Discipline

LAMP Stack

CS5002

Activity-19

Sheikh Muhammed Tadeeb (AU19B1014)

❖ Problem Statement:

1. Create a LAMP server and capture all the screens for the configuration.
2. Create a User-data script to automate this and share its images as well.

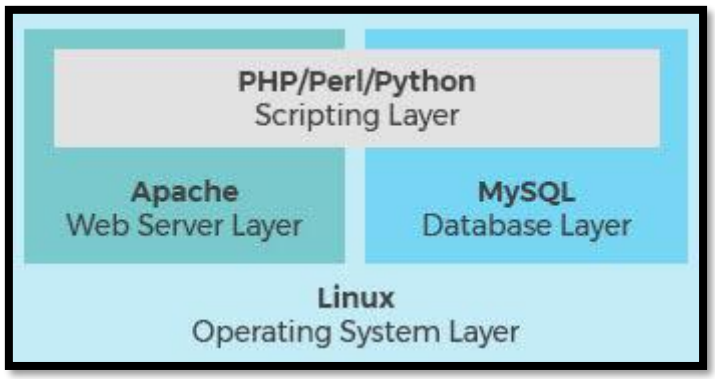
❖ Theory:

➢ Brief:

LAMP stands for Linux, Apache, MySQL, and PHP. Together, they provide a proven set of software for delivering high-performance web applications and as it contains PHP it works with Apache to help us create dynamic web pages as well.

➢ Architecture:

LAMP has a classic layered architecture, with Linux at the lowest level. The next layer is Apache and MySQL, followed by PHP. Although PHP is nominally at the top or presentation layer, the PHP component sits inside Apache.



➢ LAMP Working (layer-interoperability):

A high-level look at the LAMP stack order of execution shows how the elements interoperate. The process starts when the Apache web server receives requests for web pages from a user's browser. If the request is for a PHP file, Apache passes the request

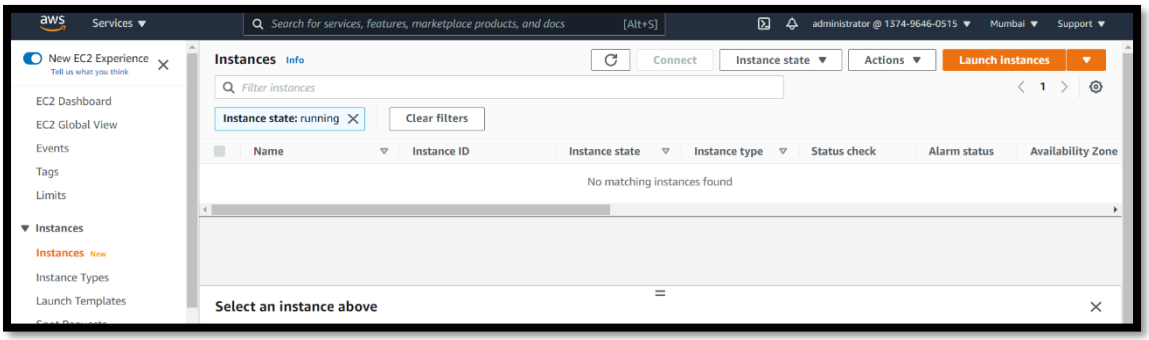
to PHP, which loads the file and executes the code contained in the file. PHP also communicates with MySQL to fetch any data referenced in the code.

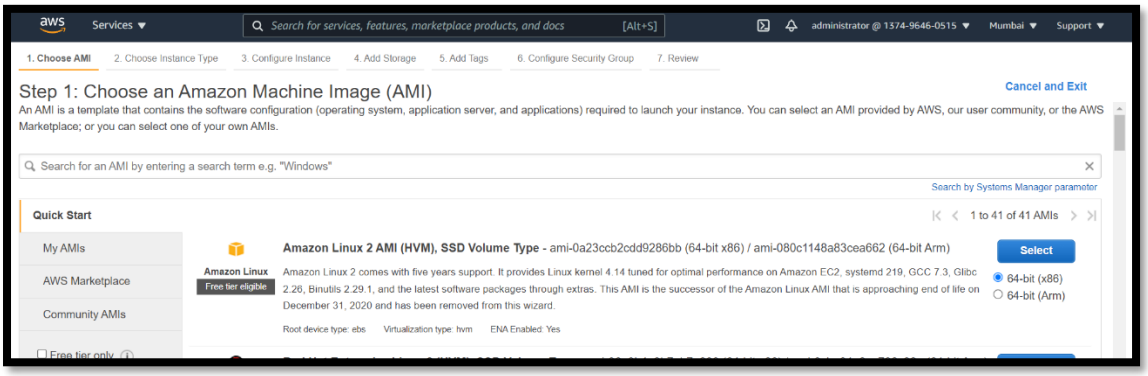
PHP then uses the code in the file and the data from the database to create the HTML that browsers require to display web pages. The LAMP stack is efficient at handling not only static web pages, but also dynamic pages where the content may change each time it is loaded depending on the date, time, user identity and other factors.

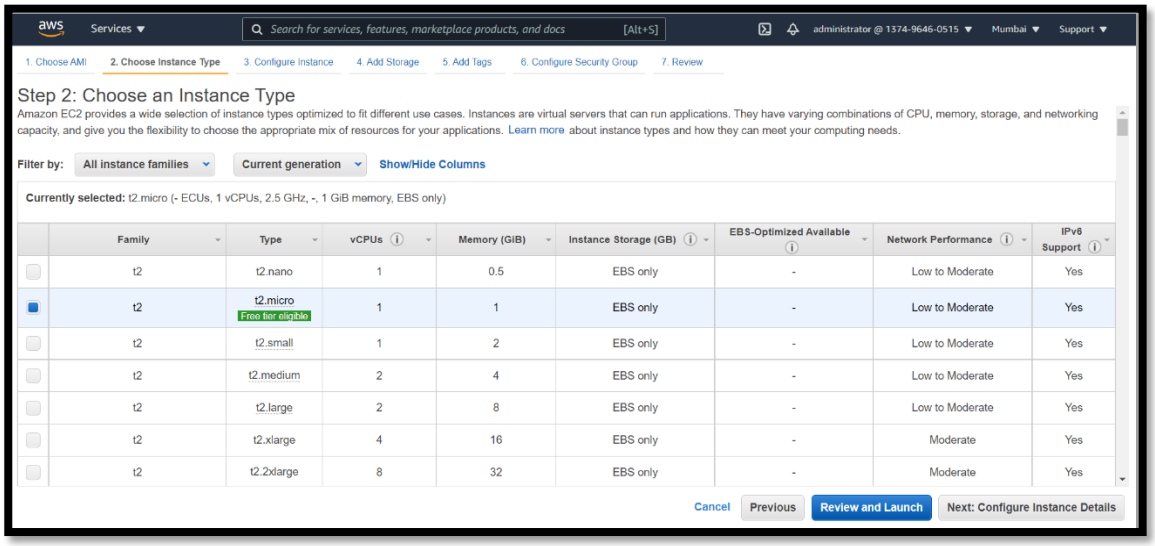
After running the file code, PHP then passes the resulting data back to the Apache web server to send to the browser. It can also store this new data in MySQL. And of course, all of these operations are enabled by the Linux operating system running at the base of the stack.

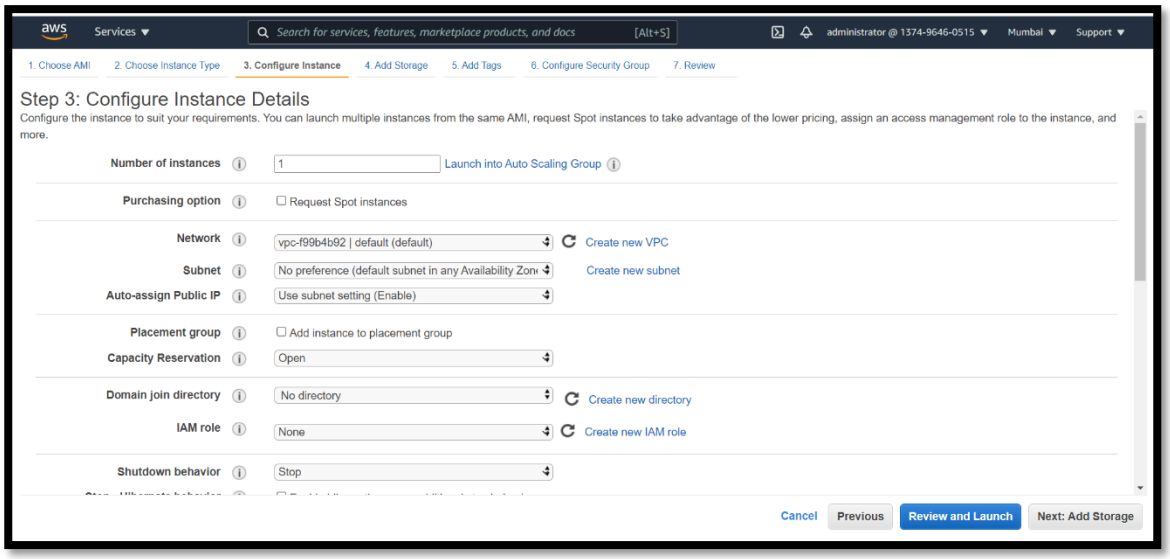
❖ Solution 1, Practical (Preparing the LAMP server):

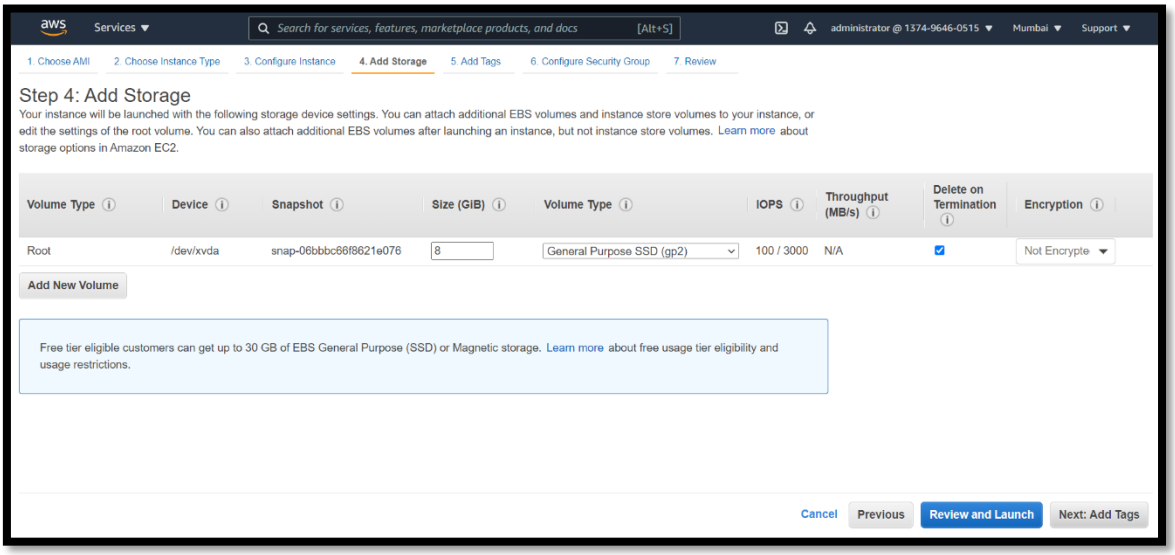
- **Step1)** Launching a new instance using Amazon Linux 2 with security group allowing SSH (port 22), HTTP (port 80), and HTTPS (port 443) connections.

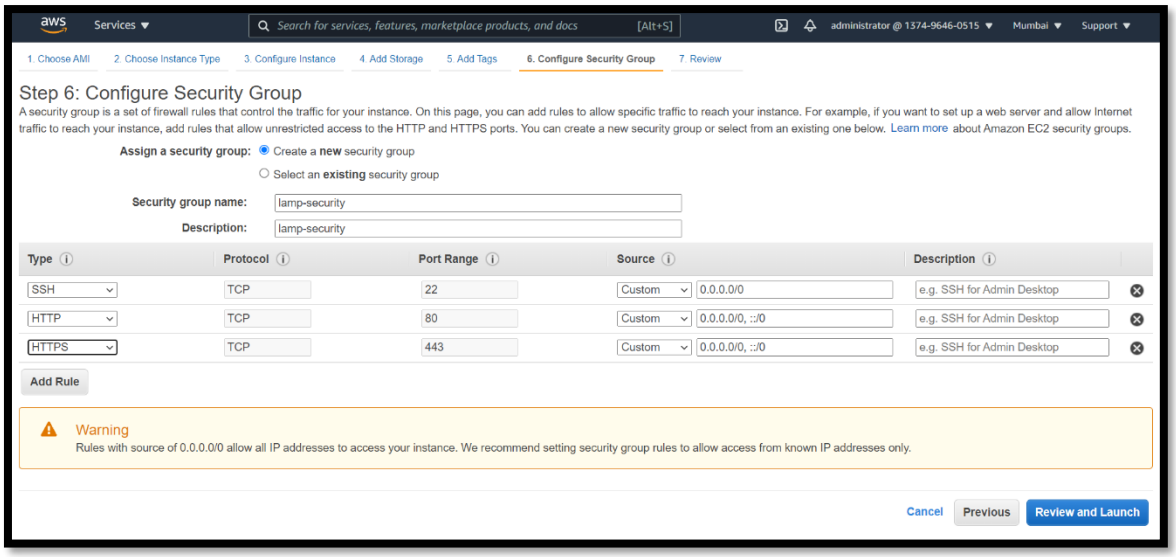


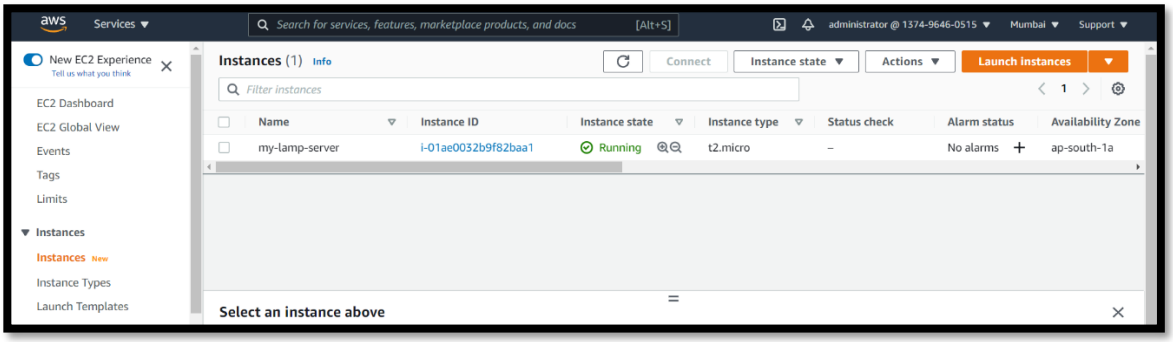



aws
Services
Search for services, features, marketplace products, and docs [Alt+S]
administrator @ 1374-9646-0515
Mumbai
Support
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 2: Choose an Instance Type
Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.
Filter by:
All instance families
Current generation
Show/Hide Columns
Currently selected: t2.micro (- ECUs, 1 vCPUs, 2.5 GHz, -, 1 GiB memory, EBS only)
Family | Type | vCPUs | Memory (GiB) | Instance Storage (GB) | EBS-Optimized Available | Network Performance | IPv6 Support
t2 | t2.nano | 1 | 0.5 | EBS only | - | Low to Moderate | Yes
t2 | t2.micro Free tier eligible | 1 | 1 | EBS only | - | Low to Moderate | Yes
t2 | t2.small | 1 | 2 | EBS only | - | Low to Moderate | Yes
t2 | t2.medium | 2 | 4 | EBS only | - | Low to Moderate | Yes
t2 | t2.large | 2 | 8 | EBS only | - | Low to Moderate | Yes
t2 | t2.xlarge | 4 | 16 | EBS only | - | Moderate | Yes
t2 | t2.2xlarge | 8 | 32 | EBS only | - | Moderate | Yes
Cancel
Previous
Review and Launch
Next: Configure Instance Details

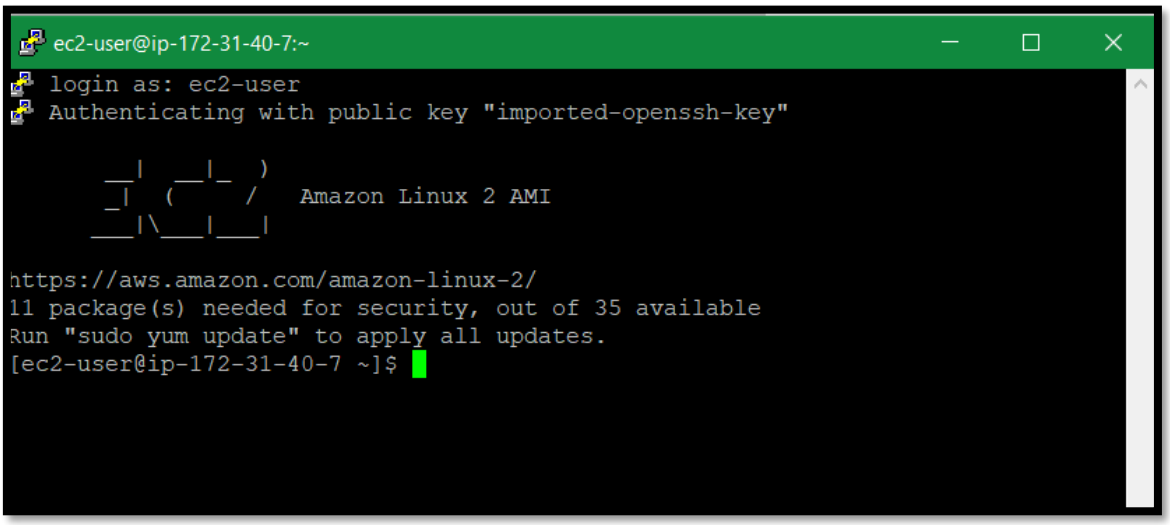

aws
Services
Search for services, features, marketplace products, and docs [Alt+S]
administrator @ 1374-9646-0515
Mumbai
Support
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 3: Configure Instance Details
Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.
Number of instances
1
Launch into Auto Scaling Group
Purchasing option
Request Spot instances
Network
vpc-f99b4b92 | default (default)
Create new VPC
Subnet
No preference (default subnet in any Availability Zone
Create new subnet
Auto-assign Public IP
Use subnet setting (Enable)
Placement group
Add instance to placement group
Capacity Reservation
Open
Domain join directory
No directory
Create new directory
IAM role
None
Create new IAM role
Shutdown behavior
Stop
Cancel
Previous
Review and Launch
Next: Add Storage


aws
Services
Search for services, features, marketplace products, and docs [Alt+S]
administrator @ 1374-9646-0515
Mumbai
Support
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 4: Add Storage
Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. Learn more about storage options in Amazon EC2.
Volume Type | Device | Snapshot | Size (GiB) | Volume Type | IOPS | Throughput (MB/s) | Delete on Termination | Encryption
Root | /dev/xvda | snap-06bbbc66f8621e076 | 8 | General Purpose SSD (gp2) | 100 / 3000 | N/A | | Not Encrypte
Add New Volume
Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. Learn more about free usage tier eligibility and usage restrictions.
Cancel
Previous
Review and Launch
Next: Add Tags



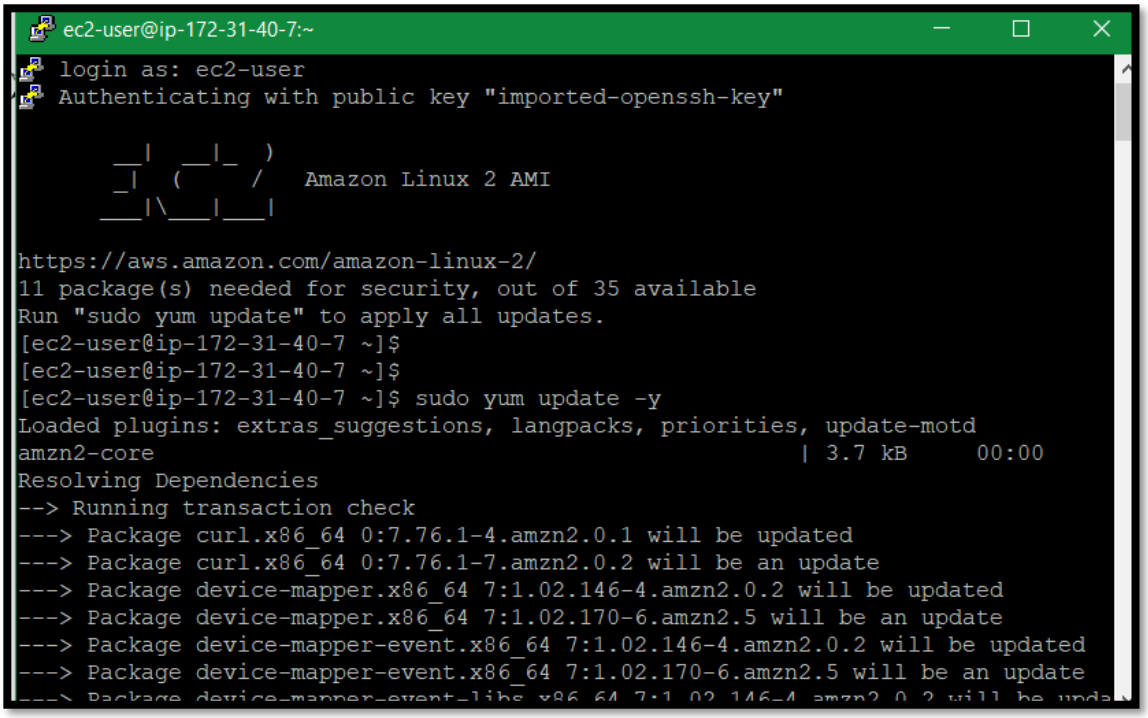


➢ **Step2)** Opening putty to check whether we are connected to our instance or not.

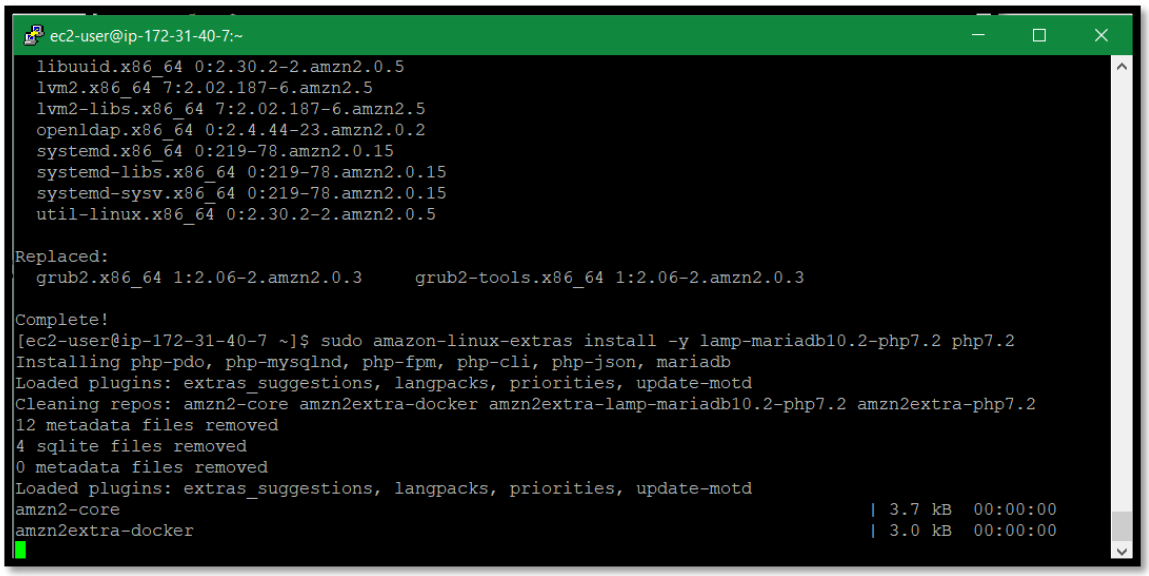


- **Step3)** To ensure that all of your software packages are up to date, perform a quick software update on your instance. This process may take a few minutes, but it is important to make sure that you have the latest security updates and bug fixes.

  The -y option installs the updates without asking for confirmation. If you would like to examine the updates before installing, you can omit this option



- **Step4)** Install the lamp-mariadb10.2-php7.2 and php7.2 Amazon Linux Extras repositories to get the latest versions of the LAMP MariaDB and PHP packages for Amazon Linux 2.



- **Step5)** We can check information about our instance using system-release.

```
ec2-user@ip-172-31-40-7:~
[ec2-user@ip-172-31-40-7 ~]$ cat /etc/system-release
Amazon Linux release 2 (Karoo)
[ec2-user@ip-172-31-40-7 ~]$
```

- **Step6)** Now that your instance is current, you can install the Apache web server, MariaDB, and PHP software packages.

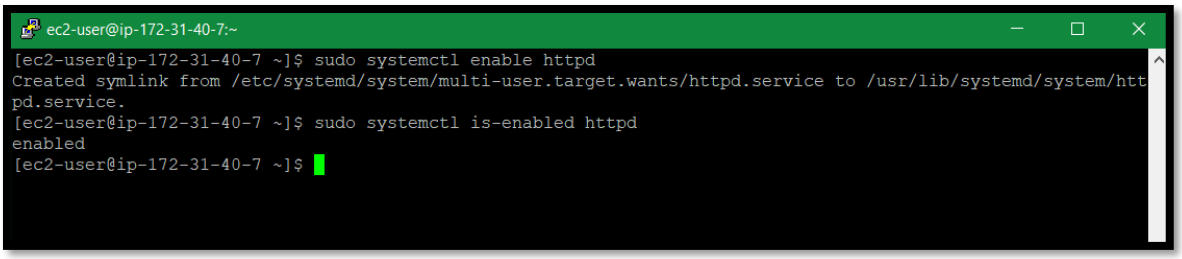
  Use the *yum install* command to install multiple software packages and all related dependencies at the same time.

```
ec2-user@ip-172-31-40-7:~
[ec2-user@ip-172-31-40-7 ~]$ sudo yum install -y httpd mariadb-server
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                                               | 3.7 kB  00:00:00
Resolving Dependencies
--> Running transaction check
---> Package httpd.x86_64 0:2.4.48-2.amzn2 will be installed
--> Processing Dependency: httpd-tools = 2.4.48-2.amzn2 for package: httpd-2.4.48-2.amzn2.x86_64
--> Processing Dependency: httpd-filesystem = 2.4.48-2.amzn2 for package: httpd-2.4.48-2.amzn2.x86_64
--> Processing Dependency: system-logos-httpd for package: httpd-2.4.48-2.amzn2.x86_64
--> Processing Dependency: mod_http2 for package: httpd-2.4.48-2.amzn2.x86_64
--> Processing Dependency: httpd-filesystem for package: httpd-2.4.48-2.amzn2.x86_64
--> Processing Dependency: /etc/mime.types for package: httpd-2.4.48-2.amzn2.x86_64
--> Processing Dependency: libaprutil-1.so.0()(64bit) for package: httpd-2.4.48-2.amzn2.x86_64
--> Processing Dependency: libapr-1.so.0()(64bit) for package: httpd-2.4.48-2.amzn2.x86_64
---> Package mariadb-server.x86_64 3:10.2.38-1.amzn2.0.1 will be installed
--> Processing Dependency: mariadb-tokudb-engine(x86-64) = 3:10.2.38-1.amzn2.0.1 for package: 3:mariadb-
server-10.2.38-1.amzn2.0.1.x86_64
--> Processing Dependency: mariadb-server-utils(x86-64) = 3:10.2.38-1.amzn2.0.1 for package: 3:mariadb-s
erver-10.2.38-1.amzn2.0.1.x86_64
--> Processing Dependency: mariadb-rocksdb-engine(x86-64) = 3:10.2.38-1.amzn2.0.1 for package: 3:mariadb
-server-10.2.38-1.amzn2.0.1.x86_64
--> Processing Dependency: mariadb-gssapi-server(x86-64) = 3:10.2.38-1.amzn2.0.1 for package: 3:mariadb-
server-10.2.38-1.amzn2.0.1.x86_64
--> Processing Dependency: mariadb-errmsg(x86-64) = 3:10.2.38-1.amzn2.0.1 for package: 3:mariadb-server-
```

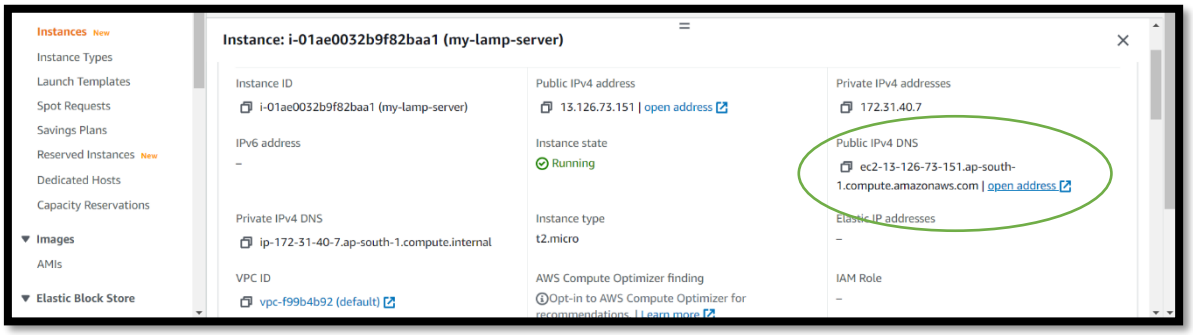
- **Step7)** Start the Apache web server and check its status.

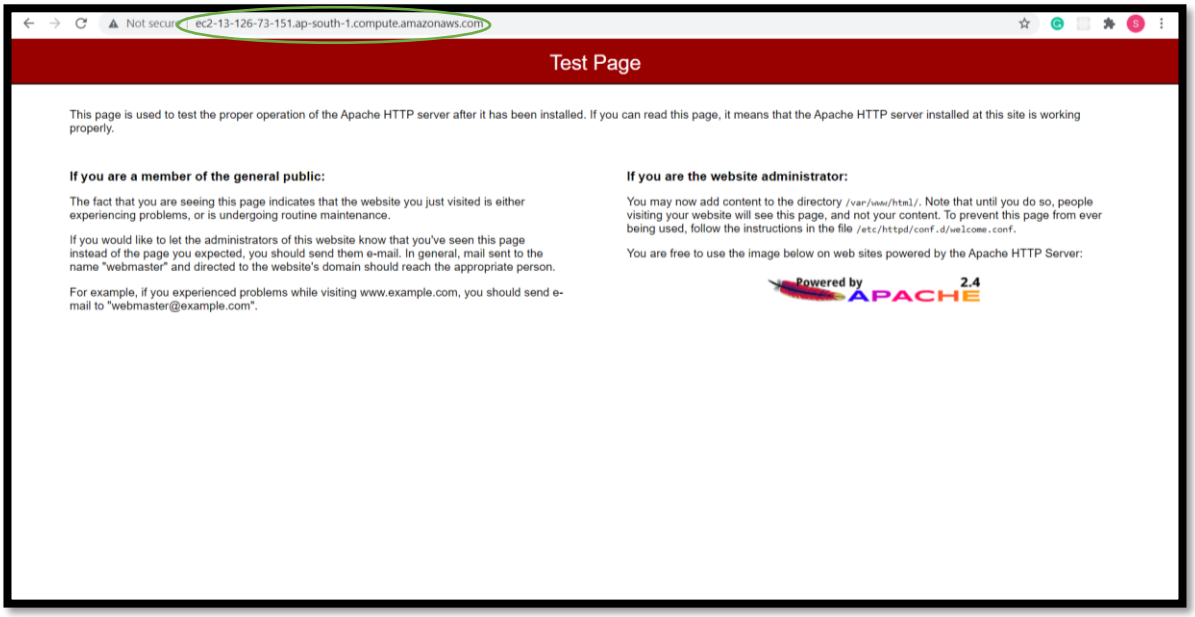
```
ec2-user@ip-172-31-40-7:~
[ec2-user@ip-172-31-40-7 ~]$ sudo systemctl start httpd
[ec2-user@ip-172-31-40-7 ~]$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
  Drop-In: /usr/lib/systemd/system/httpd.service.d
           └─php-fpm.conf
   Active: active (running) since Mon 2021-10-04 10:55:51 UTC; 53s ago
     Docs: man:httpd.service(8)
 Main PID: 8066 (httpd)
   Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec:   0 B/sec"
   CGroup: /system.slice/httpd.service
           ├─8066 /usr/sbin/httpd -DFOREGROUND
           ├─8072 /usr/sbin/httpd -DFOREGROUND
           ├─8073 /usr/sbin/httpd -DFOREGROUND
           ├─8074 /usr/sbin/httpd -DFOREGROUND
           ├─8075 /usr/sbin/httpd -DFOREGROUND
           └─8076 /usr/sbin/httpd -DFOREGROUND

Oct 04 10:55:51 ip-172-31-40-7.ap-south-1.compute.internal systemd[1]: Starting The Apache HTTP Server...
Oct 04 10:55:51 ip-172-31-40-7.ap-south-1.compute.internal systemd[1]: Started The Apache HTTP Server.
[ec2-user@ip-172-31-40-7 ~]$
```

- **Step8)** Use the *systemctl* command to configure the Apache web server to start at each system boot and check whether httpd is running or not.



- **Step9)** Test our web server. In a web browser, type the public DNS address (or the public IP address) of our instance. If there is no content in /var/www/html, we should see the Apache test page. We can get the public DNS for our instance using the Amazon EC2 console (check the Public DNS column; if this column is hidden, choose Show/Hide Columns (the gear-shaped icon) and choose Public DNS).





- **Step10)** Apache *httpd* serves files that are kept in a directory called the Apache document root. The Amazon Linux Apache document root is /var/www/html, which by default is owned by root.

  To allow the ec2-user account to manipulate files in this directory, we must modify the ownership and permissions of the directory. There are many ways to accomplish this task. We add ec2-user to the apache group, to give the apache group ownership of the /var/www directory and assign write permissions to the group.

  - **Step10.1)** Add your user (in this case, ec2-user) to the apache group and after this log-out to verify our membership.

```
ec2-user@ip-172-31-40-7:~
[ec2-user@ip-172-31-40-7 ~]$ sudo usermod -a -G apache ec2-user
[ec2-user@ip-172-31-40-7 ~]$ exit
```

  - **Step10.2)** To verify our membership in the apache group, we'll reconnect to our instance, and then run the following command:

```
ec2-user@ip-172-31-40-7:~
[ec2-user@ip-172-31-40-7 ~]$ groups
ec2-user adm wheel apache systemd-journal
[ec2-user@ip-172-31-40-7 ~]$
```

- **Step11)** Change the group ownership of /var/www and its contents to the apache group.

```
ec2-user@ip-172-31-40-7:~
[ec2-user@ip-172-31-40-7 ~]$ sudo chown -R ec2-user:apache /var/www
[ec2-user@ip-172-31-40-7 ~]$
```

- **Step12)** To add group write permissions and to set the group ID on future subdirectories, change the directory permissions of /var/www and its subdirectories.

```
ec2-user@ip-172-31-40-7:~
[ec2-user@ip-172-31-40-7 ~]$ sudo chown -R ec2-user:apache /var/www
[ec2-user@ip-172-31-40-7 ~]$
```

- **Step13)** To add group write permissions, recursively change the file permissions of /var/www and its subdirectories:

```
ec2-user@ip-172-31-40-7:~
[ec2-user@ip-172-31-40-7 ~]$ find /var/www -type f -exec sudo chmod 0664 {} \;
[ec2-user@ip-172-31-40-7 ~]$
```

NOTE: Now, ec2-user (and any future members of the apache group) can add, delete, and edit files in the Apache document root, enabling you to add content, such as a static website or a PHP application.

❖ Solution 1, Practical (Testing the LAMP server):

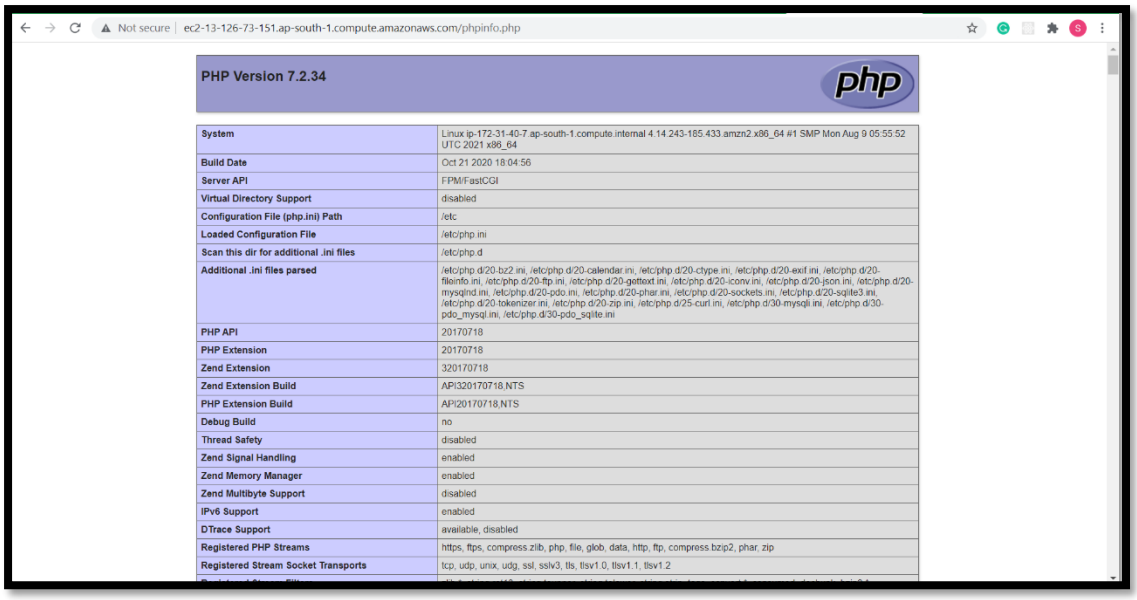
- **Brief:**

If our server is installed and running, and our file permissions are set correctly, our ec2-user account should be able to create a PHP file in the /var/www/html directory that is available from the internet.

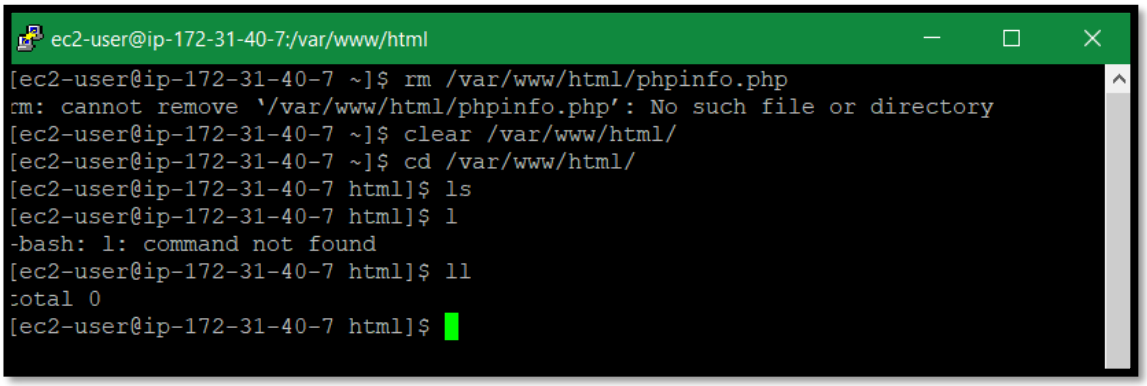
- **Step1)** Create a PHP file in the Apache document root.

```
ec2-user@ip-172-31-40-7:~
[ec2-user@ip-172-31-40-7 ~]$ echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php
[ec2-user@ip-172-31-40-7 ~]$
```

- **Step2)** In a web browser, type the URL of the file that we just created. This URL is the public DNS address of our instance followed by a forward slash and the file name. i.e., *http://ec2-13-126-73-151.ap-south-1.compute.amazonaws.com/phpinfo.php*



- **Step3)** Delete the phpinfo.php file. Although this can be useful information, it should not be broadcast to the internet for security reasons.



❖ Solution 1, Practical (Securing the database server):

- **Brief:**

The default installation of the MariaDB server has several features that are great for testing and development, but they should be disabled or removed for production servers. The *mysql_secure_installation* command walks us through the process of setting a root password and removing the insecure features from our installation. Even if we are not planning on using the MariaDB server, aws recommend performing this procedure.

➢ **Step1)** Start the MariaDB server.

```
ec2-user@ip-172-31-40-7:~
[ec2-user@ip-172-31-40-7 ~]$ sudo systemctl start mariadb
[ec2-user@ip-172-31-40-7 ~]$
```

➢ **Step2)** Run *mysql_secure_installation.*

```
ec2-user@ip-172-31-40-7:~
[ec2-user@ip-172-31-40-7 ~]$ sudo mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user.  If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.
```

- **Step2.1)** Type the current root password. By default, the root account does not have a password set. Press Enter.

```
ec2-user@ip-172-31-40-7:~
[ec2-user@ip-172-31-40-7 ~]$ sudo mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user.  If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.
```

- **Step2.2)** Type Y to set a password, and type a secure password twice.

```
ec2-user@ip-172-31-40-7:~
[ec2-user@ip-172-31-40-7 ~]$ sudo mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user.  If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.

Set root password? [Y/n] Y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
 ... Success!
```

- **Step2.3)**

  Type **Y** to remove the anonymous user accounts.

  Type **Y** to disable the remote root login.

  Type **Y** to remove the test database.

  Type **Y** to reload the privilege tables and save your changes.

```
ec2-user@ip-172-31-40-7:~
[ec2-user@ip-172-31-40-7 ~]$ sudo mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user.  If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.

Set root password? [Y/n] Y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
 ... Success!


By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them.  This is intended only for testing, and to make the installation
go a bit smoother.  You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] Y
 ... Success!

Normally, root should only be allowed to connect from 'localhost'.  This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] Y
 ... Success!

By default, MariaDB comes with a database named 'test' that anyone can
access.  This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] Y
 - Dropping test database...
 ... Success!
 - Removing privileges on test database...
 ... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] Y
 ... Success!

Cleaning up...

All done!  If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
[ec2-user@ip-172-31-40-7 ~]$
```

- **Step3)** (Optional) If you want the MariaDB server to start at every boot, type the following command.

```
ec2-user@ip-172-31-40-7:~
[ec2-user@ip-172-31-40-7 ~]$ sudo systemctl enable mariadb
Created symlink from /etc/systemd/system/multi-user.target.wants/mariadb.service
 to /usr/lib/systemd/system/mariadb.service.
[ec2-user@ip-172-31-40-7 ~]$
```

❖ Solution 1, Practical (Install phpMyAdmin):

➢ Brief:

phpMyAdmin is a web-based database management tool that you can use to view and edit the MySQL databases on your EC2 instance. Follow the steps below to install and configure phpMyAdmin on your Amazon Linux instance.

NOTE: AWS do not recommend using phpMyAdmin to access a LAMP server unless you have enabled SSL/TLS in Apache; otherwise, your database administrator password and other data are transmitted insecurely across the internet.

➢ **Step1)** Install the required dependencies.

```
ec2-user@ip-172-31-40-7:~
[ec2-user@ip-172-31-40-7 ~]$ sudo systemctl enable mariadb
Created symlink from /etc/systemd/system/multi-user.target.wants/mariadb.service
 to /usr/lib/systemd/system/mariadb.service.
[ec2-user@ip-172-31-40-7 ~]$
login as: ec2-user
Authenticating with public key "imported-openssh-key"
Last login: Mon Oct  4 16:40:18 2021 from 49.36.33.54

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-40-7 ~]$ clear
[ec2-user@ip-172-31-40-7 ~]$ sudo yum install php-mbstring php-xml -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                              | 3.7 kB     00:00
amzn2extra-docker                                       | 3.0 kB     00:00
amzn2extra-lamp-mariadb10.2-php7.2                      | 3.0 kB     00:00
amzn2extra-php7.2                                       | 3.0 kB     00:00
Resolving Dependencies
--> Running transaction check
---> Package php-mbstring.x86_64 0:7.2.34-1.amzn2 will be installed
--> Processing Dependency: libonig.so.2()(64bit) for package: php-mbstring-7.2.3
```

➢ **Step2)** Restart Apache.

```
ec2-user@ip-172-31-40-7:~
[ec2-user@ip-172-31-40-7 ~]$ sudo systemctl restart httpd
[ec2-user@ip-172-31-40-7 ~]$
```

- **Step3)** Restart php-fpm.

```
ec2-user@ip-172-31-40-7:~
[ec2-user@ip-172-31-40-7 ~]$ sudo systemctl restart php-fpm
[ec2-user@ip-172-31-40-7 ~]$
```

- **Step4)** Navigate to the Apache document root at /var/www/html.

```
ec2-user@ip-172-31-40-7:/var/www/html
[ec2-user@ip-172-31-40-7 ~]$ cd /var/www/html
[ec2-user@ip-172-31-40-7 html]$
```

- **Step5)** Select a source package for the latest phpMyAdmin release from https://www.phpmyadmin.net/downloads. To download the file directly to your instance, copy the link and paste it into a **wget** command, as in this example:

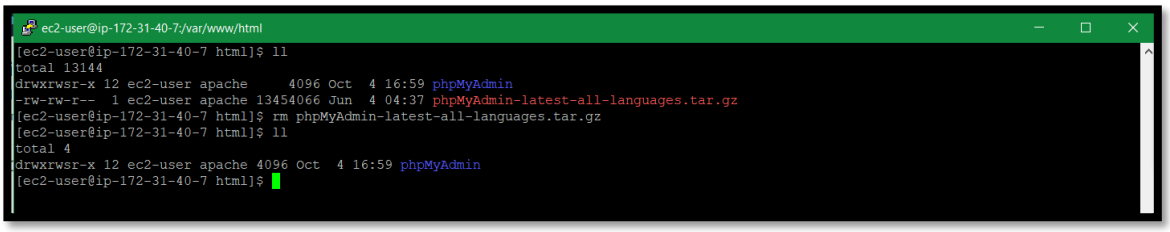
```
ec2-user@ip-172-31-40-7:/var/www/html
[ec2-user@ip-172-31-40-7 html]$ wget https://www.phpmyadmin.net/downloads/phpMyAdmin-latest-all-languages.tar.gz
--2021-10-04 16:57:16--  https://www.phpmyadmin.net/downloads/phpMyAdmin-latest-all-languages.tar.gz
Resolving www.phpmyadmin.net (www.phpmyadmin.net)... 89.187.163.85, 89.187.163.88, 89.187.163.69, ...
Connecting to www.phpmyadmin.net (www.phpmyadmin.net)|89.187.163.85|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://files.phpmyadmin.net/phpMyAdmin/5.1.1/phpMyAdmin-5.1.1-all-languages.tar.gz [following]
--2021-10-04 16:57:17--  https://files.phpmyadmin.net/phpMyAdmin/5.1.1/phpMyAdmin-5.1.1-all-languages.tar.gz
Resolving files.phpmyadmin.net (files.phpmyadmin.net)... 89.187.163.87, 89.187.163.70, 89.187.163.72, ...
Connecting to files.phpmyadmin.net (files.phpmyadmin.net)|89.187.163.87|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 13454066 (13M) [application/octet-stream]
Saving to: 'phpMyAdmin-latest-all-languages.tar.gz'

100%[===================================================================>] 13,454,066  7.53MB/s   in 1.7s

2021-10-04 16:57:20 (7.53 MB/s) - 'phpMyAdmin-latest-all-languages.tar.gz' saved [13454066/13454066]

[ec2-user@ip-172-31-40-7 html]$
```

- **Step6)** Create a phpMyAdmin folder and extract the package into it with the following command.

```
ec2-user@ip-172-31-40-7:/var/www/html
[ec2-user@ip-172-31-40-7 html]$ mkdir phpMyAdmin && tar -xvzf phpMyAdmin-latest-all-languages.tar.gz -C phpMyAdmin --strip-components
```

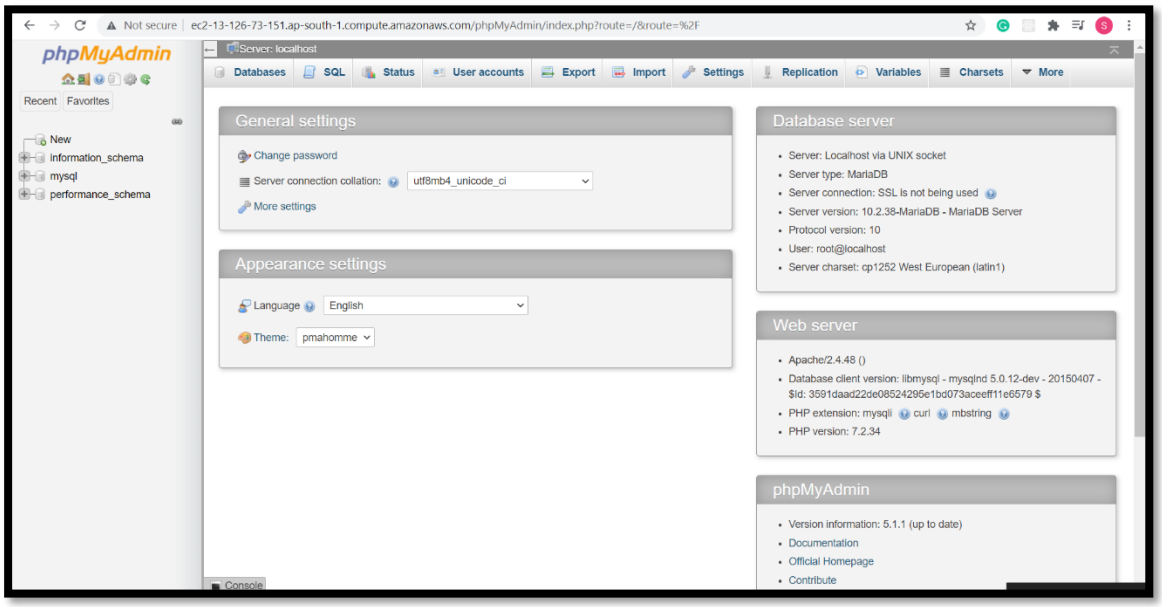
```
ec2-user@ip-172-31-40-7:/var/www/html
c2-user@ip-172-31-40-7 html]$ ll
tal 13144
wxrwsr-x 12 ec2-user apache     4096 Oct  4 16:59 phpMyAdmin
w-rw-r--  1 ec2-user apache 13454066 Jun  4 04:37 phpMyAdmin-latest-all-languages.tar.gz
c2-user@ip-172-31-40-7 html]$
```

- **Step7)** Delete the *phpMyAdmin-latest-all-languages.tar.gz* tarball.



- **Step8)** In a web browser, type the URL of our phpMyAdmin installation. This URL is the public DNS address (or the public IP address) of our instance followed by a forward slash and the name of our installation directory. i.e., *http://ec2-13-126-73-151.ap-south-1.compute.amazonaws.com/phpMyAdmin*

- **Step9)** Log in to your phpMyAdmin installation with the root user name and the MySQL root password you created earlier.



- **Step10)** Our installation must still be configured before you put it into service. Aws suggest that we begin by manually creating the configuration file, as follows:

  a. To start with a minimal configuration file, use your favorite text editor to create a new file, and then copy the contents of `config.sample.inc.php` into it.

```
ec2-user@ip-172-31-40-7:/var/www/html/phpMyAdmin
[ec2-user@ip-172-31-40-7 ~]$ ll
total 0
[ec2-user@ip-172-31-40-7 ~]$ cd /var/www/html
[ec2-user@ip-172-31-40-7 html]$ ll
total 4
drwxrwsr-x 13 ec2-user apache 4096 Oct  4 17:07 phpMyAdmin
[ec2-user@ip-172-31-40-7 html]$ cd phpMyAdmin
[ec2-user@ip-172-31-40-7 phpMyAdmin]$ ll
total 660
-rw-r--r--  1 ec2-user apache     41 Jun  4 04:15 babel.config.json
-rw-r--r--  1 ec2-user apache  49416 Jun  4 04:15 ChangeLog
-rw-r--r--  1 ec2-user apache   4064 Jun  4 04:17 composer.json
-rw-r--r--  1 ec2-user apache 204120 Jun  4 04:14 composer.lock
-rw-r--r--  1 ec2-user apache   4474 Jun  4 04:15 config.sample.inc.php
-rw-r--r--  1 ec2-user apache   2587 Jun  4 04:15 CONTRIBUTING.md
drwxr-sr-x  3 ec2-user apache     18 Jun  4 04:17 doc
drwxr-sr-x  2 ec2-user apache     99 Jun  4 04:16 examples
-rw-r--r--  1 ec2-user apache  22486 Jun  4 04:16 favicon.ico
-rw-r--r--  1 ec2-user apache    413 Jun  4 04:16 index.php
drwxr-sr-x  5 ec2-user apache     63 Jun  4 04:16 js
drwxr-sr-x  5 ec2-user apache   4096 Jun  4 04:16 libraries
-rw-r--r--  1 ec2-user apache  18092 Jun  4 04:15 LICENSE
drwxr-sr-x 45 ec2-user apache   4096 Jun  4 04:17 locale
-rw-r--r--  1 ec2-user apache   2252 Jun  4 04:16 package.json
-rw-r--r--  1 ec2-user apache   1034 Jun  4 04:16 print.css
-rw-r--r--  1 ec2-user apache   1520 Jun  4 04:15 README
-rw-r--r--  1 ec2-user apache     29 Jun  4 04:17 RELEASE-DATE-5.1.1
-rw-r--r--  1 ec2-user apache     26 Jun  4 04:16 robots.txt
drwxr-sr-x  3 ec2-user apache     90 Jun  4 04:16 setup
-rw-r--r--  1 ec2-user apache   1354 Jun  4 04:16 show_config_errors.php
drwxr-sr-x  2 ec2-user apache    141 Jun  4 04:16 sql
drwxr-sr-x 25 ec2-user apache   4096 Jun  4 04:16 templates
drwxr-sr-x  5 ec2-user apache     66 Jun  4 04:16 themes
drwxr-s---  3 apache   apache     18 Oct  4 17:07 tmp
-rw-r--r--  1 ec2-user apache   1613 Jun  4 04:17 url.php
drwxr-sr-x 17 ec2-user apache    254 Jun  4 04:17 vendor
-rw-r--r--  1 ec2-user apache 306579 Jun  4 04:17 yarn.lock
[ec2-user@ip-172-31-40-7 phpMyAdmin]$
```

  b. Save the file as `config.inc.php` in the phpMyAdmin directory that contains `index.php`.

```
ec2-user@ip-172-31-40-7:/var/www/html/phpMyAdmin
[ec2-user@ip-172-31-40-7 phpMyAdmin]$ cp config.sample.inc.php config.inc.php
[ec2-user@ip-172-31-40-7 phpMyAdmin]$ ll
total 668
-rw-r--r--  1 ec2-user apache     41 Jun  4 04:15 babel.config.json
-rw-r--r--  1 ec2-user apache  49416 Jun  4 04:15 ChangeLog
-rw-r--r--  1 ec2-user apache   4064 Jun  4 04:17 composer.json
-rw-r--r--  1 ec2-user apache 204120 Jun  4 04:14 composer.lock
-rw-r--r--  1 ec2-user apache   4474 Oct  4 17:22 config.inc.php
-rw-r--r--  1 ec2-user apache   4474 Jun  4 04:15 config.sample.inc.php
-rw-r--r--  1 ec2-user apache   2587 Jun  4 04:15 CONTRIBUTING.md
drwxr-sr-x  3 ec2-user apache     18 Jun  4 04:17 doc
drwxr-sr-x  2 ec2-user apache     99 Jun  4 04:16 examples
-rw-r--r--  1 ec2-user apache  22486 Jun  4 04:16 favicon.ico
-rw-r--r--  1 ec2-user apache    413 Jun  4 04:16 index.php
drwxr-sr-x  5 ec2-user apache     63 Jun  4 04:16 js
drwxr-sr-x  5 ec2-user apache   4096 Jun  4 04:16 libraries
-rw-r--r--  1 ec2-user apache  18092 Jun  4 04:15 LICENSE
drwxr-sr-x 45 ec2-user apache   4096 Jun  4 04:17 locale
-rw-r--r--  1 ec2-user apache   2252 Jun  4 04:16 package.json
-rw-r--r--  1 ec2-user apache   1034 Jun  4 04:16 print.css
-rw-r--r--  1 ec2-user apache   1520 Jun  4 04:15 README
-rw-r--r--  1 ec2-user apache     29 Jun  4 04:17 RELEASE-DATE-5.1.1
-rw-r--r--  1 ec2-user apache     26 Jun  4 04:16 robots.txt
drwxr-sr-x  3 ec2-user apache     90 Jun  4 04:16 setup
-rw-r--r--  1 ec2-user apache   1354 Jun  4 04:16 show_config_errors.php
drwxr-sr-x  2 ec2-user apache    141 Jun  4 04:16 sql
drwxr-sr-x 25 ec2-user apache   4096 Jun  4 04:16 templates
drwxr-sr-x  5 ec2-user apache     66 Jun  4 04:16 themes
drwxr-s---  3 apache   apache     18 Oct  4 17:07 tmp
-rw-r--r--  1 ec2-user apache   1613 Jun  4 04:17 url.php
drwxr-sr-x 17 ec2-user apache    254 Jun  4 04:17 vendor
-rw-r--r--  1 ec2-user apache 306579 Jun  4 04:17 yarn.lock
[ec2-user@ip-172-31-40-7 phpMyAdmin]$
```

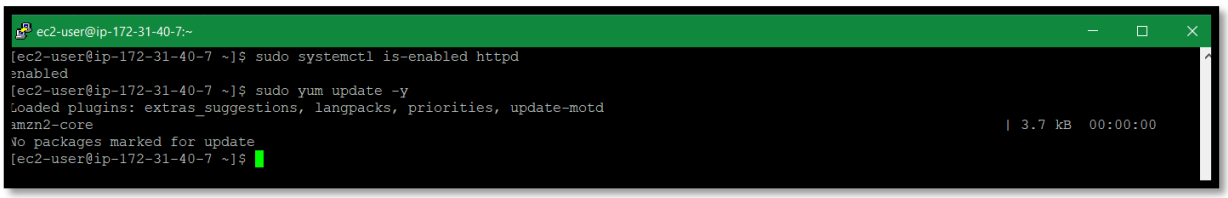
❖ Solution 1, Configure SSL/TLS on Amazon Linux 2:

- **Brief:**

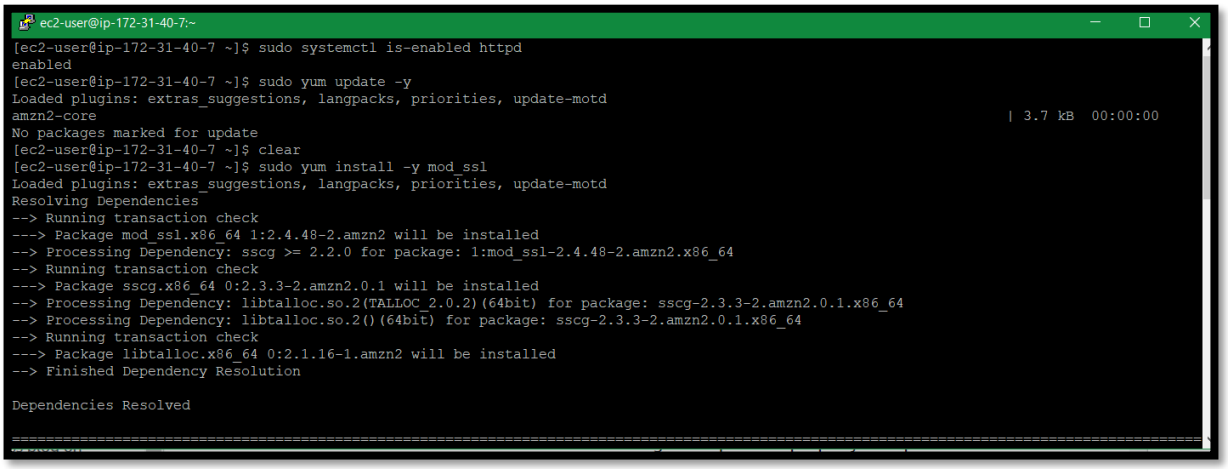
  Secure Sockets Layer/Transport Layer Security (SSL/TLS) creates an encrypted channel between a web server and web client that protects data in transit from being eavesdropped on. While web browsers still support SSL, its successor protocol TLS is less vulnerable to attack. Amazon Linux 2 disables server-side support for all versions of SSL by default

- **Step1)** To enable TLS on our server we would Connect to your instance and confirm that Apache is running.

- **Step2)** To ensure that all of your software packages are up to date, perform a quick software update on your instance. The -y option installs the updates without asking for confirmation.



- **Step3)** Now that your instance is current, add TLS support by installing the Apache module mod_ssl.



NOTE: Your instance now has the following files that you use to configure your secure server and create a certificate for testing:

- /etc/httpd/conf.d/ssl.conf

  The configuration file for mod_ssl. It contains *directives* telling Apache where to find encryption keys and certificates, the TLS protocol versions to allow, and the encryption ciphers to accept.

- /etc/pki/tls/certs/make-dummy-cert

  A script to generate a self-signed X.509 certificate and private key for your server host. This certificate is useful for testing that Apache is properly set up to use TLS. Because it offers no proof of identity, it should not be used in production. If used in production, it triggers warnings in Web browsers.

➢ **Step4)** Run the script to generate a self-signed dummy certificate and key for testing.

```
ec2-user@ip-172-31-40-7:/etc/pki/tls/certs
[ec2-user@ip-172-31-40-7 certs]$ cd /etc/pki/tls/certs
[ec2-user@ip-172-31-40-7 certs]$ sudo ./make-dummy-cert localhost.crt
[ec2-user@ip-172-31-40-7 certs]$
```

➢ **Step5)** Open the /etc/httpd/conf.d/ssl.conf file using your favorite text editor (such as **vim** or **nano**) and comment out the following line, because the self-signed dummy certificate also contains the key. If you do not comment out this line before you complete the next step, the Apache service fails to start.

```
ec2-user@ip-172-31-40-7:/etc/pki/tls/certs
[ec2-user@ip-172-31-40-7 certs]$ vi /etc/httpd/conf.d/ssl.conf.
[ec2-user@ip-172-31-40-7 certs]$ vi /etc/httpd/conf.d/ssl.conf
[ec2-user@ip-172-31-40-7 certs]$ sudo vi /etc/httpd/conf.d/ssl.conf
[ec2-user@ip-172-31-40-7 certs]$
```

```
ec2-user@ip-172-31-40-7:/etc/pki/tls/certs
ErrorLog logs/ssl_error_log
TransferLog logs/ssl_access_log
LogLevel warn

#   SSL Engine Switch:
#   Enable/Disable SSL for this virtual host.
SSLEngine on

#   SSL Protocol support:
# List the enable protocol levels with which clients will be able to
# connect.  Disable SSLv2 access by default:
SSLProtocol all -SSLv3

#   SSL Cipher Suite:
#   List the ciphers that the client is permitted to negotiate.
#   See the mod_ssl documentation for a complete list.
SSLCipherSuite HIGH:MEDIUM:!aNULL:!MD5:!SEED:!IDEA

#   Speed-optimized SSL Cipher configuration:
#   If speed is your main concern (on busy HTTPS servers e.g.),
#   you might want to force clients to specific, performance
#   optimized ciphers. In this case, prepend those ciphers
#   to the SSLCipherSuite list, and enable SSLHonorCipherOrder.
#   Caveat: by giving precedence to RC4-SHA and AES128-SHA
#   (as in the example below), most connections will no longer
#   have perfect forward secrecy - if the server's key is
#   compromised, captures of past or future traffic must be
#   considered compromised, too.
#SSLCipherSuite RC4-SHA:AES128-SHA:HIGH:MEDIUM:!aNULL:!MD5
#SSLHonorCipherOrder on

#   Server Certificate:
# Point SSLCertificateFile at a PEM encoded certificate.  If
# the certificate is encrypted, then you will be prompted for a
# pass phrase.  Note that a kill -HUP will prompt again.  A new
# certificate can be generated using the genkey(1) command.
SSLCertificateFile /etc/pki/tls/certs/localhost.crt

#   Server Private Key:
#   If the key is not combined with the certificate, use this
#   directive to point at the key file.  Keep in mind that if
#   you've both a RSA and a DSA private key you can configure
#   both in parallel (to also allow the use of DSA ciphers, etc.)
#SSLCertificateKeyFile /etc/pki/tls/private/localhost.key

#   Server Certificate Chain:
#   Point SSLCertificateChainFile at a file containing the
#   concatenation of PEM encoded CA certificates which form the
#   certificate chain for the server certificate. Alternatively
-- INSERT --
```

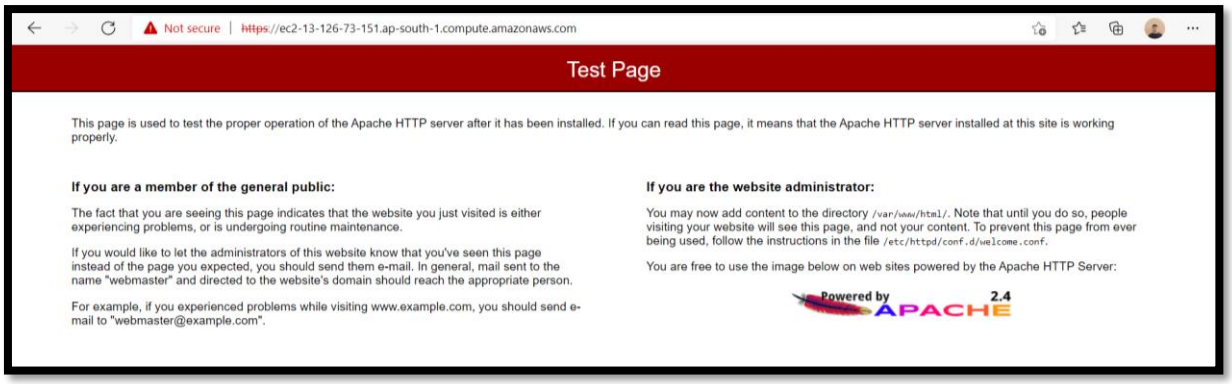
- **Step6)** Restart Apache.

```
ec2-user@ip-172-31-40-7:/etc/pki/tls/certs
[ec2-user@ip-172-31-40-7 certs]$ sudo systemctl restart httpd
[ec2-user@ip-172-31-40-7 certs]$
```

- **Step7)** Your Apache web server should now support HTTPS (secure HTTP) over port 443. Test it by entering the IP address or fully qualified domain name of your EC2 instance into a browser URL bar with the prefix **https://**.

  Because you are connecting to a site with a self-signed, untrusted host certificate, your browser may display a series of security warnings. Override the warnings and proceed to the site.

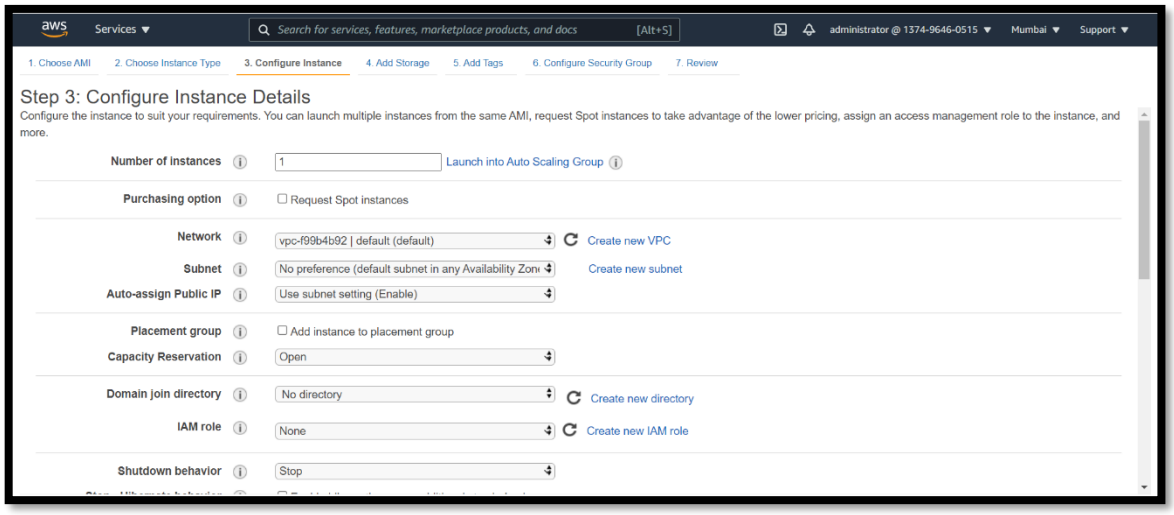
  If the default Apache test page opens, it means that you have successfully configured TLS on your server. All data passing between the browser and server is now encrypted.

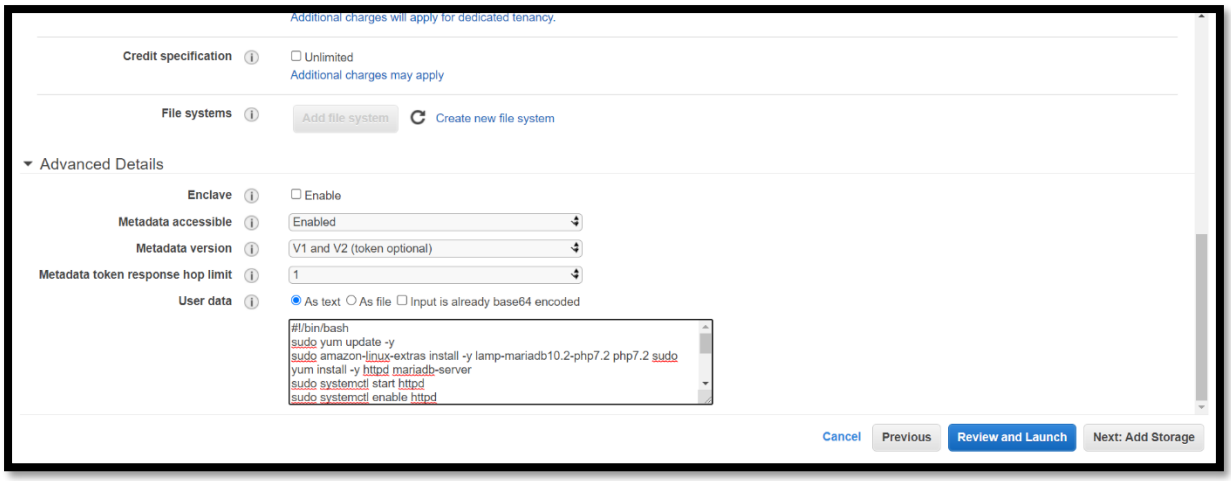


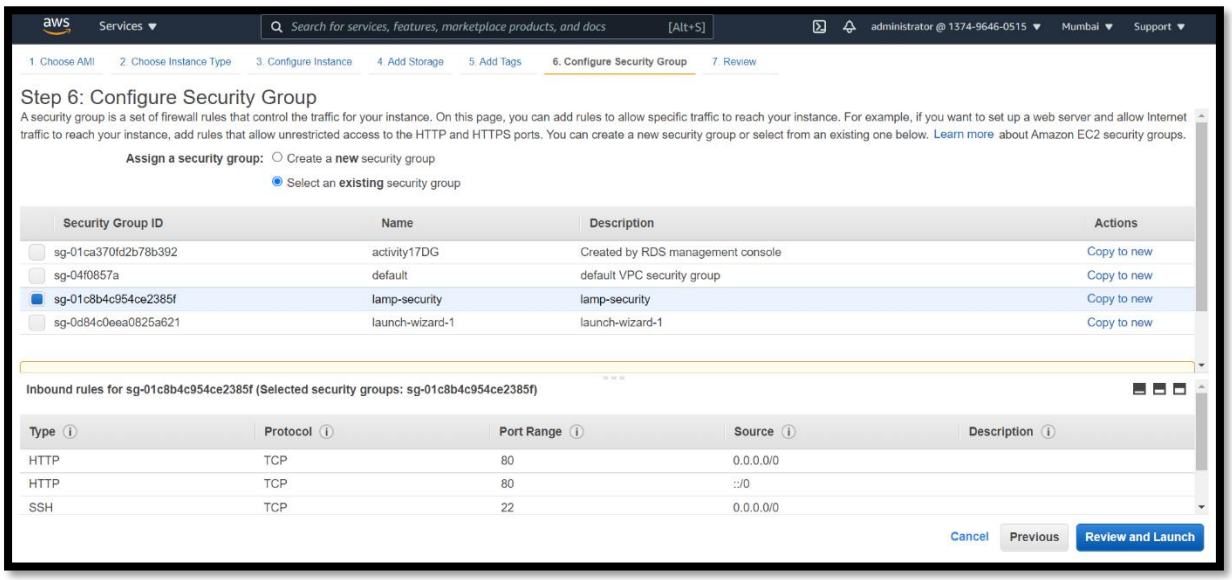
❖ Solution2 (User-Script):

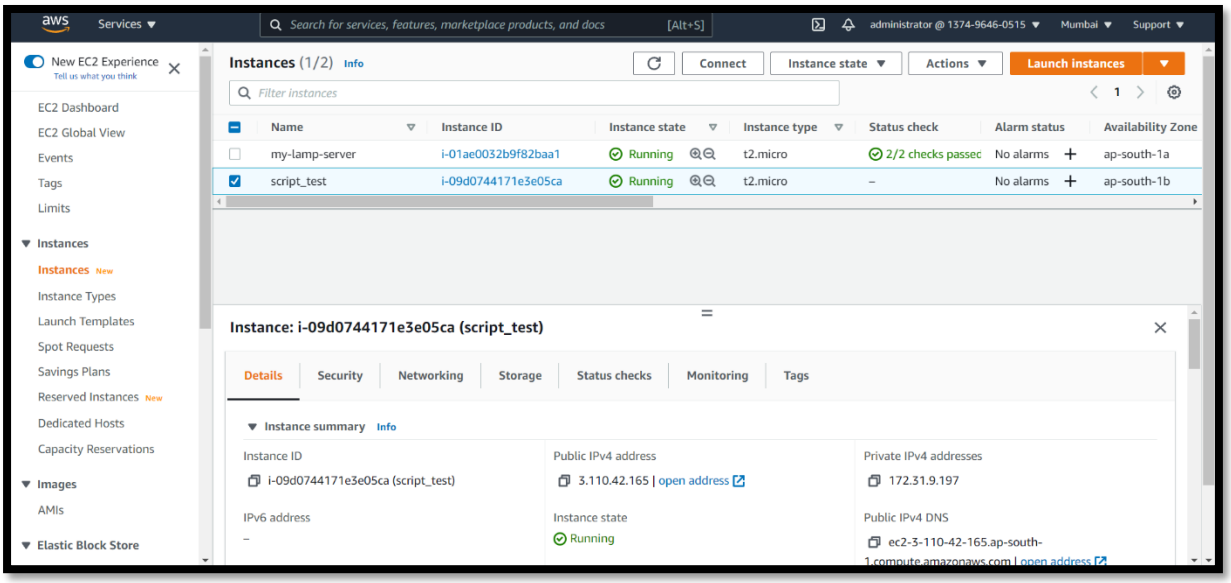
NOTE: In below Bash-Script SSL/TSL enabling is not done. We can add it later.

```
#!/bin/bash
sudo yum update -y
sudo amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2
sudo yum install -y httpd mariadb-server
sudo systemctl start httpd
sudo systemctl enable httpd
sudo usermod -a -G apache ec2-user
sudo chown -R ec2-user:apache /var/www
sudo find /var/www -type f -exec sudo chmod 0664 {} \;
```

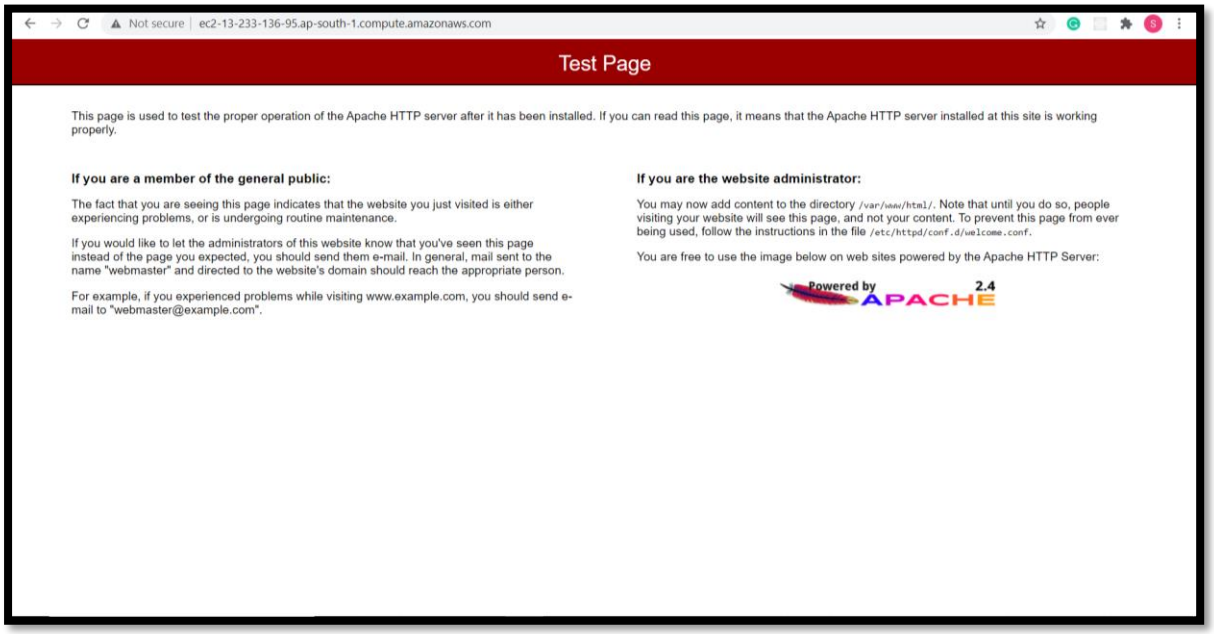



Additional charges will apply for dedicated tenancy.
Credit specification
Unlimited
Additional charges may apply
File systems
Add file system
Create new file system
Advanced Details
Enclave
Enable
Metadata accessible
Enabled
Metadata version
V1 and V2 (token optional)
Metadata token response hop limit
1
User data
As text
As file
Input is already base64 encoded
#!/bin/bash
sudo yum update -y
sudo amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2 sudo
yum install -y httpd mariadb-server
sudo systemctl start httpd
sudo systemctl enable httpd
Cancel
Previous
Review and Launch
Next: Add Storage

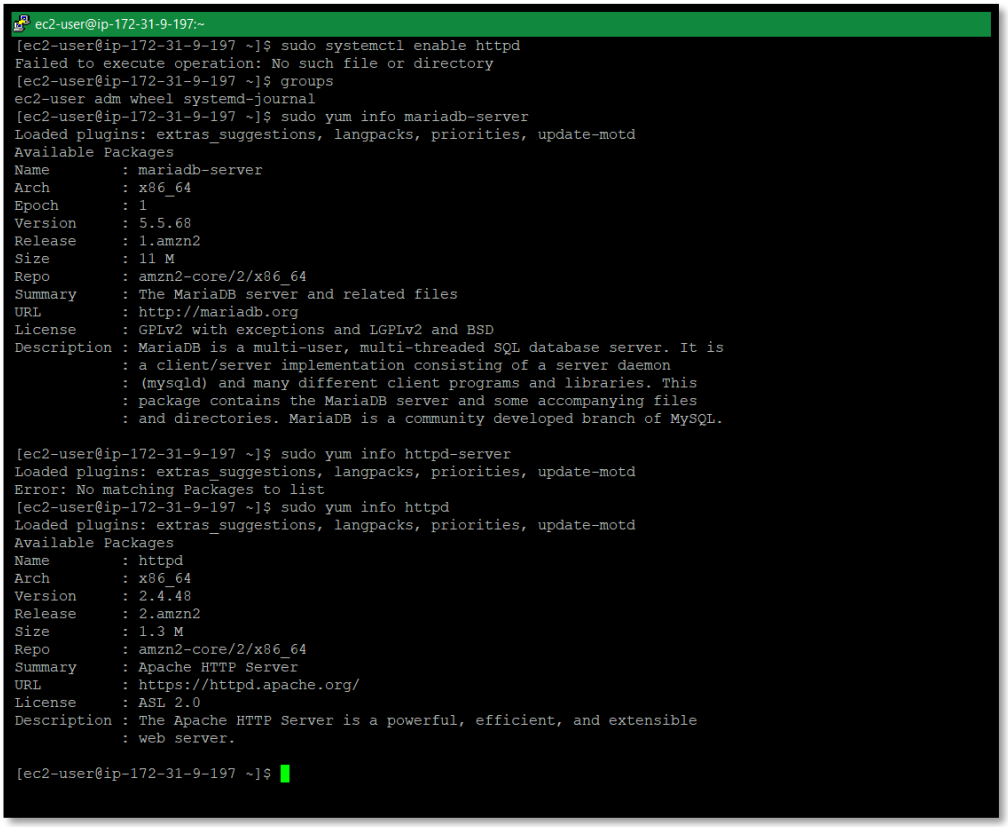

aws
Services
Search for services, features, marketplace products, and docs
[Alt+S]
administrator @ 1374-9646-0515
Mumbai
Support
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 6: Configure Security Group
A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.
Assign a security group:
Create a new security group
Select an existing security group
Security Group ID
Name
Description
Actions
sg-01ca370fd2b78b392
activity17DG
Created by RDS management console
Copy to new
sg-04f0857a
default
default VPC security group
Copy to new
sg-01c8b4c954ce2385f
lamp-security
lamp-security
Copy to new
sg-0d84c0eea0825a621
launch-wizard-1
launch-wizard-1
Copy to new
Inbound rules for sg-01c8b4c954ce2385f (Selected security groups: sg-01c8b4c954ce2385f)
Type
Protocol
Port Range
Source
Description
HTTP
TCP
80
0.0.0.0/0
HTTP
TCP
80
::/0
SSH
TCP
22
0.0.0.0/0
Cancel
Previous
Review and Launch


aws
Services
Search for services, features, marketplace products, and docs
[Alt+S]
administrator @ 1374-9646-0515
Mumbai
Support
New EC2 Experience
Tell us what you think
EC2 Dashboard
EC2 Global View
Events
Tags
Limits
Instances
Instances New
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances New
Dedicated Hosts
Capacity Reservations
Images
AMIs
Elastic Block Store
Instances (1/2) Info
Connect
Instance state
Actions
Launch instances
Filter instances
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
my-lamp-server
i-01ae0032b9f82baa1
Running
t2.micro
2/2 checks passed
No alarms
ap-south-1a
script_test
i-09d0744171e3e05ca
Running
t2.micro
–
No alarms
ap-south-1b
Instance: i-09d0744171e3e05ca (script_test)
Details
Security
Networking
Storage
Status checks
Monitoring
Tags
Instance summary Info
Instance ID
i-09d0744171e3e05ca (script_test)
Public IPv4 address
3.110.42.165 | open address
Private IPv4 addresses
172.31.9.197
IPv6 address
–
Instance state
Running
Public IPv4 DNS
ec2-3-110-42-165.ap-south-1.compute.amazonaws.com | open address

- Test our web server. In a web browser, type the public DNS address (or the public IP address) of our instance. If there is no content in /var/www/html, we should see the Apache test page. We can get the public DNS for our instance using the Amazon EC2 console (check the Public DNS column; if this column is hidden, choose Show/Hide Columns (the gear-shaped icon) and choose Public DNS).



- We can also check from putty console whether all our packages are installed in our new machine or not



```
ec2-user@ip-172-31-9-197:~
[ec2-user@ip-172-31-9-197 ~]$ sudo systemctl enable httpd
Failed to execute operation: No such file or directory
[ec2-user@ip-172-31-9-197 ~]$ groups
ec2-user adm wheel systemd-journal
[ec2-user@ip-172-31-9-197 ~]$ sudo yum info mariadb-server
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Available Packages
Name        : mariadb-server
Arch        : x86_64
Epoch       : 1
Version     : 5.5.68
Release     : 1.amzn2
Size        : 11 M
Repo        : amzn2-core/2/x86_64
Summary     : The MariaDB server and related files
URL         : http://mariadb.org
License     : GPLv2 with exceptions and LGPLv2 and BSD
Description : MariaDB is a multi-user, multi-threaded SQL database server. It is
            : a client/server implementation consisting of a server daemon
            : (mysqld) and many different client programs and libraries. This
            : package contains the MariaDB server and some accompanying files
            : and directories. MariaDB is a community developed branch of MySQL.

[ec2-user@ip-172-31-9-197 ~]$ sudo yum info httpd-server
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Error: No matching Packages to list
[ec2-user@ip-172-31-9-197 ~]$ sudo yum info httpd
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Available Packages
Name        : httpd
Arch        : x86_64
Version     : 2.4.48
Release     : 2.amzn2
Size        : 1.3 M
Repo        : amzn2-core/2/x86_64
Summary     : Apache HTTP Server
URL         : https://httpd.apache.org/
License     : ASL 2.0
Description : The Apache HTTP Server is a powerful, efficient, and extensible
            : web server.

[ec2-user@ip-172-31-9-197 ~]$
```