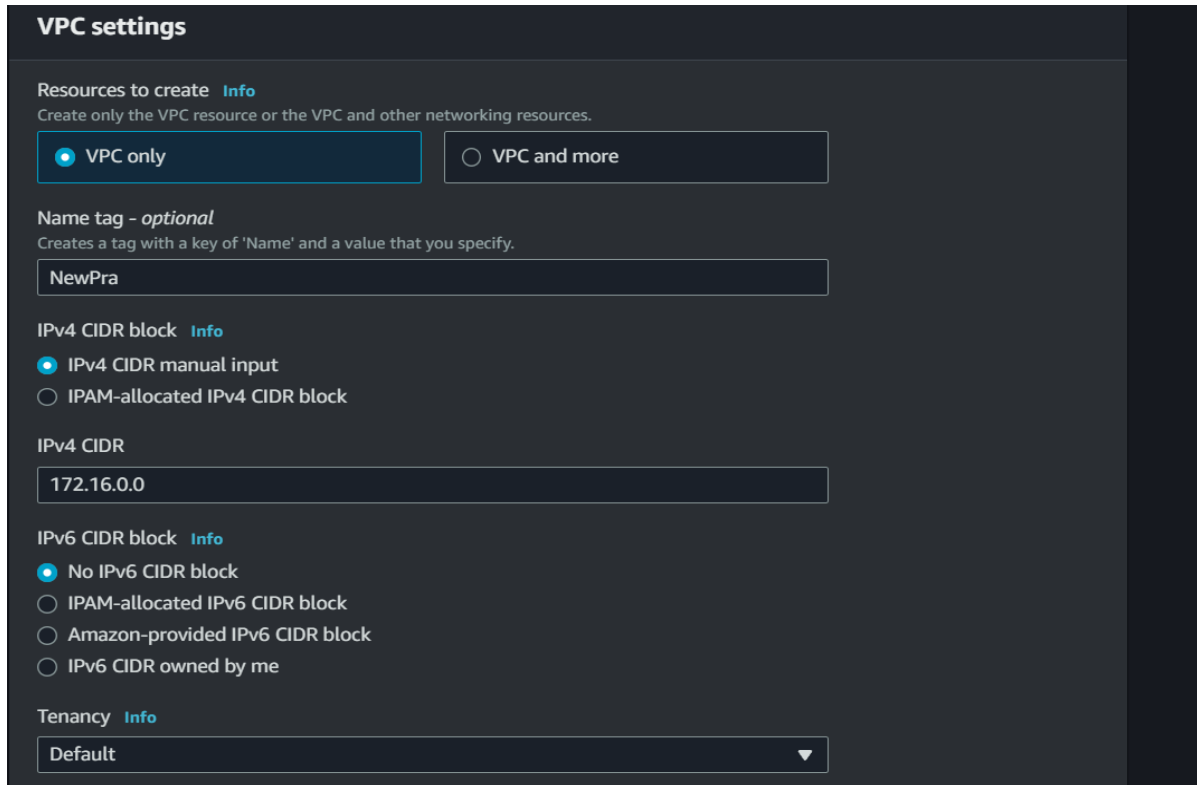


Project – Efficient Web Page Deployment on AWS with Jenkins CI/CD Pipeline

Objective – The objective of the project was to deploy a webpage on a public IP address using AWS and establish continuous deployment on a public server through the utilization of Jenkins' CI/CD pipeline.

Step 1 – A Virtual Private Cloud (VPC) was created on AWS to facilitate network isolation and provide a secure and scalable environment for the project.

- Created VPC



VPC settings

Resources to create [Info](#)
Create only the VPC resource or the VPC and other networking resources.

☒ VPC only ☐ VPC and more

Name tag - *optional*
Creates a tag with a key of 'Name' and a value that you specify.

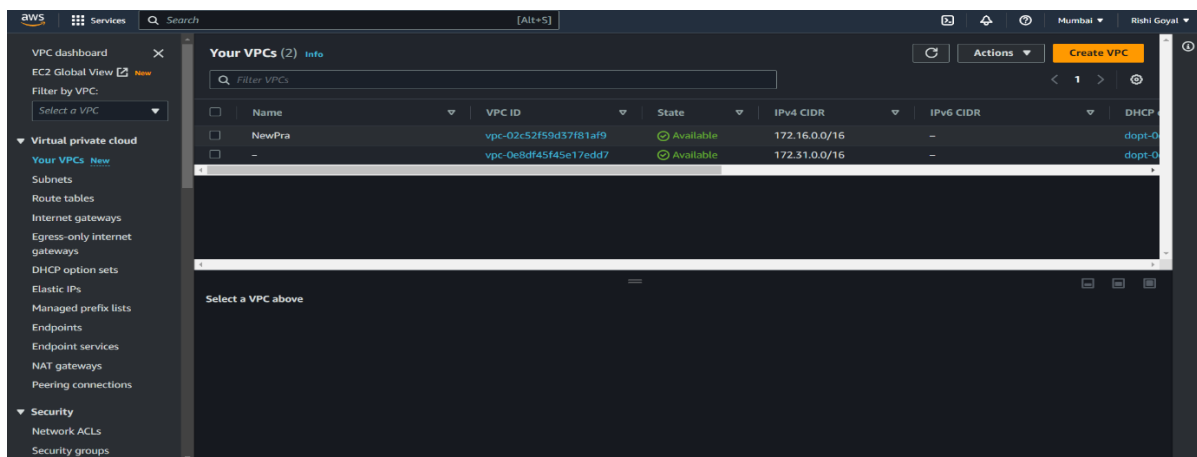
NewPra

IPv4 CIDR block [Info](#)
☒ IPv4 CIDR manual input
☐ IPAM-allocated IPv4 CIDR block

IPv4 CIDR
172.16.0.0

IPv6 CIDR block [Info](#)
☒ No IPv6 CIDR block
☐ IPAM-allocated IPv6 CIDR block
☐ Amazon-provided IPv6 CIDR block
☐ IPv6 CIDR owned by me

Tenancy [Info](#)
Default



Your VPCs (2) [info](#)

Filter by VPCs:

<input type="checkbox"/>	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP
<input type="checkbox"/>	NewPra	vpc-02c52f59d37f81af9	Available	172.16.0.0/16	-	dopt-0
<input type="checkbox"/>	-	vpc-0e8df45f45e17edd7	Available	172.31.0.0/16	-	dopt-0

Select a VPC above

- Created Subnet

VPC ID

Create subnets in this VPC.

vpc-02c52f59d37f81af9 (NewPra) ▼

Associated VPC CIDRs

IPv4 CIDRs

172.16.0.0/16

Subnet settings

Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name

Create a tag with a key of 'Name' and a value that you specify.

Subnet-NewPra

The name can be up to 256 characters long.

Availability Zone [Info](#)

Choose the zone in which your subnet will reside, or let Amazon choose one for you.

No preference ▼

IPv4 CIDR block [Info](#)

172.16.0.0/24 ✕

▼ Tags - optional

VPC dashboard

EC2 Global View New

Filter by VPC:

Select a VPC ▼

Virtual private cloud

Your VPCs New

Subnets

Route tables

Internet gateways

Egress-only internet gateways

DHCP option sets

Elastic IPs

Managed prefix lists

Endpoints

Endpoint services

NAT gateways

Peering connections

Security

Network ACLs

Security groups

You have successfully created 1 subnet: subnet-092e821e262795ea9

Subnets (4) [Info](#)

Filter subnets

< 1 > ⚙

<input type="checkbox"/>	Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR
<input type="checkbox"/>	-	subnet-0319414ae492c9517	Available	vpc-0e8df45f45e17edd7	172.31.32.0/20	-
<input type="checkbox"/>	-	subnet-0216ce5e9d96dfae5	Available	vpc-0e8df45f45e17edd7	172.31.0.0/20	-
<input type="checkbox"/>	-	subnet-0db490161dc62d857	Available	vpc-0e8df45f45e17edd7	172.31.16.0/20	-
<input type="checkbox"/>	Subnet-NewPra	subnet-092e821e262795ea9	Available	vpc-02c52f59d37f81af9 New...	172.16.0.0/24	-

Select a subnet

CloudShell

Feedback

Language

© 2023, Amazon Web Services India Private Limited or its affiliates.

Privacy

Terms

Cookie preferences

- Created Internet gateway

Internet gateways (2) [Info](#)

Filter internet gateways

< 1 > ⚙

<input type="checkbox"/>	Name ▾	Internet gateway ID ▾	State ▾	VPC ID ▾	Owner ▾
<input type="checkbox"/>	internetgateway-N...	igw-06f676cdb338bf231	⊖ Detached	-	379277208124
<input type="checkbox"/>	-	igw-0758a204e5b792d47	✔ Attached	vpc-0e8df45f45e17edd7	379277208124

VPC > Internet gateways > Attach to VPC (igw-06f676cdb338bf231)

Attach to VPC (igw-06f676cdb338bf231) [Info](#)

VPC

Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.

Available VPCs

Attach the internet gateway to this VPC.

Q vpc-02c52f59d37f81af9 X

▶ AWS Command Line Interface command

Cancel

Attach internet gateway

- Created Route Table

VPC > Route tables > Create route table

Create route table [Info](#)

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Route table settings

Name - optional
Create a tag with a key of 'Name' and a value that you specify.

RouteTable-NewPra

VPC
The VPC to use for this route table.

vpc-02c52f59d37f81af9 (NewPra)

Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key Value - optional

Name RouteTable-NewPra Remove

Add new tag

You can add 49 more tags.

Cancel Create route table

Step 2 – Three instances were created for different purposes: one for Git, another for Docker, and the third for Jenkins, and attached instances with the created VPC.

aws Services Search [Alt+S]

Mumbai Rishi Goyal

Name and tags [Info](#)

Name

Git Add additional tags

Application and OS Images (Amazon Machine Image) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Quick Start

Amazon Linux macOS Ubuntu Windows Red Hat S

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI Free tier eligible

ami-0b08bfc6ff7069aff (64-bit x86) / ami-0d583450f458cbbaa (64-bit ARM)

Virtualization: hvm ENA enabled: true Root device type: ebs

Summary

Number of instances [Info](#)

1

Software Image (AMI)

Amazon Linux 2023 AMI 2023.02...read more

ami-0b08bfc6ff7069aff

Virtual server type (instance type)

t2.micro

Firewall (security group)

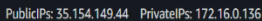
New security group

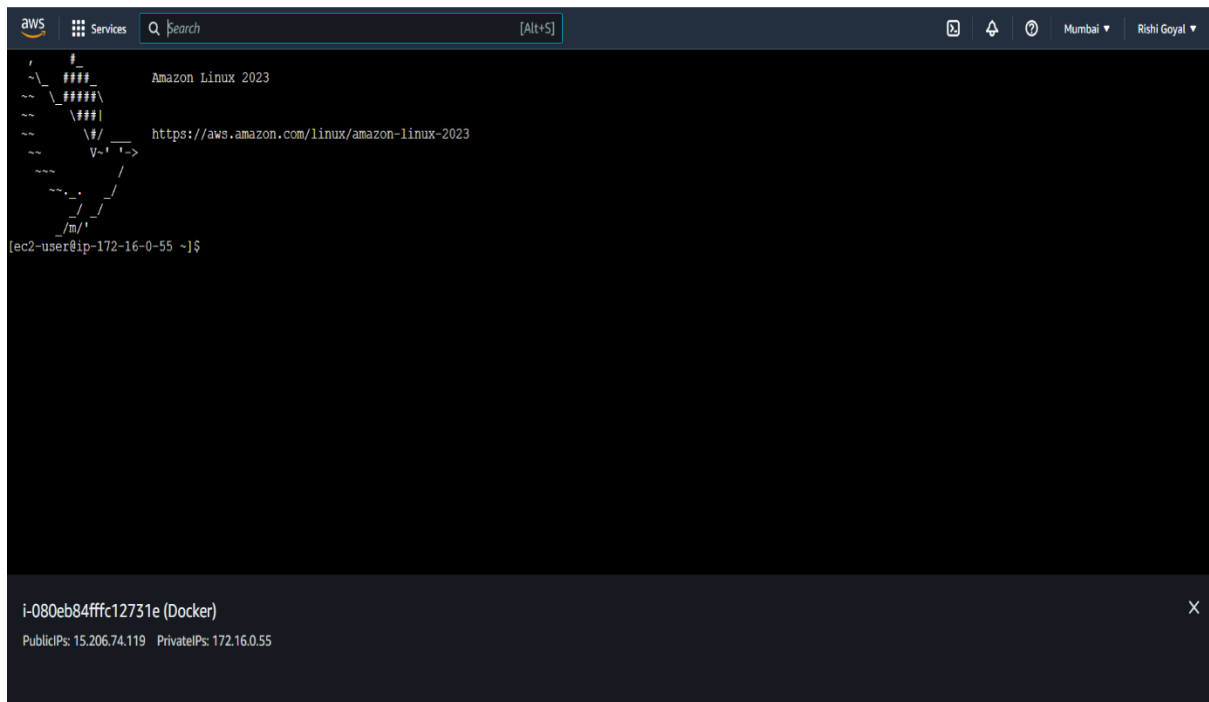
Storage (volumes)

1 volume(s) - 30 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier

Cancel Launch instance Review commands





Step 3 – All the necessary configurations were performed on the instances to ensure proper functionality and alignment with project requirements.



- Installed Git, Docker and Jenkins on instances according to the Requirements

```
aws Services Q Search [Alt+S]

[ec2-user@ip-172-16-0-136 ~]$ sudo yum install git -y
Last metadata expiration check: 0:10:17 ago on Sun May 14 09:33:38 2023.
Dependencies resolved.

=====
Package                                Architecture  Version
=====
Installing:
git                                    x86_64        2.39.2-1.amzn2023.0.1
Installing dependencies:
git-core                              x86_64        2.39.2-1.amzn2023.0.1
git-core-doc                          noarch        2.39.2-1.amzn2023.0.1
perl-Error                            noarch        1:0.17029-5.amzn2023.0.2
perl-File-Find                        noarch        1.37-477.amzn2023.0.3
perl-Git                              noarch        2.39.2-1.amzn2023.0.1
perl-TermReadKey                      x86_64        2.38-9.amzn2023.0.2
perl-lib                              x86_64        0.65-477.amzn2023.0.3
=====

Transaction Summary

Install 8 Packages

Total download size: 7.1 M
Installed size: 33 M
Downloading Packages:
(1/8): perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64.rpm
(2/8): perl-lib-0.65-477.amzn2023.0.3.x86_64.rpm
(3/8): git-2.39.2-1.amzn2023.0.1.x86_64.rpm
(4/8): perl-Error-0.17029-5.amzn2023.0.2.noarch.rpm

i-0b0f1516e84048f37 (Git)
PublicIPs: 35.154.149.44 PrivateIPs: 172.16.0.136
```

```
aws Services Q Search [Alt+S]

[ec2-user@ip-172-16-0-61 ~]$ sudo yum install git -y; sudo yum install docker -y
Last metadata expiration check: 0:05:19 ago on Sun May 14 09:39:08 2023.
Dependencies resolved.

=====
Package                                Architecture  Version
=====
Installing:
git                                    x86_64        2.39.2-1.amzn2023.0.1
Installing dependencies:
git-core                              x86_64        2.39.2-1.amzn2023.0.1
git-core-doc                          noarch        2.39.2-1.amzn2023.0.1
perl-Error                            noarch        1:0.17029-5.amzn2023.0.2
perl-File-Find                        noarch        1.37-477.amzn2023.0.3
perl-Git                              noarch        2.39.2-1.amzn2023.0.1
perl-TermReadKey                      x86_64        2.38-9.amzn2023.0.2
perl-lib                              x86_64        0.65-477.amzn2023.0.3
=====

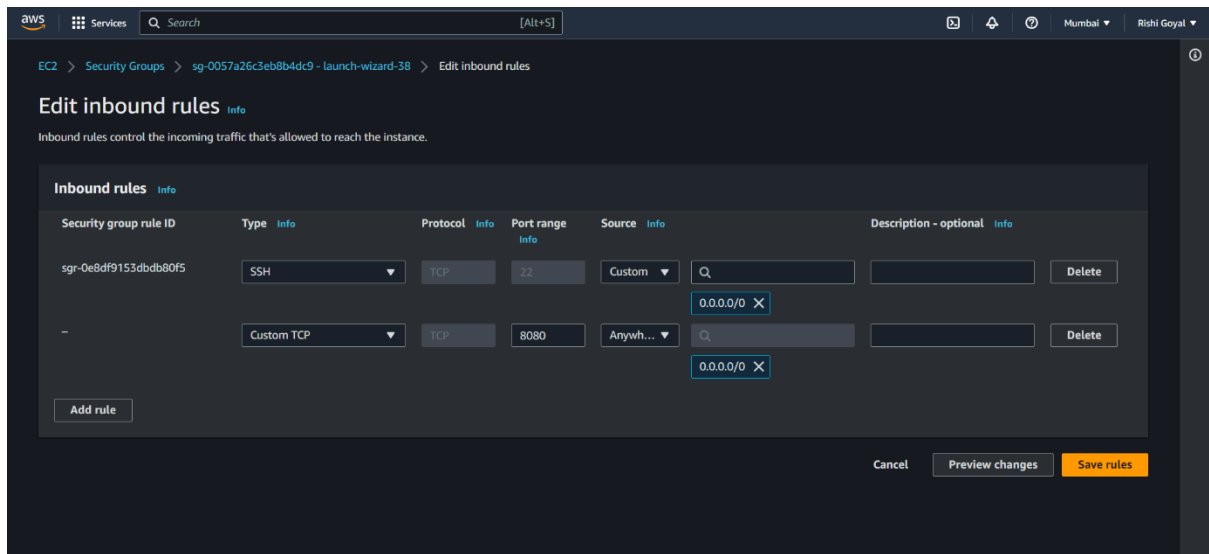
Transaction Summary

Install 8 Packages

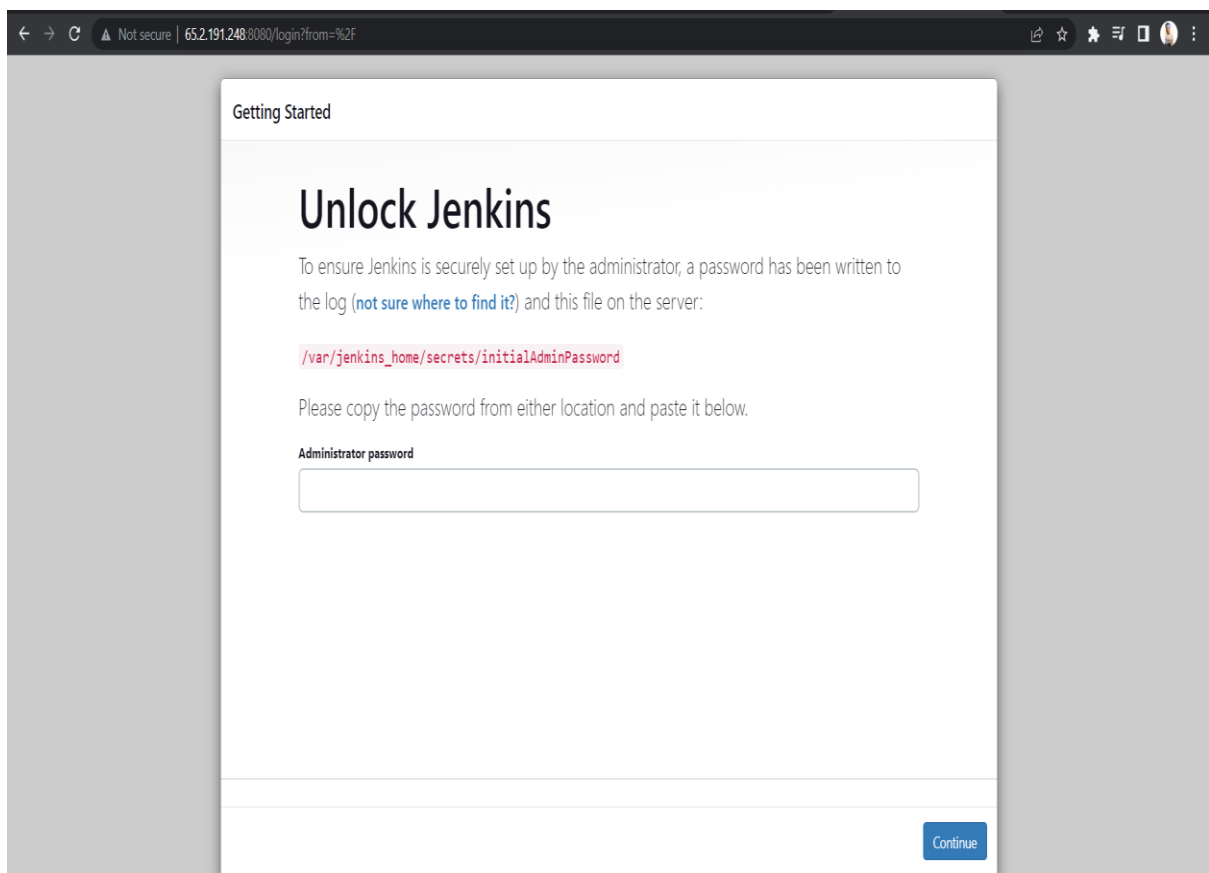
Total download size: 7.1 M
Installed size: 33 M
Downloading Packages:
(1/8): perl-lib-0.65-477.amzn2023.0.3.x86_64.rpm
(2/8): git-2.39.2-1.amzn2023.0.1.x86_64.rpm
(3/8): perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64.rpm
(4/8): perl-Error-0.17029-5.amzn2023.0.2.noarch.rpm

i-0ccdf0956e8f6656e (Jenkins)
PublicIPs: 65.2.191.248 PrivateIPs: 172.16.0.61
```

- Add the inbound rule on Jenkins instance



- Reach out the Jenkins webpage



- A central repository was created on GitHub, and the webpage was pushed to the repository.

```
[ec2-user@ip-172-16-0-136 ~]$ ls
[ec2-user@ip-172-16-0-136 ~]$ mkdir practical
[ec2-user@ip-172-16-0-136 ~]$ cd pra
-bash: cd: pra: No such file or directory
[ec2-user@ip-172-16-0-136 ~]$ cd practical/
[ec2-user@ip-172-16-0-136 practical]$ ls
[ec2-user@ip-172-16-0-136 practical]$ git clone https://github.com/rishigoyl/jenkins_lab.git
Cloning into 'jenkins_lab'...
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 15 (delta 0), reused 11 (delta 0), pack-reused 0
Receiving objects: 100% (15/15), 4.15 KiB | 4.15 MiB/s, done.
[ec2-user@ip-172-16-0-136 practical]$ ls
jenkins_lab
[ec2-user@ip-172-16-0-136 practical]$ cd jenkins_lab/
[ec2-user@ip-172-16-0-136 jenkins_lab]$
```

```
aws Services Search [Alt+S]
[ec2-user@ip-172-16-0-136 jenkins_lab]$ git commit -m "new commit"
[main f0a8985] new commit
Committer: EC2 Default User <ec2-user@ip-172-16-0-136.ap-south-1.compute.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

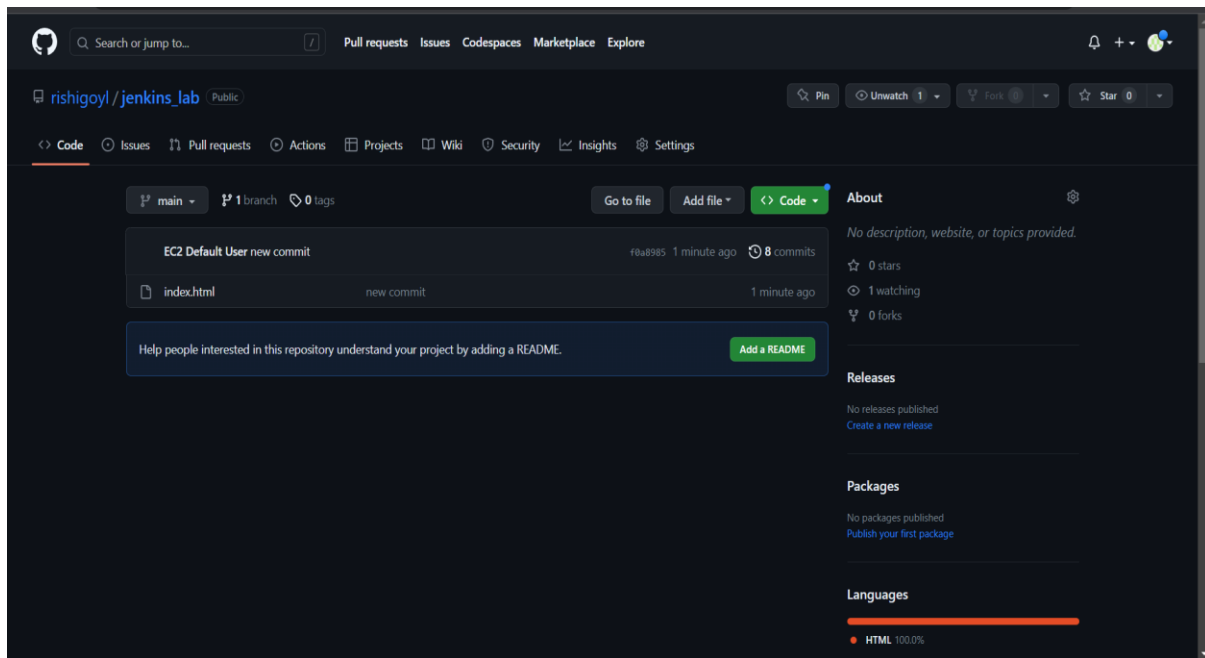
After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 14 insertions(+)
create mode 100644 index.html
[ec2-user@ip-172-16-0-136 jenkins_lab]$ git push https://github.com/rishigoyl/jenkins_lab.git
Username for 'https://github.com': rishigoyl
Password for 'https://rishigoyl@github.com':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 86.42 KiB | 10.80 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/rishigoyl/jenkins_lab.git
   3bc57ac..f0a8985  main -> main
[ec2-user@ip-172-16-0-136 jenkins_lab]$
```

i-0b0f1516e84048f37 (Git)

PublicIPs: 35.154.149.44 PrivateIPs: 172.16.0.136



- The configuration files on both the Docker and Jenkins instances were edited to establish seamless communication and enable effective coordination between the two instances.

```
#PubkeyAuthentication yes

# The default is to check both .ssh/authorized_keys and .ssh/authorized_keys2
# but this is overridden so installations will only check .ssh/authorized_keys
AuthorizedKeysFile .ssh/authorized_keys

#AuthorizedPrincipalsFile none

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# Explicitly disable PasswordAuthentication. By presetting it, we
# avoid the cloud-init set_passwords module modifying sshd_config and
# restarting sshd in the default instance launch configuration.
PasswordAuthentication yes
PermitEmptyPasswords no

# Change to no to disable s/key passwords
#KbdInteractiveAuthentication yes

# Kerberos options
-- INSERT --
```


i-080eb84fffc12731e (Docker)

PublicIPs: 15.206.74.119 PrivateIPs: 172.16.0.55

```
aws Services Search [Alt+S]
80 Defaults    env_keep += "LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET XAUTHORITY"
81
82 #
83 # Adding HOME to env_keep may enable a user to run unrestricted
84 # commands via sudo.
85 #
86 # Defaults    env_keep += "HOME"
87
88 Defaults     secure_path = /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/var/lib/snapd/snap/bin
89
90 ## Next comes the main part: which users can run what software on
91 ## which machines (the sudoers file can be shared between multiple
92 ## systems).
93 ## Syntax:
94 ##
95 ##     user    MACHINE=COMMANDS
96 ##
97 ## The COMMANDS section may have other options added to it.
98 ##
99 ## Allow root to run any commands anywhere
100 root    ALL=(ALL)    ALL
101 jenkins ALL=(ALL)    NOPASSWD:ALL
102 ## Allows members of the 'sys' group to run networking, software,
103 ## service management apps and more.
104 # %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCATE, DRIVERS
105
106 ## Allows people in group wheel to run all commands
-- INSERT -- W10: Warning: Changing a readonly file

i-0ccdf0956e8f6656e (Jenkins)
PublicIPs: 65.2.191.248 PrivateIPs: 172.16.0.61
```

- A freestyle project was created on Jenkins and linked to the GitHub repository

 **Jenkins** jenkins log out

Dashboard >

+ New Item

People

Build History

Manage Jenkins

My Views

All

+

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀	lab1	16 sec #1	N/A	4.5 sec

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

Icon: S M L

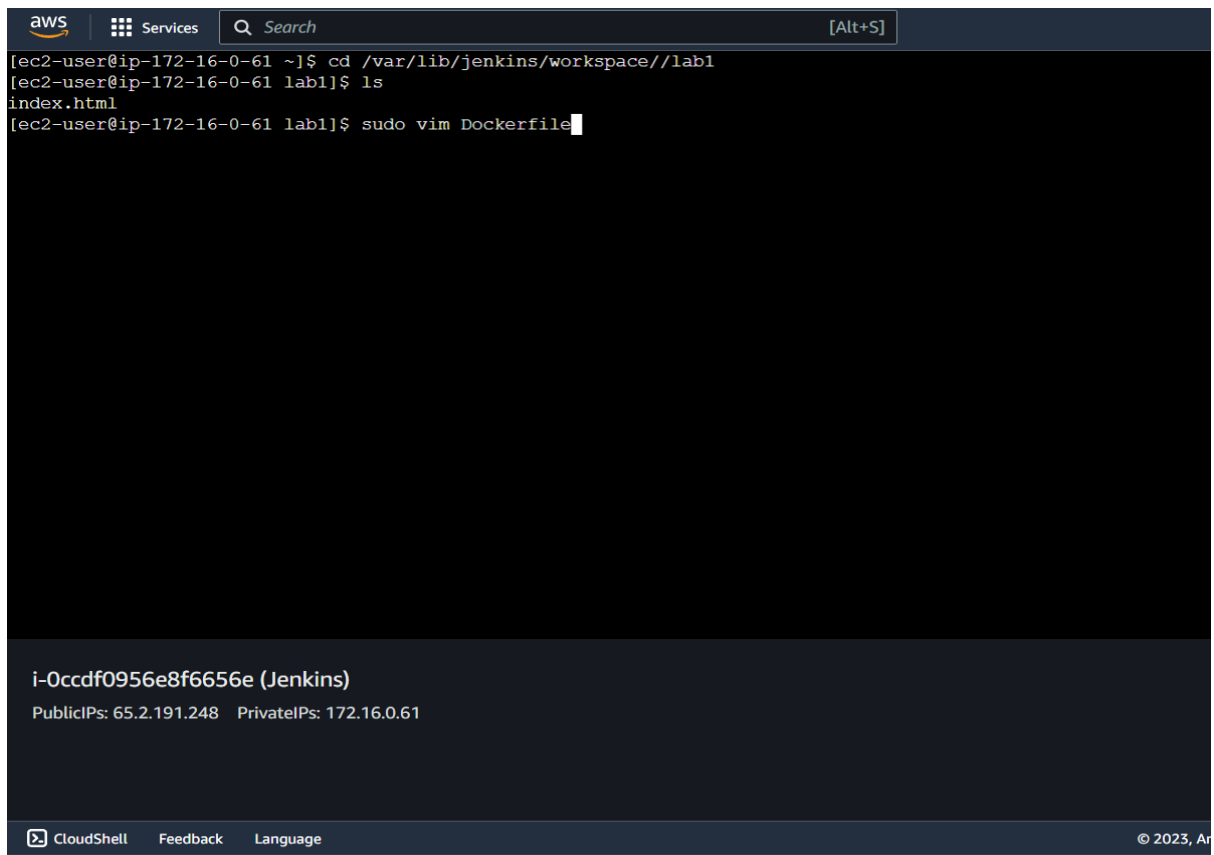
Icon legend

Atom feed for all

Atom feed for failures

Atom feed for just latest builds

- A Dockerfile was created on the Jenkins instance, and subsequently, another freestyle project was set up and linked to the first project



The screenshot shows an AWS CloudShell terminal session. At the top, there's a header with the AWS logo, 'Services', a search bar, and '[Alt+S]'. The terminal output shows the user navigating to the Jenkins workspace directory, listing files (index.html), and then opening a Dockerfile in the vim editor. The terminal prompt is '[ec2-user@ip-172-16-0-61 lab1]\$'.

```
[ec2-user@ip-172-16-0-61 ~]$ cd /var/lib/jenkins/workspace//lab1
[ec2-user@ip-172-16-0-61 lab1]$ ls
index.html
[ec2-user@ip-172-16-0-61 lab1]$ sudo vim Dockerfile
```

Below the terminal window, the instance details are shown: 'i-0ccdf0956e8f6656e (Jenkins)' with PublicIPs: 65.2.191.248 and PrivateIPs: 172.16.0.61. The footer includes 'CloudShell', 'Feedback', 'Language', and '© 2023, Ar'.

Build Steps

≡ Execute shell ?

Command

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

```
cd /var/lib/jenkins/workspace/lab1
sudo docker build -t myimage:v1 .
```

Advanced ▾

Add build step ▾

Post-build Actions

Add post-build action ▾

Search (CTRL+K)

jenkins

log out

Dashboard

+ New Item

People

Build History

Manage Jenkins

My Views

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

All

+

S	W	Name	Last Success	Last Failure	Last Duration
		lab1	4 min 8 sec #2	N/A	0.67 sec
		lab2	19 sec #1	N/A	11 sec

Icon: S M L

Icon legend

Atom feed for all

Atom feed for failures

Atom feed for just latest builds

Add description

- A third freestyle project was created to create a Docker container and it was linked to the second build

☐ With Ant

Build Steps

Execute shell

Command

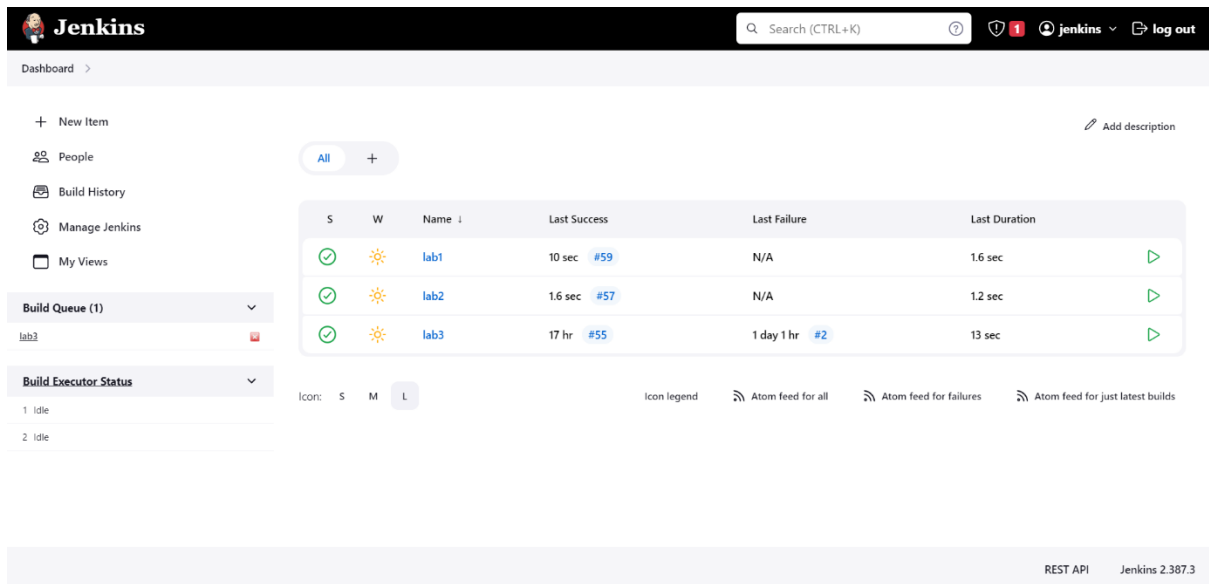
See [the list of available environment variables](#)

```

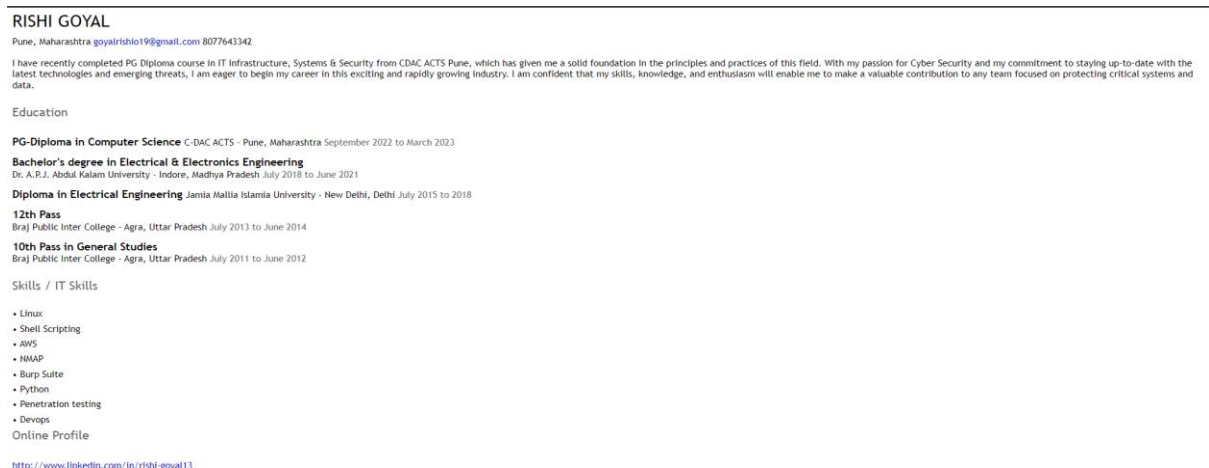
sudo docker save myimage:v1 > mypage.tar
scp mypage.tar ec2-user@172.16.0.55:/home/ec2-user
ssh ec2-user@172.16.0.55 "sudo docker load < mypage.tar"
ssh ec2-user@172.16.0.55 "sudo docker rm -f resume"
ssh ec2-user@172.16.0.55 "sudo docker run --name resume -p 8000:80 -d myimage:v1"
          
```

Advanced

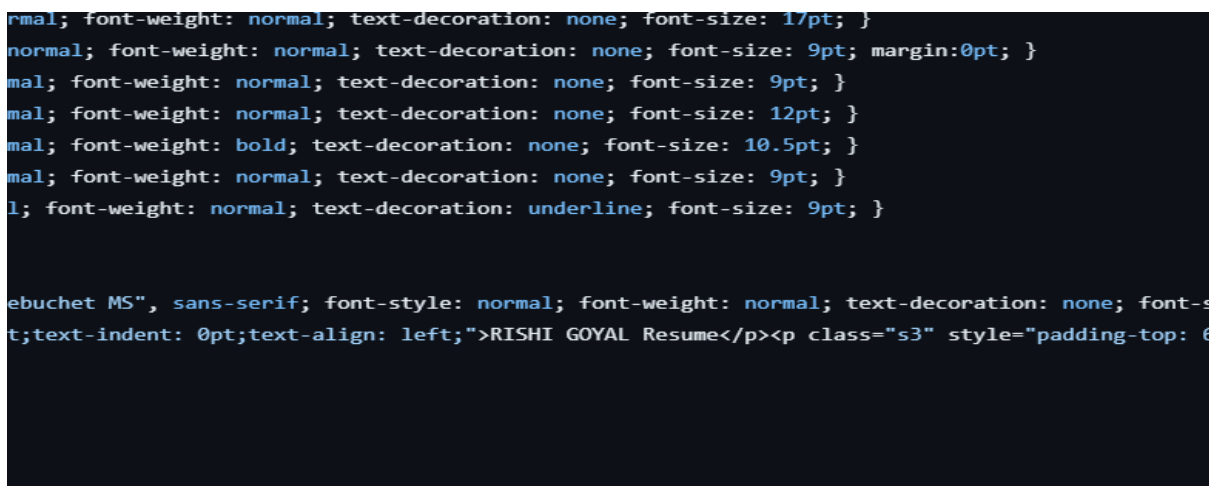
Add build step



- The webpage is now functioning seamlessly and can be accessed publicly without any issues



- Modifications have been made to the HTML code



- The webpage has been successfully updated with the modified content, reflecting the changes made to the HTML code

RISHI GOYAL Resume

Pune, Maharashtra goyalrishi19@gmail.com 8077643342

I have recently completed PG Diploma course in IT Infrastructure, Systems & Security from CDAC ACTS Pune, which has given me a solid foundation in the principles and practices of this field. With my passion for Cyber Security and my commitment to staying up-to-date with the latest technologies and emerging threats, I am eager to begin my career in this exciting and rapidly growing industry. I am confident that my skills, knowledge, and enthusiasm will enable me to make a valuable contribution to any team focused on protecting critical systems and data.

Education

PG-Diploma in Computer Science C-DAC ACTS - Pune, Maharashtra September 2022 to March 2023

Bachelor's degree in Electrical & Electronics Engineering

Dr. A.P.J. Abdul Kalam University - Indore, Madhya Pradesh July 2018 to June 2021

Diploma in Electrical Engineering Jamia Millia Islamia University - New Delhi, Delhi July 2015 to 2018

12th Pass

Brar Public Inter College - Agra, Uttar Pradesh July 2013 to June 2014

10th Pass in General Studies

Brar Public Inter College - Agra, Uttar Pradesh July 2011 to June 2012

Skills / IT Skills

- Linux
- Shell Scripting
- AWS
- NMAP
- Burp Suite
- Python
- Penetration testing
- Devops

Online Profile

<http://www.linkedin.com/in/rishi-goyal13>