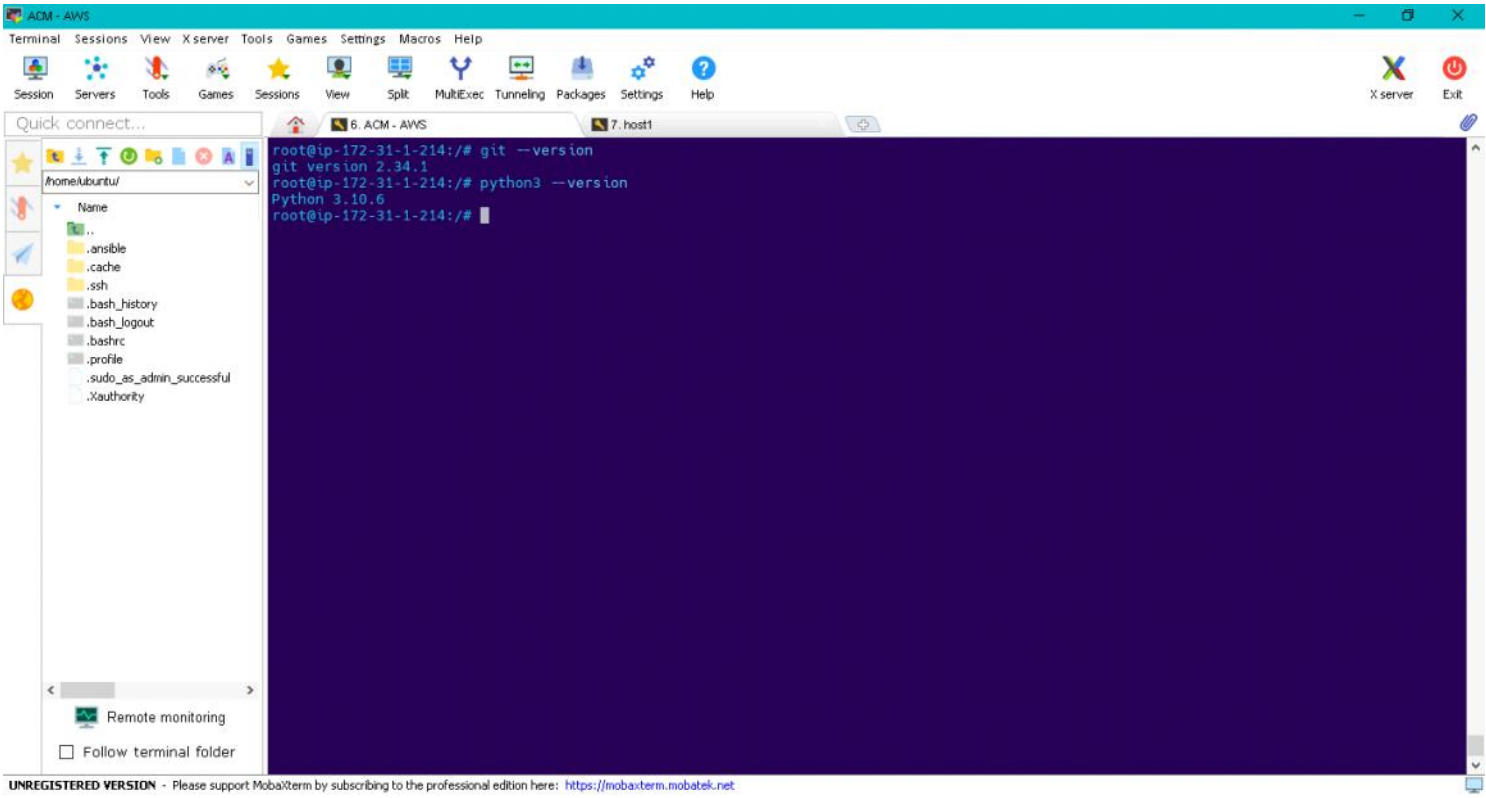
- Installed / ensured below tools on Master VM : ACM
  a. Ansible
  b. Jenkins
  c. GIT
  d. Python



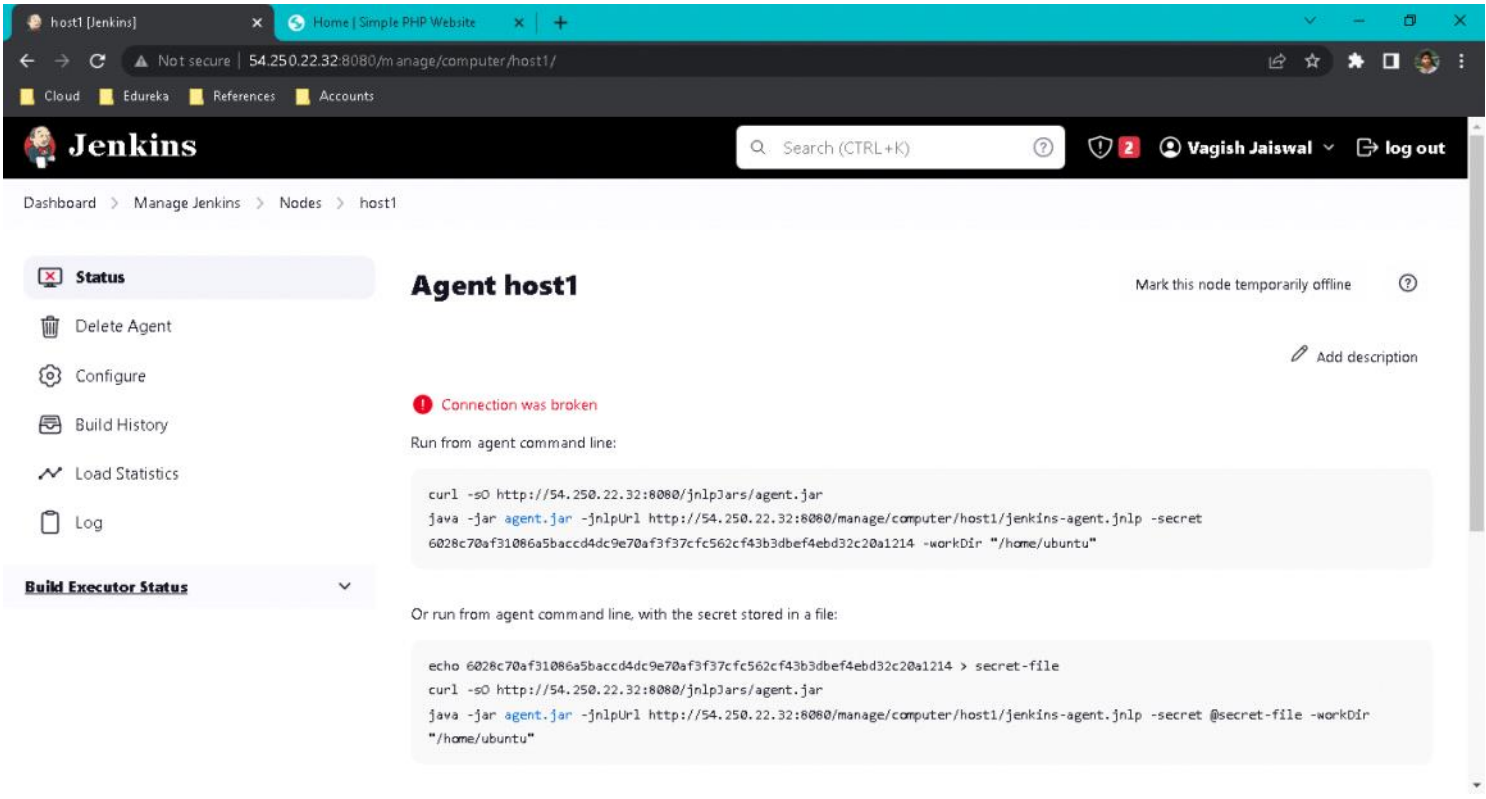

ACM - AWS
Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
X server Exit
Quick connect...
6. ACM - AWS
7. host1
/home/ubuntu/
Name
..
.ansible
.cache
.ssh
.bash_history
.bash_logout
.bashrc
.profile
.sudo_as_admin_successful
.Xauthority
Remote monitoring
Follow terminal folder
root@ip-172-31-1-214:/etc/ansible# ansible --version
ansible [core 2.14.3]
config file = /etc/ansible/ansible.cfg
configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
ansible python module location = /usr/lib/python3/dist-packages/ansible
ansible collection location = /root/.ansible/collections:/usr/share/ansible/collections
executable location = /usr/bin/ansible
python version = 3.10.6 (main, Nov 14 2022, 16:10:14) [GCC 11.3.0] (/usr/bin/python3)
jinja version = 3.0.3
libyaml = True
root@ip-172-31-1-214:/etc/ansible# ansible -m ping host1
host1 | SUCCESS => {
"ansible_facts": {
"discovered_interpreter_python": "/usr/bin/python3"
},
"changed": false,
"ping": "pong"
}
root@ip-172-31-1-214:/etc/ansible#
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net


ACM - AWS
Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
X server Exit
Quick connect...
6. ACM - AWS
7. host1
/home/ubuntu/
Name
..
.ansible
.cache
.ssh
.bash_history
.bash_logout
.bashrc
.profile
.sudo_as_admin_successful
.Xauthority
Remote monitoring
Follow terminal folder
"ping": "pong"
}
root@ip-172-31-1-214:/etc/ansible# sudo apt-get update -y
Hit:1 http://ap-northeast-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://ap-northeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://ap-northeast-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [107 kB]
Get:4 http://ap-northeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [948 kB]
Get:5 http://ap-northeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [884 kB]
Get:6 http://ap-northeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [18.0 kB]
Get:7 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Hit:8 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu jammy InRelease
Fetched 2186 kB in 1s (1584 kB/s)
Reading package lists... Done
root@ip-172-31-1-214:/etc/ansible# curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo tee \
/usr/share/keyrings/jenkins-keyring.asc > /dev/null
root@ip-172-31-1-214:/etc/ansible# 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
root@ip-172-31-1-214:/etc/ansible# sudo apt-get update
Hit:1 http://ap-northeast-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://ap-northeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://ap-northeast-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Ign:4 https://pkg.jenkins.io/debian-stable binary/ InRelease
Get:5 https://pkg.jenkins.io/debian-stable binary/ Release [2044 B]
Get:6 https://pkg.jenkins.io/debian-stable binary/ Release.gpg [833 B]
Hit:7 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:8 https://pkg.jenkins.io/debian-stable binary/ Packages [24.5 kB]
Hit:9 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu jammy InRelease
Fetched 27.4 kB in 1s (21.6 kB/s)
Reading package lists... Done
root@ip-172-31-1-214:/etc/ansible# sudo apt-get install jenkins
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
net-tools
The following NEW packages will be installed:
jenkins net-tools
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

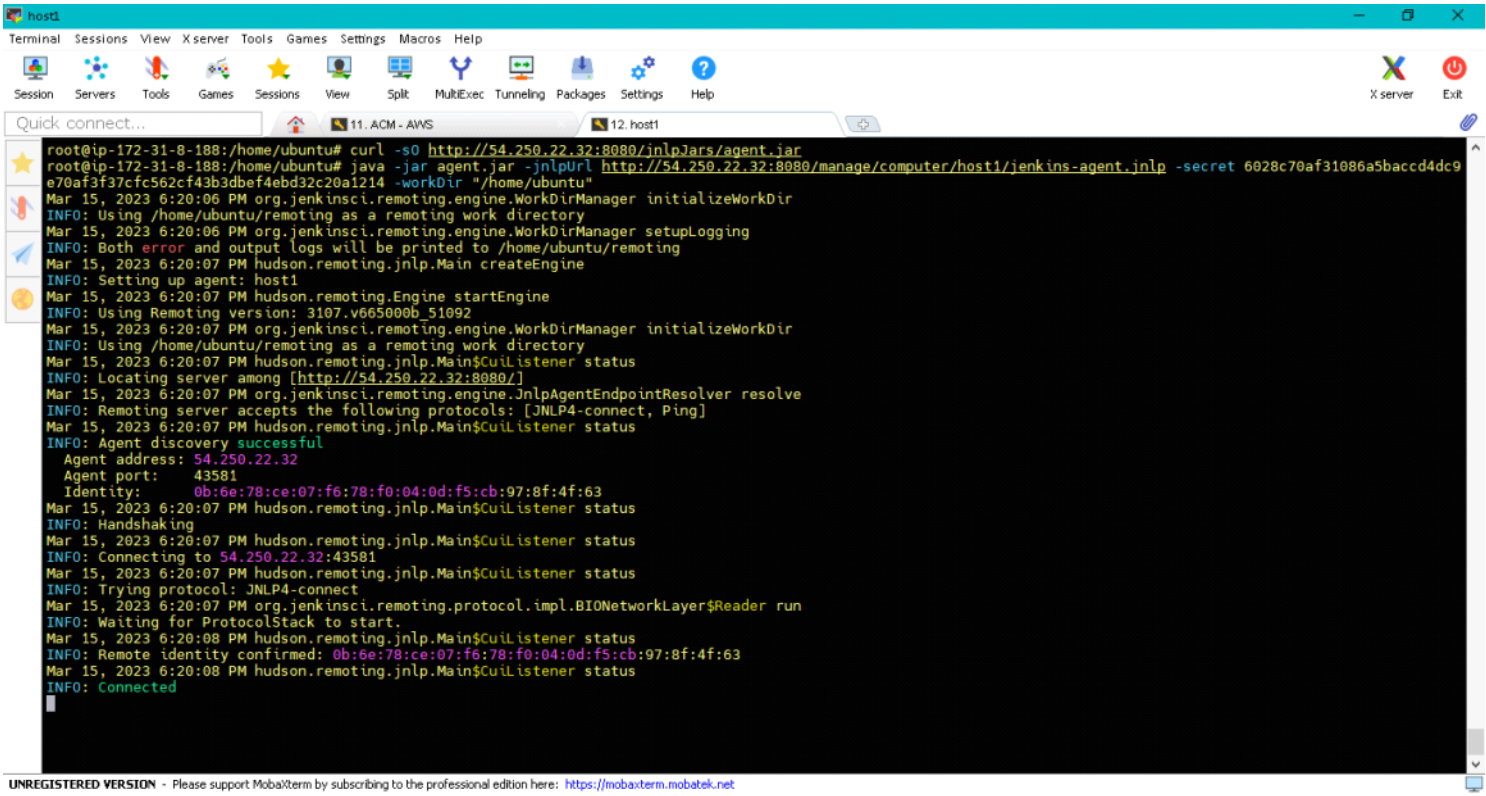




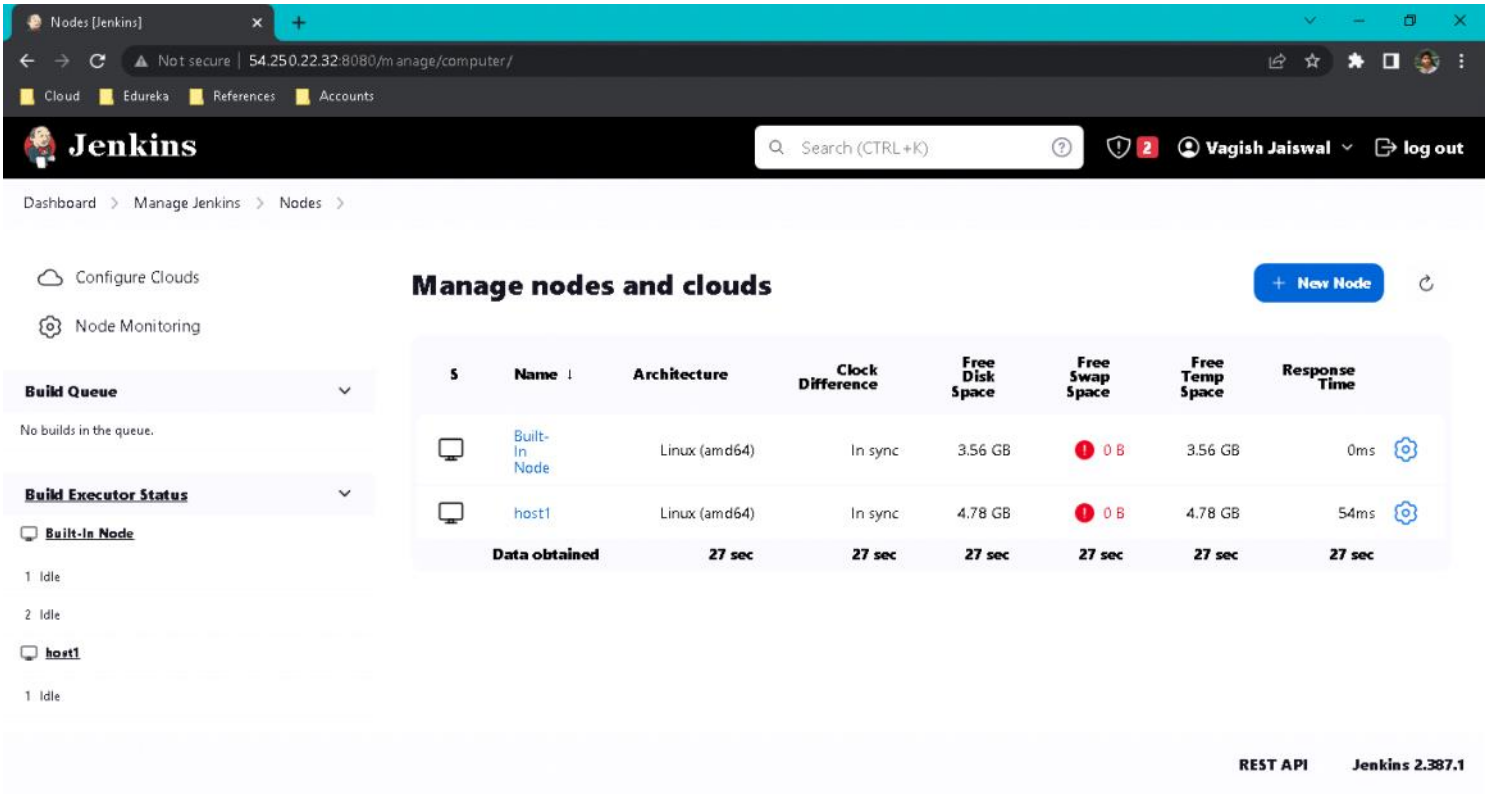
- Started Jenkins on Master VM: ACM and Created Agent Node to connect with Slave Server: host1



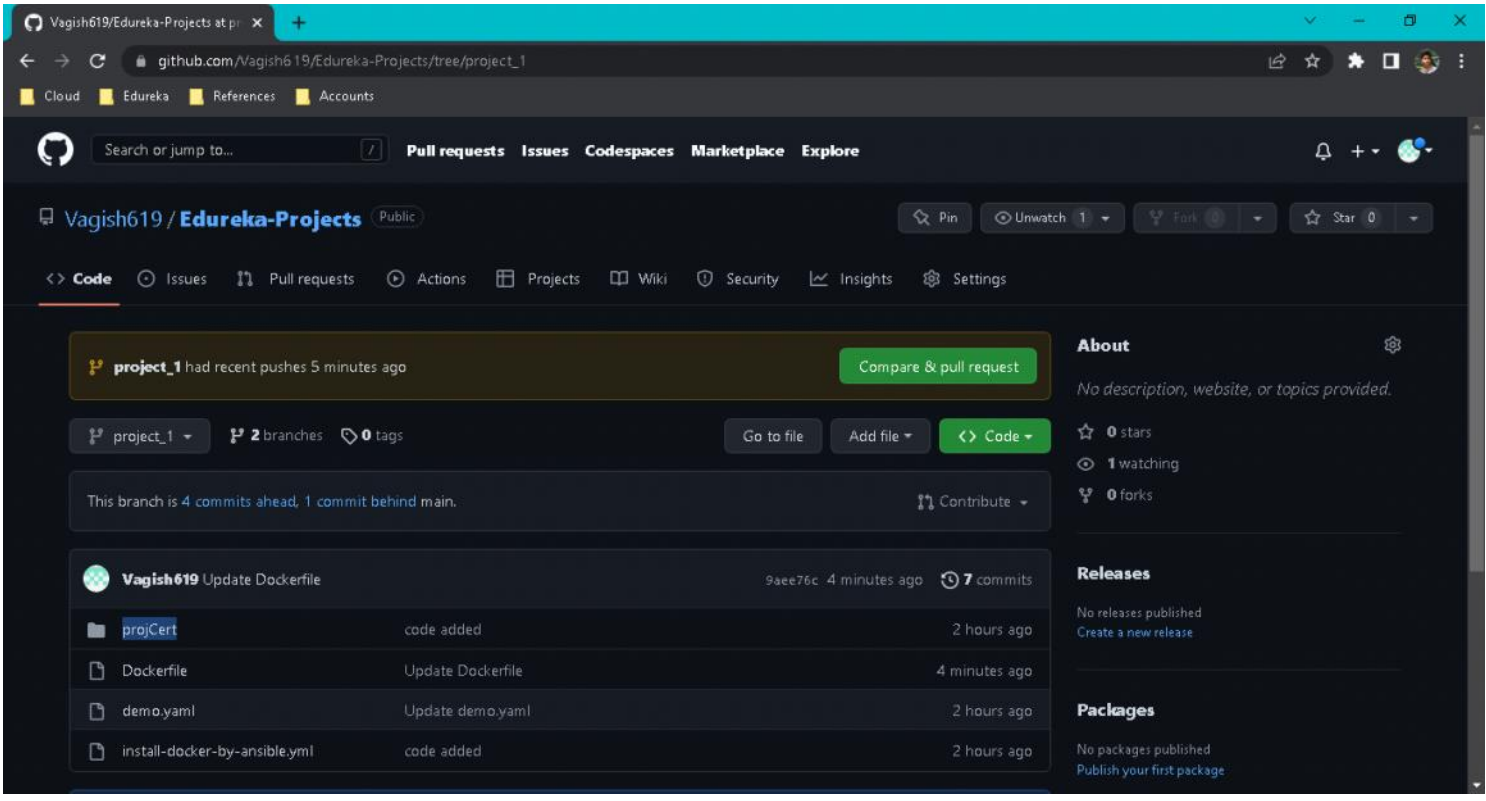
- On Slave-Server, ran Master VM Jenkins connection commands



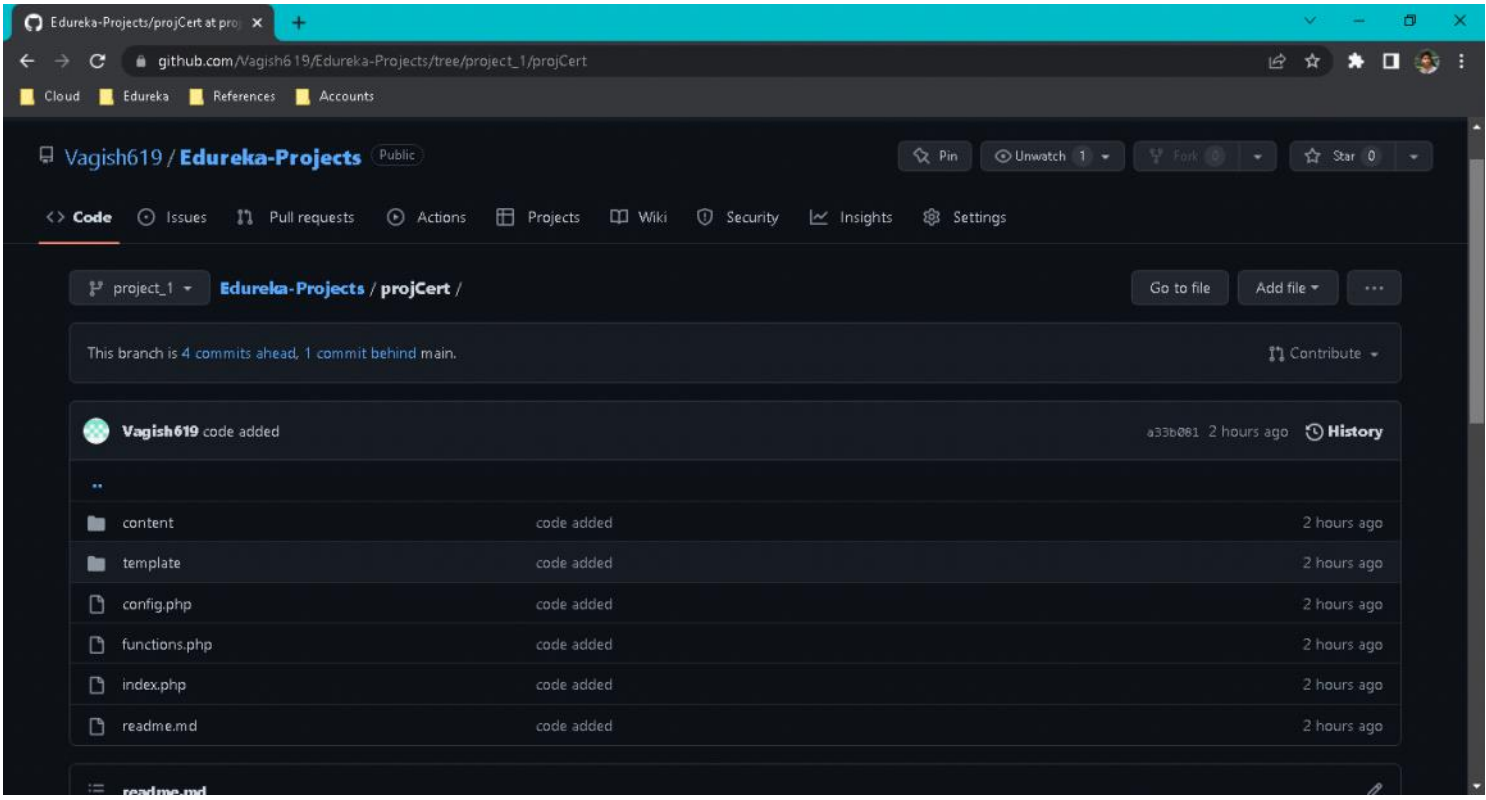
- Slave-server:host1 connected as Agent on Master VM Jenkins



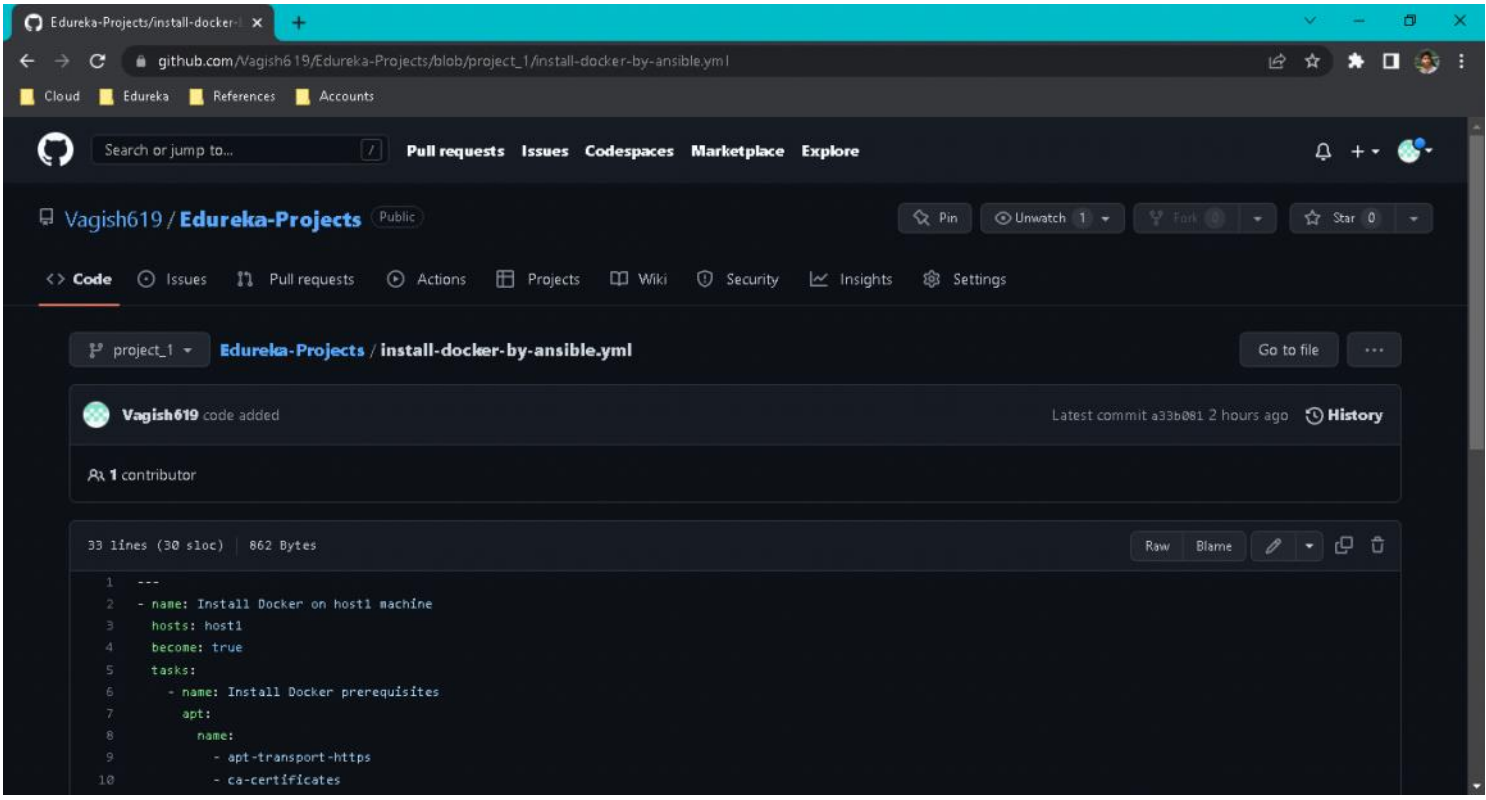
- GIT Repo Screenshot: Fresh Branch Created



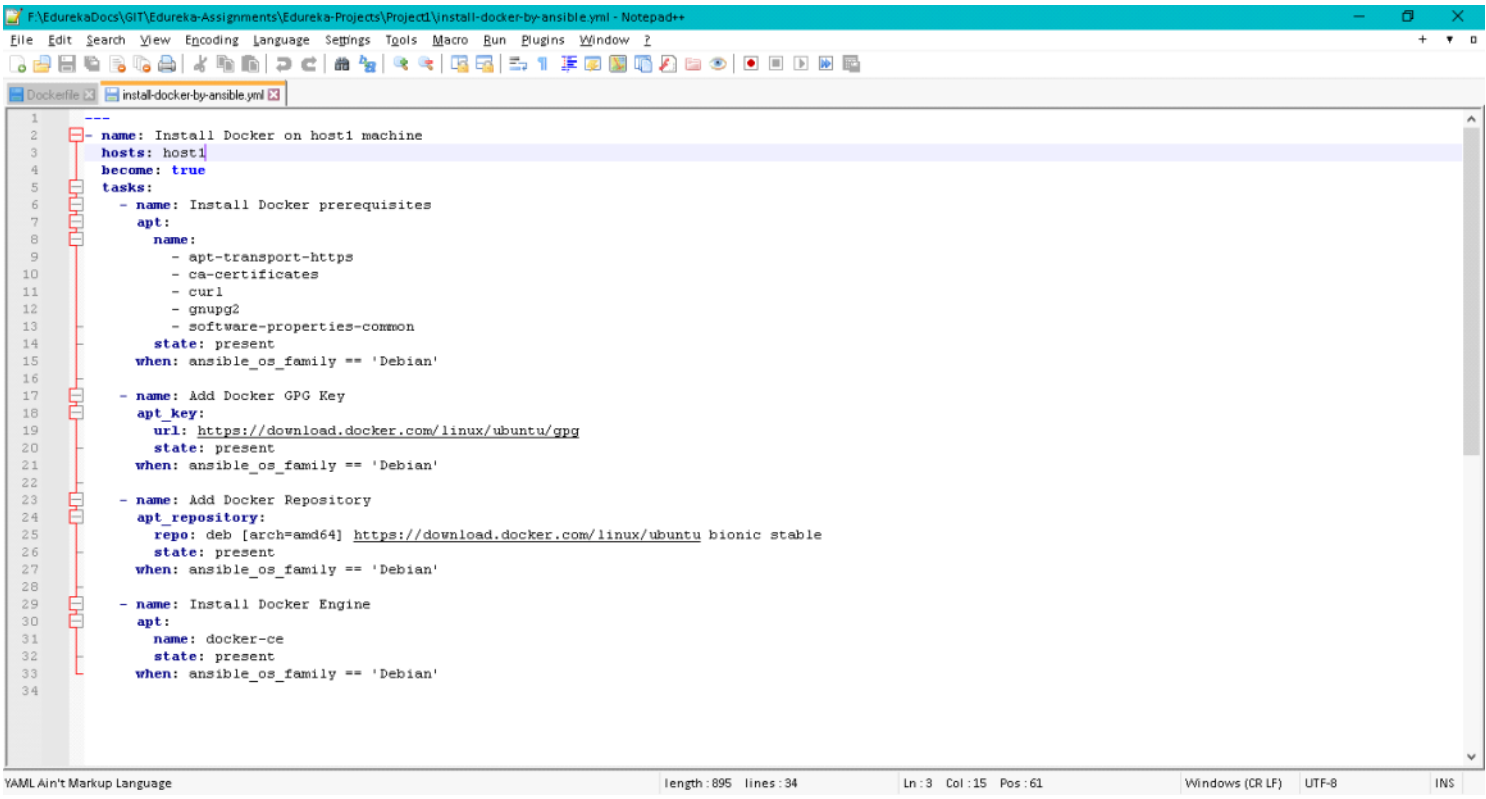
- GIT Repo Screenshot: Shared Code uploaded



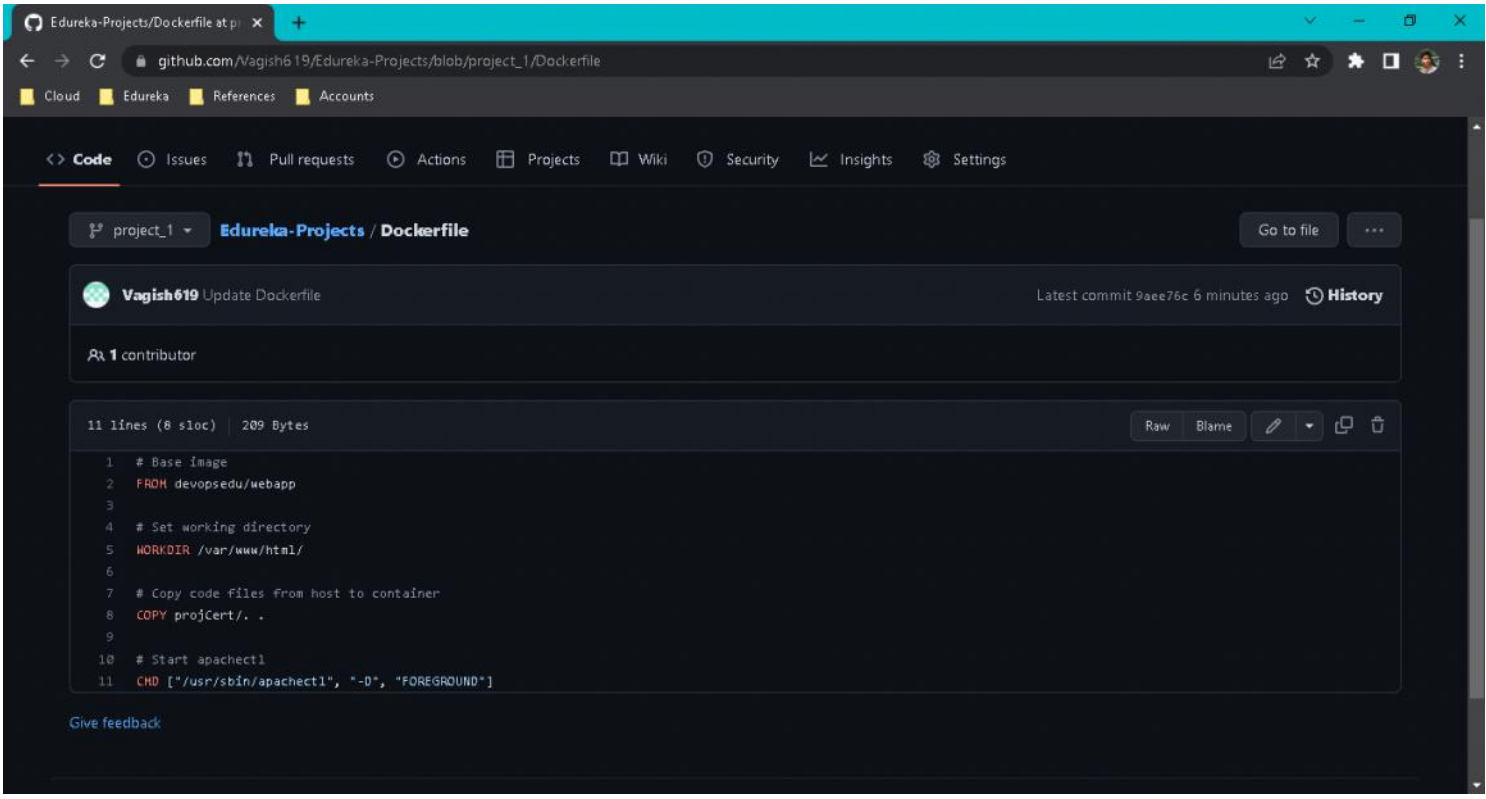
- GIT Repo Screenshot: Ansible-Playbook created to install Docker on Slave Server: host1 from Master VM : ACM
    - File-Name: install-docker-by-ansible.yml



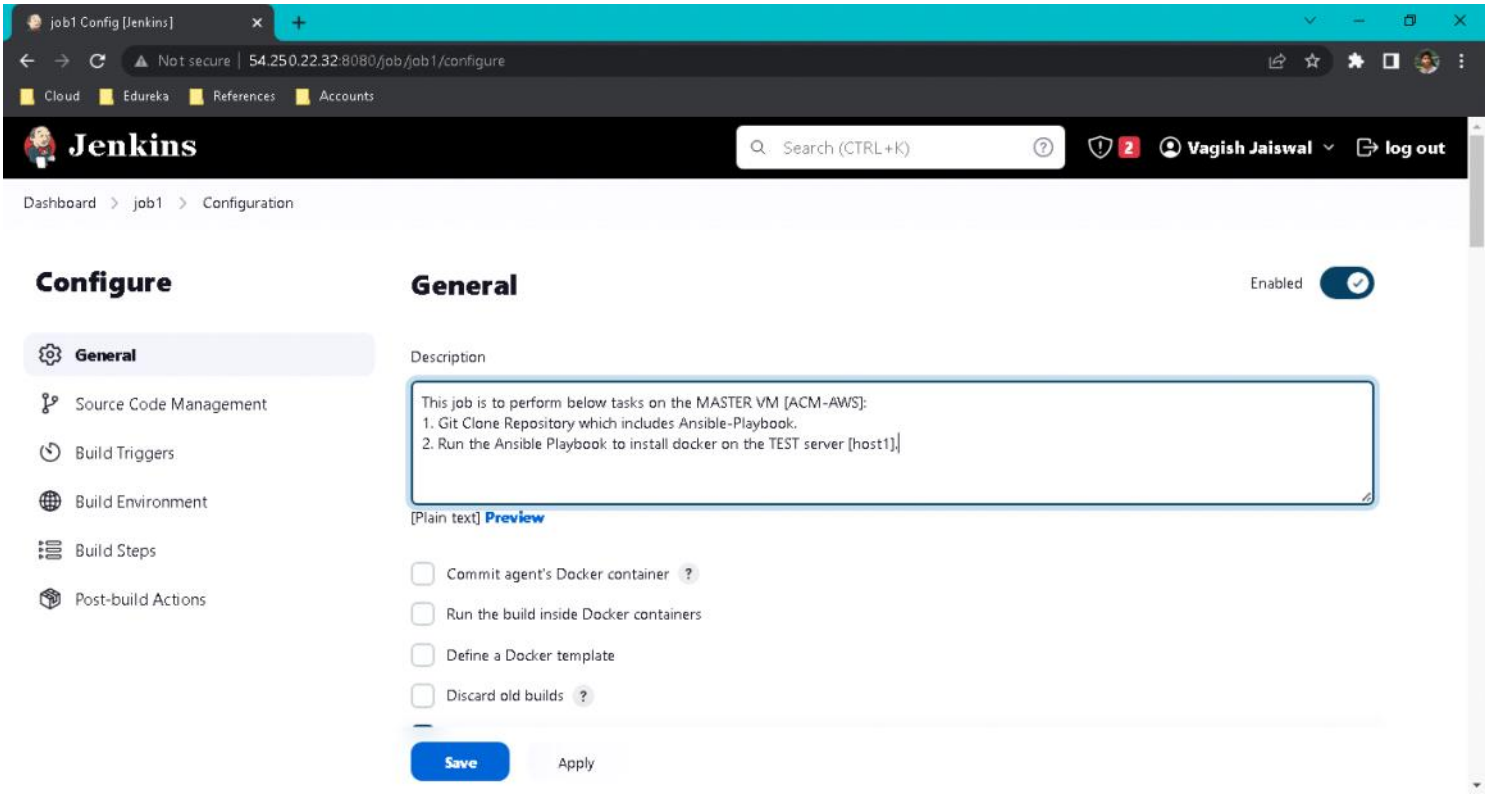
- Ansible-Playbook: install-docker-by-ansible.yml -- To be ran on Master VM [ACM] by Jenkins

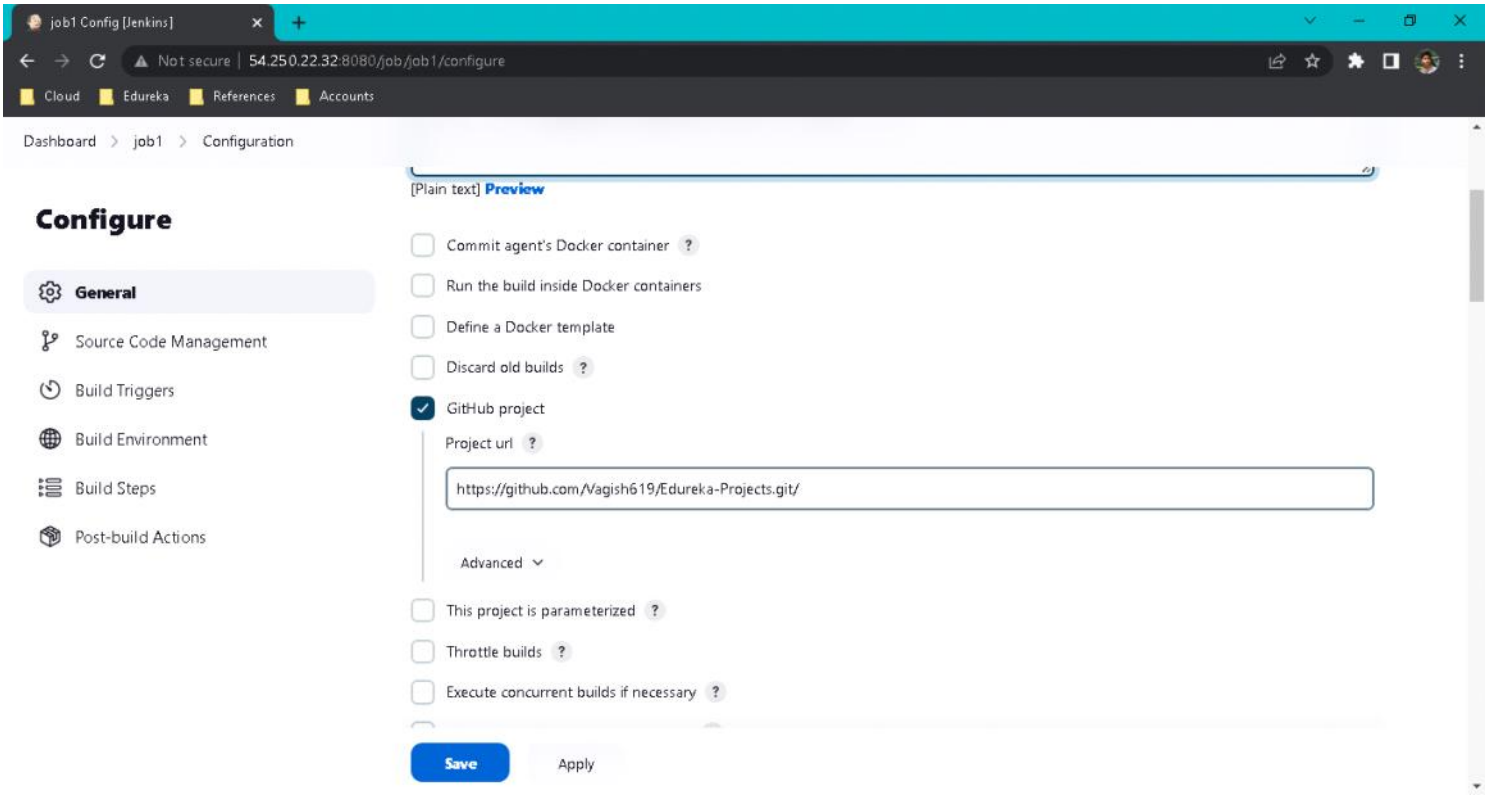


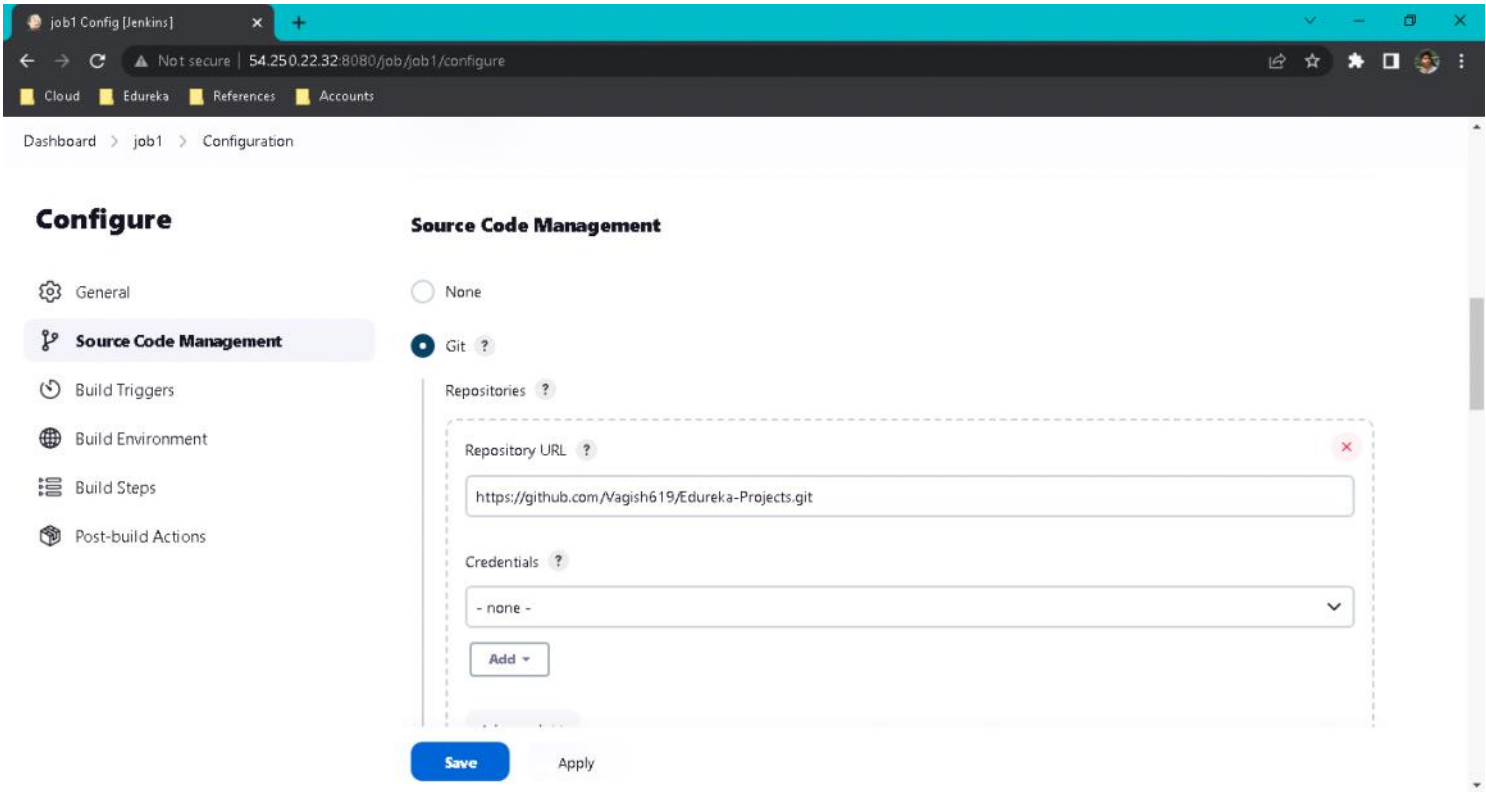
- Dockerfile: To create container (as independent Test-Server) image on Slave-server [host1]

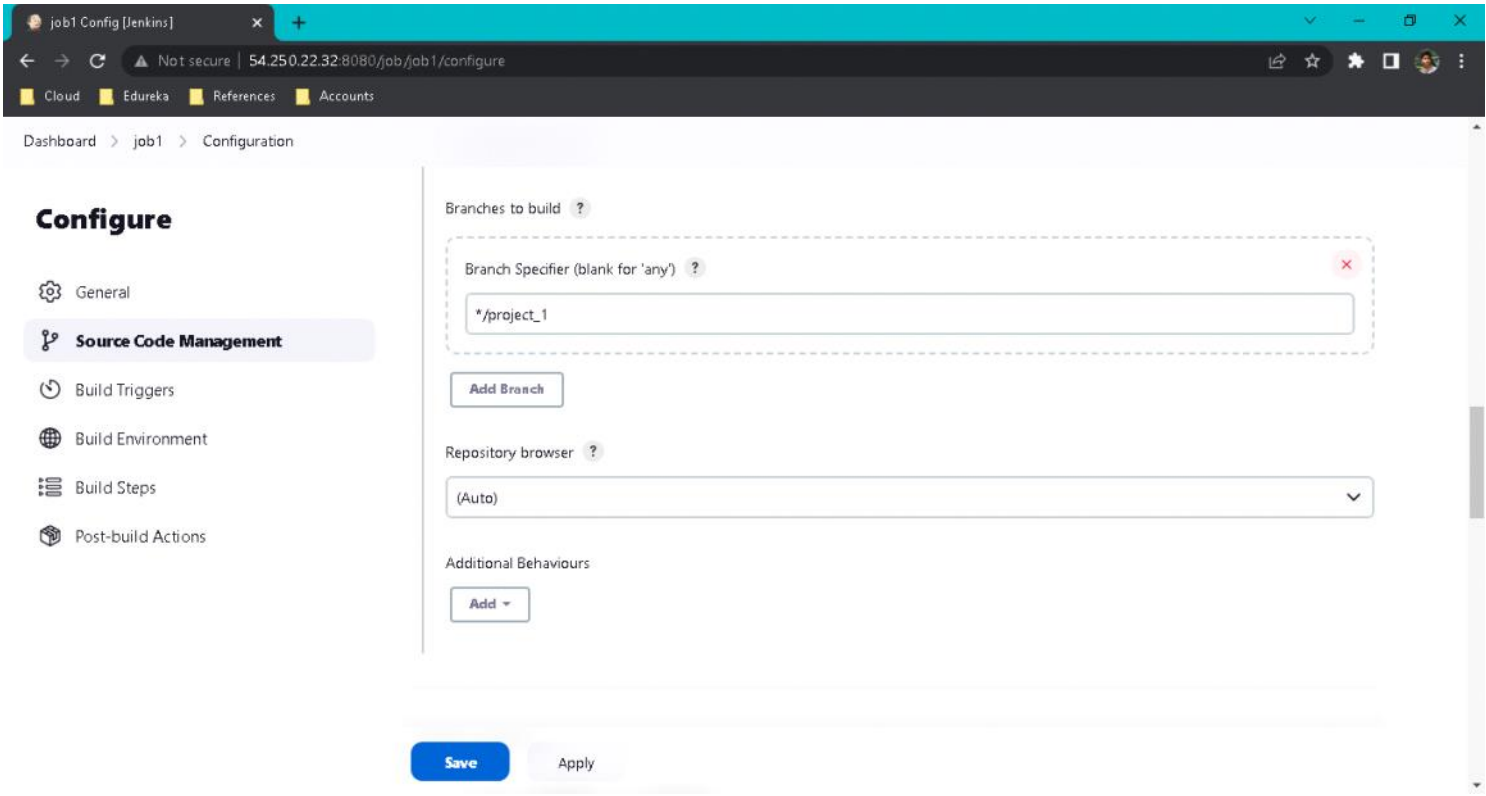


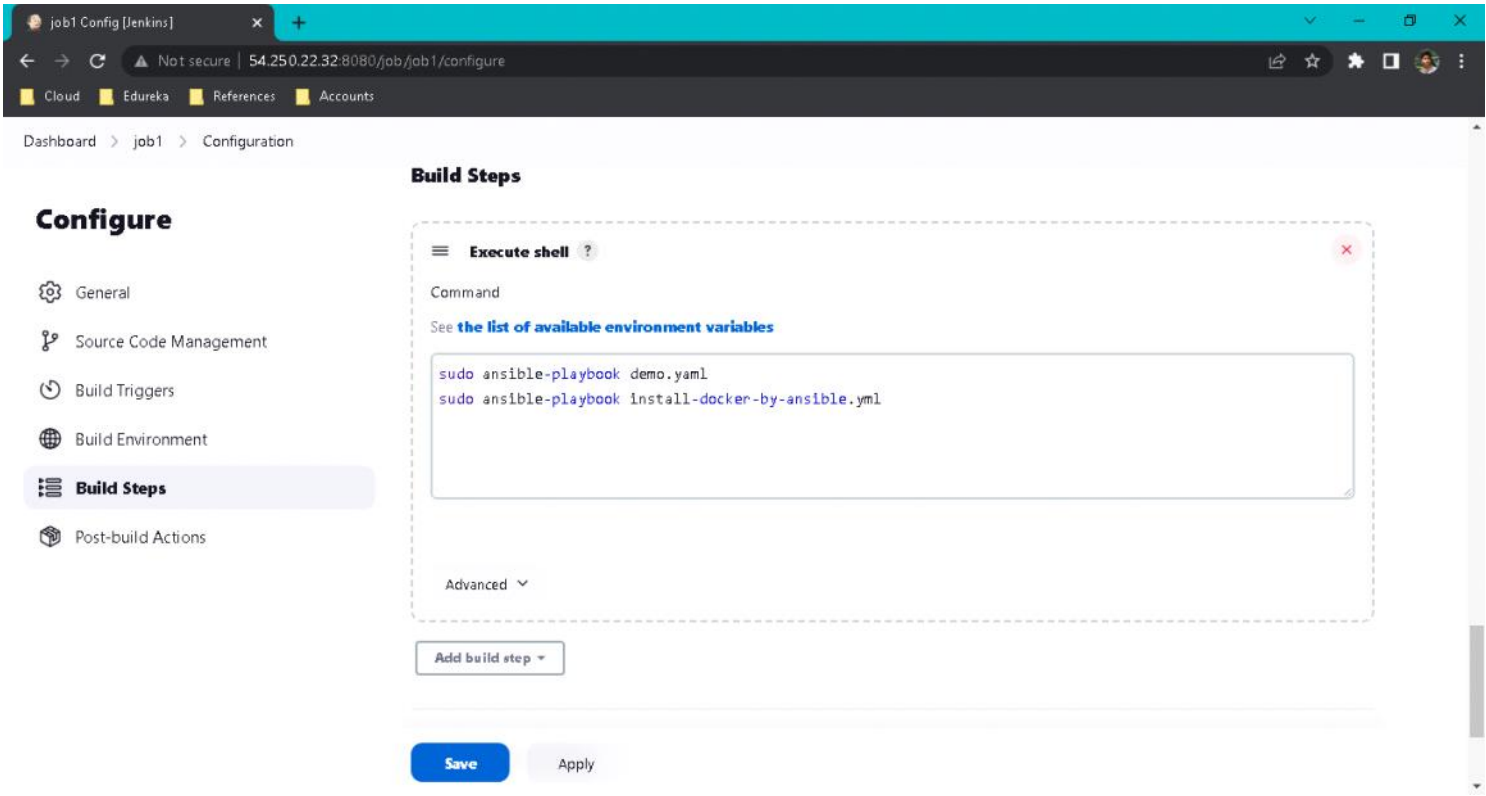
- Jenkins JOB1 : To install Docker on Slave-Server[host1], using Ansible-Playbook downloaded on Master-VM



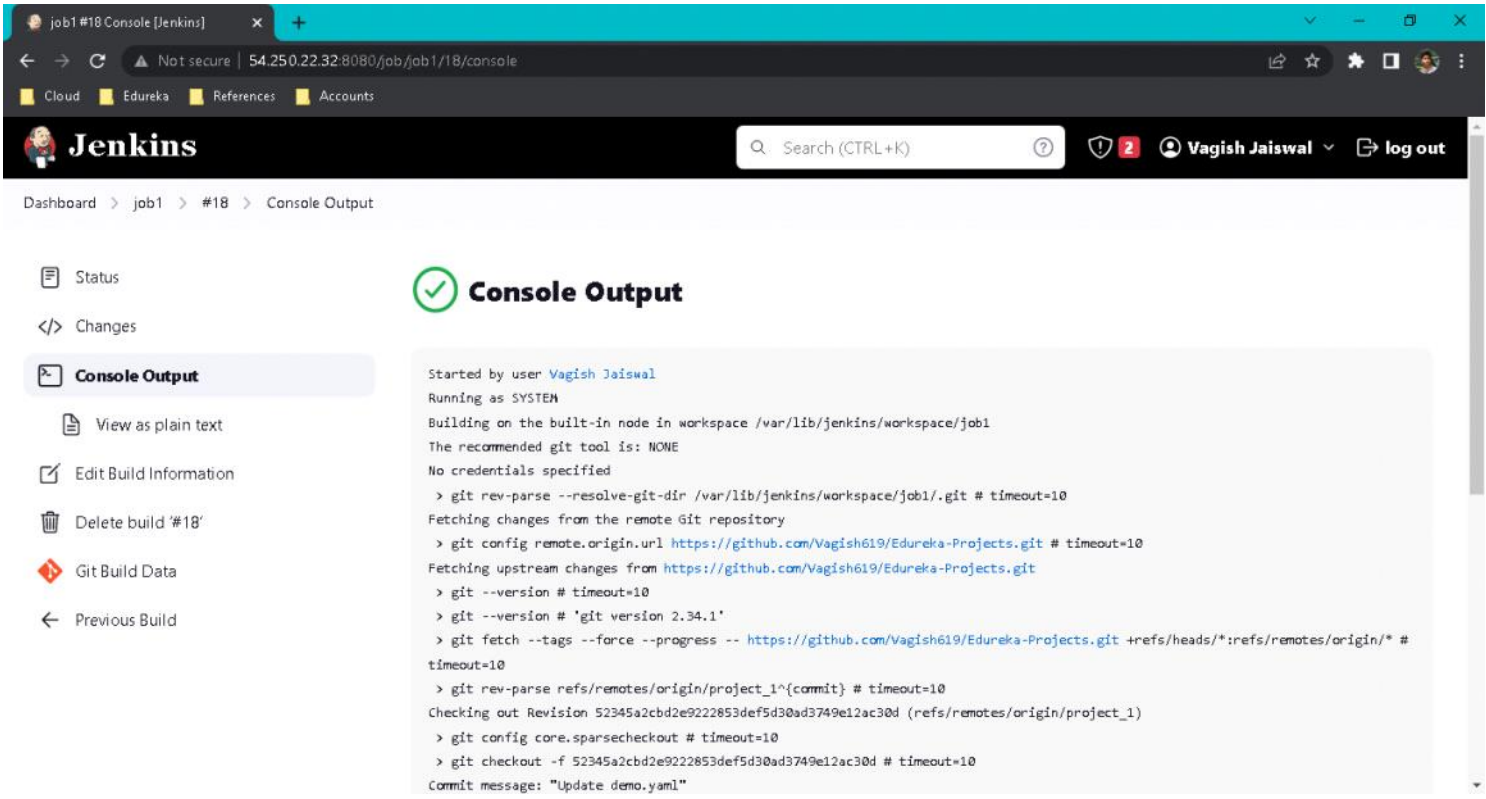

job1 Config [Jenkins]
Not secure | 54.250.22.32:8080/job/job1/configure
Cloud
Edureka
References
Accounts
Dashboard > job1 > Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
[Plain text] Preview
Commit agent's Docker container
Run the build inside Docker containers
Define a Docker template
Discard old builds
GitHub project
Project url
https://github.com/Vagish619/Edureka-Projects.git/
Advanced
This project is parameterized
Throttle builds
Execute concurrent builds if necessary
Save
Apply

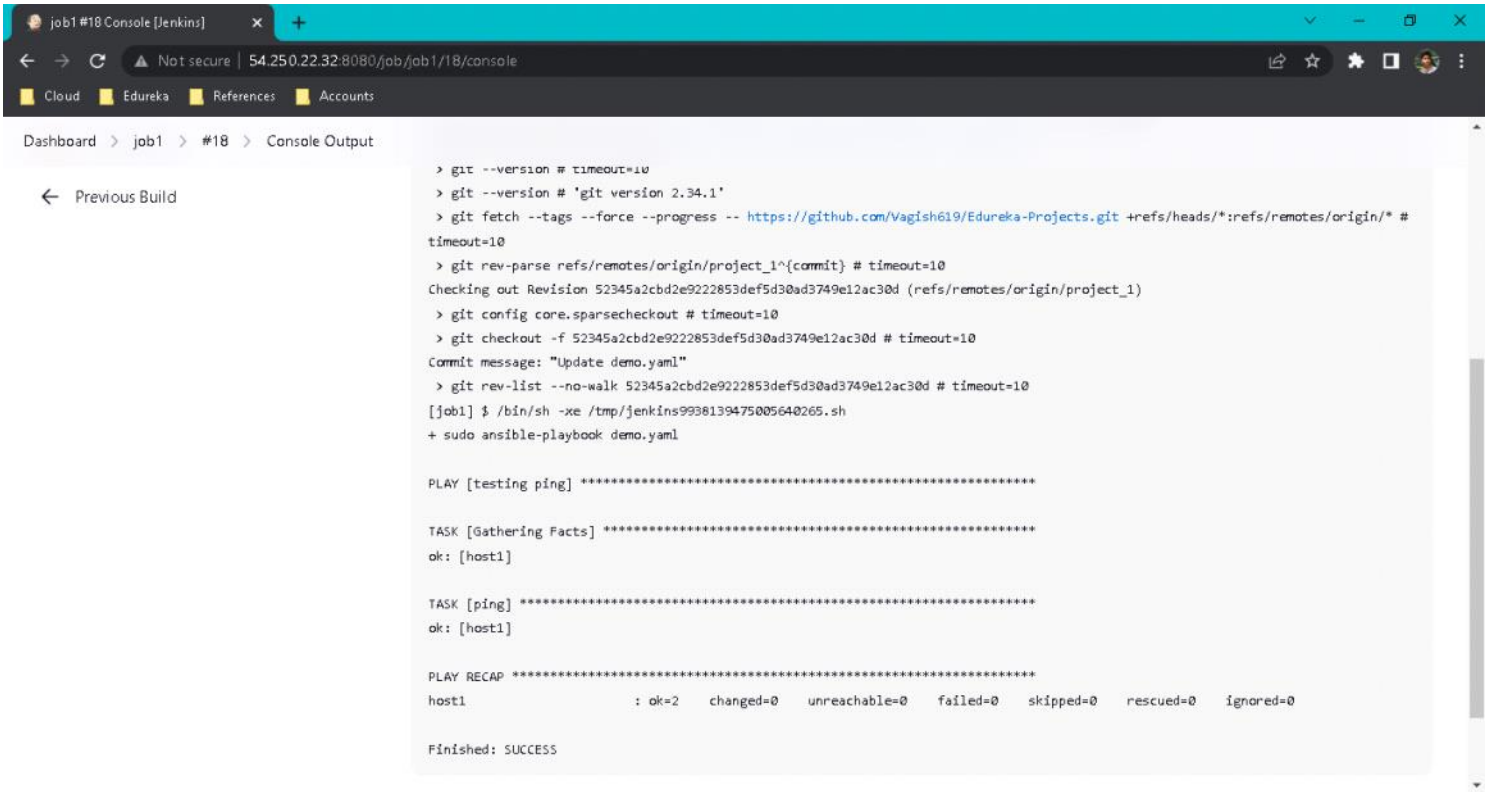

job1 Config [Jenkins]
Not secure | 54.250.22.32:8080/job/job1/configure
Cloud
Edureka
References
Accounts
Dashboard > job1 > Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
Source Code Management
None
Git
Repositories
Repository URL
https://github.com/Vagish619/Edureka-Projects.git
Credentials
- none -
Add
Save
Apply

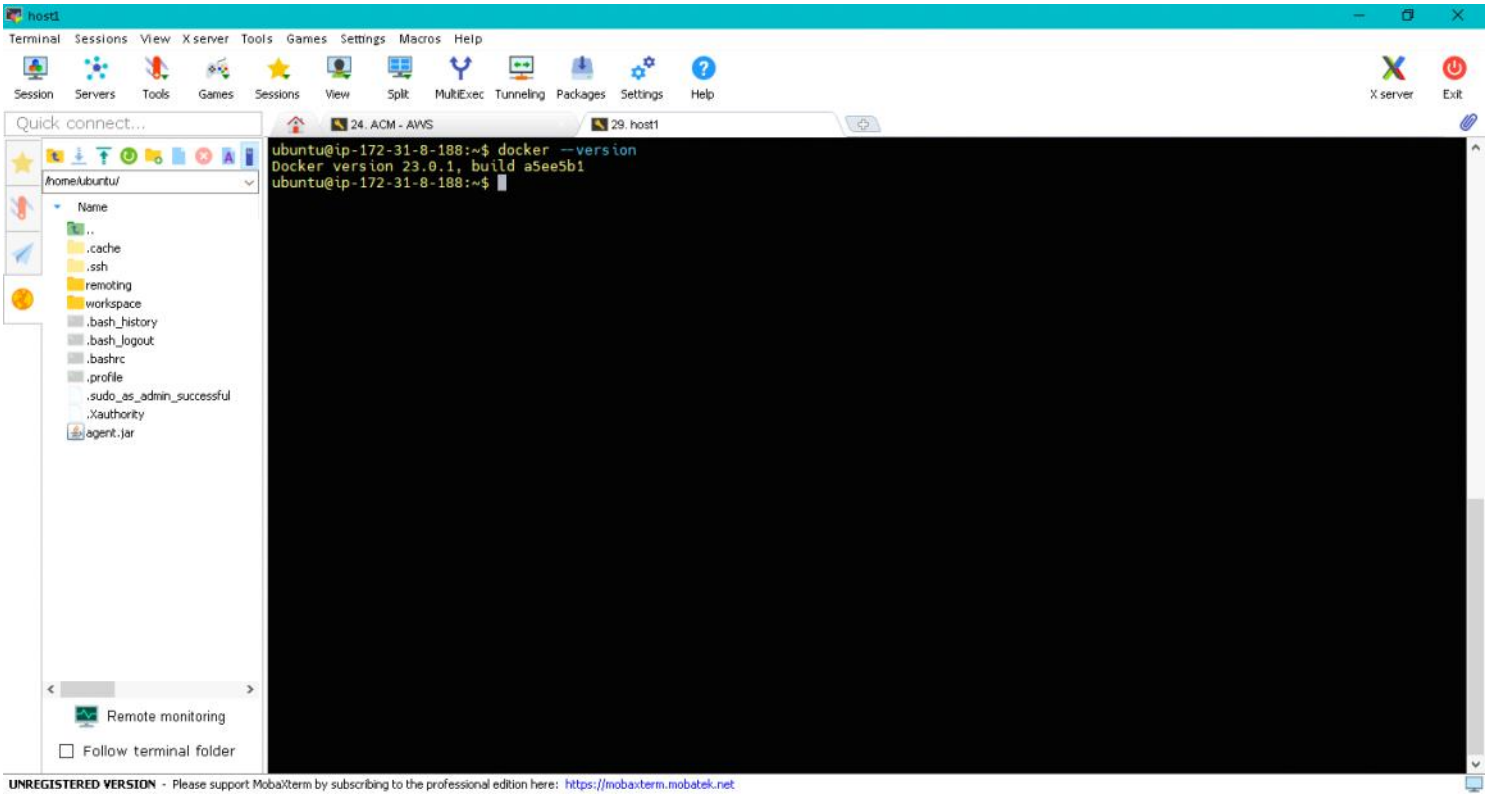




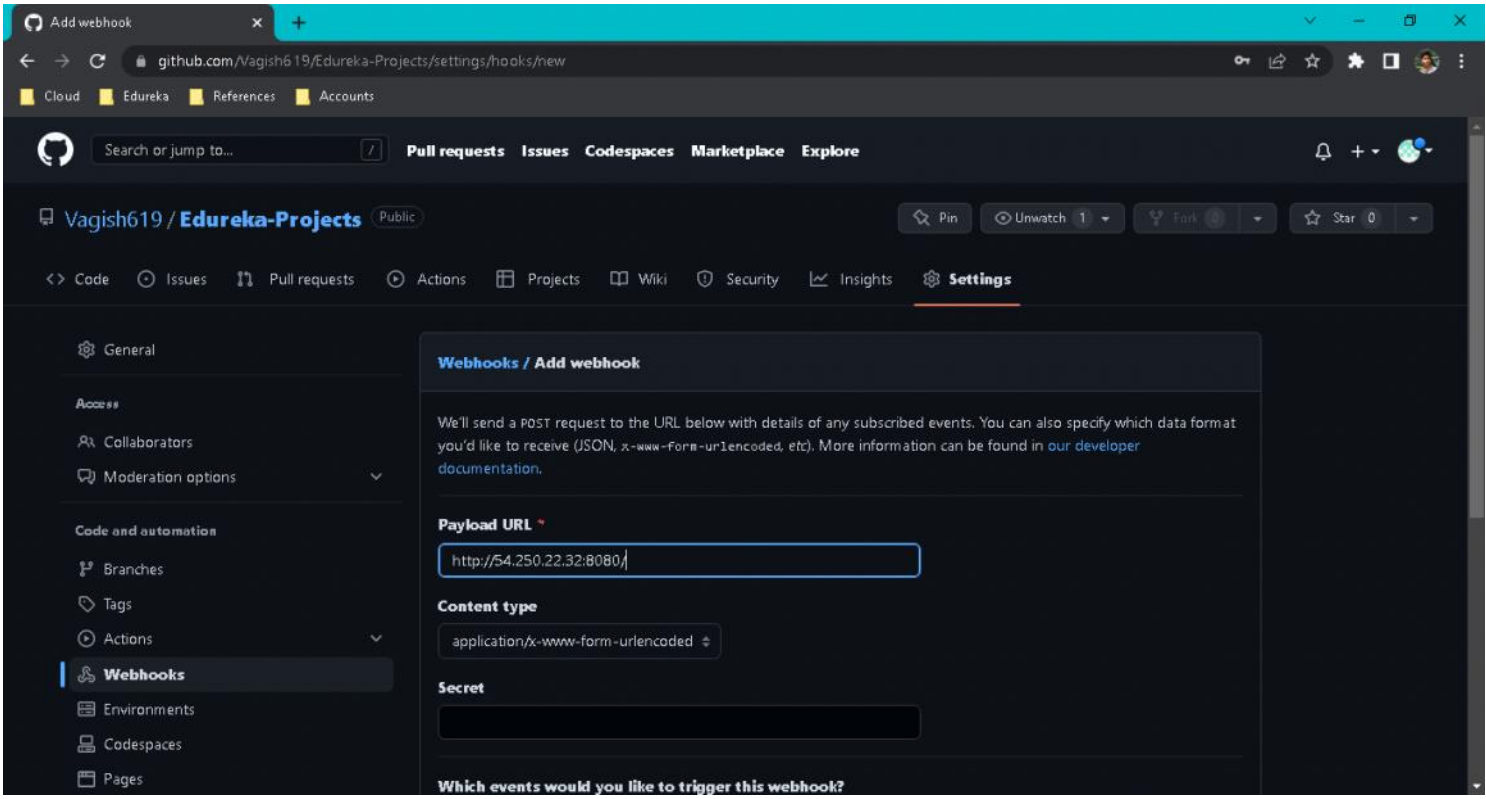
- Jenkins JOB1 : Console Output

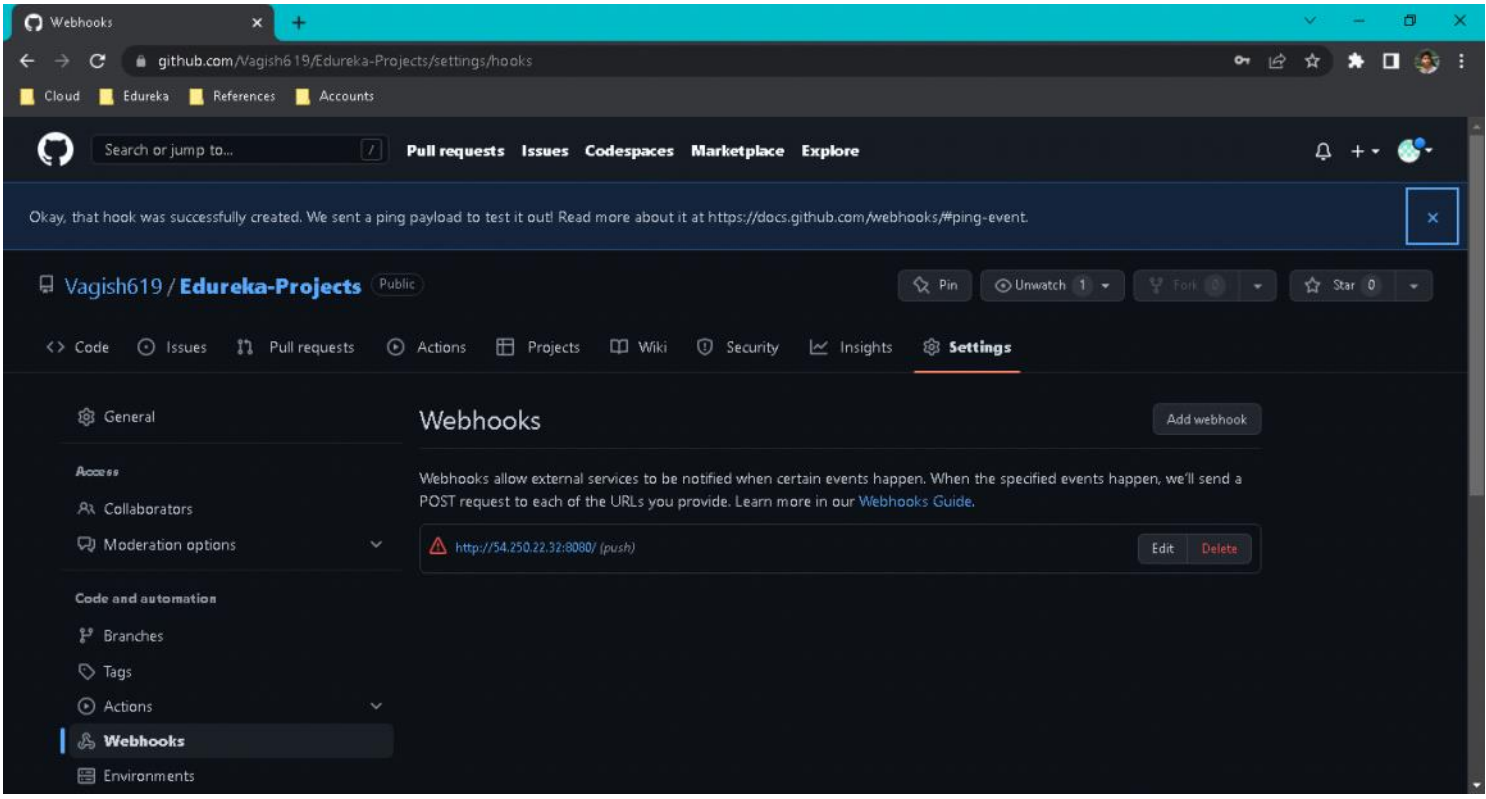

job1 #18 Console [Jenkins]
Not secure | 54.250.22.32:8080/job/job1/18/console
Cloud
Edureka
References
Accounts
Jenkins
Search (CTRL+K)
2
Vagish Jaiswal
log out
Dashboard > job1 > #18 > Console Output
Status
Changes
Console Output
View as plain text
Edit Build Information
Delete build '#18'
Git Build Data
Previous Build
Console Output
Started by user Vagish Jaiswal
Running as SYSTEM
Building on the built-in node in workspace /var/lib/jenkins/workspace/job1
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/job1/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/Vagish619/Edureka-Projects.git # timeout=10
Fetching upstream changes from https://github.com/Vagish619/Edureka-Projects.git
> git --version # timeout=10
> git --version # 'git version 2.34.1'
> git fetch --tags --force --progress -- https://github.com/Vagish619/Edureka-Projects.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/project_1^{commit} # timeout=10
Checking out Revision 52345a2cbd2e9222853def5d30ad3749e12ac30d (refs/remotes/origin/project_1)
> git config core.sparsecheckout # timeout=10
> git checkout -f 52345a2cbd2e9222853def5d30ad3749e12ac30d # timeout=10
Commit message: "Update demo.yaml"


job1 #18 Console [Jenkins]
Not secure | 54.250.22.32:8080/job/job1/18/console
Cloud
Edureka
References
Accounts
Dashboard > job1 > #18 > Console Output
Previous Build
> git --version # timeout=10
> git --version # 'git version 2.34.1'
> git fetch --tags --force --progress -- https://github.com/Vagish619/Edureka-Projects.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/project_1^{commit} # timeout=10
Checking out Revision 52345a2cbd2e9222853def5d30ad3749e12ac30d (refs/remotes/origin/project_1)
> git config core.sparsecheckout # timeout=10
> git checkout -f 52345a2cbd2e9222853def5d30ad3749e12ac30d # timeout=10
Commit message: "Update demo.yaml"
> git rev-list --no-walk 52345a2cbd2e9222853def5d30ad3749e12ac30d # timeout=10
[job1] $ /bin/sh -xe /tmp/jenkins9938139475005640265.sh
+ sudo ansible-playbook demo.yaml
PLAY [testing ping] **
TASK [Gathering Facts] ***
ok: [host1]
TASK [ping] **
ok: [host1]
PLAY RECAP ***
host1 : ok=2 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
Finished: SUCCESS

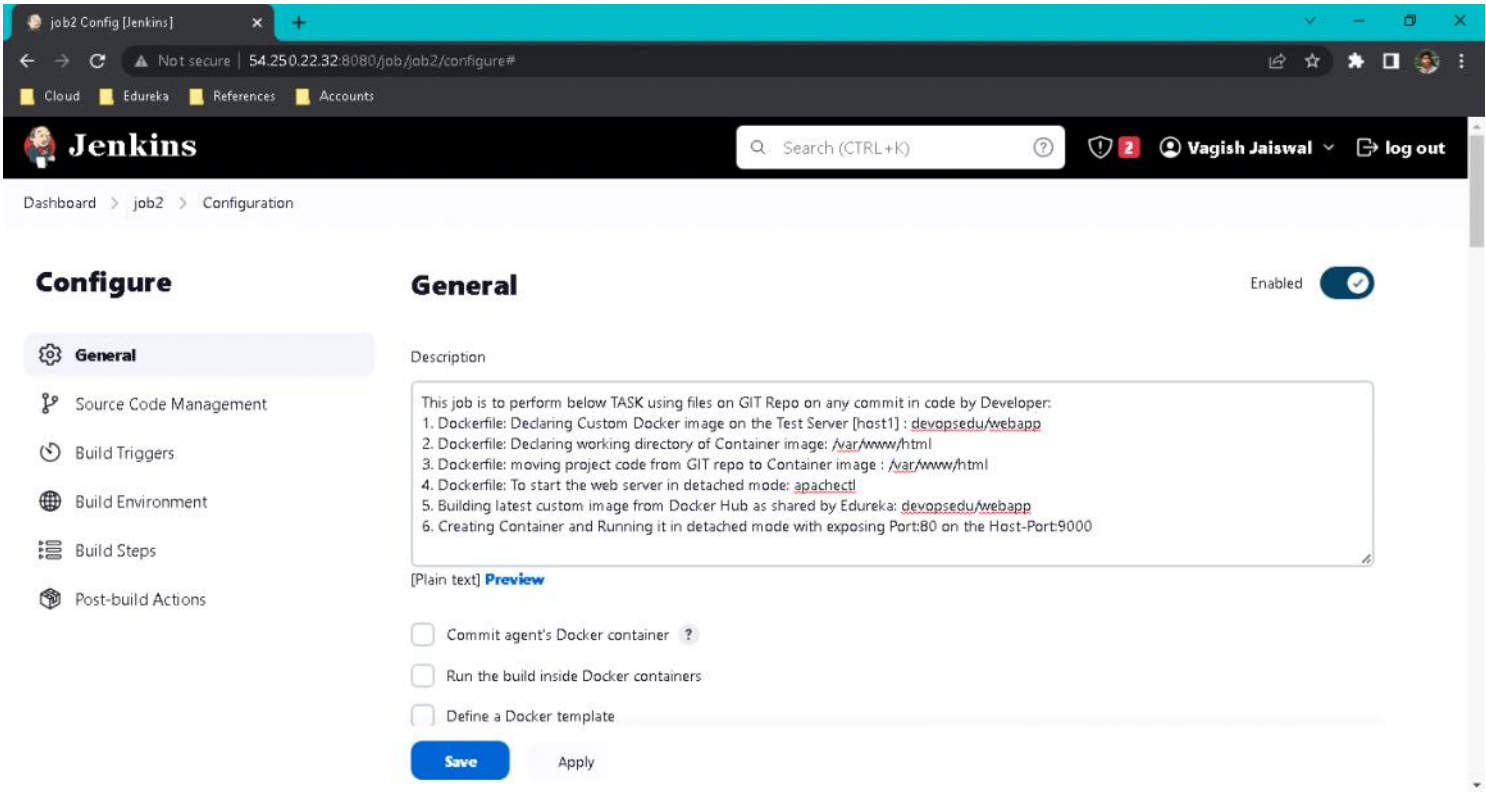


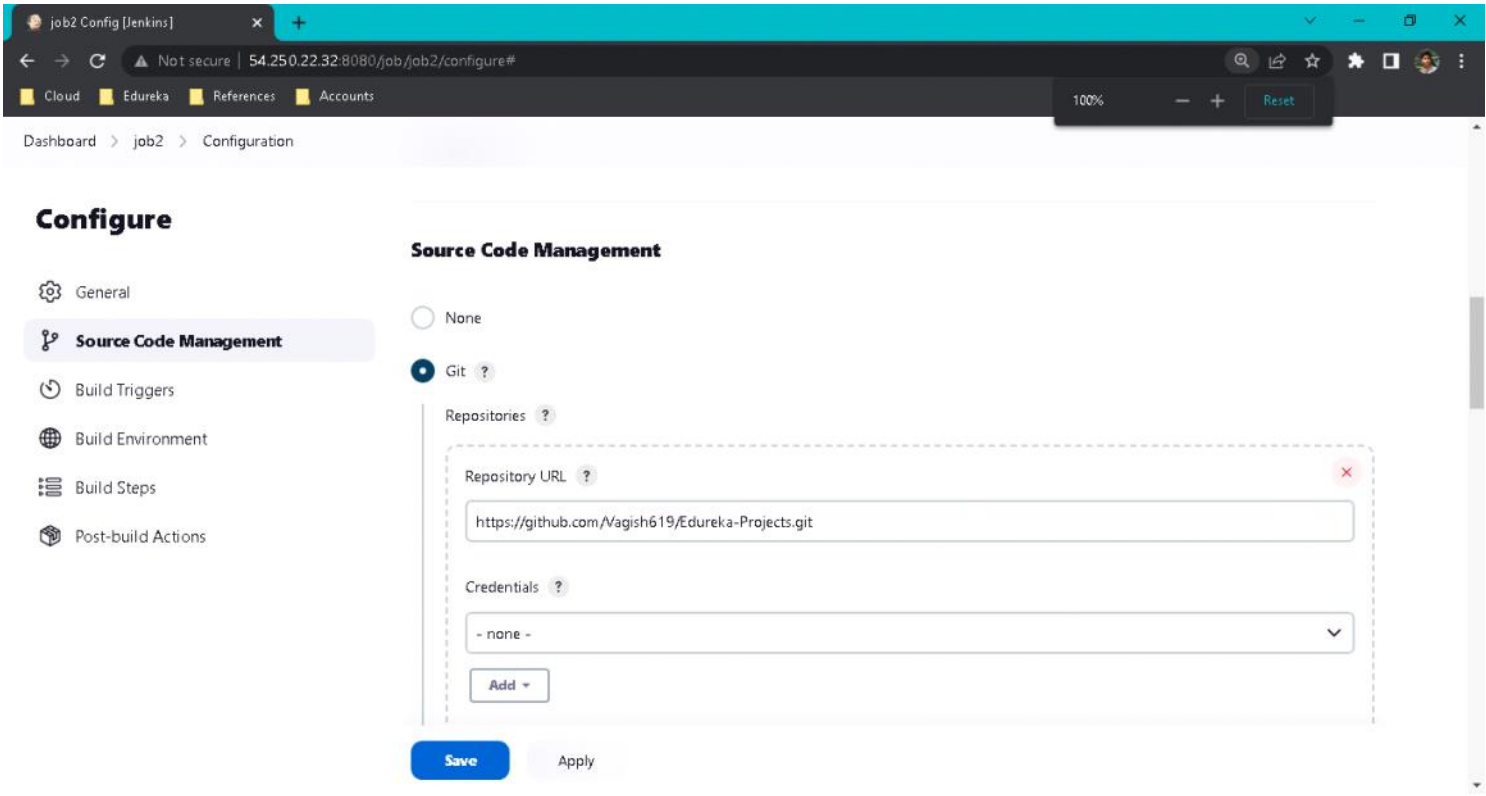
- GIT Repo Screenshot: Created Webhook for Job2

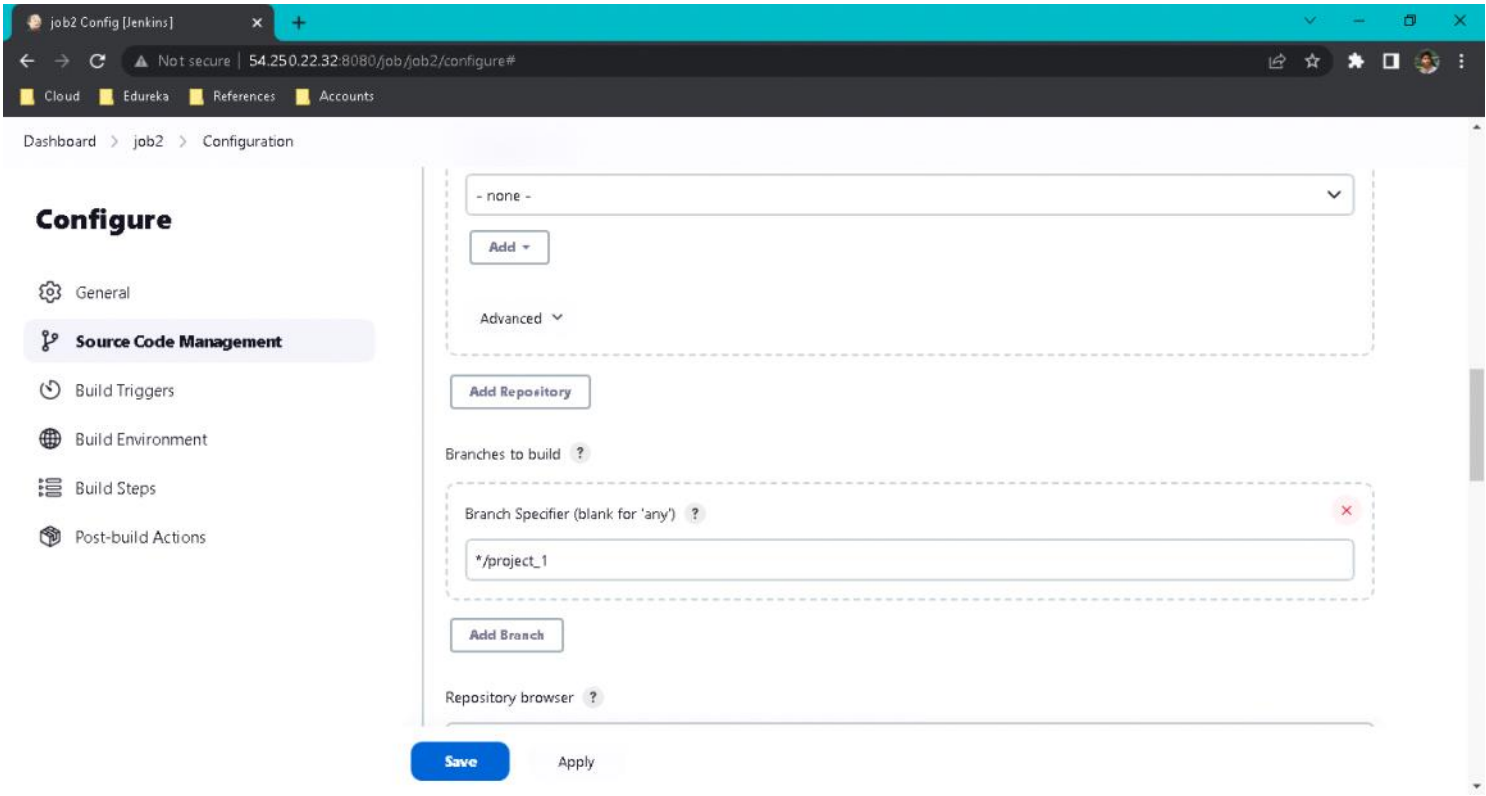


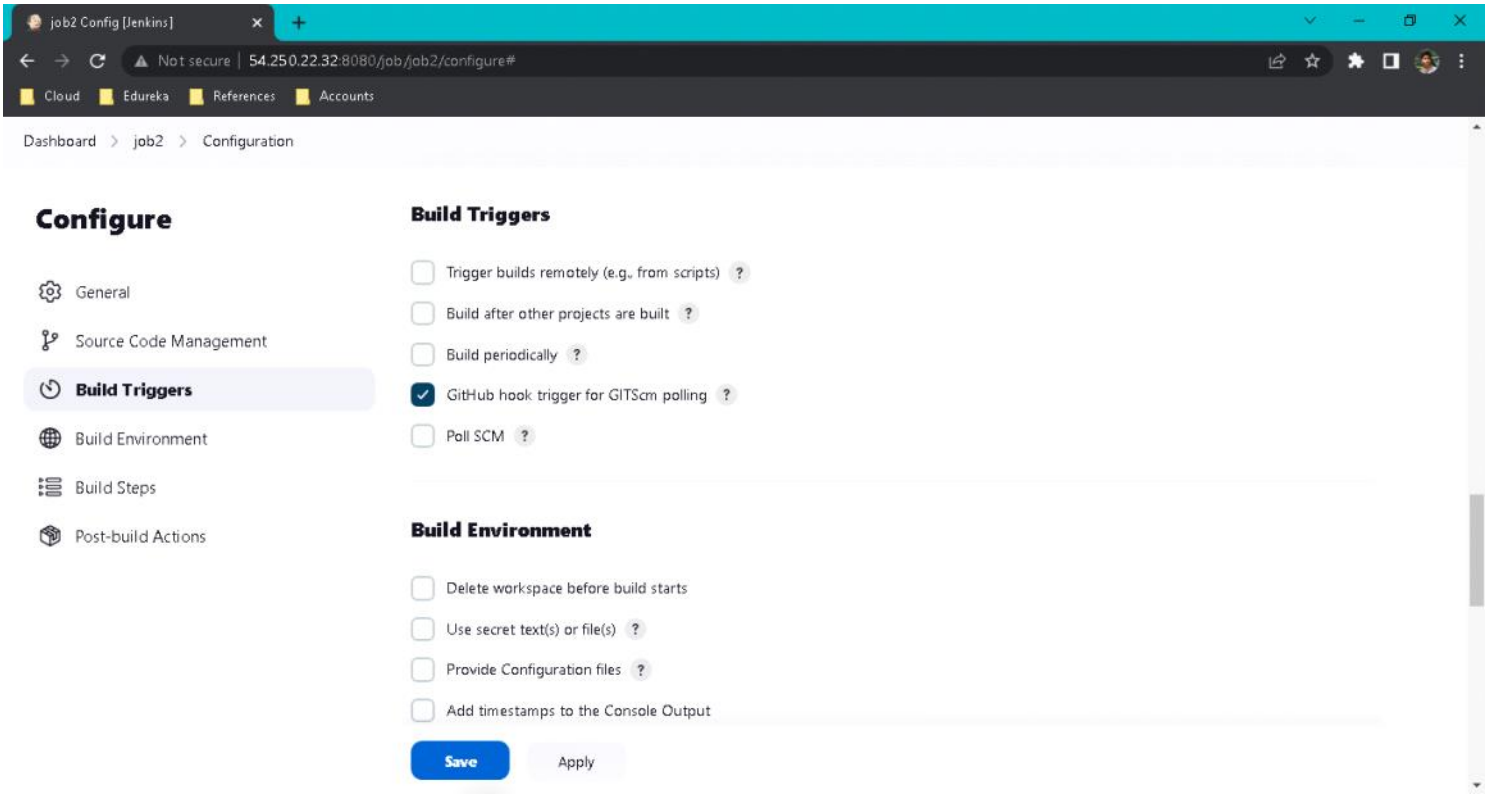


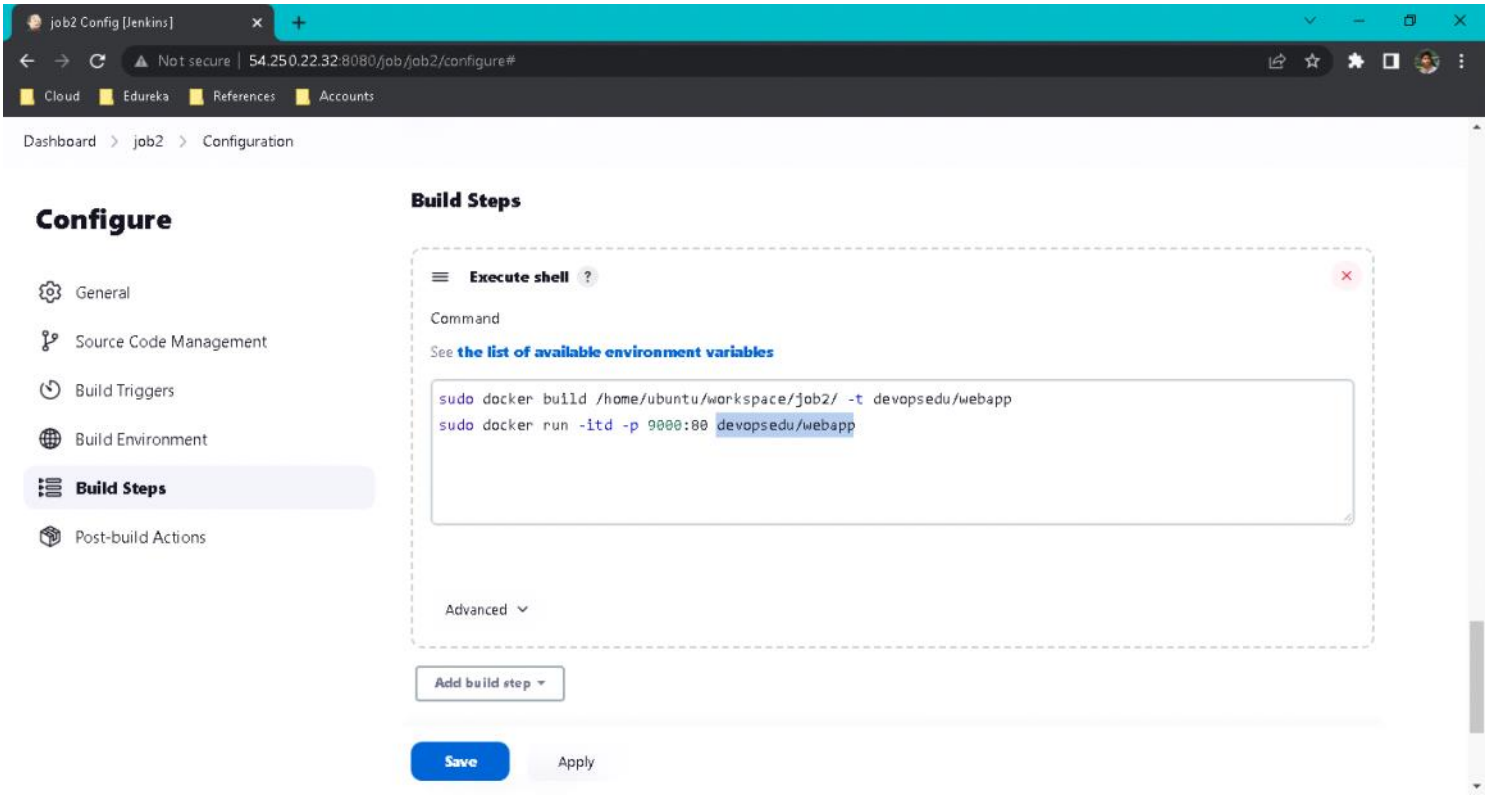
- Jenkins JOB2: To create Container on Slave-Server[host1]



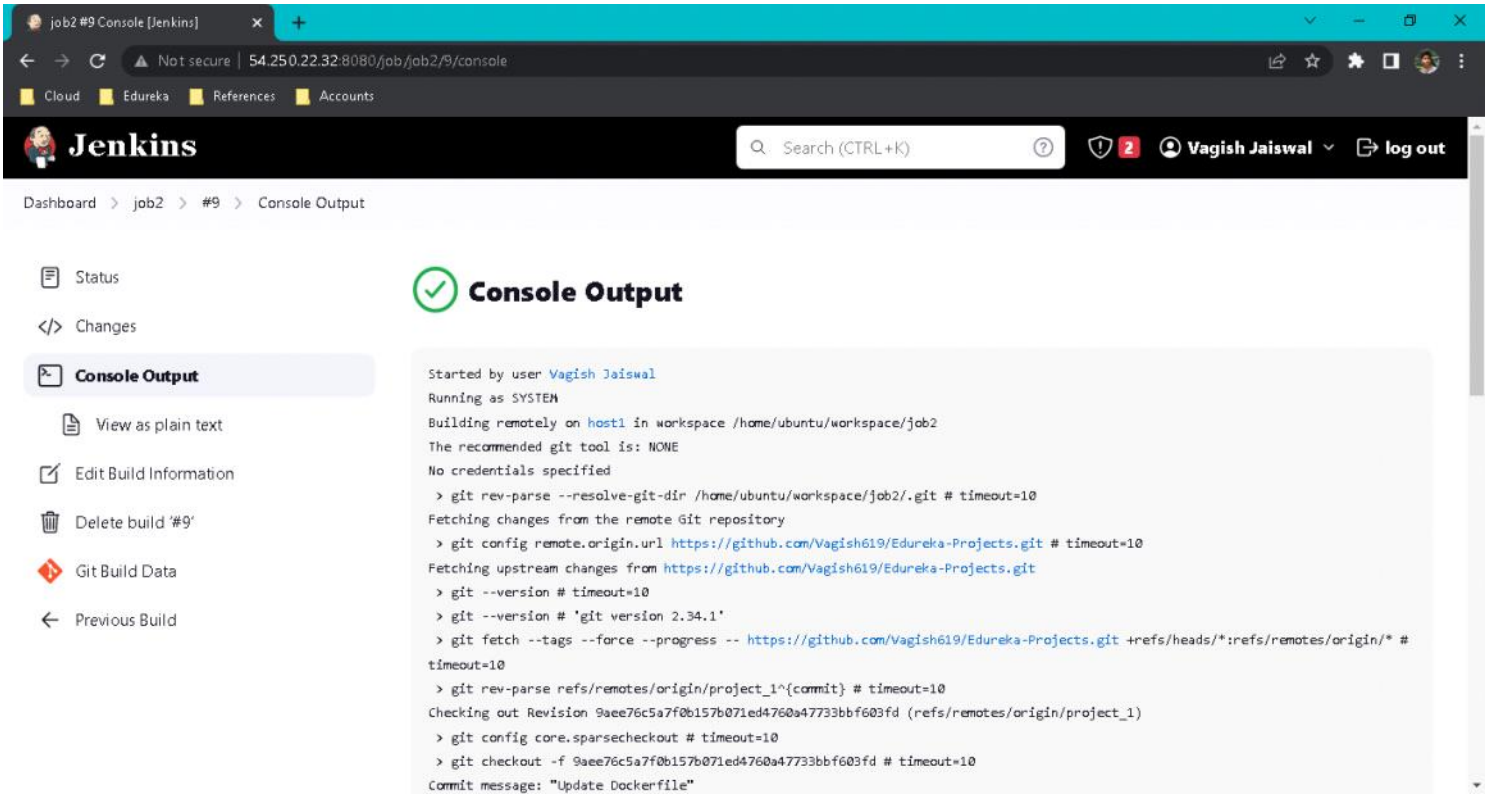

job2 Config [Jenkins]
Not secure | 54.250.22.32:8080/job/job2/configure#
Cloud
Edureka
References
Accounts
100%
Reset
Dashboard > job2 > Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
Source Code Management
None
Git
Repositories
Repository URL
https://github.com/Vagish619/Edureka-Projects.git
Credentials
- none -
Add
Save
Apply

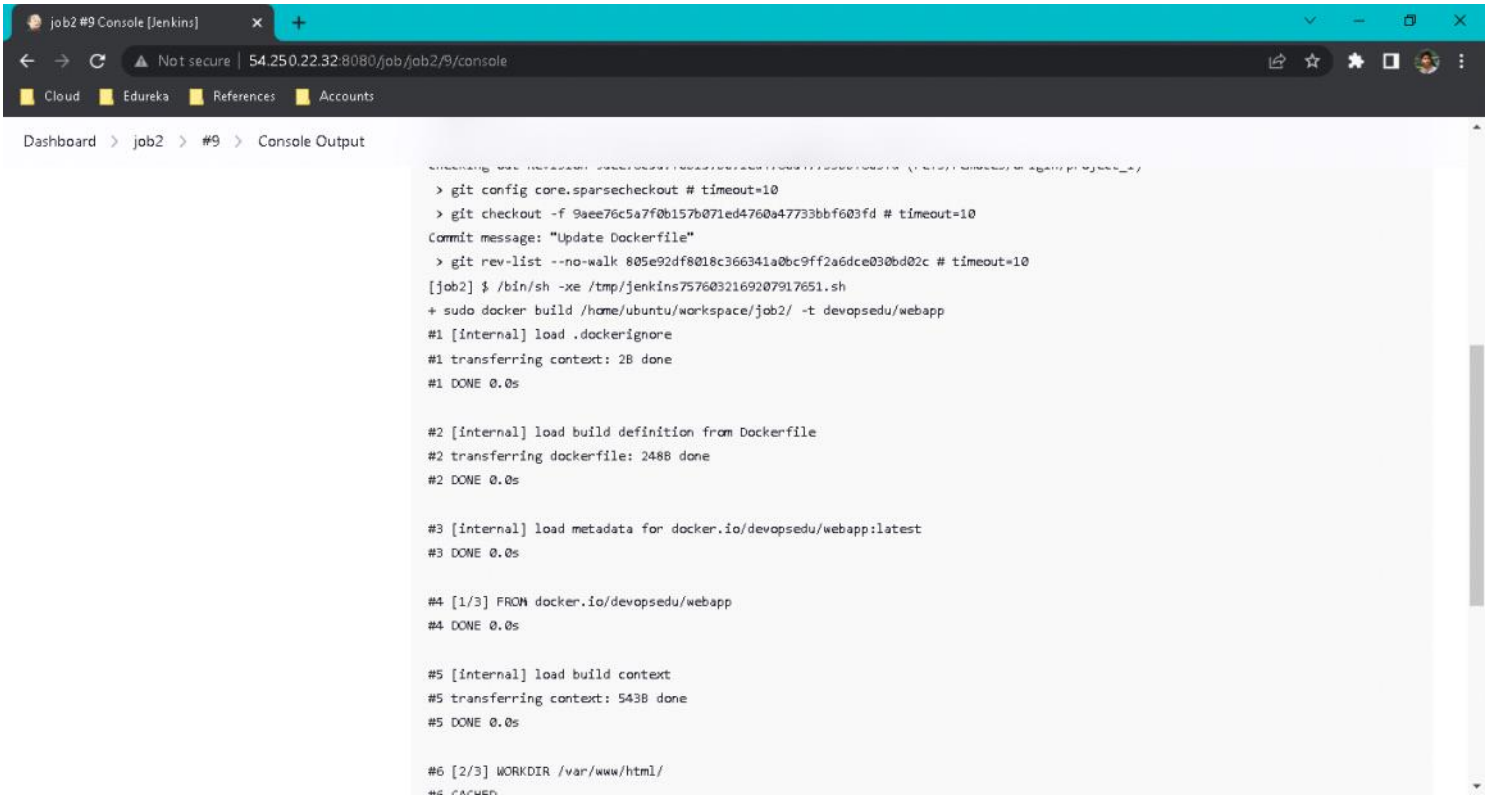

job2 Config [Jenkins]
Not secure | 54.250.22.32:8080/job/job2/configure#
Cloud
Edureka
References
Accounts
Dashboard > job2 > Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
- none -
Add
Advanced
Add Repository
Branches to build
Branch Specifier (blank for 'any')
*/project_1
Add Branch
Repository browser
Save
Apply



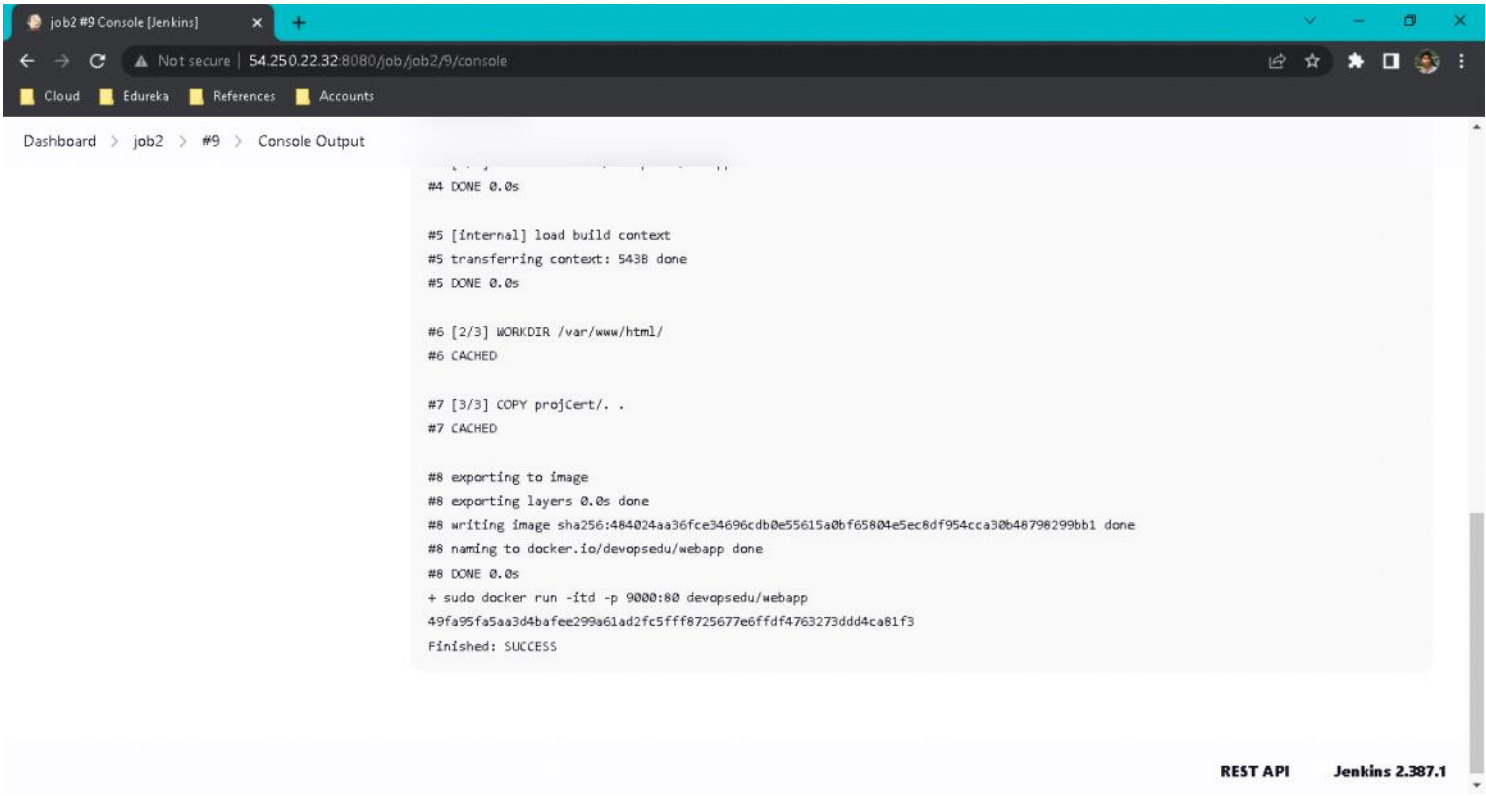


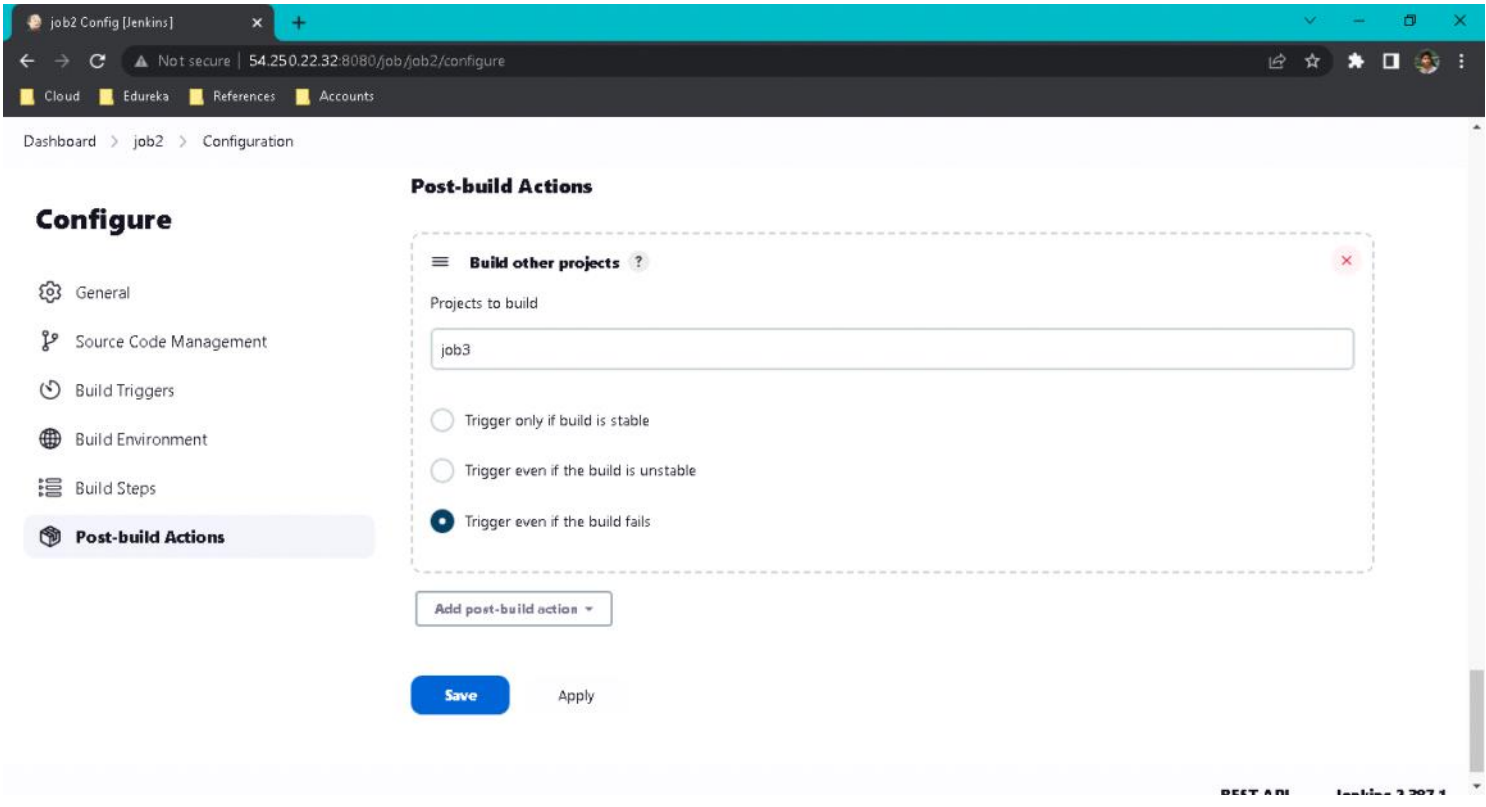
- Jenkins JOB2: Console Output

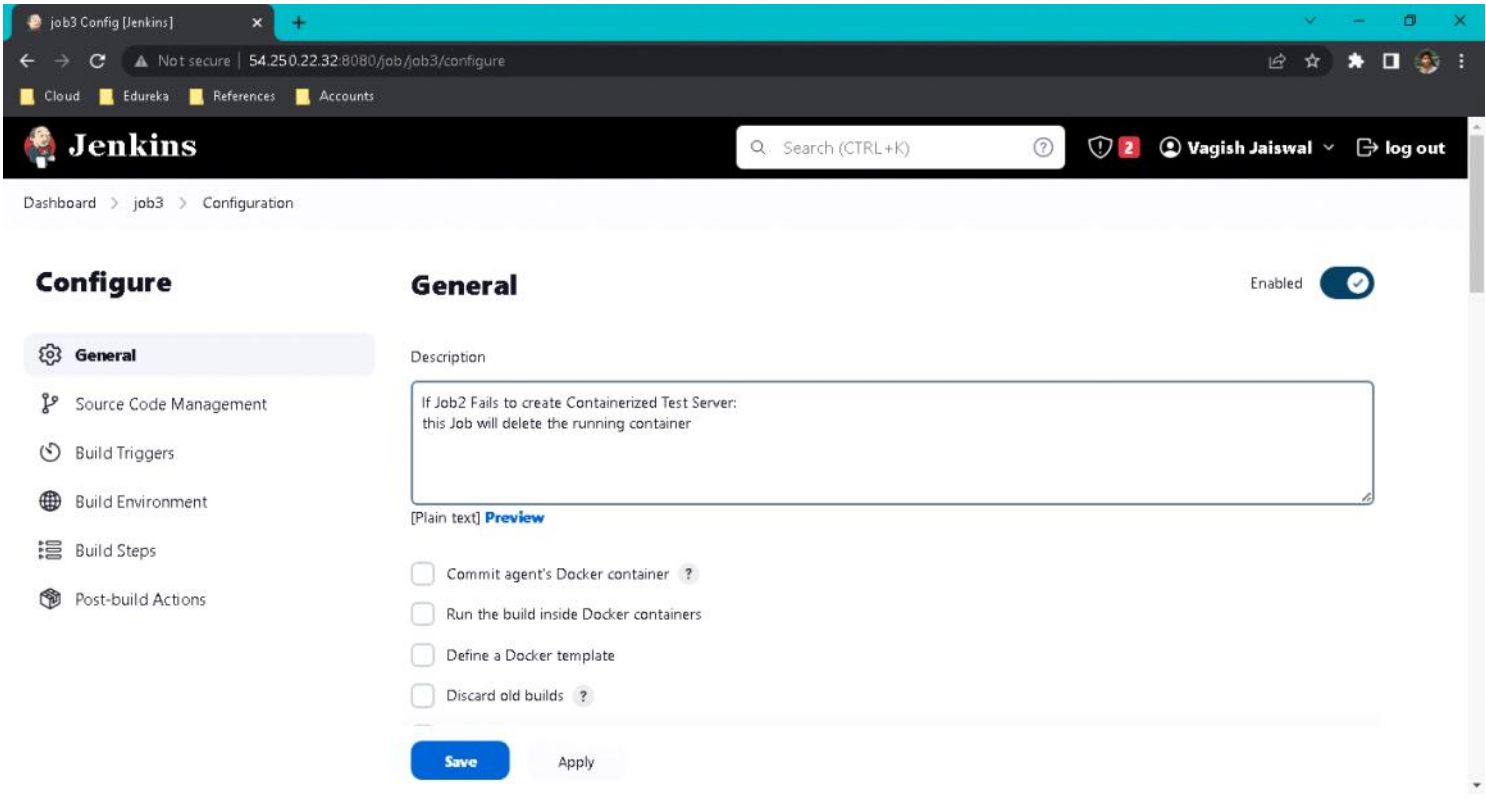


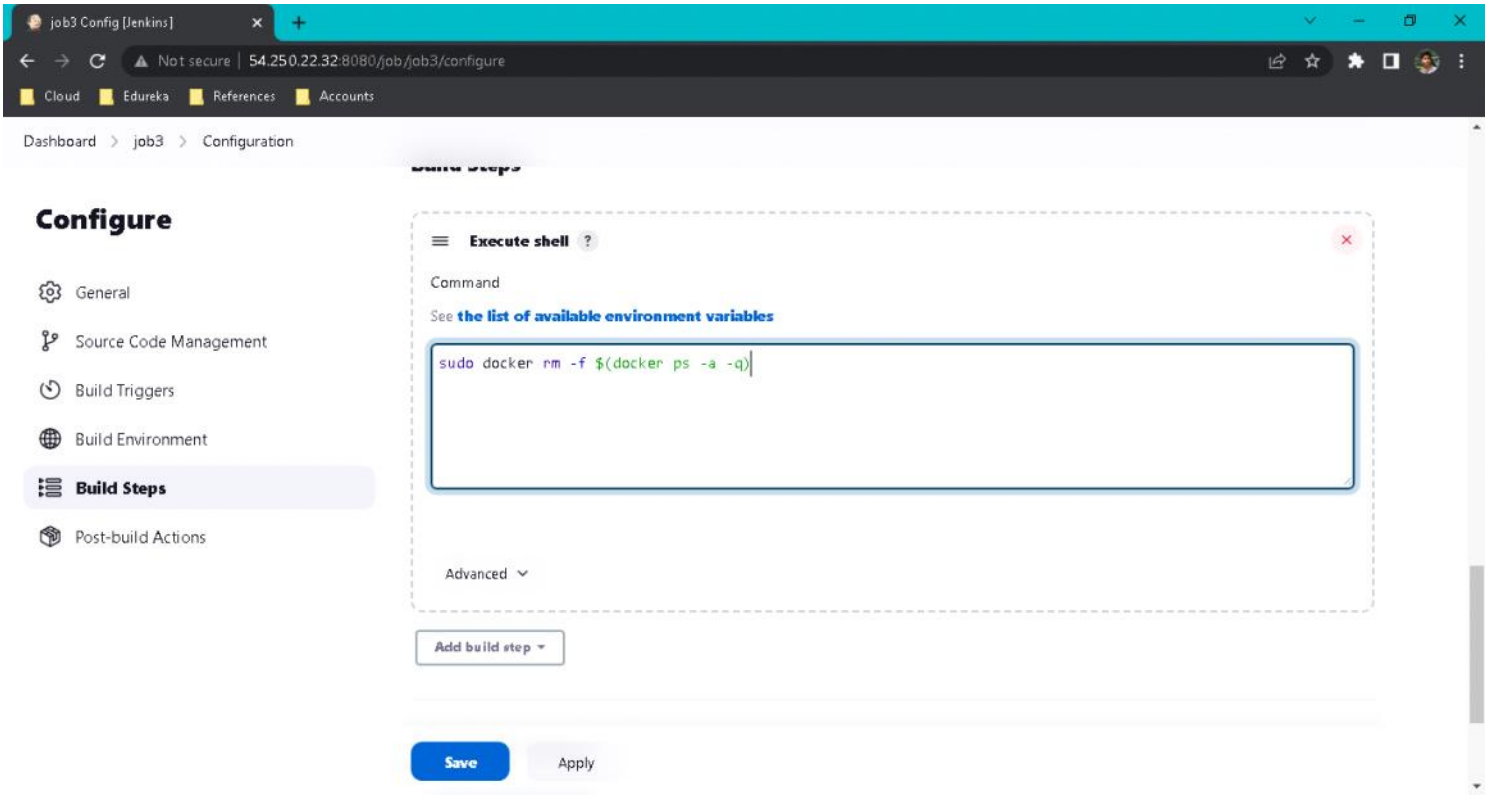


- Jenkins JOB3: if Job2 Fails it will delete the already running Containers

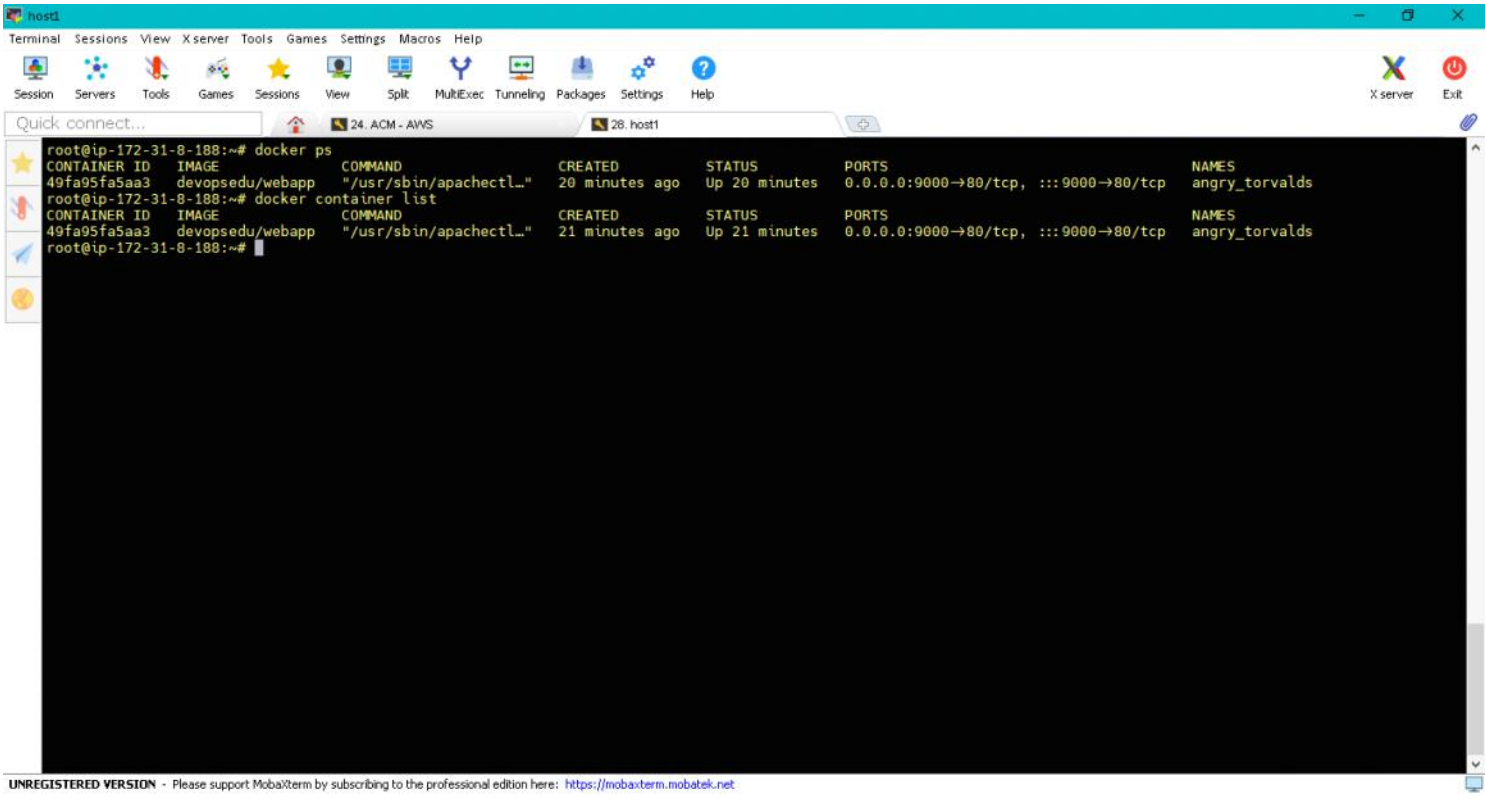

job2 #9 Console [Jenkins]
Not secure | 54.250.22.32:8080/job/job2/9/console
Cloud
Edureka
References
Accounts
Dashboard > job2 > #9 > Console Output
#4 DONE 0.0s
#5 [internal] load build context
#5 transferring context: 543B done
#5 DONE 0.0s
#6 [2/3] WORKDIR /var/www/html/
#6 CACHED
#7 [3/3] COPY projCert/. .
#7 CACHED
#8 exporting to image
#8 exporting layers 0.0s done
#8 writing image sha256:484024aa36fce34696cdb0e55615a0bf65804e5ec8df954cca30b48798299bb1 done
#8 naming to docker.io/devopsedu/webapp done
#8 DONE 0.0s
+ sudo docker run -itd -p 9000:80 devopsedu/webapp
49fa95fa5aa3d4bafee299a61ad2fc5fff8725677e6ffdf4763273ddd4ca81f3
Finished: SUCCESS
REST API
Jenkins 2.387.1


job2 Config [Jenkins]
Not secure | 54.250.22.32:8080/job/job2/configure
Cloud
Edureka
References
Accounts
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

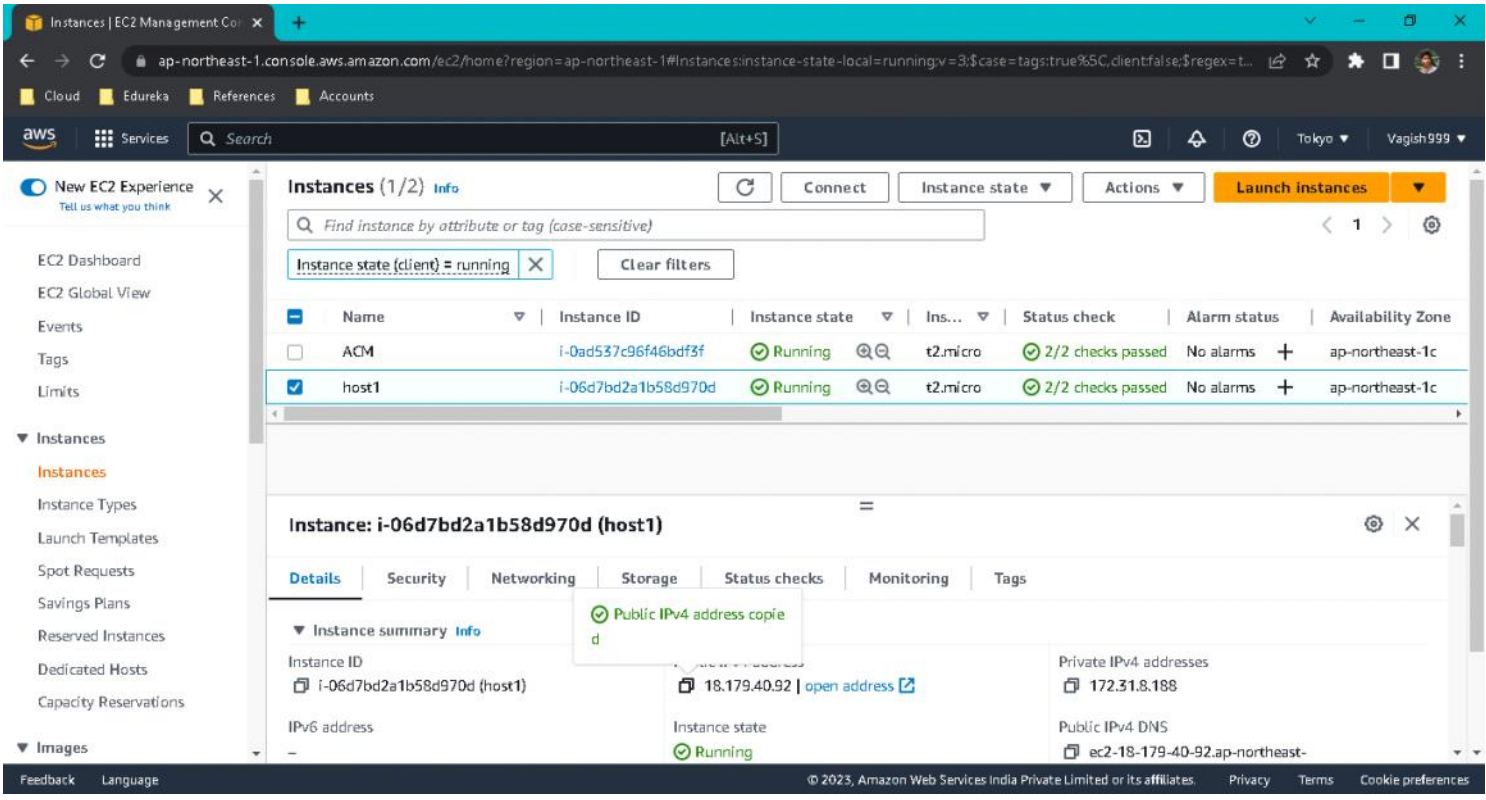




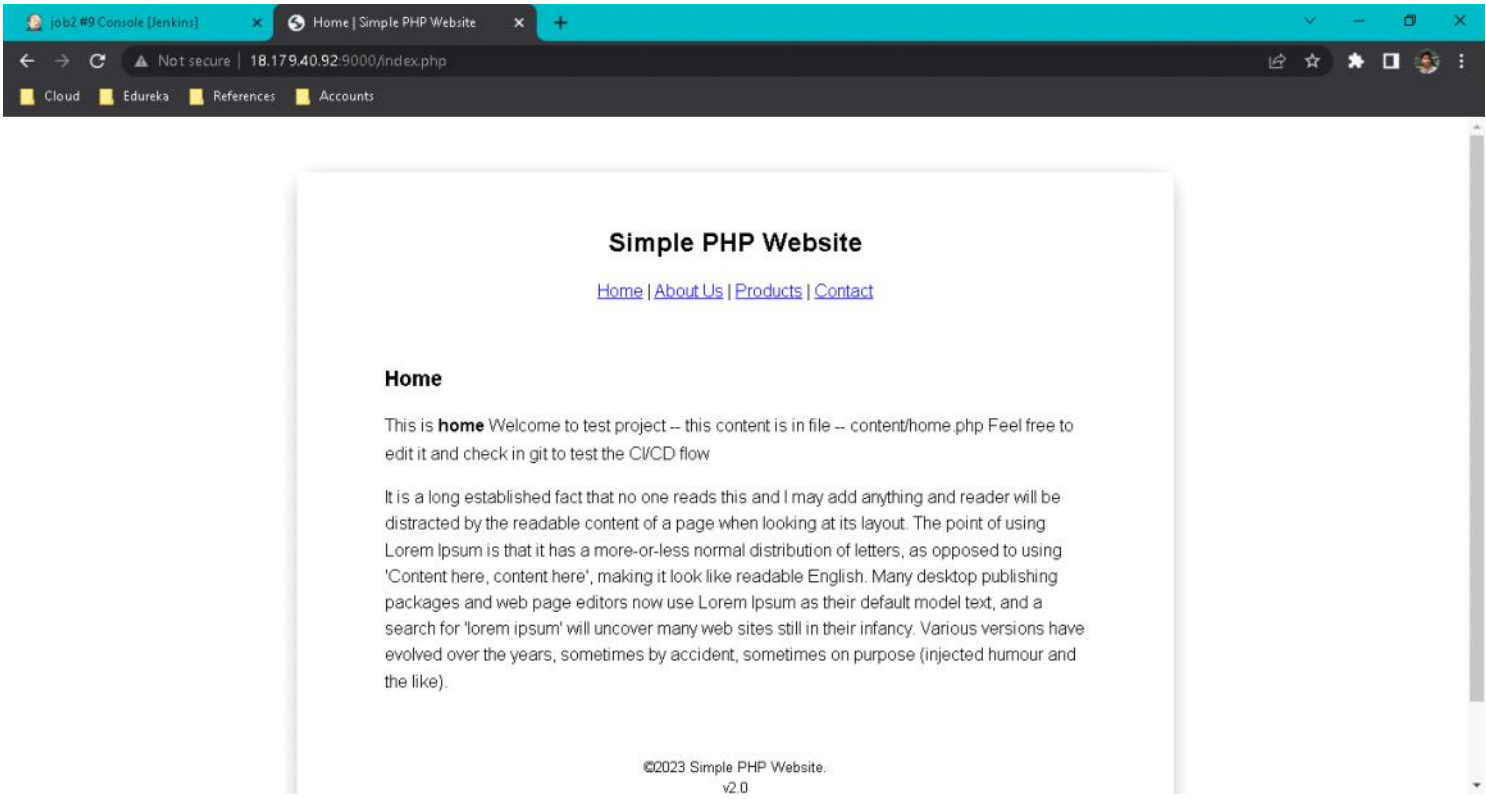
- Slave-Server Screenshot: Created Container (Test-Server)



- Accessing Sample Project on Containerized Test Server on Port:9000



- End Result


job2 #9 Console [Jenkins]
Home | Simple PHP Website
Not secure | 18.179.40.92:9000/index.php
Cloud
Edureka
References
Accounts
Simple PHP Website
Home | About Us | Products | Contact
Home
This is home Welcome to test project -- this content is in file -- content/home.php Feel free to edit it and check in git to test the CI/CD flow
It is a long established fact that no one reads this and I may add anything and reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using 'Content here, content here', making it look like readable English. Many desktop publishing packages and web page editors now use Lorem Ipsum as their default model text, and a search for 'lorem ipsum' will uncover many web sites still in their infancy. Various versions have evolved over the years, sometimes by accident, sometimes on purpose (injected humour and the like).
©2023 Simple PHP Website.
v2.0