



LAB 2

Popular Cloud Computing Services

Fullname: Nguyen Chi Hoang Minh

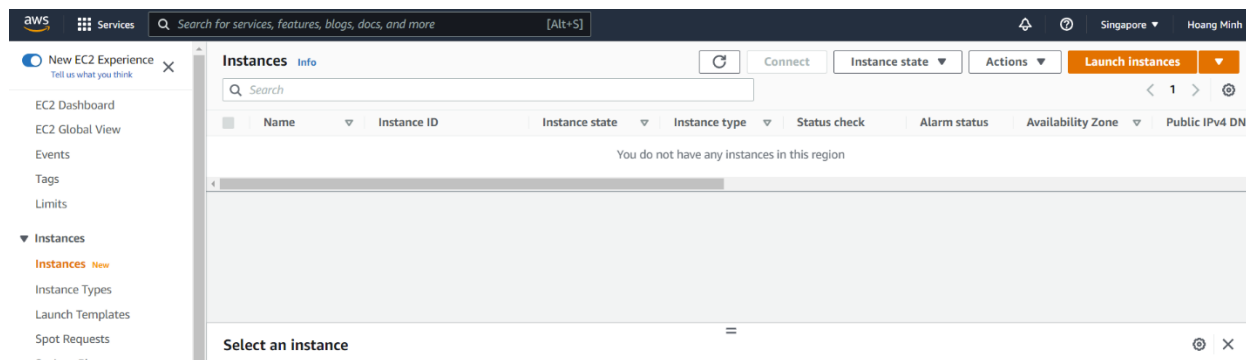
Student ID: B1809707

1. Amazon Web Services (AWS)

1.1. Launch a Linux Virtual Machine using EC2

To launch an instance

From the console dashboard, choose Launch Instance



Choose an Amazon Machine Image (AMI)

Choose an Instance Type

Choose Review and Launch to let the wizard complete the other configuration settings.

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

Improve your instances' security. Your security group, launch-wizard-2, is open to the world.

Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only. You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

▼ AMI Details [Edit AMI](#)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type - ami-02a45d709a415958a

Free tier eligible: Amazon Linux 2 comes with five years support. It provides Linux kernel 5.10 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is n...

Root Device Type: ebs Virtualization type: hvm

▼ Instance Type [Edit instance type](#)

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	-	1	1	EBS only	-	Low to Moderate

▼ Security Groups [Edit security groups](#)

Security group name: launch-wizard-2

Description: launch-wizard-2 created 2022-02-28T23:42:32.706+07:00

[Cancel](#) [Previous](#) [Launch](#)

Choose Edit security groups.

Finally, choose Launch.

Step 7: Review Instance Launch

▼ Security Groups [Edit security groups](#)

Security group name: launch-wizard-2

Description: launch-wizard-2 created 2022-02-28T23:42:32.706+07:00

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	0.0.0.0/0	
HTTP	TCP	80	0.0.0.0/0	
HTTP	TCP	80	:::0	
Custom TCP Rule	TCP	8080	0.0.0.0/0	
Custom TCP Rule	TCP	8080	:::0	

► Instance Details [Edit instance details](#)

▼ Storage [Edit storage](#)

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/xvda	snap-03591dfbda535ac8d	8	gp2	100 / 3000	N/A	Yes	Not Encrypted

► Tags [Edit tags](#)

[Cancel](#) [Previous](#) [Launch](#)

When prompted for a key pair, select Choose an existing key pair, then select the key pair that created.

Select an existing key pair or create a new key pair ✕

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance. Amazon EC2 supports ED25519 and RSA key pair types.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Select a key pair

☐ I acknowledge that I have access to the corresponding private key file, and that without this file, I won't be able to log into my instance.

Cancel
Launch Instances

When ready, select the acknowledgement check box, and then choose Launch Instances.

On the Instances screen, view the status of the launch.

The screenshot shows the AWS Management Console interface. On the left, there is a navigation menu with options like 'EC2 Dashboard', 'Events', 'Tags', 'Limits', 'Instances', 'Images', and 'Elastic Block Store'. The main content area is titled 'Instances (1)' and shows a table with the following data:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
b1809707_ins...	i-0a2d482af22b3e598	Running	t2.micro	2/2 checks passed	No alarms	ap-southeast-1b	ec2-54-251-146-

Below the table, there is a section titled 'Select an instance' with a search bar and a list of instances.

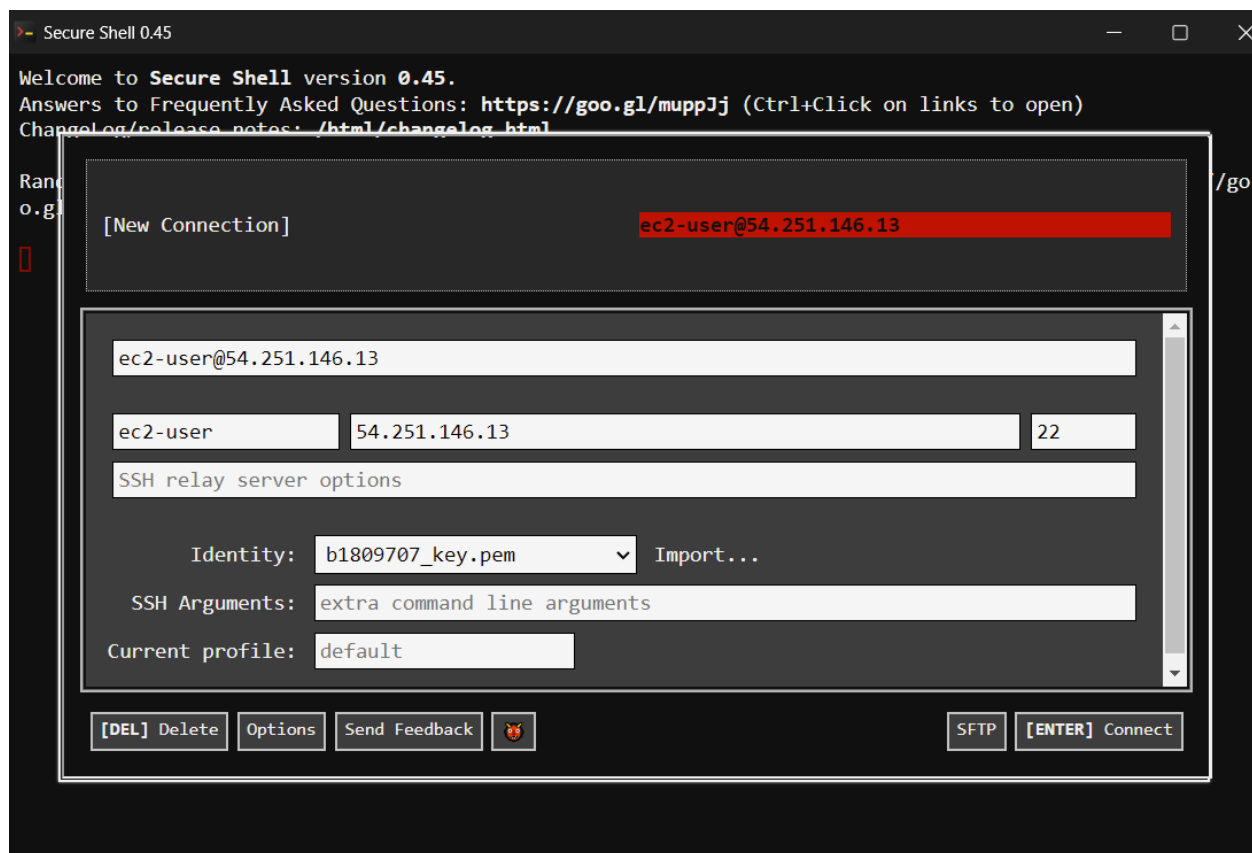
To connect to an instance

Get the public DNS name and user name to connect to instance

Locate the private key

Installing the Secure Shell Chrome Extension

<https://chrome.google.com/webstore/detail/secure-shell/iodihamcpbpeioajjeobimgagajmlibd>



1.2. Install a LAMP Web Server with the Amazon Linux 2 AMI

Prepare the LAMP server

```
sudo yum update -y
```

```
ec2-user@ip-172-31-25-103:~  
Connecting to ec2-user@54.251.146.13...  
The authenticity of host '54.251.146.13 (54.251.146.13)' can't be established.  
ED25519 key fingerprint is SHA256:8TPmW3mo1WUnYpACSU9C+SnhWtcsb4022Rkcgwzv0vI.  
This key is not known by any other names  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '54.251.146.13' (ED25519) to the list of known hosts.  
  
  _ | _ | _ )  
  _ | ( _ /  Amazon Linux 2 AMI  
  _ | \ _ | _ |  
  
https://aws.amazon.com/amazon-linux-2/  
8 package(s) needed for security, out of 14 available  
Run "sudo yum update" to apply all updates.  
[ec2-user@ip-172-31-25-103 ~]$ ls  
[ec2-user@ip-172-31-25-103 ~]$ sudo yum update -y  
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd  
amzn2-core | 3.7 kB 00:00:00  
Resolving Dependencies  
--> Running transaction check  
--> Package aws-cfn-bootstrap.noarch 0:2.0-9.amzn2 will be updated  
--> Package aws-cfn-bootstrap.noarch 0:2.0-10.amzn2 will be an update  
--> Package ca-certificates.noarch 0:2021.2.50-72.amzn2.0.2 will be updated  
--> Package ca-certificates.noarch 0:2021.2.50-72.amzn2.0.3 will be an update  
--> Package cloud-init.noarch 0:19.3-44.amzn2 will be updated  
--> Package cloud-init.noarch 0:19.3-45.amzn2 will be an update  
--> Package ec2-net-utils.noarch 0:1.5-3.amzn2 will be updated  
--> Package ec2-net-utils.noarch 0:1.6-1.amzn2 will be an update  
--> Package ec2-utils.noarch 0:1.2-45.amzn2 will be updated  
--> Package ec2-utils.noarch 0:1.2-47.amzn2 will be an update
```

```
sudo amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2
```

```
ec2-user@ip-172-31-25-103:~  
[ec2-user@ip-172-31-25-103 ~]$ sudo amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2  
Installing php-pdo, php-mysqldb, php-fpm, php-cli, php-json, mariadb  
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd  
Cleaning repos: amzn2-core amzn2extra-docker amzn2extra-kernel-5.10 amzn2extra-lamp-mariadb10.2-php7.2  
: amzn2extra-php7.2  
17 metadata files removed  
6 sqlite files removed  
0 metadata files removed  
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd  
amzn2-core | 3.7 kB 00:00:00  
amzn2extra-docker | 3.0 kB 00:00:00  
amzn2extra-kernel-5.10 | 3.0 kB 00:00:00  
amzn2extra-lamp-mariadb10.2-php7.2 | 3.0 kB 00:00:00  
amzn2extra-php7.2 | 3.0 kB 00:00:00  
(1/11): amzn2-core/2/x86_64/group_gz | 2.5 kB 00:00:00  
(2/11): amzn2-core/2/x86_64/updateinfo | 447 kB 00:00:00  
(3/11): amzn2extra-docker/2/x86_64/updateinfo | 4.7 kB 00:00:00  
(4/11): amzn2extra-kernel-5.10/2/x86_64/primary_db | 6.9 MB 00:00:00  
(5/11): amzn2extra-lamp-mariadb10.2-php7.2/2/x86_64/updateinfo | 76 B 00:00:00  
(6/11): amzn2extra-kernel-5.10/2/x86_64/updateinfo | 11 kB 00:00:00  
(7/11): amzn2extra-docker/2/x86_64/primary_db | 83 kB 00:00:00  
(8/11): amzn2extra-php7.2/2/x86_64/updateinfo | 76 B 00:00:00  
(9/11): amzn2extra-lamp-mariadb10.2-php7.2/2/x86_64/primary_db | 506 kB 00:00:00  
(10/11): amzn2extra-php7.2/2/x86_64/primary_db | 580 kB 00:00:00  
(11/11): amzn2-core/2/x86_64/primary_db | 60 MB 00:00:01  
Resolving Dependencies  
--> Running transaction check  
--> Package mariadb.x86_64 3:10.2.38-1.amzn2.0.1 will be installed  
--> Processing Dependency: mariadb-libs(x86-64) = 3:10.2.38-1.amzn2.0.1 for package: 3:mariadb-10.2.38-1.amzn2.0.1.x86_64
```

```
sudo yum install -y httpd mariadb-server
```

```
ec2-user@ip-172-31-25-103:~  
[ec2-user@ip-172-31-25-103 ~]$ 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 0:2.4.52-1.amzn2 will be installed  
--> Processing Dependency: httpd-tools = 2.4.52-1.amzn2 for package: httpd-2.4.52-1.amzn2.x86_64  
--> Processing Dependency: httpd filesystem = 2.4.52-1.amzn2 for package: httpd-2.4.52-1.amzn2.x86_64  
--> Processing Dependency: system-logos-httpd for package: httpd-2.4.52-1.amzn2.x86_64  
--> Processing Dependency: mod_http2 for package: httpd-2.4.52-1.amzn2.x86_64  
--> Processing Dependency: httpd filesystem for package: httpd-2.4.52-1.amzn2.x86_64  
--> Processing Dependency: /etc/mime.types for package: httpd-2.4.52-1.amzn2.x86_64  
--> Processing Dependency: libaprutil-1.so.0()(64bit) for package: httpd-2.4.52-1.amzn2.x86_64  
--> Processing Dependency: libapr-1.so.0()(64bit) for package: httpd-2.4.52-1.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-server-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-10.2.38-1.amzn2.0.1.x86_64  
--> Processing Dependency: mariadb-cracklib-password-check(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-backup(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: bison for package: 3:mariadb-server-10.2.38-1.amzn2.0.1.x86_64
```

```
sudo systemctl start httpd
sudo systemctl enable httpd
sudo systemctl is-enabled httpd
```

```
ec2-user@ip-172-31-25-103:~$ sudo systemctl start httpd
[ec2-user@ip-172-31-25-103 ~]$ sudo systemctl enable httpd
[ec2-user@ip-172-31-25-103 ~]$ sudo systemctl is-enabled httpd
enabled
```

Test the LAMP server

```
echo "<?php phpinfo(); ?>" > /var/www/html/index.php
```


```
ec2-user@ip-172-31-25-103:~/lab02$ ll /var/www/html
total 4
-rw-r--r-- 1 root root 21 Feb 28 16:56 index.php
```

In a web browser, type the URL of the file that just created.

phpinfo()


Not secure | 54.251.146.13

PHP Version 7.2.34



System	Linux ip-172-31-25-103.ap-southeast-1.compute.internal 5.10.96-90.amzn2.x86_64 #1 SMP Fri Feb 4 17:12:04 UTC 2022 x86_64
Build Date	Oct 21 2020 18:04:56
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional .ini files parsed	/etc/php.d/20-bz2.ini, /etc/php.d/20-calendar.ini, /etc/php.d/20-ctype.ini, /etc/php.d/20-exif.ini, /etc/php.d/20-fileinfo.ini, /etc/php.d/20-ftp.ini, /etc/php.d/20-gettext.ini, /etc/php.d/20-iconv.ini, /etc/php.d/20-json.ini, /etc/php.d/20-mysqld.ini, /etc/php.d/20-pdo.ini, /etc/php.d/20-phar.ini, /etc/php.d/20-sockets.ini, /etc/php.d/20-sqlite3.ini, /etc/php.d/20-tokenizer.ini, /etc/php.d/20-zip.ini, /etc/php.d/25-curl.ini, /etc/php.d/30-mysql.ini, /etc/php.d/30-pdo_mysql.ini, /etc/php.d/30-pdo_sqlite.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718.NTS
PHP Extension Build	API20170718.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, compress.bzip2, phar, zip
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, sslv3, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, bzip2.*, convert.iconv.*

This program makes use of the Zend Scripting Language Engine:



1.3. Hosting a WordPress Blog with Amazon Linux 2 AMI

To download and unzip the WordPress installation package

wget <https://wordpress.org/latest.tar.gz>

tar -xzf latest.tar.gz

```
[ec2-user@ip-172-31-25-103 lab02]$ ls
latest.tar.gz
[ec2-user@ip-172-31-25-103 lab02]$ tar -xzf latest.tar.gz
[ec2-user@ip-172-31-25-103 lab02]$ ls
latest.tar.gz  wordpress
```

To create a database user and database for the WordPress installation

sudo systemctl start mariadb

sudo mysql_secure_installation

```
> ec2-user@ip-172-31-25-103:~/lab02

[ec2-user@ip-172-31-25-103 lab02]$ sudo systemctl start mariadb
[ec2-user@ip-172-31-25-103 lab02]$ 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] n
... skipping.

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
```

```
mysql -u root -p
CREATE USER 'wordpress-user'@'localhost' IDENTIFIED BY 'wordpress
_password';
CREATE DATABASE `wordpress-db`;
GRANT ALL PRIVILEGES ON `wordpress-db`.* TO "wordpress-
user"@"localhost";
FLUSH PRIVILEGES;
exit
```

```
MariaDB [(none)]> CREATE USER 'wordpress-user'@'localhost' IDENTIFIED BY 'wordpress@2022';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> CREATE DATABASE `wordpress-db`;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON `wordpress-db`.* TO "wordpress-user"@"localhost";
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)
```

To create and edit the wp-config.php file

```
cp wordpress/wp-config-sample.php wordpress/wp-config.php
```

```
nano wordpress/wp-config.php
```

```
ec2-user@ip-172-31-25-103:~/lab02
GNU nano 2.9.8 wordpress/wp-config.php

* * ABSPATH
*
* @link https://wordpress.org/support/article/editing-wp-config-php/
*
* @package WordPress
*/

// ** Database settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress-db' );

/** Database username */
define( 'DB_USER', 'wordpress-user' );

/** Database password */
define( 'DB_PASSWORD', 'wordpress@2022' );

/** Database hostname */
define( 'DB_HOST', 'localhost' );

/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );

/** The database collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );
```

Visit <https://api.wordpress.org/secret-key/1.1/salt/> to randomly generate a set of key values that copy and paste into wp-config.php file.

```

define('AUTH_KEY',         '$',ZAL~?fprGcP{u:.&4K,`[Y!@.o>+Co+5~nB|K#=R$+p6c<14bftdP<#~f}:GF');
define('SECURE_AUTH_KEY', '*y!@:nH@q-jLuZ!qxc:%t%,{a/2S#aXf.lHA^afL K rozup1%Q1*Z<wE2@5N+KH');
define('LOGGED_IN_KEY',    'YOmNt?nvcE3S%pD M<lnJ+GX2U-Z7h-[1-,H9iB(Q65:p{g!C;-m~TL6T}qMP%mw');
define('NONCE_KEY',        'M8^%mzOj^h{4.U8Mv_#jj3*-JP[5U]@P1AI+!|bD-Tyw_OXQxO_BTtV`vGdIh');
define('AUTH_SALT',        '7#b{:L+4 O|*,f[+m[[ck91j`{-A)nWASfUzMZ9[? 3qQOuW+3&0gh^zYnq/^`i');
define('SECURE_AUTH_SALT', '72R||$[v$}0uK=:@$cut0S2$^U##/JS({X;t&z040~}f$wQ=)}F:M1Dshj~3Rua');
define('LOGGED_IN_SALT',   'Kwb4p_,9xP!H+8]w{b0&`?EQ5#Y+uC]=Y lDm>/>Db_-|&}_7]7|o=J:6h=8Q-:');
define('NONCE_SALT',       'LrinfJIG)lUgY`I&S@Oe*Mnh6*8P/IFL,ACrrt-.{tA,iF|?V2ms8_`PDR9G8NsN');

```

```

ec2-user@ip-172-31-25-103:~/lab02
GNU nano 2.9.8                               wordpress/wp-config.php

/**#@+
 * Authentication unique keys and salts.
 *
 * Change these to different unique phrases! You can generate these using
 * the {@link https://api.wordpress.org/secret-key/1.1/salt/ WordPress.org secret-key service}.
 *
 * You can change these at any point in time to invalidate all existing cookies.
 * This will force all users to have to log in again.
 *
 * @since 2.6.0
 */
define( 'AUTH_KEY',         '$',ZAL~?fprGcP{u:.&4K,`[Y!@.o>+Co+5~nB|K#=R$+p6c<14bftdP<#~f}:GF' );
define( 'SECURE_AUTH_KEY', '*y!@:nH@q-jLuZ!qxc:%t%,{a/2S#aXf.lHA^afL K rozup1%Q1*Z<wE2@5N+KH' );
define( 'LOGGED_IN_KEY',    'YOmNt?nvcE3S%pD M<lnJ+GX2U-Z7h-[1-,H9iB(Q65:p{g!C;-m~TL6T}qMP%mw');
define( 'NONCE_KEY',        'M8^%mzOj^h{4.U8Mv_#jj3*-JP[5U]@P1AI+!|bD-Tyw_OXQxO_BTtV`vGdIh');
define( 'AUTH_SALT',        '7#b{:L+4 O|*,f[+m[[ck91j`{-A)nWASfUzMZ9[? 3qQOuW+3&0gh^zYnq/^`i');
define( 'SECURE_AUTH_SALT', '72R||$[v$}0uK=:@$cut0S2$^U##/JS({X;t&z040~}f$wQ=)}F:M1Dshj~3Rua');
define( 'LOGGED_IN_SALT',   'Kwb4p_,9xP!H+8]w{b0&`?EQ5#Y+uC]=Y lDm>/>Db_-|&}_7]7|o=J:6h=8Q-:');
define( 'NONCE_SALT',       'LrinfJIG)lUgY`I&S@Oe*Mnh6*8P/IFL,ACrrt-.{tA,iF|?V2ms8_`PDR9G8NsN');

/**#@-*/

/**
 * WordPress database table prefix.
 */

```

To install WordPress files under the Apache document root

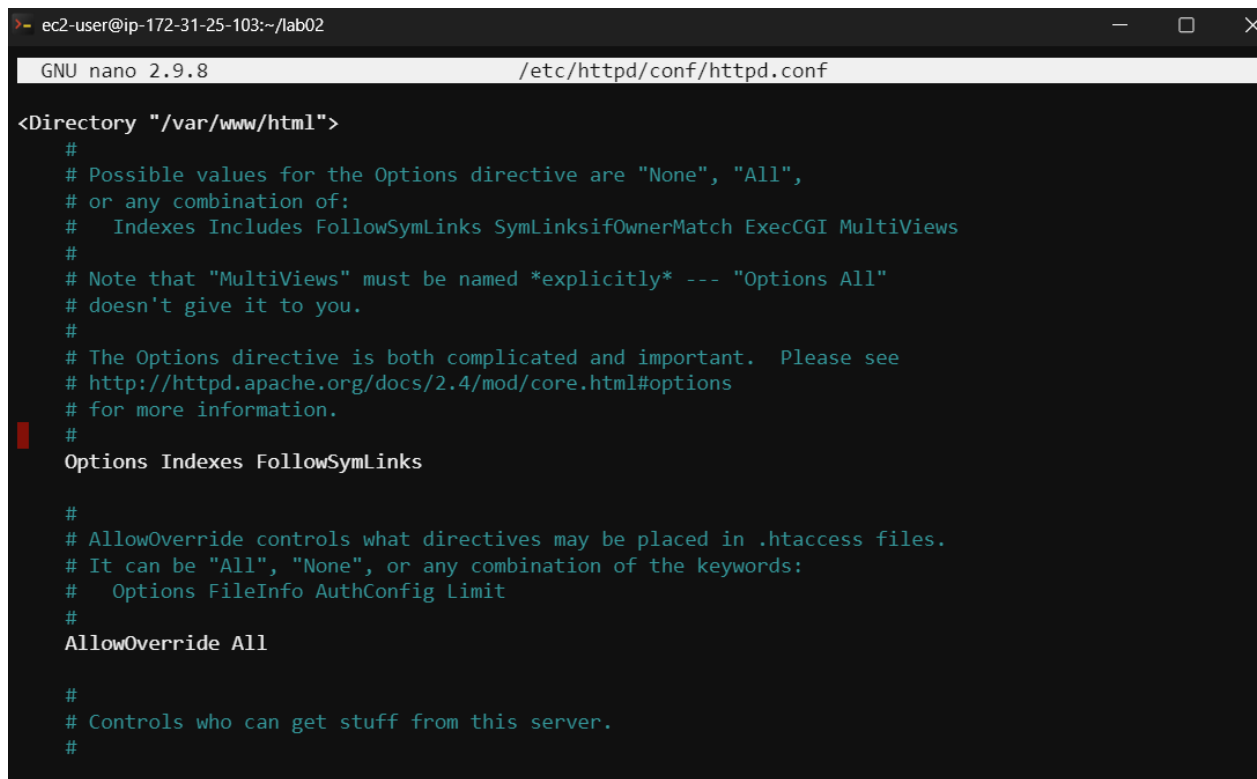
```
sudo mkdir /var/www/html/blog
```

```
sudo cp -r wordpress/* /var/www/html/blog/
```

```
[ec2-user@ip-172-31-25-103 lab02]$ sudo mkdir /var/www/html/blog
[ec2-user@ip-172-31-25-103 lab02]$ sudo cp -r wordpress/* /var/www/html/blog/
```

To allow WordPress to use permalinks

Change the AllowOverride None line in the <Directory "/var/www/html"> section to read AllowOverride All.



```
ec2-user@ip-172-31-25-103:~/lab02
GNU nano 2.9.8 /etc/httpd/conf/httpd.conf

<Directory "/var/www/html">
#
# Possible values for the Options directive are "None", "All",
# or any combination of:
#   Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI MultiViews
#
# Note that "MultiViews" must be named *explicitly* --- "Options All"
# doesn't give it to you.
#
# The Options directive is both complicated and important. Please see
# http://httpd.apache.org/docs/2.4/mod/core.html#options
# for more information.
#
Options Indexes FollowSymLinks

#
# AllowOverride controls what directives may be placed in .htaccess files.
# It can be "All", "None", or any combination of the keywords:
#   Options FileInfo AuthConfig Limit
#
AllowOverride All

#
# Controls who can get stuff from this server.
#
```

To install the PHP graphics drawing library on Amazon Linux 2

```
sudo yum install php-gd
```

```
ec2-user@ip-172-31-25-103:~/lab02
[ec2-user@ip-172-31-25-103 lab02]$ sudo yum install php-gd
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core | 3.7 kB 00:00:00
amzn2extra-docker | 3.0 kB 00:00:00
amzn2extra-kernel-5.10 | 3.0 kB 00:00:00
amzn2extra-lamp-mariadb10.2-php7.2 | 3.0 kB 00:00:00
amzn2extra-php7.2 | 3.0 kB 00:00:00
Resolving Dependencies
--> Running transaction check
--> Package php-gd.x86_64 0:7.2.34-1.amzn2 will be installed
--> Processing Dependency: libXpm.so.4()(64bit) for package: php-gd-7.2.34-1.amzn2.x86_64
--> Processing Dependency: libX11.so.6()(64bit) for package: php-gd-7.2.34-1.amzn2.x86_64
--> Running transaction check
--> Package libX11.x86_64 0:1.6.7-3.amzn2.0.2 will be installed
--> Processing Dependency: libX11-common >= 1.6.7-3.amzn2.0.2 for package: libX11-1.6.7-3.amzn2.0.2.x86_64
--> Processing Dependency: libxcb.so.1()(64bit) for package: libX11-1.6.7-3.amzn2.0.2.x86_64
--> Package libXpm.x86_64 0:3.5.12-1.amzn2.0.2 will be installed
--> Running transaction check
--> Package libX11-common.noarch 0:1.6.7-3.amzn2.0.2 will be installed
--> Package libxcb.x86_64 0:1.12-1.amzn2.0.2 will be installed
--> Processing Dependency: libXau.so.6()(64bit) for package: libxcb-1.12-1.amzn2.0.2.x86_64
--> Running transaction check
--> Package libXau.x86_64 0:1.0.8-2.1.amzn2.0.2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package Arch Version Repository Size
```

```
sudo yum list installed | grep php-gd
```

```
[ec2-user@ip-172-31-25-103 lab02]$ sudo yum list installed | grep php-gd
php-gd.x86_64 7.2.34-1.amzn2 @amzn2extra-php7.2
```

To set file permissions for the Apache web server

```
sudo chown -R apache /var/www
sudo chgrp -R apache /var/www
sudo chmod 2775 /var/www
find /var/www -type d -exec sudo chmod 2775 {} \;
find /var/www -type f -exec sudo chmod 0664 {} \;
sudo systemctl restart httpd
```

```
[ec2-user@ip-172-31-25-103 lab02]$ sudo chown -R apache /var/www
[ec2-user@ip-172-31-25-103 lab02]$ sudo chgrp -R apache /var/www
[ec2-user@ip-172-31-25-103 lab02]$ sudo chmod 2775 /var/www
[ec2-user@ip-172-31-25-103 lab02]$ find /var/www -type d -exec sudo chmod 2775 {} \;
[ec2-user@ip-172-31-25-103 lab02]$ find /var/www -type f -exec sudo chmod 0664 {} \;
[ec2-user@ip-172-31-25-103 lab02]$ sudo systemctl restart httpd
```

Test the WordPress installation

```
sudo systemctl start mariadb && sudo systemctl start httpd
```

```
sudo systemctl status mariadb && sudo systemctl status httpd
```

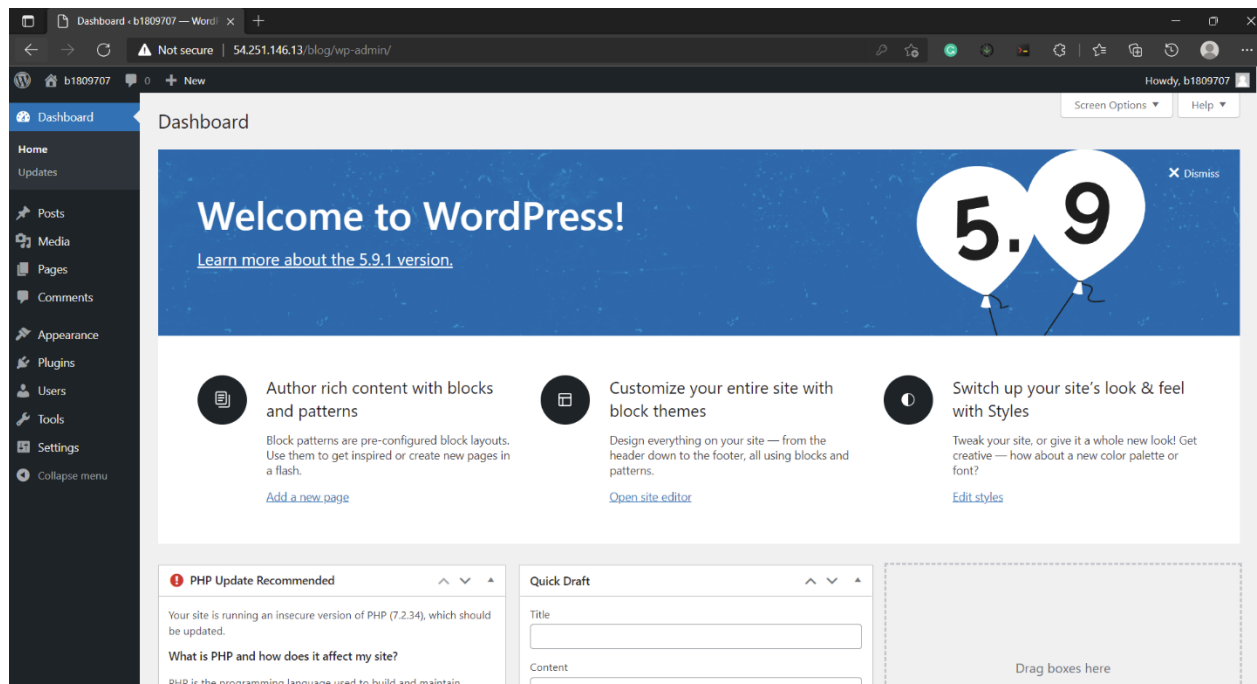
```
[ec2-user@ip-172-31-25-103 lab02]$ sudo systemctl enable httpd && sudo systemctl enable mariadb
Created symlink from /etc/systemd/system/multi-user.target.wants/mariadb.service to /usr/lib/systemd/system/mariadb.service.
[ec2-user@ip-172-31-25-103 lab02]$ sudo systemctl status mariadb
● mariadb.service - MariaDB 10.2 database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; vendor preset: disabled)
   Drop-In: /usr/lib/systemd/system/mariadb.service.d
            └─tokudb.conf
   Active: active (running) since Tue 2022-03-01 00:58:25 UTC; 33min ago
   Main PID: 4315 (mysqld)
     Status: "Taking your SQL requests now..."
    CGroup: /system.slice/mariadb.service
            └─4315 /usr/libexec/mysqld --basedir=/usr

Mar 01 00:58:25 ip-172-31-25-103.ap-southeast-1.compute.internal mysql-check-upgrade[4386]: ERROR: ld...
Mar 01 00:58:25 ip-172-31-25-103.ap-southeast-1.compute.internal mysql-check-upgrade[4386]: ERROR: ld...
Mar 01 00:58:25 ip-172-31-25-103.ap-southeast-1.compute.internal mysql-check-upgrade[4386]: ERROR: ld...
Mar 01 00:58:25 ip-172-31-25-103.ap-southeast-1.compute.internal mysql-check-upgrade[4386]: ERROR: ld...
Mar 01 00:58:25 ip-172-31-25-103.ap-southeast-1.compute.internal mysql-check-upgrade[4386]: ERROR: ld...
Mar 01 00:58:25 ip-172-31-25-103.ap-southeast-1.compute.internal mysql-check-upgrade[4386]: ERROR: ld...
Mar 01 00:58:25 ip-172-31-25-103.ap-southeast-1.compute.internal mysql-check-upgrade[4386]: ERROR: ld...
Mar 01 00:58:25 ip-172-31-25-103.ap-southeast-1.compute.internal mysql-check-upgrade[4386]: ERROR: ld...
Mar 01 00:58:25 ip-172-31-25-103.ap-southeast-1.compute.internal mysql-check-upgrade[4386]: ERROR: ld...
Mar 01 00:58:25 ip-172-31-25-103.ap-southeast-1.compute.internal systemd[1]: Started MariaDB 10.2 dat...
Hint: Some lines were ellipsized, use -l to show in full.
```

```
[ec2-user@ip-172-31-25-103 lab02]$ sudo systemctl start mariadb
[ec2-user@ip-172-31-25-103 lab02]$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Drop-In: /usr/lib/systemd/system/httpd.service.d
            └─php-fpm.conf
   Active: active (running) since Tue 2022-03-01 01:32:01 UTC; 1min 26s ago
     Docs: man:httpd.service(8)
   Main PID: 17034 (httpd)
     Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec:  0 B/sec"
    CGroup: /system.slice/httpd.service
            └─17034 /usr/sbin/httpd -DFOREGROUND
              └─17035 /usr/sbin/httpd -DFOREGROUND
                └─17036 /usr/sbin/httpd -DFOREGROUND
                  └─17037 /usr/sbin/httpd -DFOREGROUND
                    └─17038 /usr/sbin/httpd -DFOREGROUND
                      └─17039 /usr/sbin/httpd -DFOREGROUND

Mar 01 01:32:01 ip-172-31-25-103.ap-southeast-1.compute.internal systemd[1]: Stopped The Apache HTTP ...
Mar 01 01:32:01 ip-172-31-25-103.ap-southeast-1.compute.internal systemd[1]: Starting The Apache HTTP...
Mar 01 01:32:01 ip-172-31-25-103.ap-southeast-1.compute.internal systemd[1]: Started The Apache HTTP ...
Hint: Some lines were ellipsized, use -l to show in full.
```

In a web browser, type the URL of the WordPress blog



b1809707

Sample Page

Hello world!

Welcome to WordPress. This is your first post. Edit or delete it, then start writing!

March 1, 2022 b1809707 Uncategorized

--END--