

## 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.

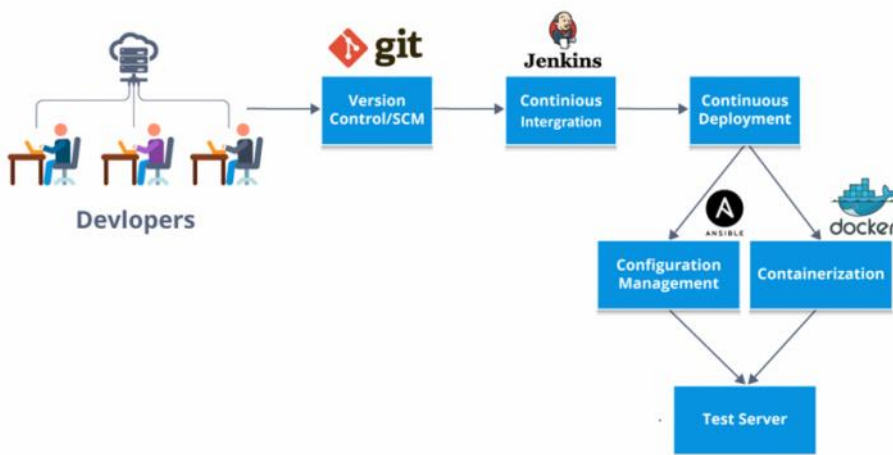
All this should happen automatically and should be triggered from a push to the GitHub master branch.

### 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 & Slave 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

The screenshot shows the AWS Management Console for the 'Instances' page. The left sidebar contains navigation links for EC2 Dashboard, EC2 Global View, Events, Tags, Limits, and a dropdown for 'Instances' which includes 'Instances', 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', and 'Capacity Reservations'. The main content area shows a list of instances with columns: Name, Instance ID, Instance state, Ins..., Status check, Alarm status, and Availability Zone. Two instances are listed: 'ACM' (Instance ID: i-0ad537c96f46bdf3f) and 'host1' (Instance ID: i-06d7bd2a1b58d970d). The 'ACM' instance is selected, and its details are shown in the right pane. The details include the Instance ID, IP address (54.250.22.32), Instance state (Running), and Public IPv4 address (172.31.1.214).

Name	Instance ID	Instance state	Ins...	Status check	Alarm status	Availability Zone
ACM	i-0ad537c96f46bdf3f	Running	t2.micro	2/2 checks passed	No alarms	ap-northeast-1c
host1	i-06d7bd2a1b58d970d	Running	t2.micro	2/2 checks passed	No alarms	ap-northeast-1c

Instance: i-0ad537c96f46bdf3f (ACM)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

Instance summary Info

Instance ID: i-0ad537c96f46bdf3f (ACM)

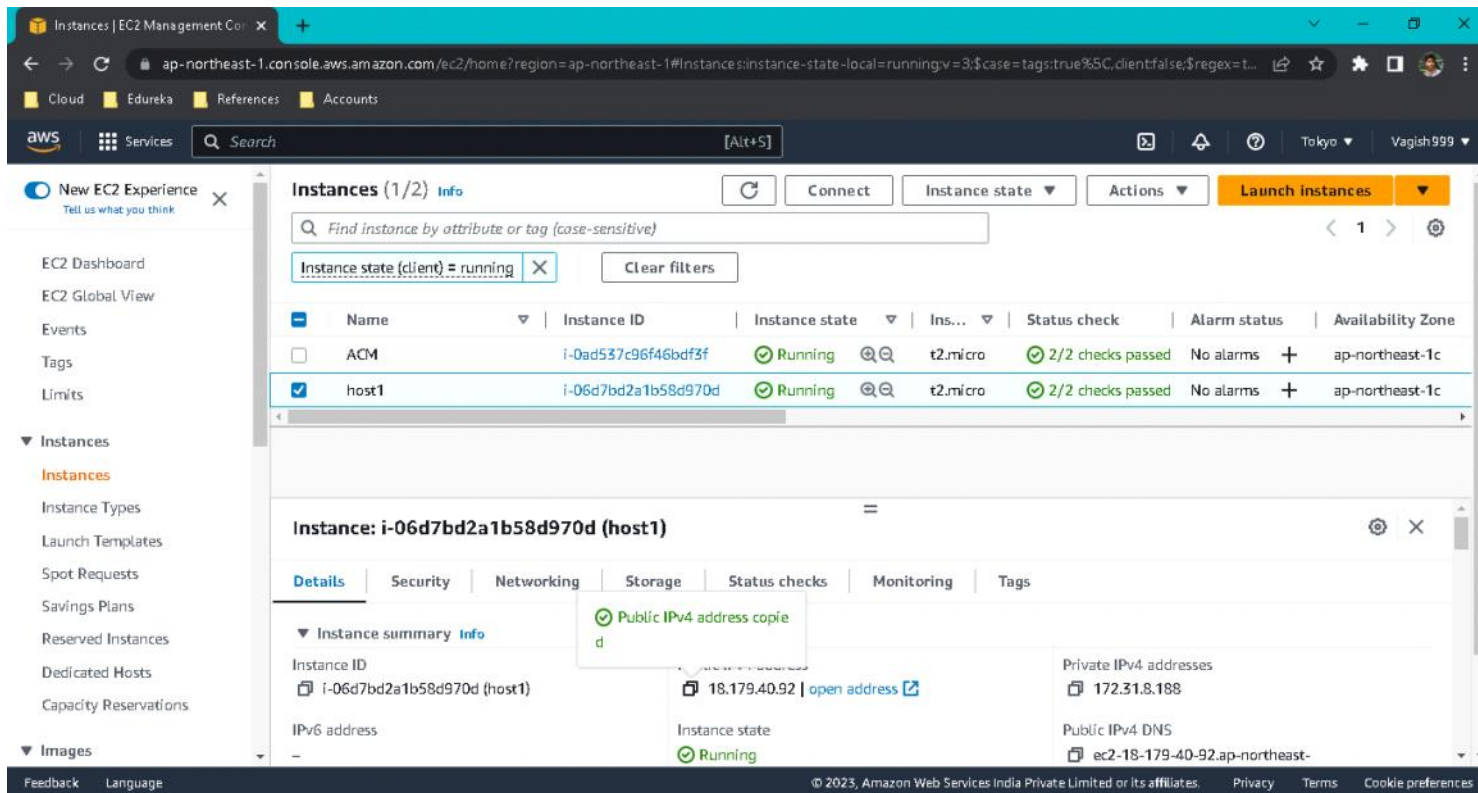
IPV6 address: -

Public IPv4 address: 54.250.22.32 | open address

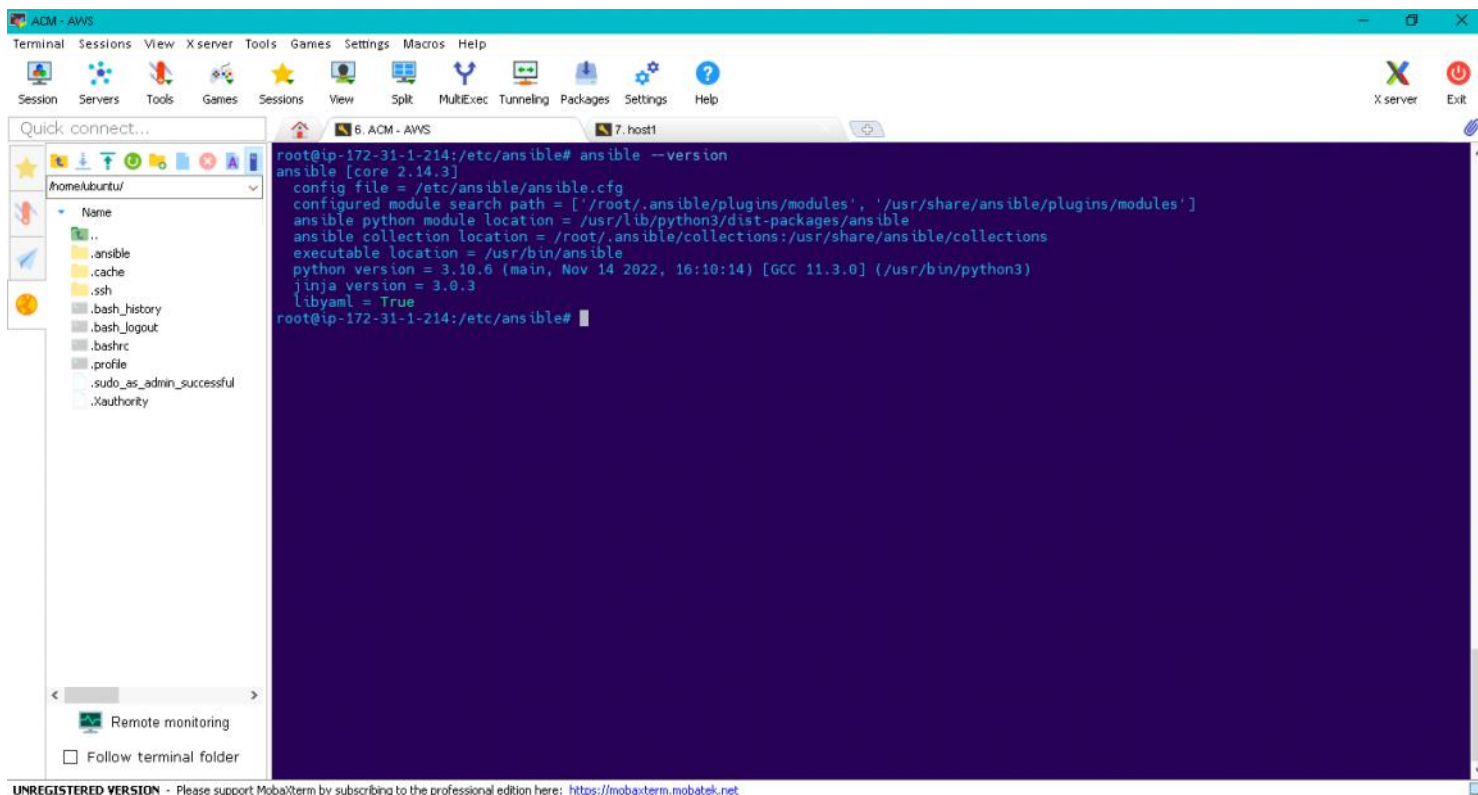
Instance state: Running

Private IPv4 addresses: 172.31.1.214

Public IPv4 DNS: ec2-54-250-22-32.ap-northeast-



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





The screenshot shows the MobaXterm interface with a terminal window titled '7. host1'. The terminal output shows the Ansible version command being executed, displaying the configuration file, search path, module location, collection location, executable location, Python version, Jinja version, and Libyaml status. Following this, a ping command is executed, resulting in a success message and a JSON output showing the discovered interpreter path and the 'ping' result.

```
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>

The screenshot shows the MobaXterm interface with a terminal window titled '7. host1'. The terminal output shows the installation of Jenkins on Ubuntu. It starts with updating the package lists, then downloading the Jenkins keyring and adding it to the system. After updating the package lists again, Jenkins is installed using apt-get. The output shows the packages being fetched, the package lists being read, and the installation of Jenkins and its dependencies.

```
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>

The screenshot displays the MobaXterm interface. The top menu bar includes Terminal, Sessions, View, X server, Tools, Games, Settings, Macros, and Help. Below the menu is a toolbar with icons for Session, Servers, Tools, Games, Sessions, View, Split, MultiExec, Tunneling, Packages, Settings, and Help. A 'Quick connect...' search bar is located on the left. The main terminal window shows a command prompt on a remote host (ip-172-31-1-214) where the command 'service jenkins status' has been executed. The output shows that Jenkins is loaded and active, with details about its PID, memory usage, CPU, and the service group. Below this, a series of log messages from the Jenkins service are visible, including startup messages and status updates. The bottom status bar indicates that the version is 'UNREGISTERED VERSION' and provides a link to the professional edition.

ACM - AWS

Terminal Sessions View X server Tools Games Settings Macros Help

Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help

Quick connect...

home/ubuntu/

Name

- ...
- .ansible
- .cache
- .ssh
- .bash\_history
- .bash\_logout
- .bashrc
- .profile
- .sudo\_as\_admin\_successful
- .Xauthority

```

root@ip-172-31-1-214:/# service jenkins status
● jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2023-03-15 17:42:17 UTC; 13s ago
     Main PID: 6073 (java)
       Tasks: 44 (limit: 1141)
      Memory: 345.4M
         CPU: 42.438s
    CGroup: /system.slice/jenkins.service
            └─6073 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080

Mar 15 17:41:43 ip-172-31-1-214 jenkins[6073]: d3b9ebbacbea4da494fa90f438dd9c7a
Mar 15 17:41:43 ip-172-31-1-214 jenkins[6073]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Mar 15 17:41:43 ip-172-31-1-214 jenkins[6073]: *****
Mar 15 17:41:43 ip-172-31-1-214 jenkins[6073]: *****
Mar 15 17:42:17 ip-172-31-1-214 jenkins[6073]: 2023-03-15 17:42:17.630+0000 [id=28] INFO jenkins.InitReactorRunner$1onAtt
Mar 15 17:42:17 ip-172-31-1-214 jenkins[6073]: 2023-03-15 17:42:17.668+0000 [id=22] INFO hudson.lifecycle.Lifecycle#onRead
Mar 15 17:42:17 ip-172-31-1-214 systemd[1]: Started Jenkins Continuous Integration Server.
Mar 15 17:42:18 ip-172-31-1-214 jenkins[6073]: 2023-03-15 17:42:18.329+0000 [id=44] INFO h.m.DownloadService$Downloadable#
Mar 15 17:42:18 ip-172-31-1-214 jenkins[6073]: 2023-03-15 17:42:18.330+0000 [id=44] INFO hudson.util.Retrier#start: Perfor

lines 1-20/20 (END)
  
```

Remote monitoring

☐ Follow terminal folder

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

host1 [Jenkins] x Home | Simple PHP Website x +

Not secure | 54.250.22.32:8080/manage/computer/host1/

Cloud Edureka References Accounts

**Jenkins** Search (CTRL+K) Vagish Jaiswal log out

Dashboard > Manage Jenkins > Nodes > host1

**Status** Agent host1 Mark this node temporarily offline ?

Delete Agent

Configure

Build History

Load Statistics

Log

**Build Executor Status**

**Connection was broken**

Run from agent command line:

```
curl -sO http://54.250.22.32:8080/jnlpJars/agent.jar
java -jar agent.jar -jnlpUrl http://54.250.22.32:8080/manage/computer/host1/jenkins-agent.jnlp -secret 6028c70af31086a5bacc4dc9e70af37cf37cf562cf43b3dbef4ebd32c20a1214 -workDir "/home/ubuntu"
```

Or run from agent command line, with the secret stored in a file:

```
echo 6028c70af31086a5bacc4dc9e70af37cf37cf562cf43b3dbef4ebd32c20a1214 > secret-file
curl -sO http://54.250.22.32:8080/jnlpJars/agent.jar
java -jar agent.jar -jnlpUrl http://54.250.22.32:8080/manage/computer/host1/jenkins-agent.jnlp -secret @secret-file -workDir "/home/ubuntu"
```

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

host1

Terminal Sessions View X server Tools Games Settings Macros Help

Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help

Quick connect... 11. ACM - AWS 12. host1

```
root@ip-172-31-8-188:/home/ubuntu# curl -sO http://54.250.22.32:8080/jnlpJars/agent.jar
root@ip-172-31-8-188:/home/ubuntu# java -jar agent.jar -jnlpUrl http://54.250.22.32:8080/manage/computer/host1/jenkins-agent.jnlp -secret 6028c70af31086a5bacc4dc9e70af37cf37cf562cf43b3dbef4ebd32c20a1214 -workDir "/home/ubuntu"
Mar 15, 2023 6:20:06 PM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /home/ubuntu/remoting as a remoting work directory
Mar 15, 2023 6:20:06 PM org.jenkinsci.remoting.engine.WorkDirManager setupLogging
INFO: Both error and output logs will be printed to /home/ubuntu/remoting
Mar 15, 2023 6:20:07 PM hudson.remoting.jnlp.Main createEngine
INFO: Setting up agent: host1
Mar 15, 2023 6:20:07 PM hudson.remoting.Engine startEngine
INFO: Using Remoting version: 3107.v665000b_51092
Mar 15, 2023 6:20:07 PM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /home/ubuntu/remoting as a remoting work directory
Mar 15, 2023 6:20:07 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Locating server among [http://54.250.22.32:8080/]
Mar 15, 2023 6:20:07 PM org.jenkinsci.remoting.engine.JnlpAgentEndpointResolver resolve
INFO: Remoting server accepts the following protocols: [JNLP4-connect, Ping]
Mar 15, 2023 6:20:07 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Agent discovery successful
Agent address: 54.250.22.32
Agent port: 43581
Identity: 0b:6e:78:ce:07:f6:78:f0:04:0d:f5:cb:97:8f:4f:63
Mar 15, 2023 6:20:07 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Handshaking
Mar 15, 2023 6:20:07 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Connecting to 54.250.22.32:43581
Mar 15, 2023 6:20:07 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Trying protocol: JNLP4-connect
Mar 15, 2023 6:20:07 PM org.jenkinsci.remoting.protocol.impl.BIONetworkLayer$Reader run
INFO: Waiting for ProtocolStack to start.
Mar 15, 2023 6:20:08 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Remote identity confirmed: 0b:6e:78:ce:07:f6:78:f0:04:0d:f5:cb:97:8f:4f:63
Mar 15, 2023 6:20:08 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Connected
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

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



**Manage nodes and clouds**

Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
Built-In Node	Linux (amd64)	In sync	3.56 GB	0 B	3.56 GB	0ms
host1	Linux (amd64)	In sync	4.78 GB	0 B	4.78 GB	54ms
<b>Data obtained</b>		<b>27 sec</b>	<b>27 sec</b>	<b>27 sec</b>	<b>27 sec</b>	<b>27 sec</b>

REST API Jenkins 2.387.1

- GIT Repo Screenshot: Fresh Branch Created

**project\_1** had recent pushes 5 minutes ago

Compare & pull request

project\_1 2 branches 0 tags

This branch is 4 commits ahead, 1 commit behind main.

**Vagish619** Update Dockerfile 9see76c 4 minutes ago 7 commits

- projCert code added 2 hours ago
- Dockerfile Update Dockerfile 4 minutes ago
- demo.yaml Update demo.yaml 2 hours ago
- install-docker-by-ansible.yml code added 2 hours ago

**About** No description, website, or topics provided.

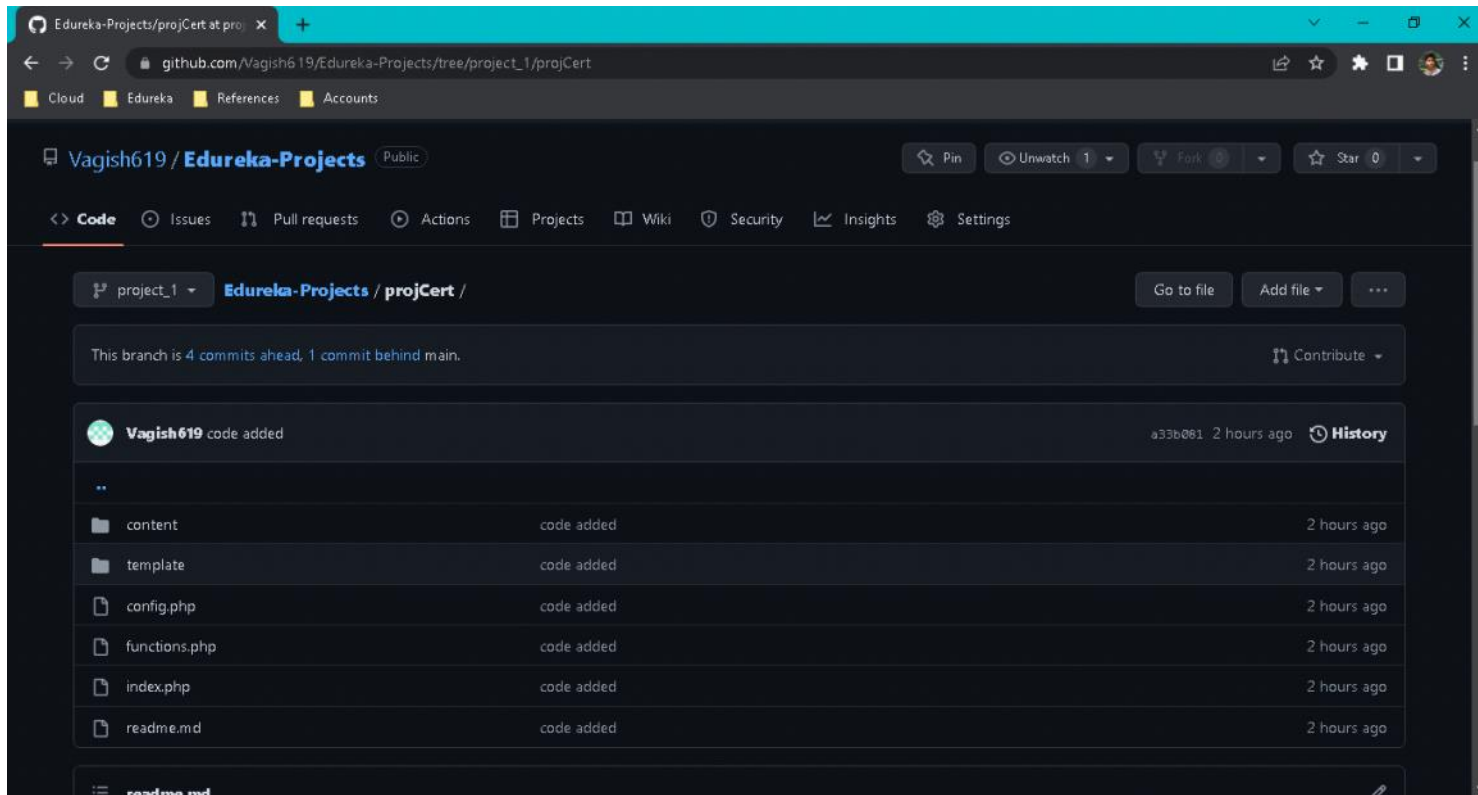
0 stars 1 watching 0 forks

**Releases** No releases published. Create a new release

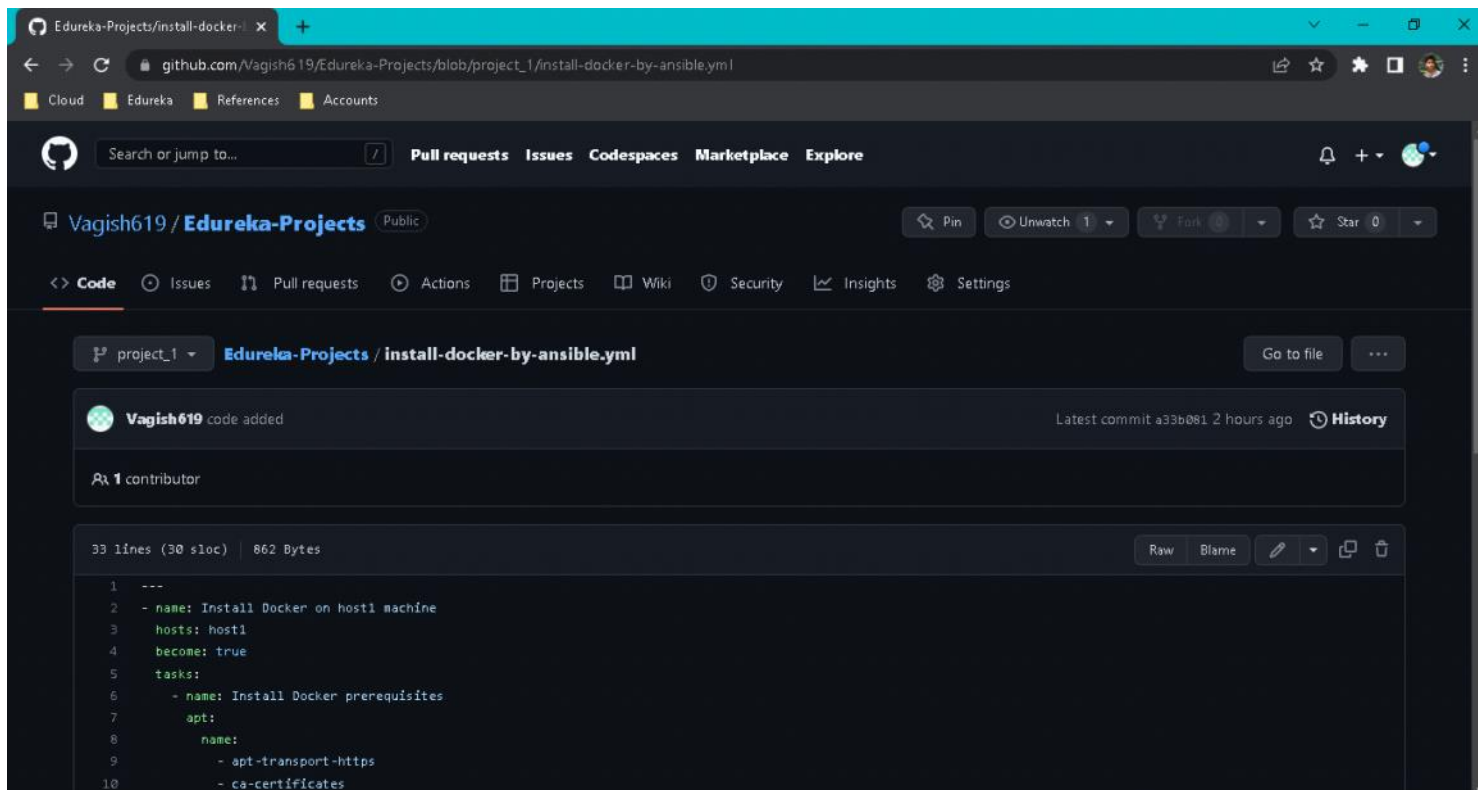
**Packages** No packages published. Publish your first package

- GIT Repo Screenshot: Shared Code uploaded

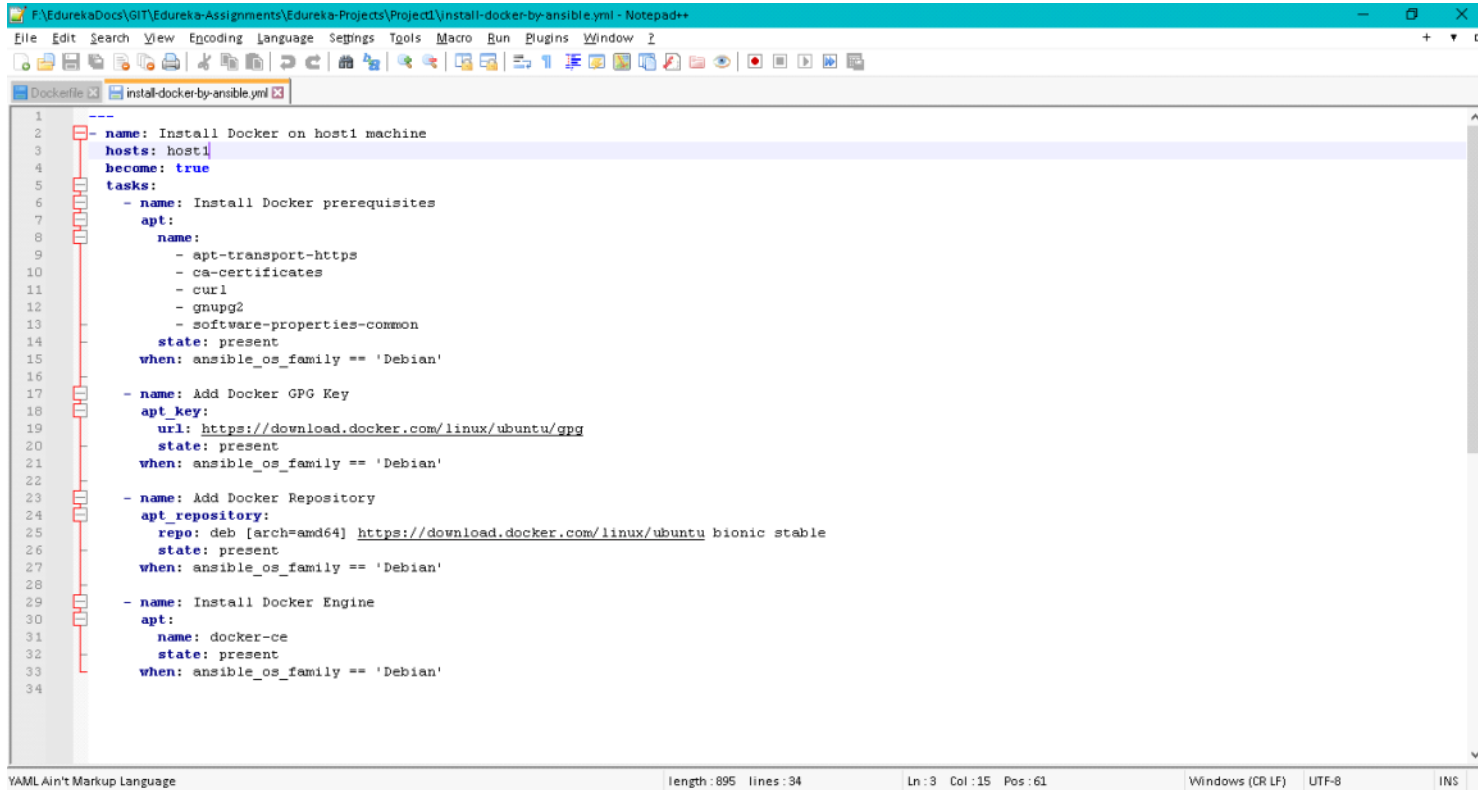




- GIT Repo Screenshot: Ansible-Playbook created to install Docker on Slave Server: host1 from Master VM : ACM
  - File-Name: install-docker-by-ansible.yml



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



```
1  ---
2  - name: Install Docker on host1 machine
3    hosts: host1
4    become: true
5    tasks:
6      - name: Install Docker prerequisites
7        apt:
8          name:
9            - apt-transport-https
10           - ca-certificates
11           - curl
12           - gnupg2
13           - software-properties-common
14          state: present
15          when: ansible_os_family == 'Debian'
16      - name: Add Docker GPG Key
17        apt_key:
18          url: https://download.docker.com/linux/ubuntu/gpg
19          state: present
20          when: ansible_os_family == 'Debian'
21      - name: Add Docker Repository
22        apt_repository:
23          repo: deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable
24          state: present
25          when: ansible_os_family == 'Debian'
26      - name: Install Docker Engine
27        apt:
28          name: docker-ce
29          state: present
30          when: ansible_os_family == 'Debian'
```

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

The screenshot shows a GitHub repository page for 'Edureka-Projects / Dockerfile'. The repository is owned by 'Vagish619'. The latest commit is '9aee76c' from 6 minutes ago. The Dockerfile content is as follows:

```
1 # Base image
2 FROM devopsedu/webapp
3
4 # Set working directory
5 WORKDIR /var/www/html/
6
7 # Copy code files from host to container
8 COPY projCert/. .
9
10 # Start apachectl
11 CMD ["/usr/sbin/apachectl", "-D*", "FOREGROUND"]
```

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

The screenshot shows the Jenkins 'Configure' page for job 'job1'. The 'General' tab is selected. The job is enabled. The description field contains the following text:

```
This job is to perform below tasks on the MASTER VM [ACM-AWS]:
1. Git Clone Repository which includes Ansible-Playbook.
2. Run the Ansible Playbook to install docker on the TEST server [host1]
```

The 'Save' button is highlighted in blue.

job1 Config [Jenkins] x +

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] x +

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`

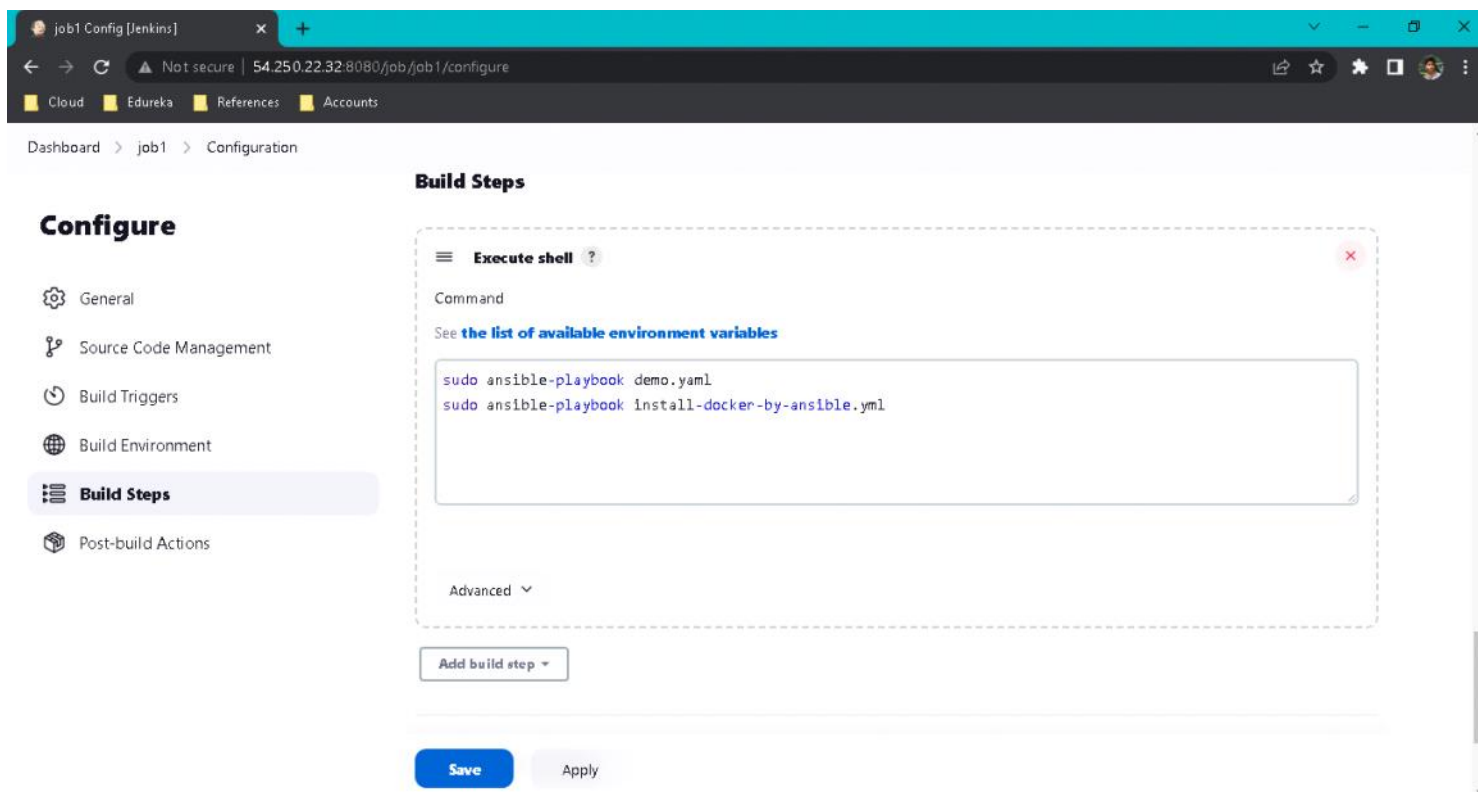
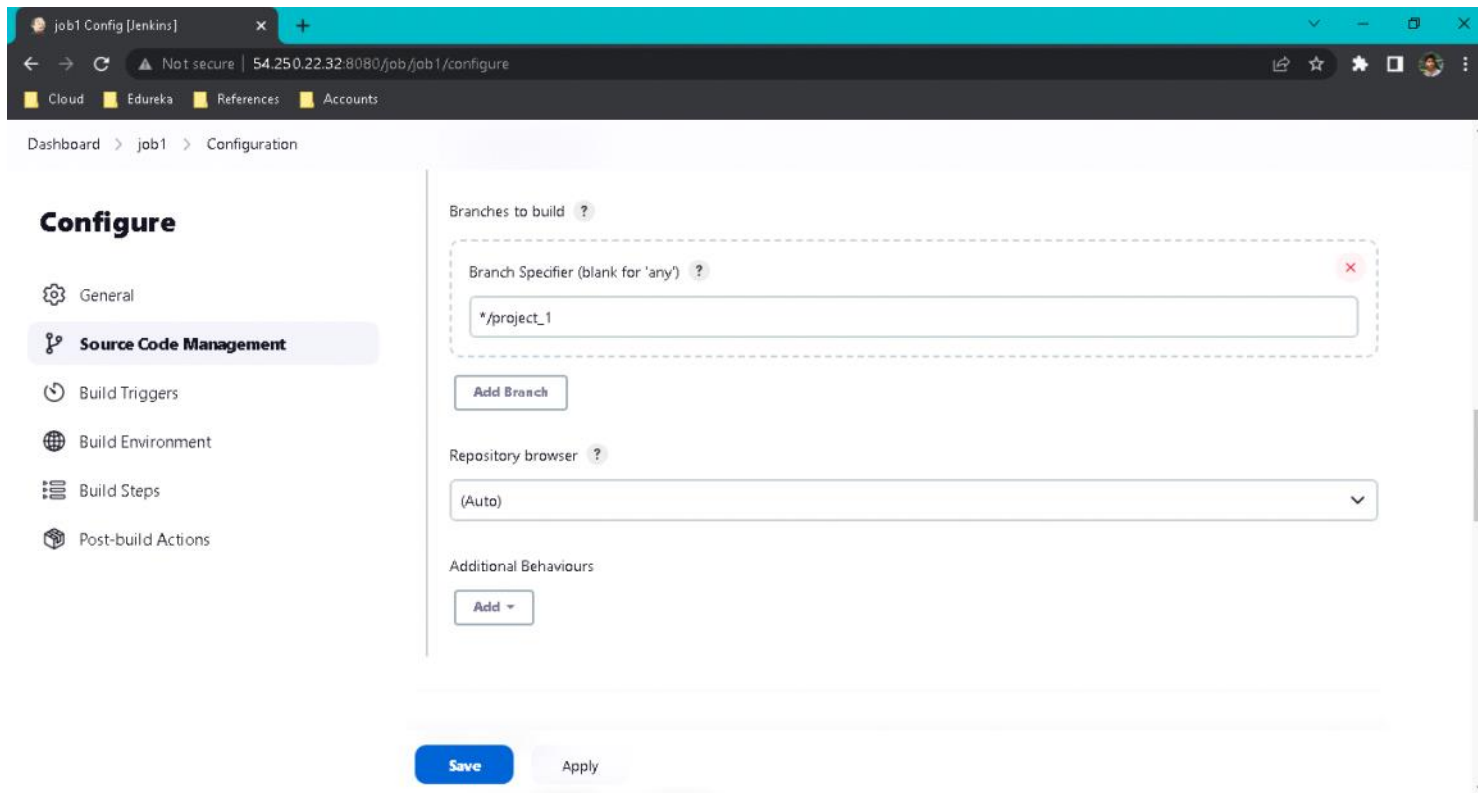
Credentials ?

- none - ▾

[Add ▾](#)

[Save](#) [Apply](#)





- Jenkins JOB1 : Console Output

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"
```

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

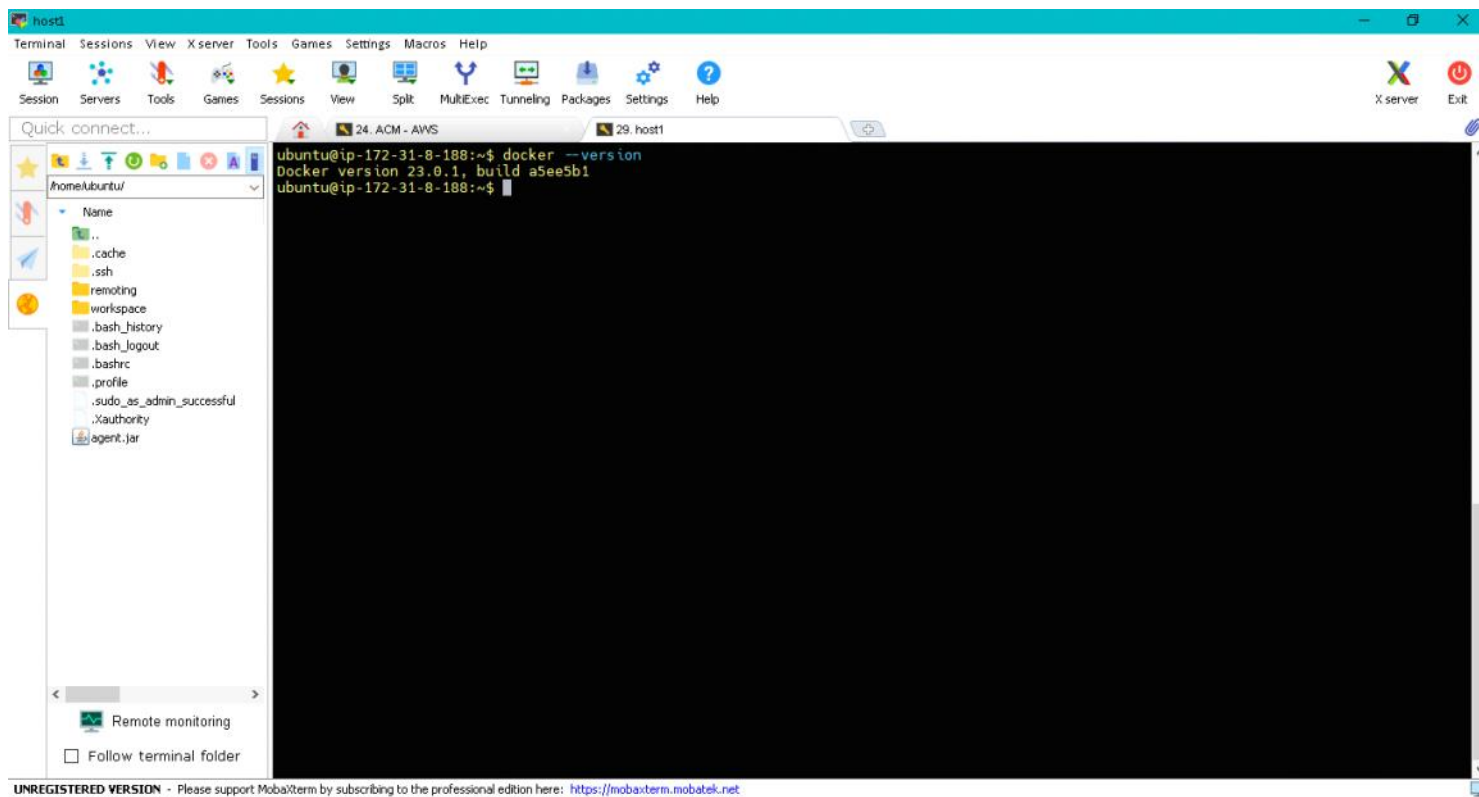
PLAY [testing ping] *****

TASK [Gathering Facts] *****
ok: [host1]

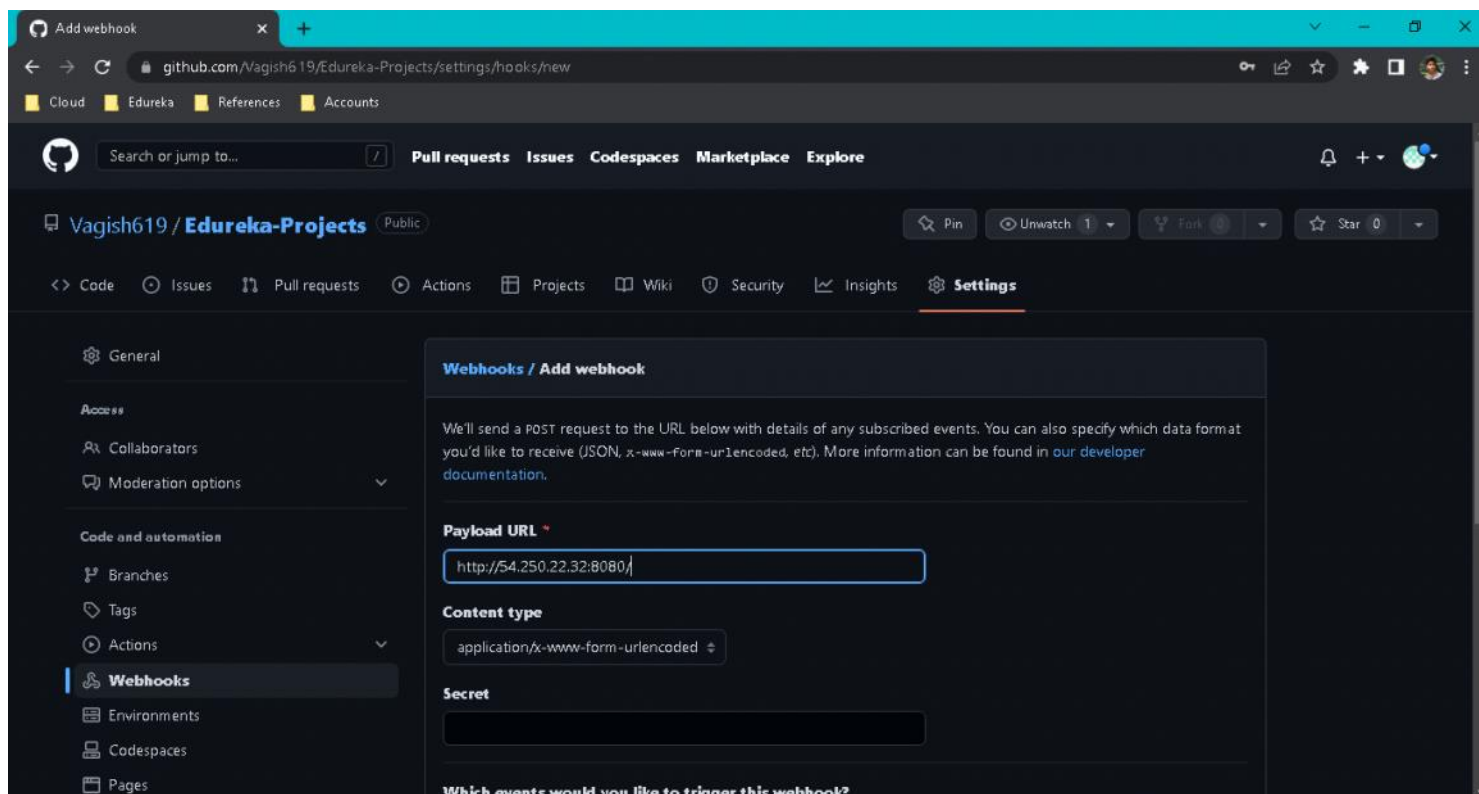
TASK [ping] *****
ok: [host1]

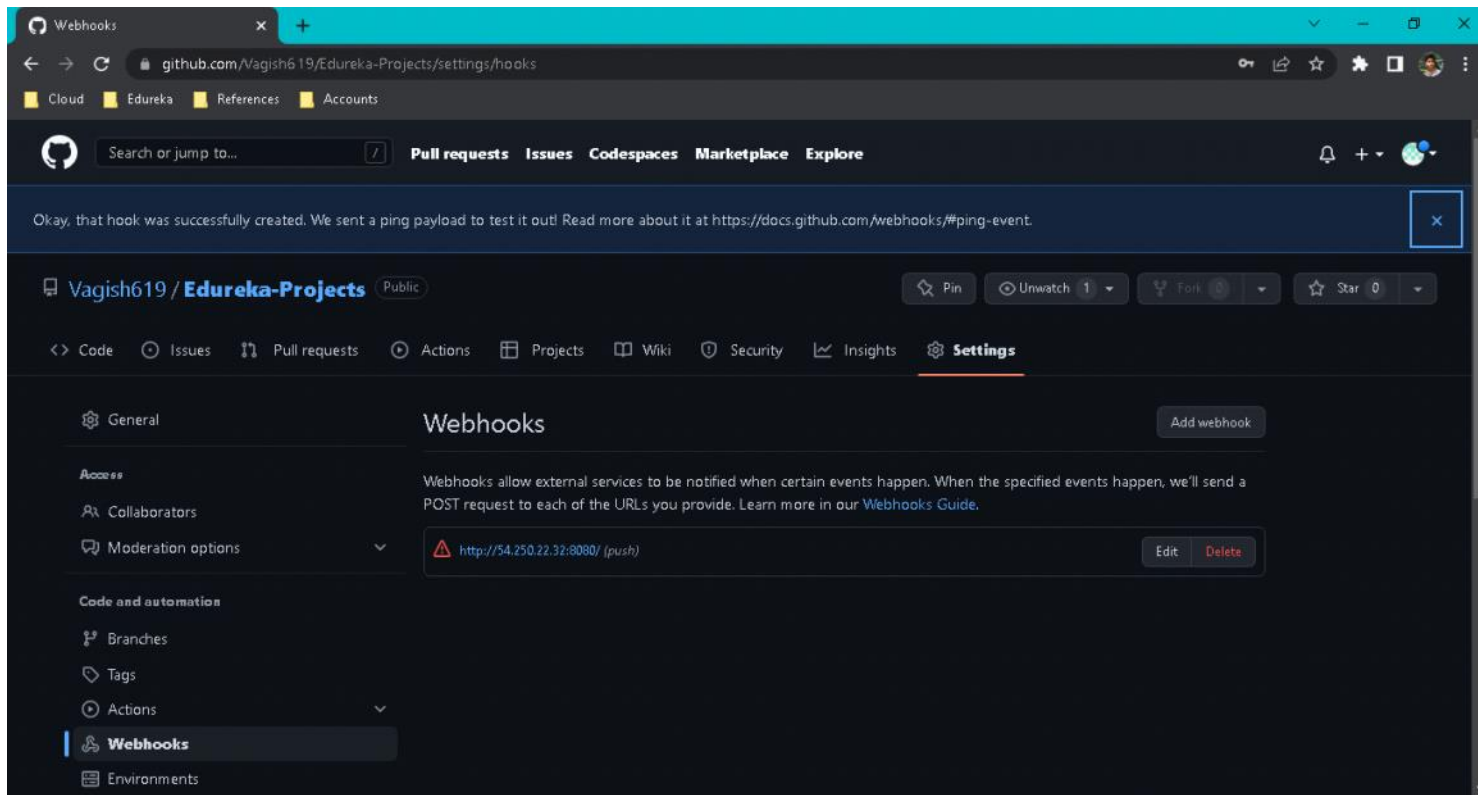
PLAY RECAP *****
host1                : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

Finished: SUCCESS
```

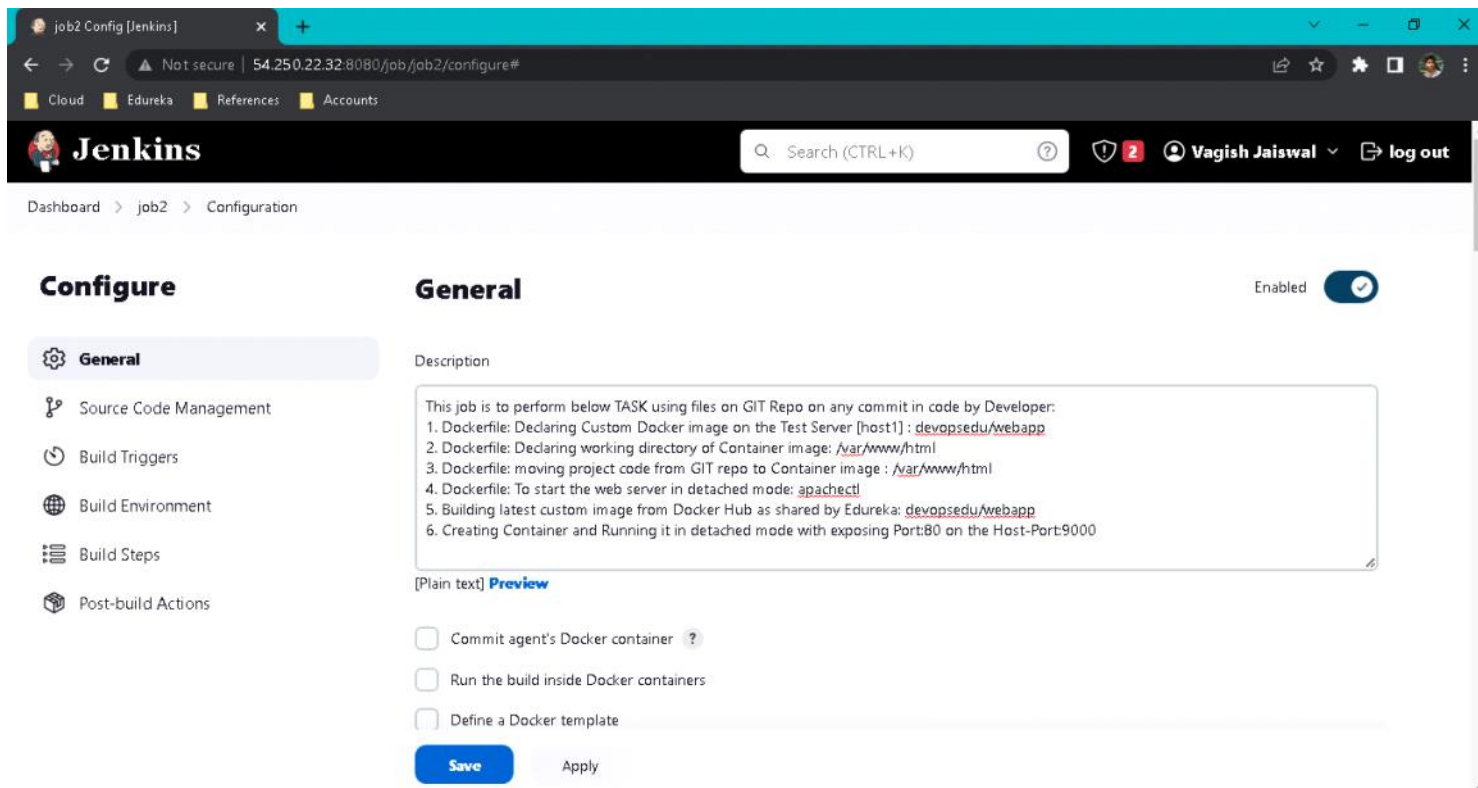


- GIT Repo Screenshot: Created Webhook for Job2





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





job2 Config [Jenkins] x +

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] x +

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

job2 Config [Jenkins] x +

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

### Build Triggers

- ☐ Trigger builds remotely (e.g., from scripts) ?
- ☐ Build after other projects are built ?
- ☐ Build periodically ?
- ☒ GitHub hook trigger for GITScm polling ?
- ☐ Poll SCM ?

### Build Environment

- ☐ Delete workspace before build starts
- ☐ Use secret text(s) or file(s) ?
- ☐ Provide Configuration files ?
- ☐ Add timestamps to the Console Output

**Save** Apply

job2 Config [Jenkins] x +

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

### Build Steps

#### Execute shell ?

Command

See [the list of available environment variables](#)

```
sudo docker build /home/ubuntu/workspace/job2/ -t devopsedu/webapp
sudo docker run -itd -p 9000:80 devopsedu/webapp
```

Advanced ▾

**Add build step ▾**

**Save** Apply

- Jenkins JOB2: Console Output

job2 #9 Console [Jenkins] x +

Not secure | 54.250.22.32:8080/job/job2/9/console

Cloud Edureka References Accounts

**Jenkins** Search (CTRL+K) Vagish Jaiswal log out

Dashboard > job2 > #9 > Console Output

Status

Changes

**Console Output**

View as plain text

Edit Build Information

Delete build '#9'

Git Build Data

Previous Build

**Console Output**

```

Started by user Vagish Jaiswal
Running as SYSTEM
Building remotely on host1 in workspace /home/ubuntu/workspace/job2
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /home/ubuntu/workspace/job2/.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 9aee76c5a7f0b157b071ed4760a47733bbf603fd (refs/remotes/origin/project_1)
> git config core.sparsecheckout # timeout=10
> git checkout -f 9aee76c5a7f0b157b071ed4760a47733bbf603fd # timeout=10
Commit message: "Update Dockerfile"

```

job2 #9 Console [Jenkins] x +

Not secure | 54.250.22.32:8080/job/job2/9/console

Cloud Edureka References Accounts

Dashboard > job2 > #9 > Console Output

```

> git config core.sparsecheckout # timeout=10
> git checkout -f 9aee76c5a7f0b157b071ed4760a47733bbf603fd # timeout=10
Commit message: "Update Dockerfile"
> git rev-list --no-walk 805e92df8018c366341a0bc9ff2a6dce030bd02c # timeout=10
[job2] $ /bin/sh -xe /tmp/jenkins7576032169207917651.sh
+ sudo docker build /home/ubuntu/workspace/job2/ -t devopsedu/webapp
#1 [internal] load .dockerignore
#1 transferring context: 2B done
#1 DONE 0.0s

#2 [internal] load build definition from Dockerfile
#2 transferring dockerfile: 248B done
#2 DONE 0.0s

#3 [internal] load metadata for docker.io/devopsedu/webapp:latest
#3 DONE 0.0s

#4 [1/3] FROM docker.io/devopsedu/webapp
#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

```

- 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

REST API Jenkins 2.387.1



job3 Config [Jenkins] x +

Not secure | 54.250.22.32:8080/job/job3/configure

Cloud Edureka References Accounts

**Jenkins** Search (CTRL+K) Vagish Jaiswal log out

Dashboard > job3 > Configuration

### Configure

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

### General

Enabled

Description

If Job2 Fails to create Containerized Test Server:  
this Job will delete the running container

[Plain text] [Preview](#)

- ☐ Commit agent's Docker container ?
- ☐ Run the build inside Docker containers
- ☐ Define a Docker template
- ☐ Discard old builds ?

[Save](#) [Apply](#)

job3 Config [Jenkins] x +

Not secure | 54.250.22.32:8080/job/job3/configure

Cloud Edureka References Accounts

Dashboard > job3 > Configuration

### Configure

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

### Build Steps

**Execute shell** ?

Command

See [the list of available environment variables](#)

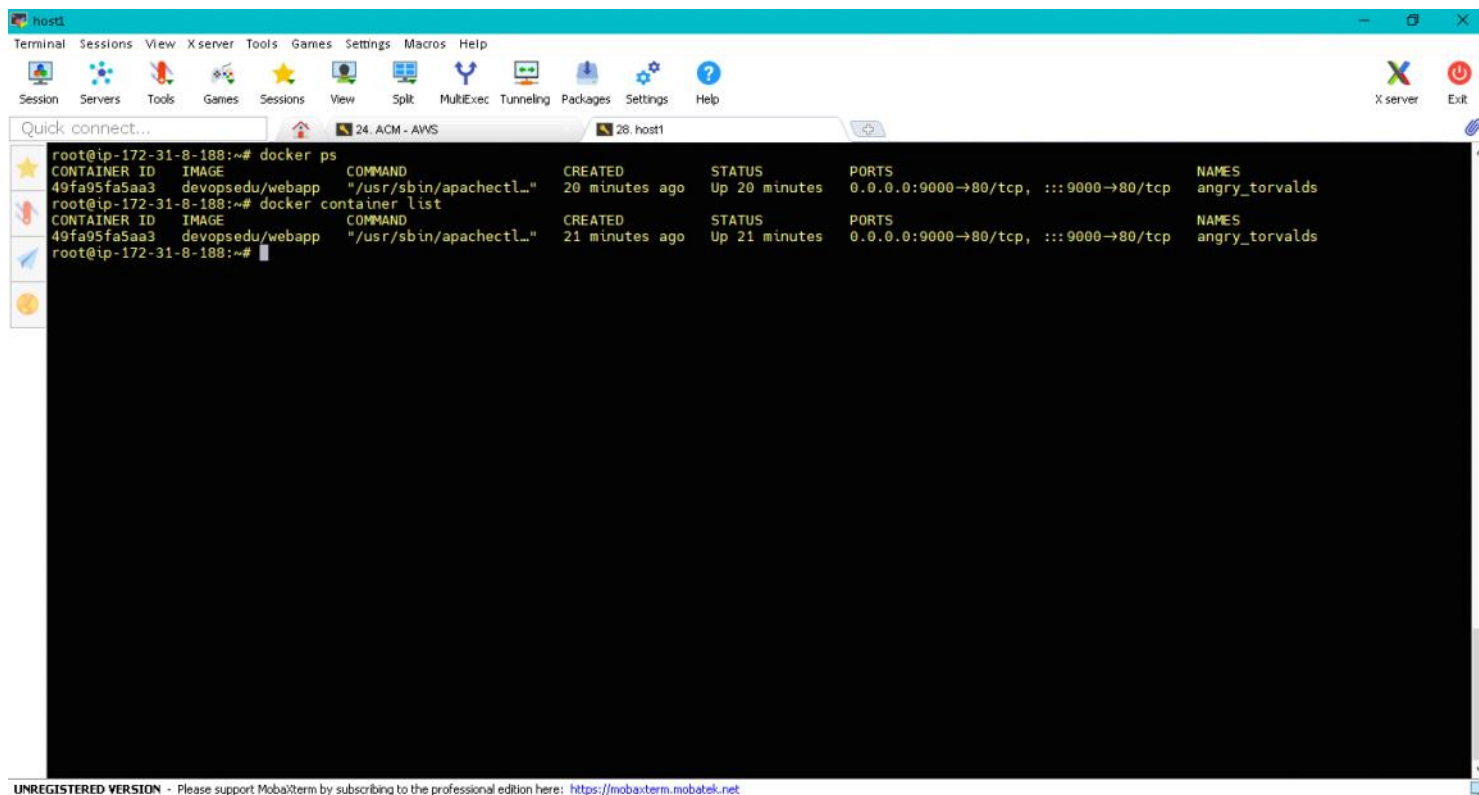
```
sudo docker rm -f $(docker ps -a -q)
```

Advanced

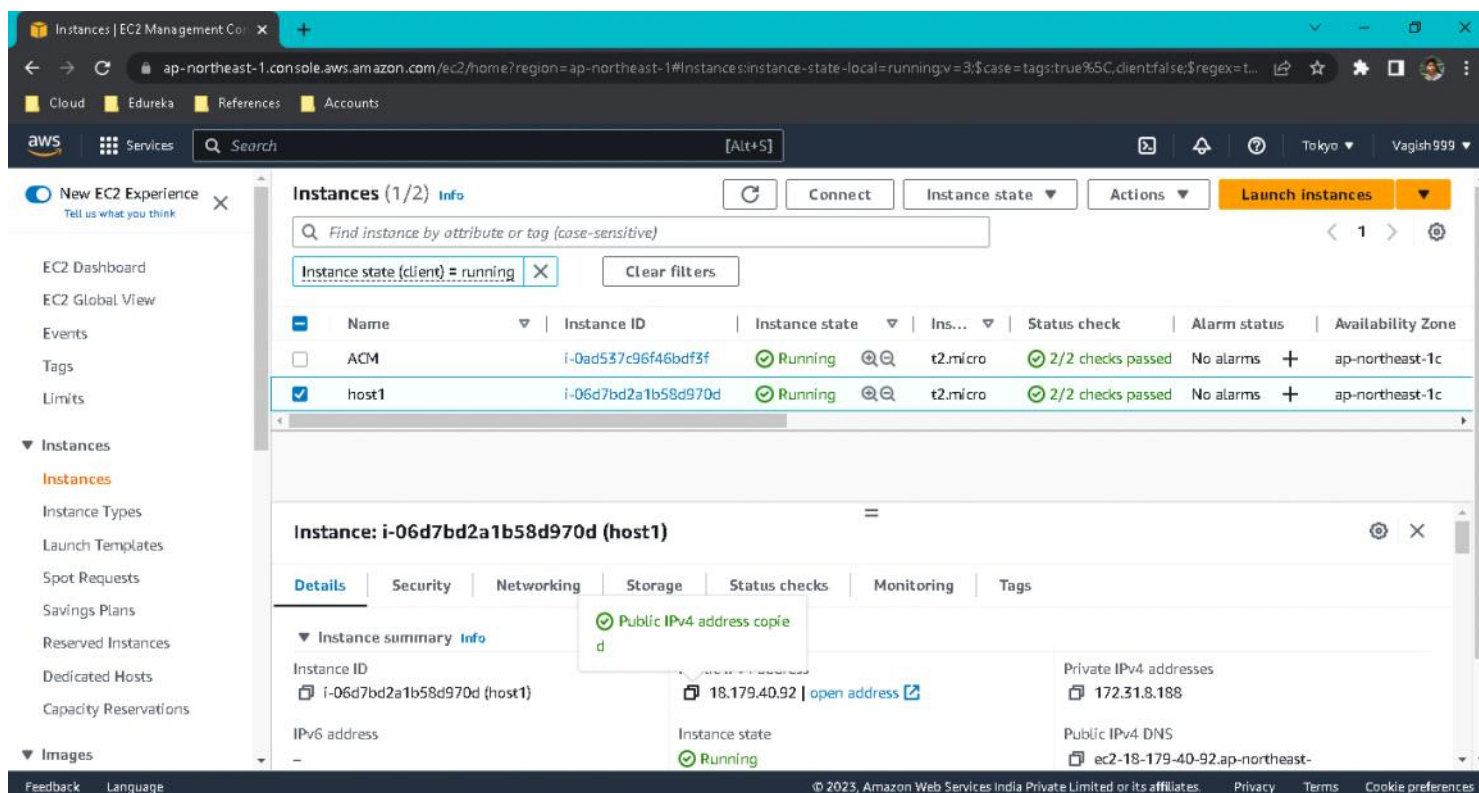
[Add build step](#)

[Save](#) [Apply](#)

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



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



- End Result

## 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