Question/Assignment 1

1. Kill all processes/zombie processes of service called “gunicorn” in a single command.

Command:-

```
pkill -9 -f gunicorn
```

Or

```
ps aux | grep gunicorn | awk '{print $2}' | xargs kill –9
```

2. MySQL shell command to show the unique IPs from where MySQL connections are being made to the Database.

Command :-

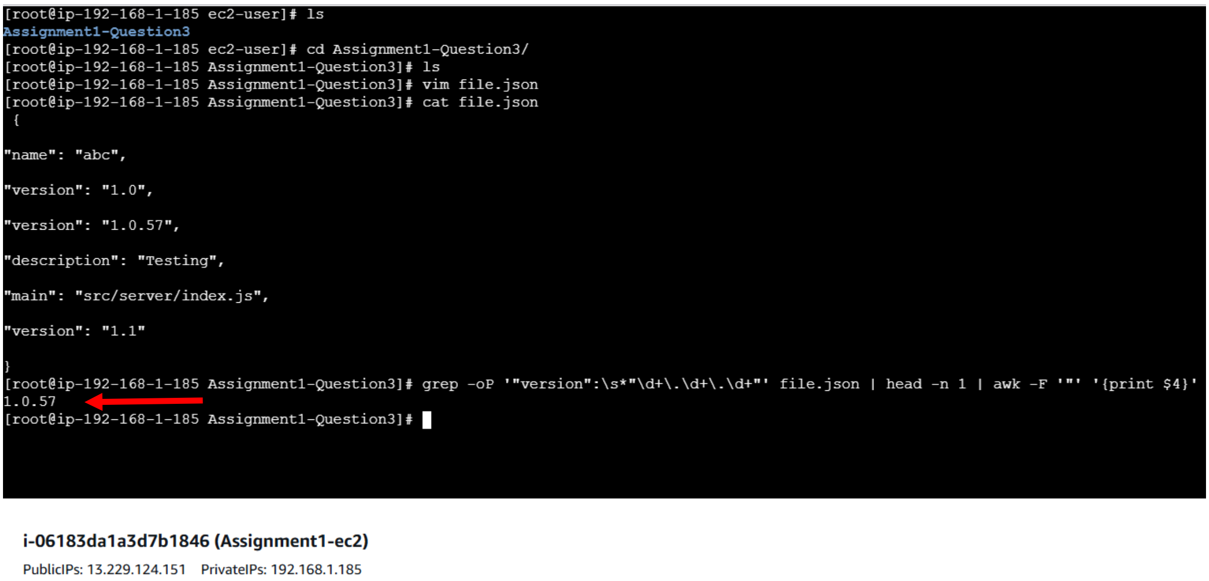
```
SELECT DISTINCT HOST FROM information_schema.processlist;
```

3. Bash command to get value of version number of 3 decimal points (first occurrence) from a file containing the JSON:

```
 {
"name": "abc",
"version": "1.0",
"version": "1.0.57",
"description": "Testing",
"main": "src/server/index.js",
"version": "1.1"
}
```

Command :-

```
grep -oP '"version":\s*"\d+\.\d+\.\d+"' file.json | head -n 1 | awk -F '"' '{print $4}'
```



```
[root@ip-192-168-1-185 ec2-user]# ls
Assignment1-Question3
[root@ip-192-168-1-185 ec2-user]# cd Assignment1-Question3/
[root@ip-192-168-1-185 Assignment1-Question3]# ls
[root@ip-192-168-1-185 Assignment1-Question3]# vim file.json
[root@ip-192-168-1-185 Assignment1-Question3]# cat file.json
 {

"name": "abc",

"version": "1.0",

"version": "1.0.57",

"description": "Testing",

"main": "src/server/index.js",

"version": "1.1"

}
[root@ip-192-168-1-185 Assignment1-Question3]# grep -oP '"version":\s*"\d+\.\d+\.\d+"' file.json | head -n 1 | awk -F '"' '{print $4}'
1.0.57
[root@ip-192-168-1-185 Assignment1-Question3]#
```

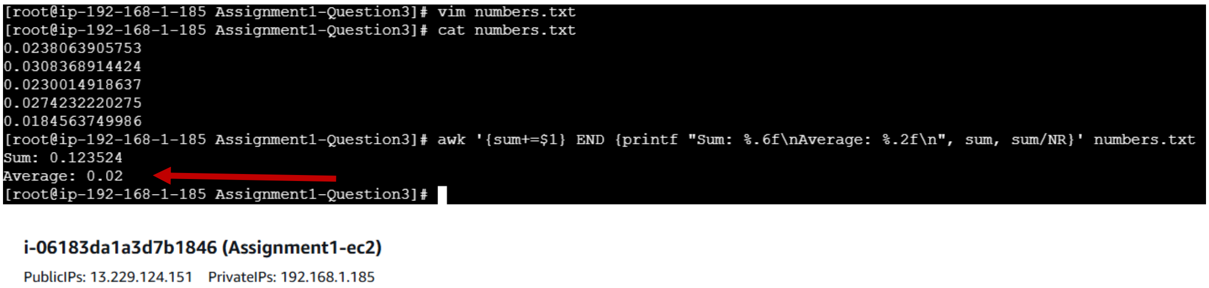
i-06183da1a3d7b1846 (Assignment1-ec2)
PublicIPs: 13.229.124.151   PrivateIPs: 192.168.1.185

4. Bash command to add these numbers from a file and find average upto 2 decimal points:

0.0238063905753 0.0308368914424 0.0230014918637 0.0274232220275 0.0184563749986

Command :-   awk '{sum += $1} END {printf "%.2f\n", sum / NR}' file.txt

Output:-



```
[root@ip-192-168-1-185 Assignment1-Question3]# vim numbers.txt
[root@ip-192-168-1-185 Assignment1-Question3]# cat numbers.txt
0.0238063905753
0.0308368914424
0.0230014918637
0.0274232220275
0.0184563749986
[root@ip-192-168-1-185 Assignment1-Question3]# awk '{sum+=$1} END {printf "Sum: %.6f\nAverage: %.2f\n", sum, sum/NR}' numbers.txt
Sum: 0.123524
Average: 0.02
[root@ip-192-168-1-185 Assignment1-Question3]#
```

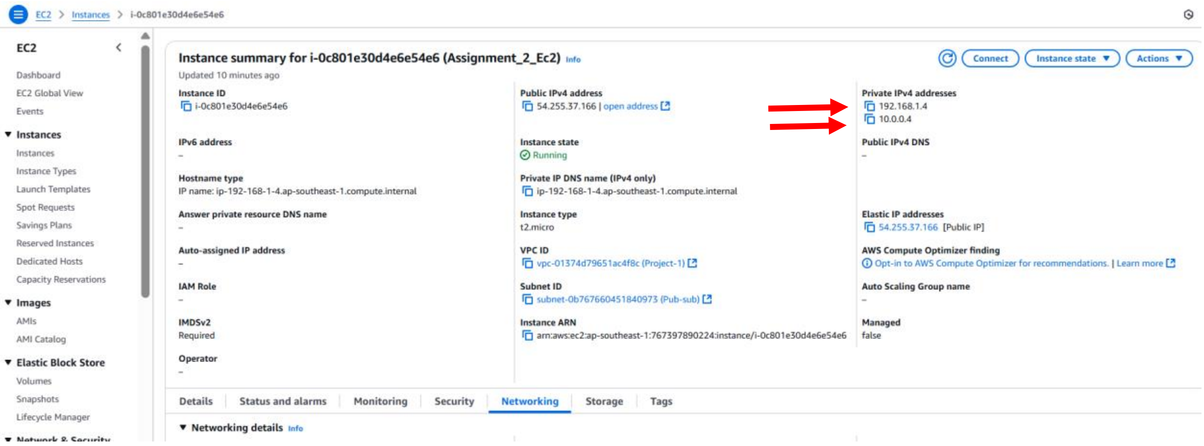
i-06183da1a3d7b1846 (Assignment1-ec2)
PublicIPs: 13.229.124.151   PrivateIPs: 192.168.1.185

Question/Assignment 2

Create a Virtual Machine:
● Set up a VM (e.g., using VirtualBox, VMware, or a cloud provider like AWS, GCP, or Azure).
● The VM should have at least two network interfaces: eth0 (for the default network) and eth1 (for a second network).

Solution :- I have deployed an EC2 instance in AWS with **two network interfaces** (ENIs) assigned from different CIDR blocks (`192.168.1.0/24` and `10.0.0.0/24`), enabling multi-VPC connectivity **through AWS Transit Gateway.** Additionally, I have assigned an Elastic IP as a **public IPv4 address** for **internet connectivity**.

Please Find the Screenshot for your reference--



Instance summary for i-0c801e30d4e6e54e6 (Assignment_2_Ec2) Info

Updated 10 minutes ago

Instance ID
i-0c801e30d4e6e54e6

IPv6 address
–

Hostname type
IP name: ip-192-168-1-4.ap-southeast-1.compute.internal

Answer private resource DNS name
–

Auto-assigned IP address
–

IAM Role
–

IMDSv2
Required

Operator
–

Public IPv4 address
54.255.37.166 | open address

Instance state
Running

Private IP DNS name (IPv4 only)
ip-192-168-1-4.ap-southeast-1.compute.internal

Instance type
t2.micro

VPC ID
vpc-01374d79651ac4f8c (Project-1)

Subnet ID
subnet-0b767660451840973 (Pub-sub)

Instance ARN
arn:aws:ec2:ap-southeast-1:767397890224:instance/i-0c801e30d4e6e54e6

Private IPv4 addresses
192.168.1.4
10.0.0.4

Public IPv4 DNS
–

Elastic IP addresses
54.255.37.166 [Public IP]

AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. | Learn more

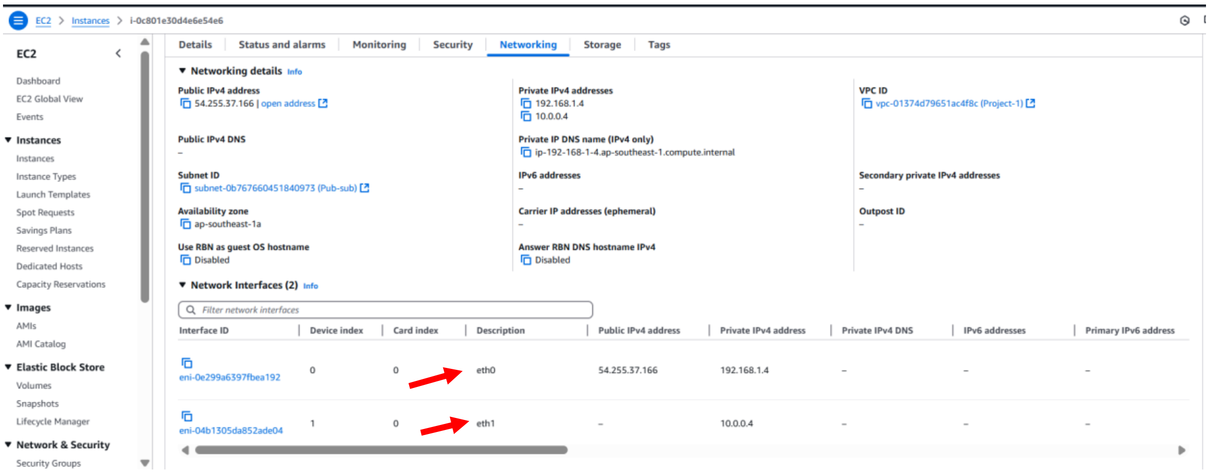
Auto Scaling Group name
–

Managed
false

▼ Networking details Info

Configure IP Addressing:

you can view the Network Interfaces where I have assigned the first network interface (**eth0**) with an IP address **192.168.1.4 in the range of 192.168.1.0/24** & assigned the second network interface **(eth1)** with an IP address **10.0.0.4 in the range of 10.0.0.0/24 as instructed.**



Networking details

Public IPv4 address
54.255.37.166 | open address

Public IPv4 DNS
–

Subnet ID
subnet-0b767660451840973 (Pub-sub)

Availability zone
ap-southeast-1a

Use RBN as guest OS hostname
Disabled

Private IPv4 addresses
192.168.1.4
10.0.0.4

Private IP DNS name (IPv4 only)
ip-192-168-1-4.ap-southeast-1.compute.internal

IPv6 addresses
–

Carrier IP addresses (ephemeral)
–

Answer RBN DNS hostname IPv4
Disabled

VPC ID
vpc-01374d79651ac4f8c (Project-1)

Secondary private IPv4 addresses
–

Outpost ID
–

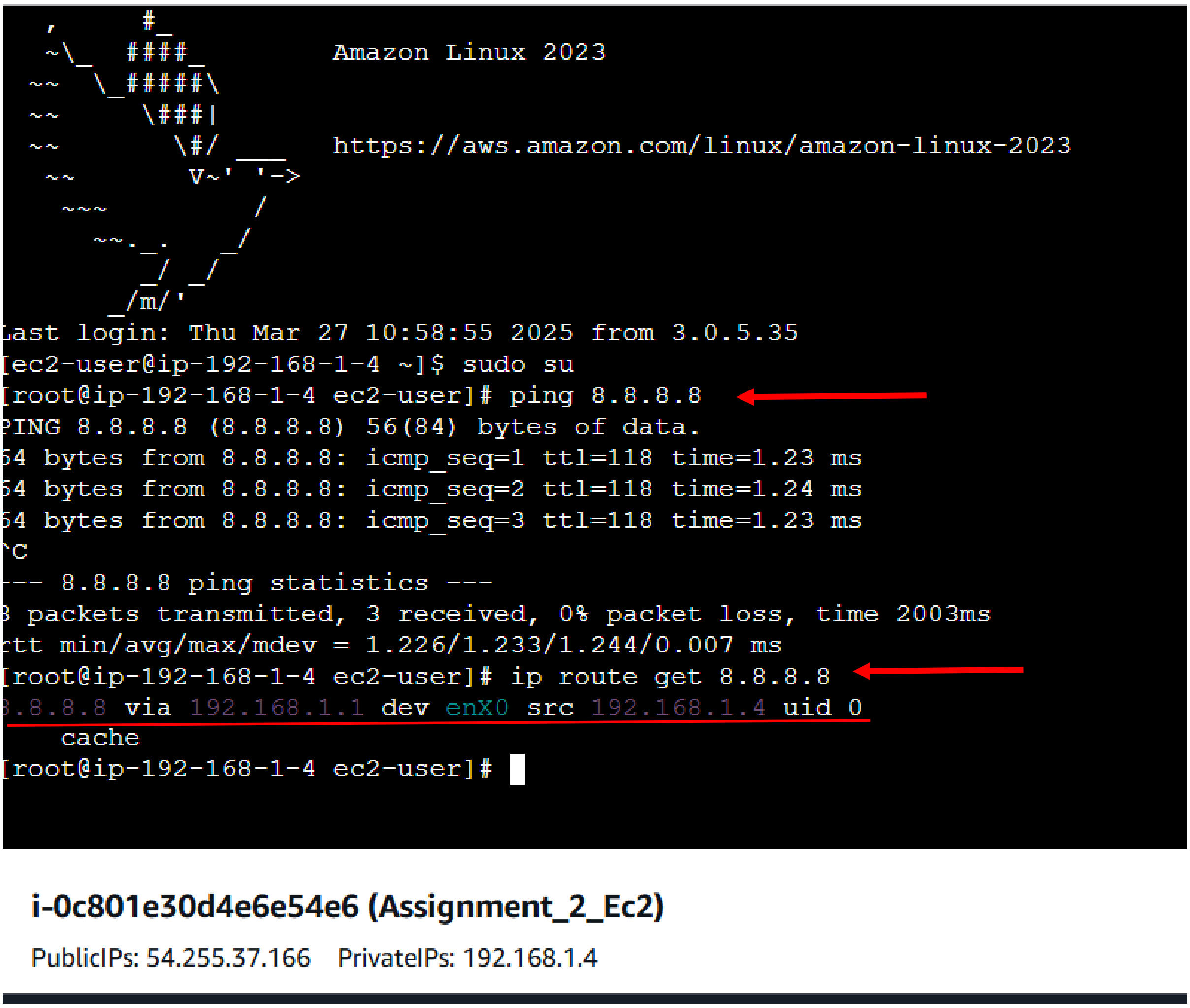
Network Interfaces (2)

Interface ID	Device index	Card index	Description	Public IPv4 address	Private IPv4 address	Private IPv4 DNS	IPv6 addresses	Primary IPv6 address
eni-0e299a6397fbea192	0	0	eth0	54.255.37.166	192.168.1.4	–	–	–
eni-04b1305da852ade04	1	0	eth1	–	10.0.0.4	–	–	–

Please Find Below--

Network Routing Setup & Testing the Network Configuration:

You can see that the **instance can access the external website (google.com) via the `eth0` interface**, as I have previously mentioned.



```
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Thu Mar 27 10:58:55 2025 from 3.0.5.35
[ec2-user@ip-192-168-1-4 ~]$ sudo su
[root@ip-192-168-1-4 ec2-user]# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=118 time=1.23 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=118 time=1.24 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=118 time=1.23 ms
^C
--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 1.226/1.233/1.244/0.007 ms
[root@ip-192-168-1-4 ec2-user]# ip route get 8.8.8.8
8.8.8.8 via 192.168.1.1 dev enX0 src 192.168.1.4 uid 0
    cache
[root@ip-192-168-1-4 ec2-user]#
```

i-0c801e30d4e6e54e6 (Assignment_2_Ec2)

PublicIPs: 54.255.37.166   PrivateIPs: 192.168.1.4

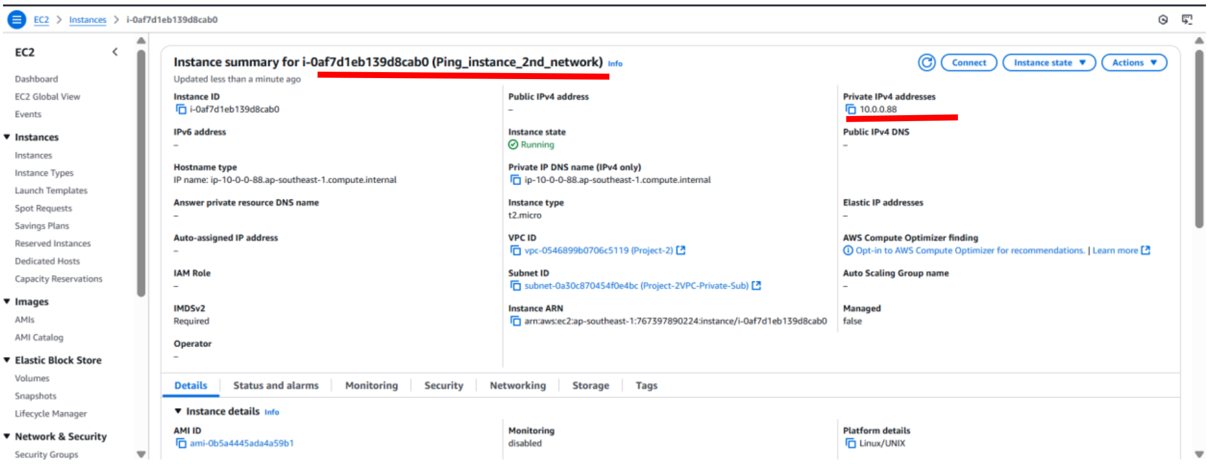
- Set up a static route on the EC2 to route traffic destined for 10.0.0.0/24 through the eth1 interface.

Command :-   ip route add 10.0.0.0/24 dev eth1

- Test the connectivity by pinging other machines on the same network.

(attached screenshot)

You can see another EC2 instance (Host) from the **10.0.0.0/24** network with the IP **10.0.0.88** in the screenshot.



Instance summary for i-0af7d1eb139d8cab0 (Ping_instance_2nd_network) Info

Updated less than a minute ago

Instance ID
i-0af7d1eb139d8cab0

IPv6 address
–

Hostname type
IP name: ip-10-0-0-88.ap-southeast-1.compute.internal

Answer private resource DNS name
–

Auto-assigned IP address
–

IAM Role
–

IMDSv2
Required

Operator
–

Public IPv4 address
–

Instance state
Running

Private IP DNS name (IPv4 only)
ip-10-0-0-88.ap-southeast-1.compute.internal

Instance type
t2.micro

VPC ID
vpc-0546899b0706c5119 (Project-2)

Subnet ID
subnet-0a30c870454f0e4bc (Project-2VPC-Private-Sub)

Instance ARN
arn:aws:ec2:ap-southeast-1:767397890224:instance/i-0af7d1eb139d8cab0

Private IPv4 addresses
10.0.0.88

Public IPv4 DNS
–

Elastic IP addresses
–

AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. | Learn more

Auto Scaling Group name
–

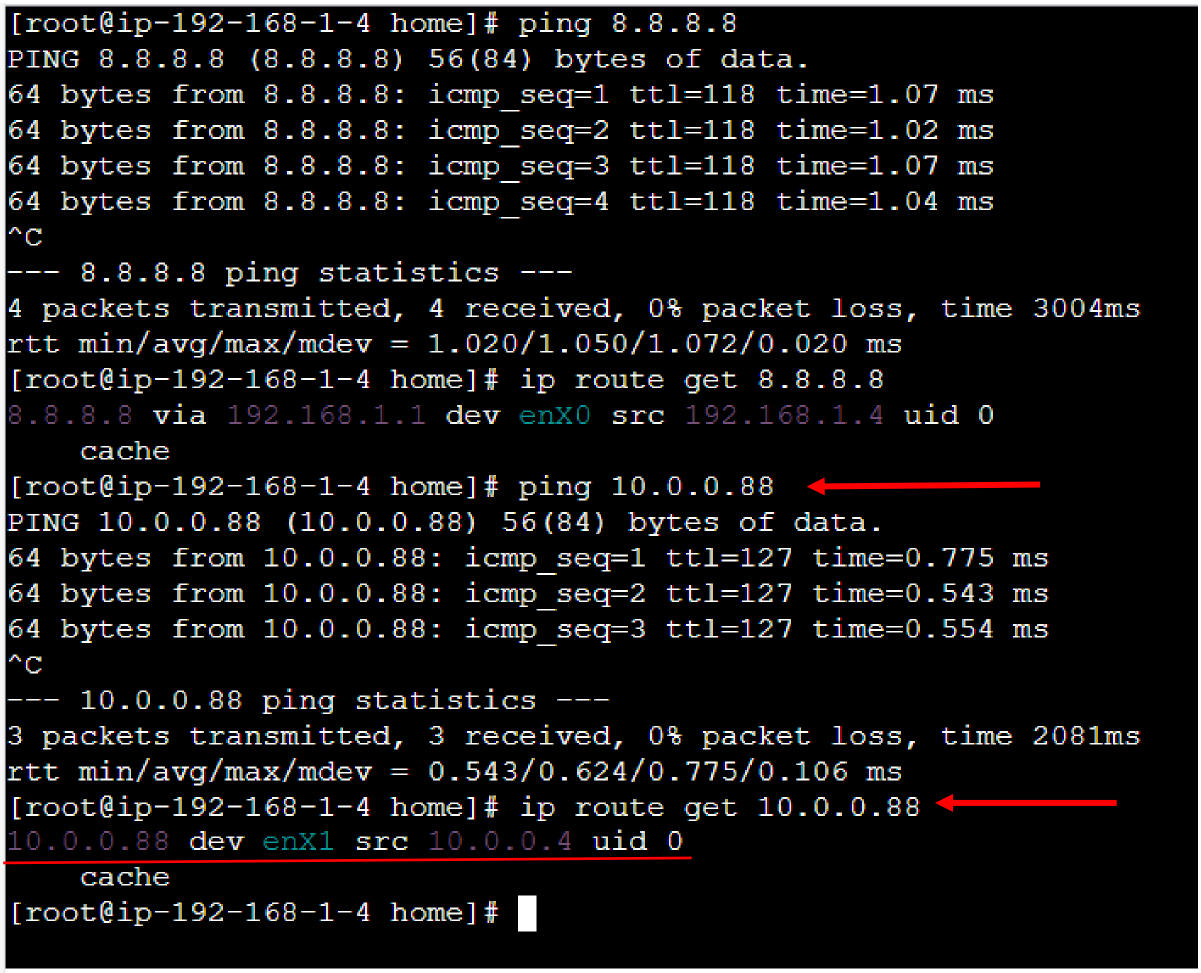
Managed
false

Instance details Info

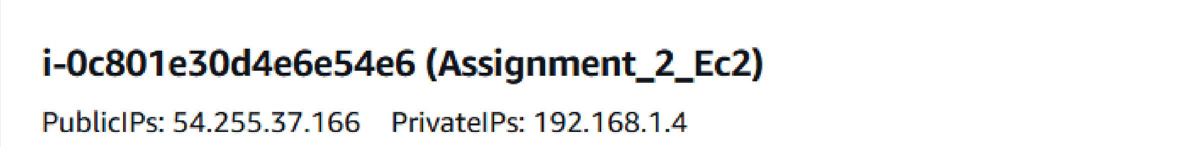
AMI ID
ami-0b5a4445ada4a59b1

Monitoring
disabled

Platform details
Linux/UNIX



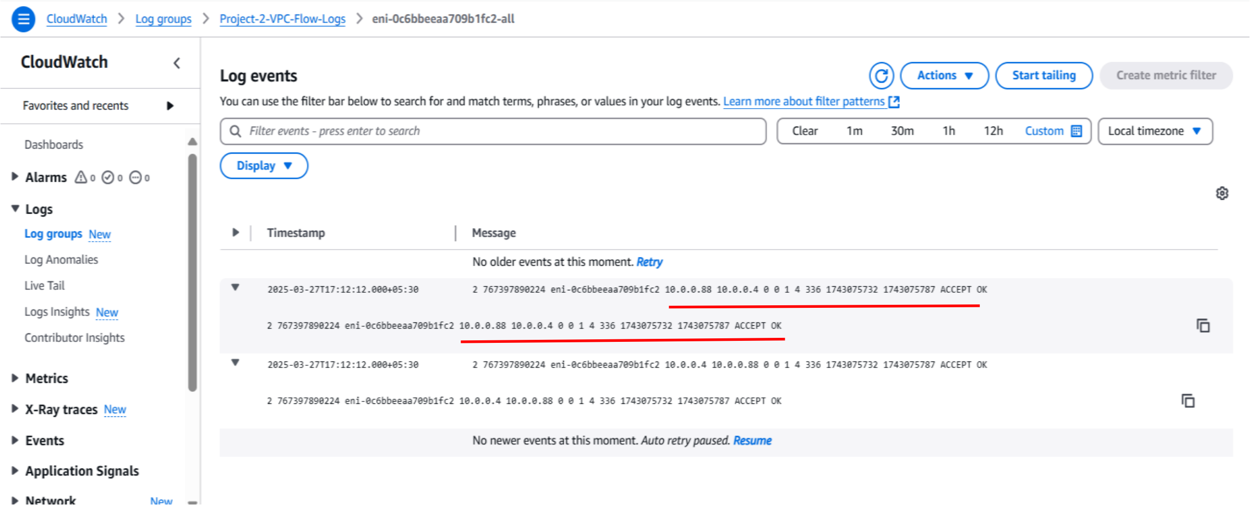
```
[root@ip-192-168-1-4 home]# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=118 time=1.07 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=118 time=1.02 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=118 time=1.07 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=118 time=1.04 ms
^C
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 1.020/1.050/1.072/0.020 ms
[root@ip-192-168-1-4 home]# ip route get 8.8.8.8
8.8.8.8 via 192.168.1.1 dev enX0 src 192.168.1.4 uid 0
    cache
[root@ip-192-168-1-4 home]# ping 10.0.0.88
PING 10.0.0.88 (10.0.0.88) 56(84) bytes of data.
64 bytes from 10.0.0.88: icmp_seq=1 ttl=127 time=0.775 ms
64 bytes from 10.0.0.88: icmp_seq=2 ttl=127 time=0.543 ms
64 bytes from 10.0.0.88: icmp_seq=3 ttl=127 time=0.554 ms
^C
--- 10.0.0.88 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2081ms
rtt min/avg/max/mdev = 0.543/0.624/0.775/0.106 ms
[root@ip-192-168-1-4 home]# ip route get 10.0.0.88
10.0.0.88 dev enX1 src 10.0.0.4 uid 0
    cache
[root@ip-192-168-1-4 home]#
```



i-0c801e30d4e6e54e6 (Assignment_2_Ec2)

PublicIPs: 54.255.37.166   PrivateIPs: 192.168.1.4

I have also attached a screenshot of the **VPC Flow Logs** for your reference. Please review it.



You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns

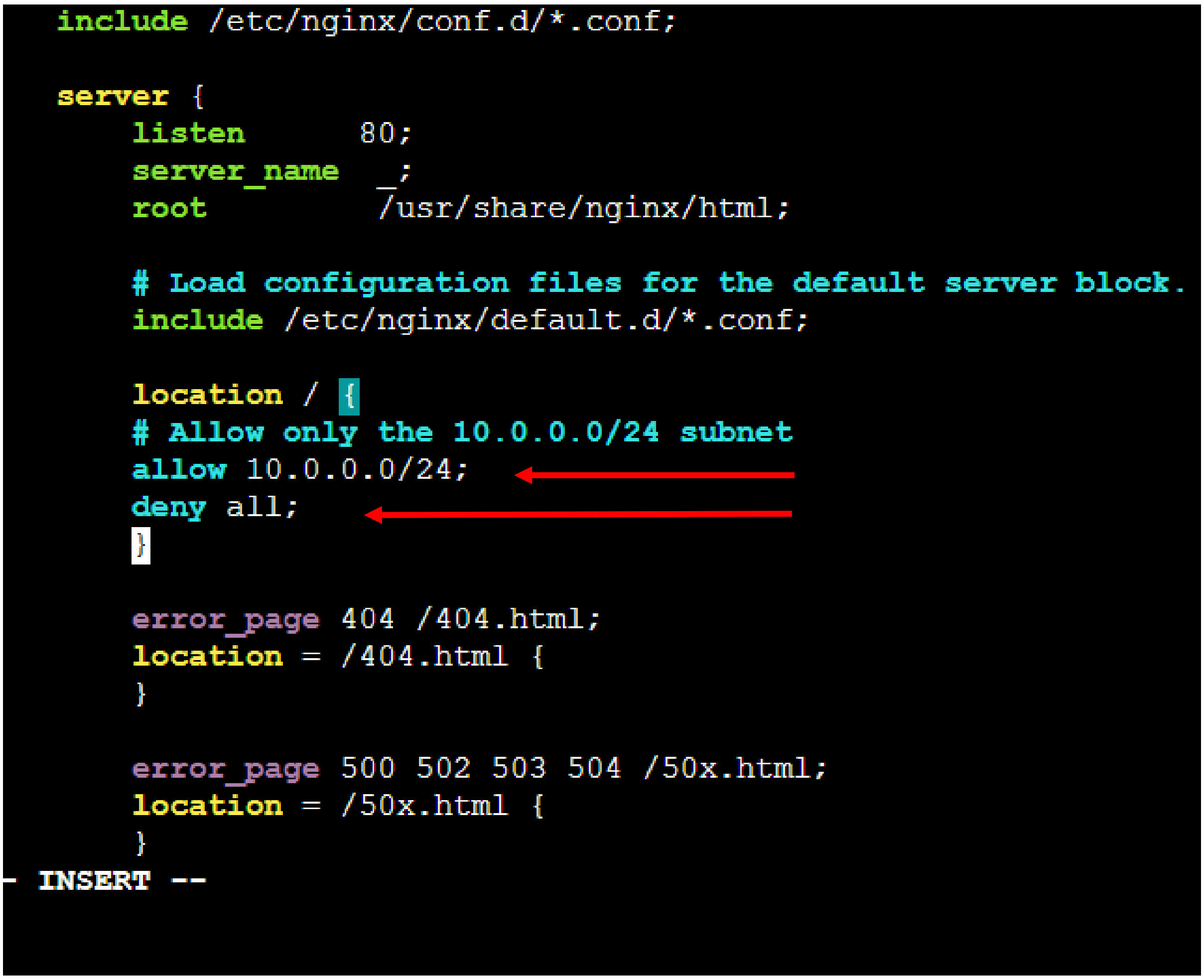
```
Timestamp                        Message
                                 No older events at this moment. Retry
2025-03-27T17:12:12.000+05:30    2 767397890224 eni-0c6bbeeaa709b1fc2 10.0.0.88 10.0.0.4 0 0 1 4 336 1743075732 1743075787 ACCEPT OK
2 767397890224 eni-0c6bbeeaa709b1fc2 10.0.0.88 10.0.0.4 0 0 1 4 336 1743075732 1743075787 ACCEPT OK
2025-03-27T17:12:12.000+05:30    2 767397890224 eni-0c6bbeeaa709b1fc2 10.0.0.4 10.0.0.88 0 0 1 4 336 1743075732 1743075787 ACCEPT OK
2 767397890224 eni-0c6bbeeaa709b1fc2 10.0.0.4 10.0.0.88 0 0 1 4 336 1743075732 1743075787 ACCEPT OK
                                 No newer events at this moment. Auto retry paused. Resume
```

- Set up a simple web server (Apache) on the Ec2, and configure the server to only be accessible via the 10.0.0.0/24 network interface. Ensure that the server cannot be accessed through the eth0 interface.

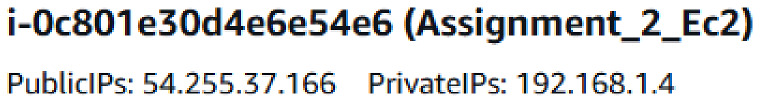
Solution :-

I have updated the **nginx.conf** configuration file to restrict access exclusively to my private IP assigned to the **eth1** network interface, allowing access only from the **10.0.0.0/24** network range on the default **port 80**. All **other access** attempts will be

denied.



```
   include /etc/nginx/conf.d/*.conf;

   server {
       listen       80;
       server_name  _;
       root         /usr/share/nginx/html;

       # Load configuration files for the default server block.
       include /etc/nginx/default.d/*.conf;

       location / {
       # Allow only the 10.0.0.0/24 subnet
       allow 10.0.0.0/24;
       deny all;
       }

       error_page 404 /404.html;
       location = /404.html {
       }

       error_page 500 502 503 504 /50x.html;
       location = /50x.html {
       }
- INSERT --
```

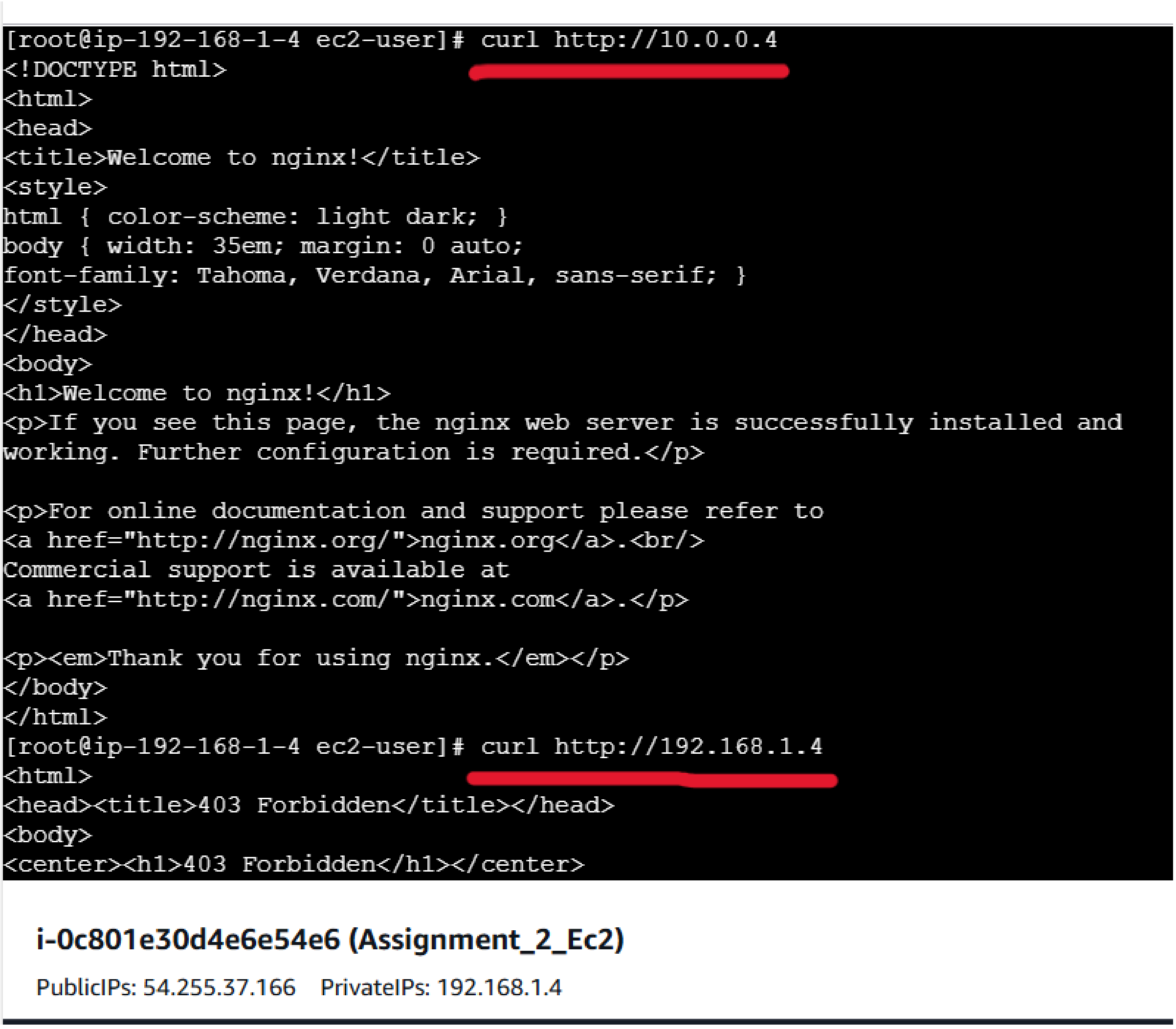


i-0c801e30d4e6e54e6 (Assignment_2_Ec2)

PublicIPs: 54.255.37.166   PrivateIPs: 192.168.1.4

Please find below the screenshot showing --

access **allowed** from the IP of the **eth1** network interface and **denied** from the IP of the **eth0** network interface.



```
[root@ip-192-168-1-4 ec2-user]# curl http://10.0.0.4
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
[root@ip-192-168-1-4 ec2-user]# curl http://192.168.1.4
<html>
<head><title>403 Forbidden</title></head>
<body>
<center><h1>403 Forbidden</h1></center>
```

i-0c801e30d4e6e54e6 (Assignment_2_Ec2)

PublicIPs: 54.255.37.166 PrivateIPs: 192.168.1.4

Question/Assignment 3

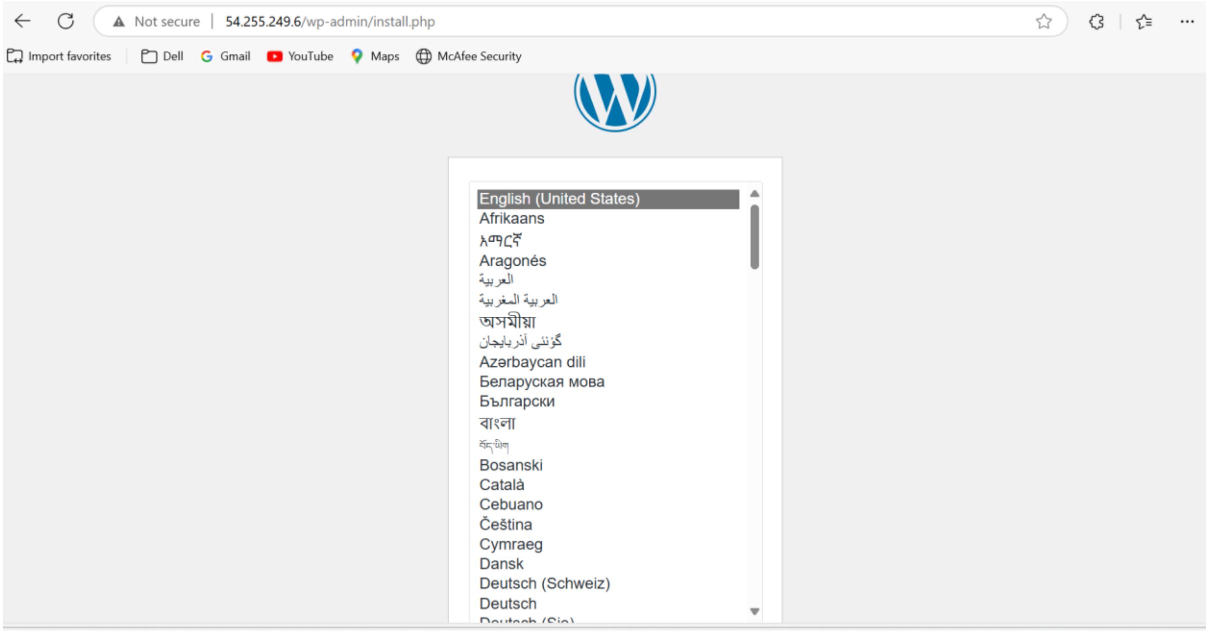
Write an executable bash script to set up a whole LAMP stack, PHP app can be Wordpress and DB can be MySQL.

The script should meet the below requirements :

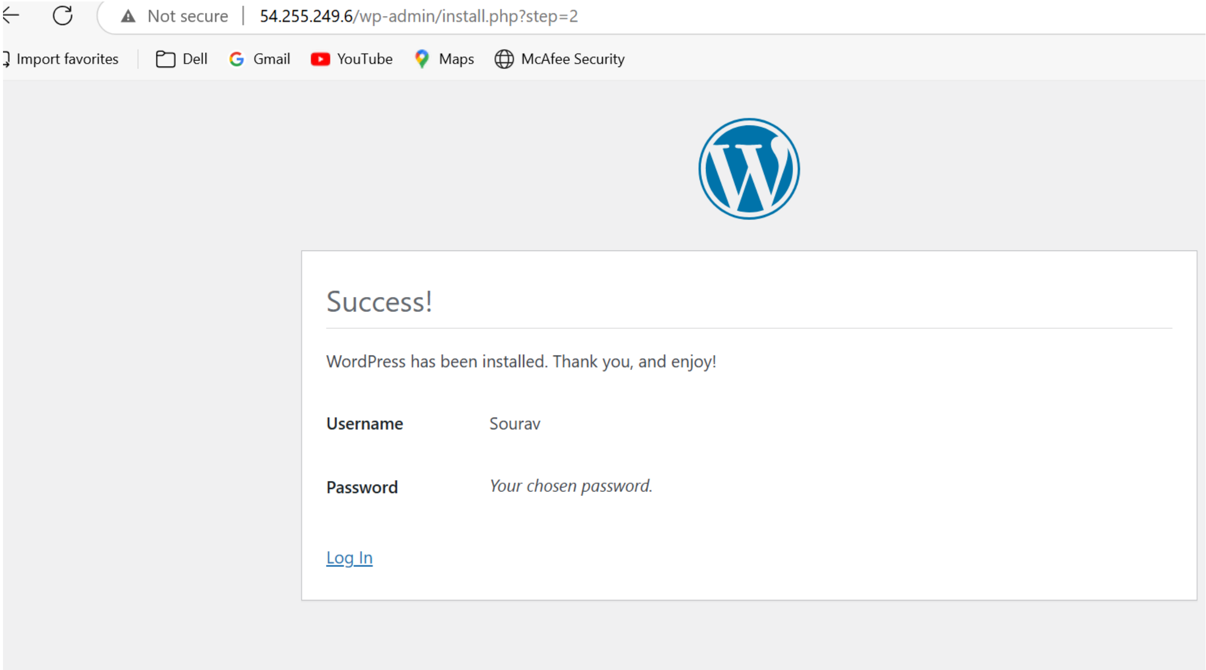
● This script should **install all components needed for a Wordpress website**.

● You can run this script on a local machine or server and after the execution of the script, it should have Wordpress **Running via Apache**.

● A database user for Wordpress can be **made automatically from within the script** and the same can be set in **Wordpress conf file**. Also, the script **showing output the database user details** at the end after the successful installation **as a MySQL connection string.**



English (United States)
Afrikaans
አማርኛ
Aragonés
العربية
العربية المغربية
অসমীয়া
گؤنئی آذربایجان
Azərbaycan dili
Беларуская мова
Български
বাংলা
བོད་ཡིག
Bosanski
Català
Cebuano
Čeština
Cymraeg
Dansk
Deutsch (Schweiz)
Deutsch
Deutsch (Sie)



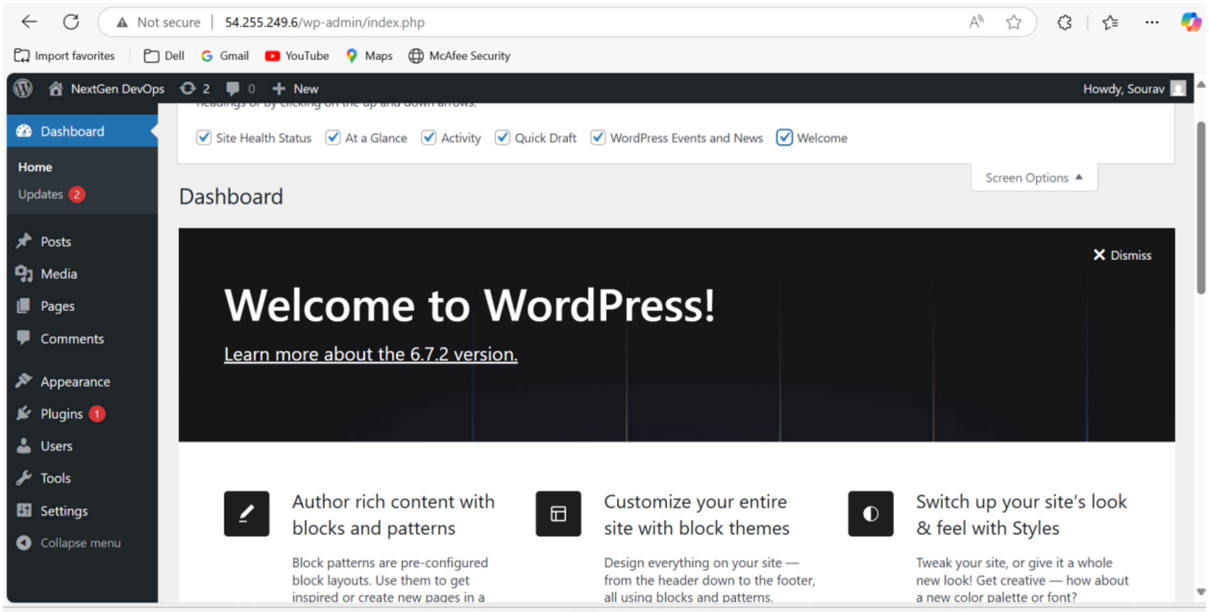
Success!

WordPress has been installed. Thank you, and enjoy!

Username Sourav

Password *Your chosen password.*

PFBS



Dashboard

Welcome to WordPress!

Learn more about the 6.7.2 version.

Author rich content with blocks and patterns

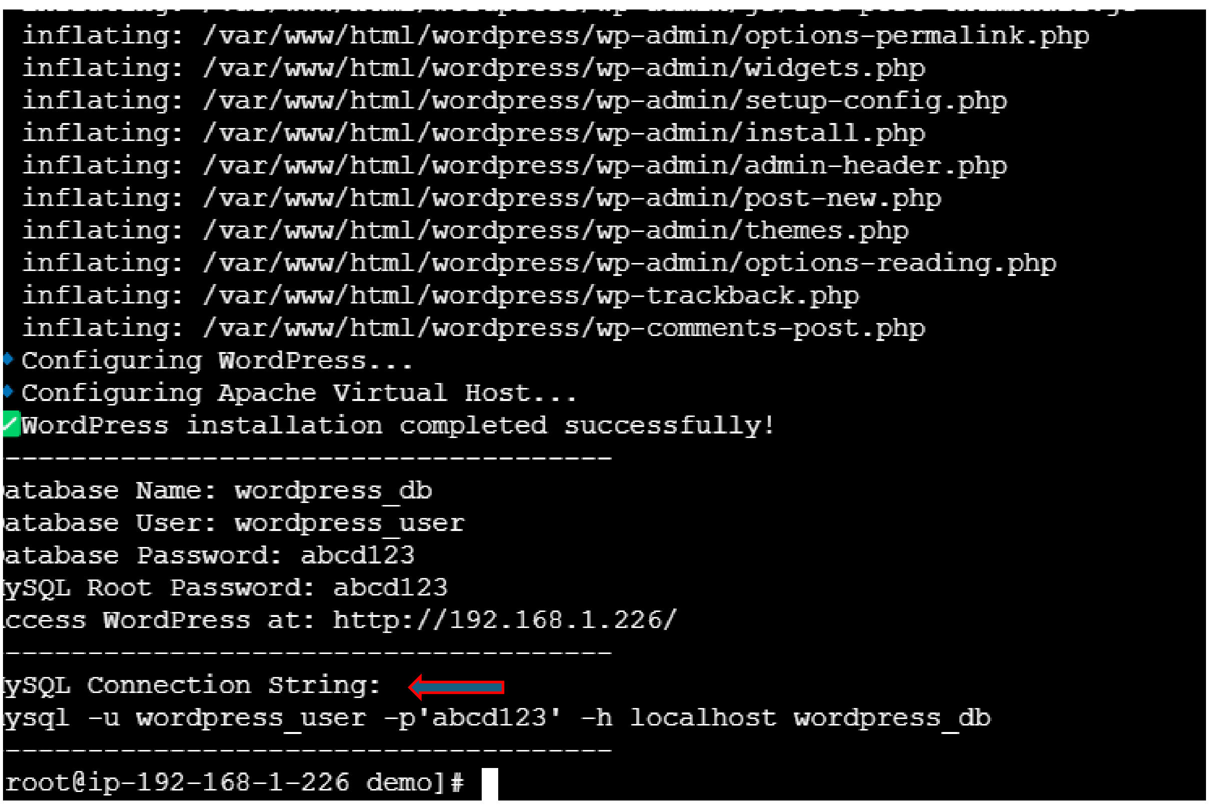
Block patterns are pre-configured block layouts. Use them to get inspired or create new pages in a

Customize your entire site with block themes

Design everything on your site — from the header down to the footer, all using blocks and patterns.

Switch up your site's look & feel with Styles

Tweak your site, or give it a whole new look! Get creative — how about a new color palette or font?



```
 inflating: /var/www/html/wordpress/wp-admin/options-permalink.php
 inflating: /var/www/html/wordpress/wp-admin/widgets.php
 inflating: /var/www/html/wordpress/wp-admin/setup-config.php
 inflating: /var/www/html/wordpress/wp-admin/install.php
 inflating: /var/www/html/wordpress/wp-admin/admin-header.php
 inflating: /var/www/html/wordpress/wp-admin/post-new.php
 inflating: /var/www/html/wordpress/wp-admin/themes.php
 inflating: /var/www/html/wordpress/wp-admin/options-reading.php
 inflating: /var/www/html/wordpress/wp-trackback.php
 inflating: /var/www/html/wordpress/wp-comments-post.php
 Configuring WordPress...
 Configuring Apache Virtual Host...
✅WordPress installation completed successfully!
-------------------------------------
atabase Name: wordpress_db
atabase User: wordpress_user
atabase Password: abcd123
ySQL Root Password: abcd123
ccess WordPress at: http://192.168.1.226/
-------------------------------------
ySQL Connection String:
ysql -u wordpress_user -p'abcd123' -h localhost wordpress_db
-------------------------------------
root@ip-192-168-1-226 demo]#
```

i-0c0e76e0b78f81462 (demo-ec2)

PublicIPs: 54.255.249.6 PrivateIPs: 192.168.1.226

Script:

#!/bin/bash

Exit script on error

```
set -e
```

Define variables

```
DB_NAME="wordpress_db" DB_USER="wordpress_user" DB_PASS="abcd123"
MYSQL_ROOT_PASS="abcd123" WP_DIR="/var/www/html/wordpress"
APACHE_CONF="/etc/httpd/conf.d/wordpress.conf"
```

Stop and clean up previous installations

```
echo "□ Cleaning up old installations..." sudo systemctl stop httpd mariadb
2>/dev/null || true sudo rm -rf $WP_DIR sudo rm -f $APACHE_CONF sudo dnf
remove -y mariadb* httpd php* 2>/dev/null || true sudo rm -rf /var/lib/mysql
/etc/my.cnf
```

Update system

```
echo "□ Updating system packages..." sudo dnf update -y
```

Install Apache, MariaDB, PHP, and required extensions

```
echo "□ Installing Apache, MariaDB, and PHP..." sudo dnf install -y httpd
mariadb105-server php8.3 php8.3-mysqlnd php8.3-xml php8.3-mbstring php8.3-
common unzip wget
```

Start and enable services

```
echo "□ Starting and enabling services..." sudo systemctl enable --now httpd
mariadb
```

Secure MariaDB installation

```
echo "□ Securing MariaDB..." sudo mysqladmin -u root password
"$MYSQL_ROOT_PASS" echo -e "$MYSQL_ROOT_PASS\nn\ny\ny\ny\ny" | sudo
mysql_secure_installation
```

Restart MariaDB after securing

```
sudo systemctl restart mariadb
```

Create WordPress Database & User

```
echo "□ Creating MySQL Database and User..." sudo mysql -u root -
p"$MYSQL_ROOT_PASS" <<MYSQL_SCRIPT CREATE DATABASE IF NOT
EXISTS $DB_NAME; CREATE USER IF NOT EXISTS '$DB_USER'@'localhost'
```

IDENTIFIED BY '$DB_PASS'; GRANT ALL PRIVILEGES ON $DB_NAME.* TO '$DB_USER'@'localhost'; FLUSH PRIVILEGES; MYSQL_SCRIPT

Download and setup WordPress

echo "□ Downloading and configuring WordPress..." wget https://wordpress.org/latest.zip -O /tmp/wordpress.zip sudo unzip /tmp/wordpress.zip -d /var/www/html/ sudo chown -R apache:apache $WP_DIR sudo chmod -R 755 $WP_DIR

Configure wp-config.php

echo "□ Configuring WordPress..." sudo cp $WP_DIR/wp-config-sample.php $WP_DIR/wp-config.php sudo sed -i "s/database_name_here/$DB_NAME/" $WP_DIR/wp-config.php sudo sed -i "s/username_here/$DB_USER/" $WP_DIR/wp-config.php sudo sed -i "s/password_here/$DB_PASS/" $WP_DIR/wp-config.php

Configure Apache Virtual Host

echo "□ Configuring Apache Virtual Host..." sudo bash -c "cat > $APACHE_CONF <<EOF <VirtualHost *:80> ServerAdmin admin@example.com DocumentRoot $WP_DIR <Directory $WP_DIR> AllowOverride All Require all granted ErrorLog /var/log/httpd/wordpress_error.log CustomLog /var/log/httpd/wordpress_access.log combined EOF"

Restart Apache to apply changes

sudo systemctl restart httpd

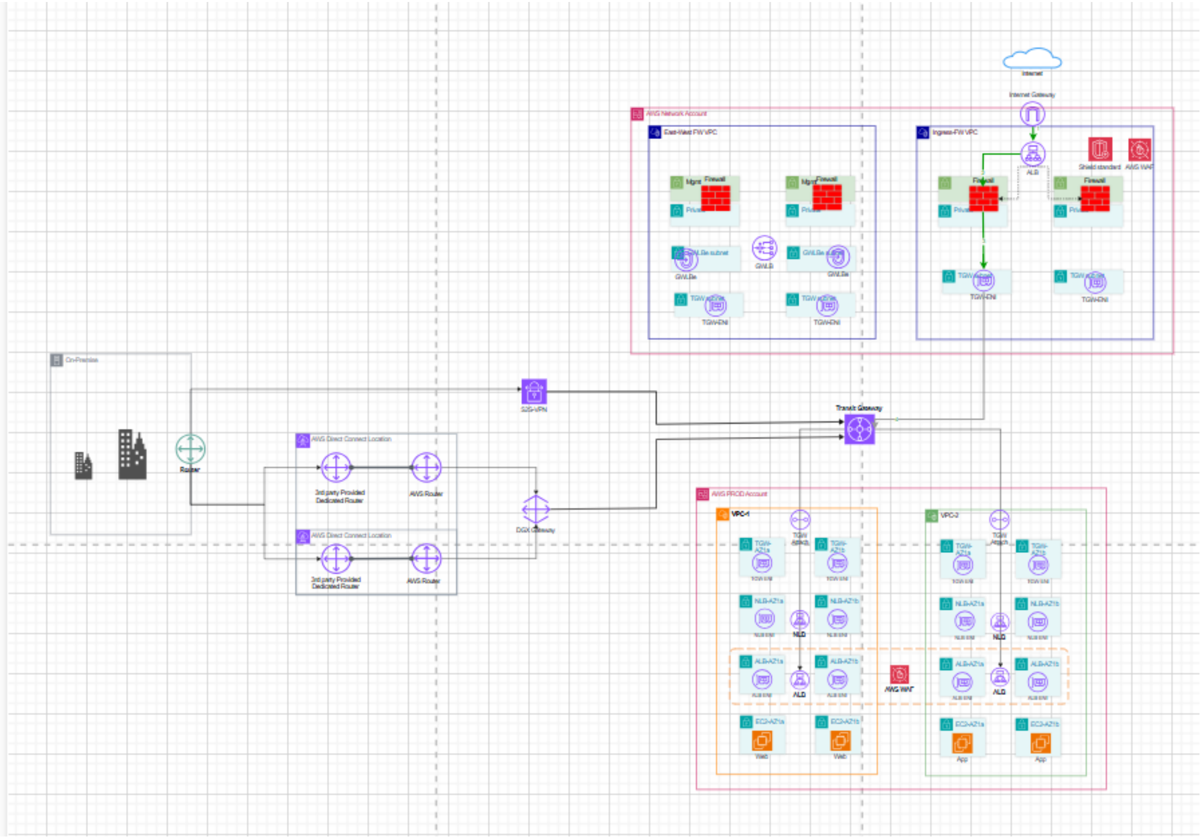
Output MySQL Connection String

echo "✅WordPress installation completed successfully!" echo "-------------------------------------" echo "Database Name: $DB_NAME" echo "Database User: $DB_USER" echo "Database Password: $DB_PASS" echo "MySQL Root Password: $MYSQL_ROOT_PASS" echo "Access WordPress at: http://$(hostname -I | awk '{print $1}')/" echo "-------------------------------------"

echo "MySQL Connection String:" echo "mysql -u $DB_USER -p'$DB_PASS' -h localhost $DB_NAME" echo "-------------------------------------"

Question/Assignment 4

Let’s assume,we are working on an application which is hosted on AWS . Drawn anarchitecture diagram for a PHP/JAVA/Python-based application to be hosted on AWS with all mentions like VPC, AWS,well-defined network segregation.Any more details that you think are necessary please do include them.



BONU QUESTIONS

1. Write a script which will **based on "Number of requests"** metric of the ALB/ELB scale up web-app EC2 instances under the Load Balancer, increase AWS Elasticsearch Nodes count, and change the instance size of a MongoDB EC2 instance from m4.large to m4.xlarge. (without using ASG) (Can be done for any cloud platform)

Prerequisites:

- AWS CLI must be installed and configured with the necessary permissions.
- jq must be installed (`sudo yum install -y jq` on Amazon Linux).
- IAM user/role must have permissions for EC2, ELB, and OpenSearch.

Script:

```
#!/bin/bash

#Exit on error

set -e

#Variables

ALB_NAME="load-balancer-name"             # Replace with  ALB/ELB name
WEB_APP_AMI_ID="ami-xxxxxxxxxxxxxxx" # Replace with your AMI ID for new
EC2 instances INSTANCE_TYPE="t2.micro"       # Instance type for new web-app
instances SECURITY_GROUP="sg-xxx"                          # Security Group ID

SUBNET_ID="subnet-xxxx"                              # Subnet ID

ALB_TARGET_GROUP="target-group"          # Target Group name for Load
Balancer

MONGO_INSTANCE_ID="i-xxxxx"                   # MongoDB EC2 Instance ID

 ES_DOMAIN="es-domain"                               # Elasticsearch Domain Name

#Function to get ALB Request Count

get_alb_request_count() {

METRIC_VALUE=$(aws cloudwatch get-metric-statistics --namespace
AWS/ApplicationELB  \

--metric-name RequestCount --dimensions
Name=LoadBalancer,Value=$ALB_NAME \

--statistics Sum --period 300 --start-time $(date -u -d '-5 minutes'
+%Y-%m-%dT%H:%M:%SZ)  \

--end-time $(date -u +%Y-%m-%dT%H:%M:%SZ) --region us-east-1 | jq -r
'.Datapoints[0].Sum')

echo "${METRIC_VALUE:-0}"


}

#Function to launch a new Web-App EC2 instance
```

```
scale_up_web_app() {

echo "Scaling up Web-App EC2 Instance..."

INSTANCE_ID=$(aws ec2 run-instances --image-id $WEB_APP_AMI_ID --instance-
type $INSTANCE_TYPE

--security-group-ids $SECURITY_GROUP --subnet-id $SUBNET_ID --count 1

--query 'Instances[0].InstanceId' --output text)

echo "New Web-App instance launched: $INSTANCE_ID"

# Register the new instance to Target Group
aws elbv2 register-targets --target-group-arn $ELB_TARGET_GROUP --
targets Id=$INSTANCE_ID
echo "Instance added to Load Balancer Target Group."


}
```

#Function to increase Elasticsearch Nodes

```
increase_es_nodes() {

echo "Increasing Elasticsearch Nodes..."

CURRENT_INSTANCE_COUNT=$(aws opensearch describe-domain --domain-
name $ES_DOMAIN --query 'DomainStatus.ClusterConfig.InstanceCount' --output
text) NEW_INSTANCE_COUNT=$((CURRENT_INSTANCE_COUNT + 1))

aws opensearch update-domain-config --domain-name $ES_DOMAIN --
cluster-config InstanceCount=$NEW_INSTANCE_COUNT
echo "Elasticsearch node count updated to $NEW_INSTANCE_COUNT."


}
```

#Function to resize MongoDB EC2 instance

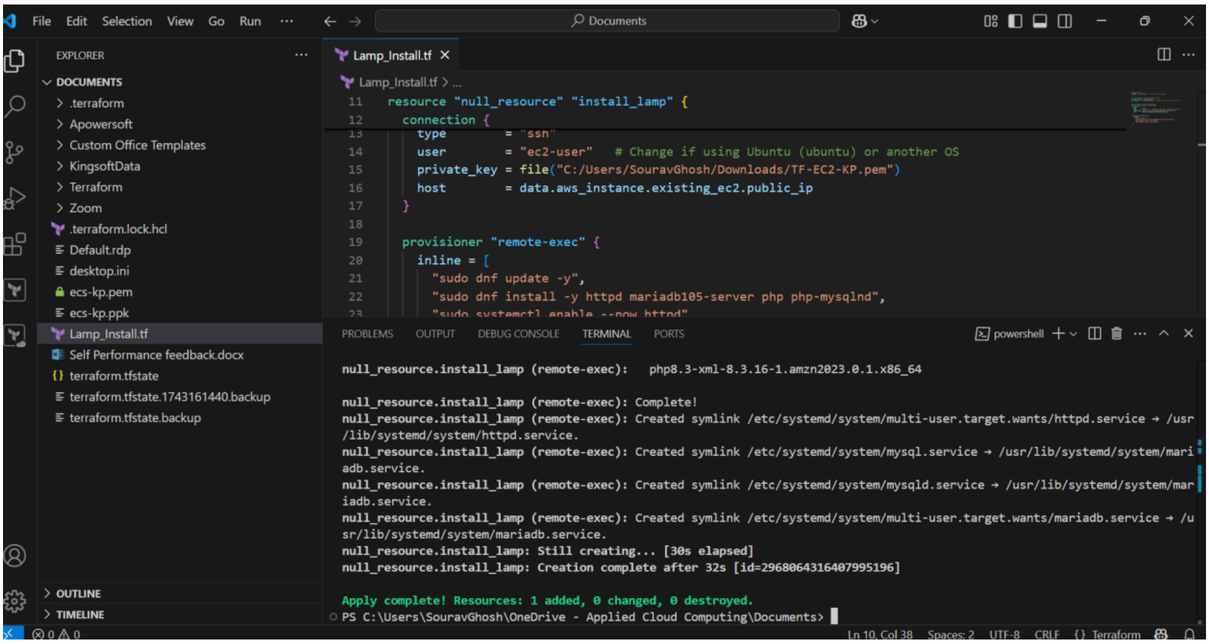
```
resize_mongo_instance() {

echo "Changing MongoDB instance type..."

aws ec2 stop-instances --instance-ids $MONGO_INSTANCE_ID

echo " Waiting for MongoDB instance to stop..."
aws ec2 wait instance-stopped --instance-ids $MONGO_INSTANCE_ID

aws ec2 modify-instance-attribute --instance-id $MONGO_INSTANCE_ID --instance-type "{\"Value\": \"m4.xlarge\"}"
echo "MongoDB instance type updated to m4.xlarge."

aws ec2 start-instances --instance-ids $MONGO_INSTANCE_ID
echo "MongoDB instance restarted."


}

#Main Execution Logic

REQUEST_COUNT=$(get_alb_request_count)

echo "Current ALB Request Count: $REQUEST_COUNT"

if [[ $REQUEST_COUNT -gt 1000 ]]; then

 scale_up_web_app

increase_es_nodes

resize_mongo_instance


else

echo "No scaling required. Request count is below threshold."

 fi
```

2. Write a Terraform/Cloud Formation template for the LAMP stack in Question 2

Terraform Script :

```
provider "aws" {
  region = "ap-southeast-1"  # Change to your AWS region
}
```

```
# ? Get details of the existing EC2 instance
data "aws_instance" "existing_ec2" {
  instance_id = "i-05e0193509e74d96f"  # Replace with your actual instance ID
}

# ? Provisioner to install LAMP stack
resource "null_resource" "install_lamp" {
  connection {
    type        = "ssh"
    user        = "ec2-user"   # Change if using Ubuntu (ubuntu) or another OS
    private_key = file("C:/Users/SouravGhosh/Downloads/TF-EC2-KP.pem")
    host        = data.aws_instance.existing_ec2.public_ip
  }

  provisioner "remote-exec" {
    inline = [
      "sudo dnf update -y",
      "sudo dnf install -y httpd mariadb105-server php php-mysqlnd",
      "sudo systemctl enable --now httpd",
      "sudo systemctl enable --now mariadb"
    ]
  }
}
```

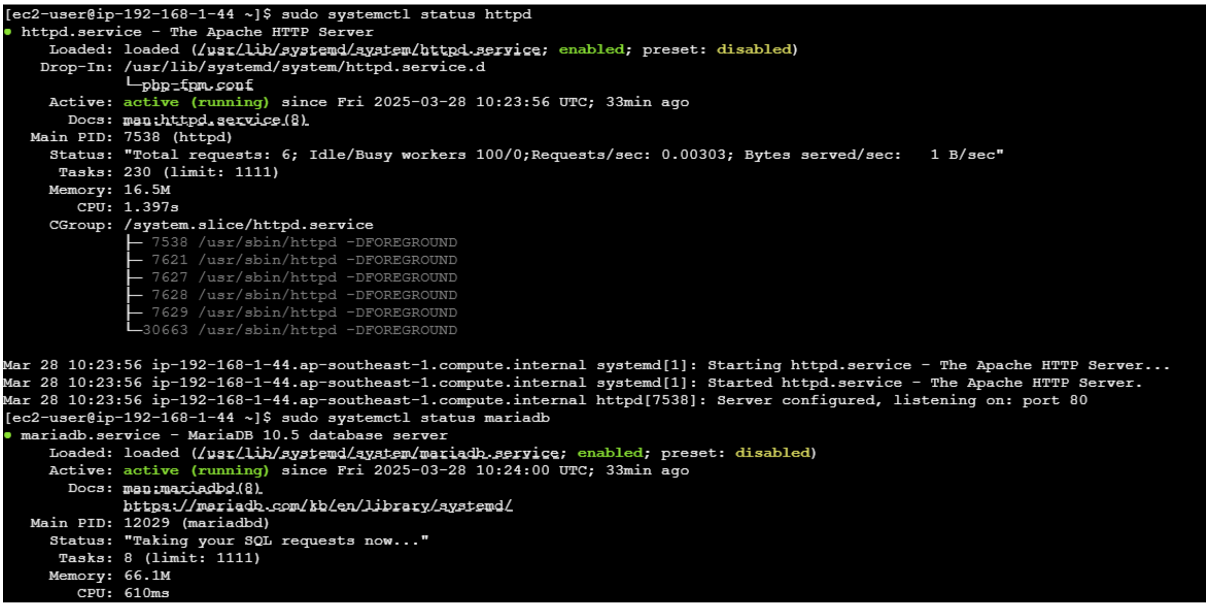


```
resource "null_resource" "install_lamp" {
  connection {
    type        = "ssh"
    user        = "ec2-user"   # Change if using Ubuntu (ubuntu) or another OS
    private_key = file("C:/Users/SouravGhosh/Downloads/TF-EC2-KP.pem")
    host        = data.aws_instance.existing_ec2.public_ip
  }

  provisioner "remote-exec" {
    inline = [
      "sudo dnf update -y",
      "sudo dnf install -y httpd mariadb105-server php php-mysqlnd",
      "sudo systemctl enable --now httpd"
```

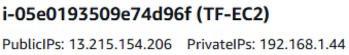
```
null_resource.install_lamp (remote-exec):   php8.3-xml-8.3.16-1.amzn2023.0.1.x86_64

null_resource.install_lamp (remote-exec): Complete!
null_resource.install_lamp (remote-exec): Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr
/lib/systemd/system/httpd.service.
null_resource.install_lamp (remote-exec): Created symlink /etc/systemd/system/mysql.service → /usr/lib/systemd/system/mari
adb.service.
null_resource.install_lamp (remote-exec): Created symlink /etc/systemd/system/mysqld.service → /usr/lib/systemd/system/mar
iadb.service.
null_resource.install_lamp (remote-exec): Created symlink /etc/systemd/system/multi-user.target.wants/mariadb.service → /u
sr/lib/systemd/system/mariadb.service.
null_resource.install_lamp: Still creating... [30s elapsed]
null_resource.install_lamp: Creation complete after 32s [id=2968064316407995196]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS C:\Users\SouravGhosh\OneDrive - Applied Cloud Computing\Documents>
```

PFB-

Showing **httpd & Mariadb is installed & Active.**



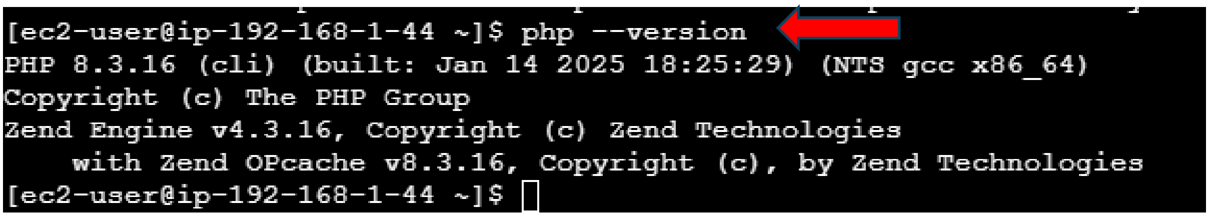
```
[ec2-user@ip-192-168-1-44 ~]$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
    Drop-In: /usr/lib/systemd/system/httpd.service.d
             └─php-fpm.conf
     Active: active (running) since Fri 2025-03-28 10:23:56 UTC; 33min ago
       Docs: man:httpd.service(8)
   Main PID: 7538 (httpd)
     Status: "Total requests: 6; Idle/Busy workers 100/0;Requests/sec: 0.00303; Bytes served/sec:   1 B/sec"
      Tasks: 230 (limit: 1111)
     Memory: 16.5M
        CPU: 1.397s
     CGroup: /system.slice/httpd.service
             ├─ 7538 /usr/sbin/httpd -DFOREGROUND
             ├─ 7621 /usr/sbin/httpd -DFOREGROUND
             ├─ 7627 /usr/sbin/httpd -DFOREGROUND
             ├─ 7628 /usr/sbin/httpd -DFOREGROUND
             ├─ 7629 /usr/sbin/httpd -DFOREGROUND
             └─30663 /usr/sbin/httpd -DFOREGROUND

Mar 28 10:23:56 ip-192-168-1-44.ap-southeast-1.compute.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
Mar 28 10:23:56 ip-192-168-1-44.ap-southeast-1.compute.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
Mar 28 10:23:56 ip-192-168-1-44.ap-southeast-1.compute.internal httpd[7538]: Server configured, listening on: port 80
[ec2-user@ip-192-168-1-44 ~]$ sudo systemctl status mariadb
● mariadb.service - MariaDB 10.5 database server
     Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset: disabled)
     Active: active (running) since Fri 2025-03-28 10:24:00 UTC; 33min ago
       Docs: man:mariadbd(8)
             https://mariadb.com/kb/en/library/systemd/
   Main PID: 12029 (mariadbd)
     Status: "Taking your SQL requests now..."
      Tasks: 8 (limit: 1111)
     Memory: 66.1M
        CPU: 610ms
```



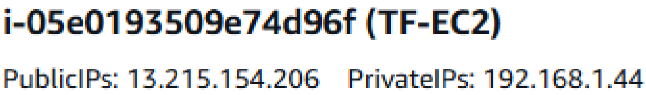
i-05e0193509e74d96f (TF-EC2)

PublicIPs: 13.215.154.206   PrivateIPs: 192.168.1.44

Showing php is also installed

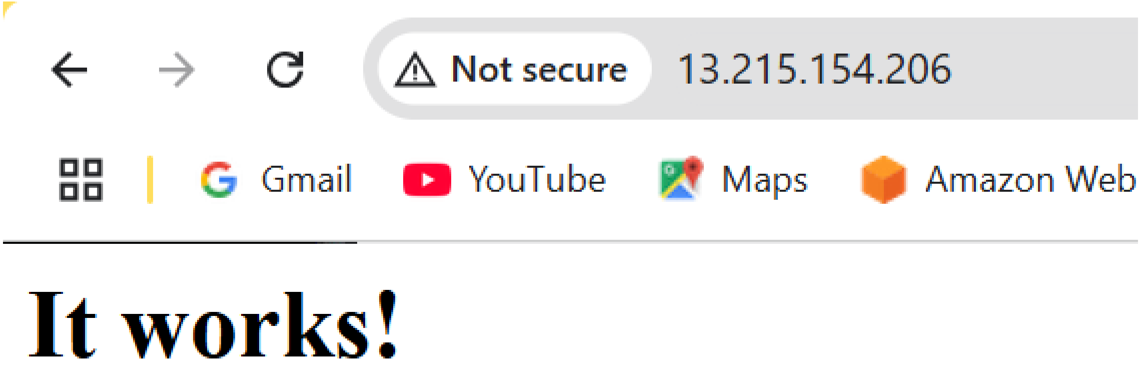


```
[ec2-user@ip-192-168-1-44 ~]$ php --version
PHP 8.3.16 (cli) (built: Jan 14 2025 18:25:29) (NTS gcc x86_64)
Copyright (c) The PHP Group
Zend Engine v4.3.16, Copyright (c) Zend Technologies
    with Zend OPcache v8.3.16, Copyright (c), by Zend Technologies
[ec2-user@ip-192-168-1-44 ~]$
```



i-05e0193509e74d96f (TF-EC2)

PublicIPs: 13.215.154.206   PrivateIPs: 192.168.1.44



It works!

--Thank You---