

LAB 2

Content Management Systems (CMS)



Drupal, Wordpress, and Joomla should ring the bell by now, these are examples of popular content management systems. These are not the only ones out there, but the functionality is more or less the same. You can do research to discover others. Visit this link on Youtube to get an overview of CMSs <https://www.youtube.com/watch?v=cb8MkRr9gu0> or read from <https://kinsta.com/knowledgebase/content-management-system/>. For this lab, we will not go into the specifics, because the assumption is that we are all familiar with AWS, installation of packages, permissions, and directory manipulation.

Installation of WordPress

Prerequisites;

- Redhat 8 on AWS
- Php, apache, mysql/mariadb, curl, tar

```
php-mysqlnd php-fpm mariadb-server httpd tar curl
```

Installation of packages

Copy and paste the above packages to your terminal for installation. NB this is not the full command, so use the correct package manager, and command syntax.

Start both the Apache webserver and the MariaDB services

```
# systemctl start mariadb  
  
# systemctl start httpd
```

We want apache and the database to start up automatically when we power on the server. Therefore enable MariaDB and httpd to start after system reboot.

```
# systemctl enable mariadb

# systemctl enable httpd
```

Check the status of mariadb and httpd to make sure that no errors are there.

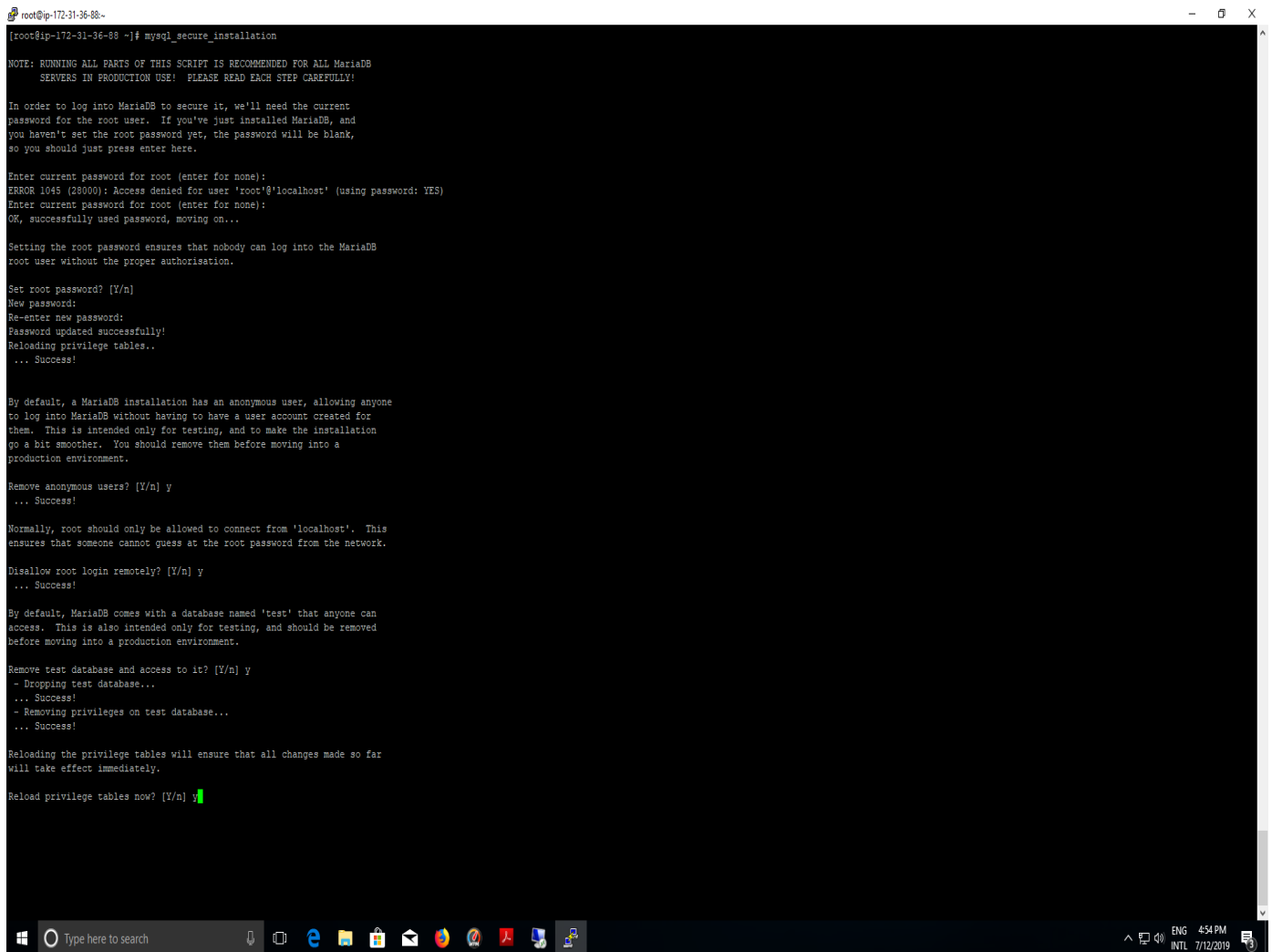
```
root@ip-172-31-36-88~
[root@ip-172-31-36-88 ~]# systemctl status mariadb
● mariadb.service - MariaDB 10.3 database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; disabled; vendor preset: disabled)
   Active: active (running) since Fri 2019-07-12 14:52:02 UTC; 42s ago
     Docs: man:mysqld(8)
           https://mariadb.com/kb/en/library/systemd/
   Process: 25979 ExecStartPost=/usr/libexec/mysql-check-upgrade (code=exited, status=0/SUCCESS)
   Process: 25745 ExecStartPre=/usr/libexec/mysql-prepare-db-dir mariadb.service (code=exited, status=0/SUCCESS)
   Process: 25721 ExecStartPre=/usr/libexec/mysql-check-socket (code=exited, status=0/SUCCESS)
  Main PID: 25847 (mysqld)
    Status: "Taking your SQL requests now..."
     Tasks: 30 (limit: 4998)
    Memory: 83.4M
    CGroup: /system.slice/mariadb.service
            └─25847 /usr/libexec/mysqld --basedir=/usr

Jul 12 14:52:01 ip-172-31-36-88.us-east-2.compute.internal mysql-prepare-db-dir[25745]: Please report any problems at http://mariadb.org/life
Jul 12 14:52:01 ip-172-31-36-88.us-east-2.compute.internal mysql-prepare-db-dir[25745]: The latest information about MariaDB is available at http://mariadb.org/.
Jul 12 14:52:01 ip-172-31-36-88.us-east-2.compute.internal mysql-prepare-db-dir[25745]: You can find additional information about the MySQL part at:
Jul 12 14:52:01 ip-172-31-36-88.us-east-2.compute.internal mysql-prepare-db-dir[25745]: http://dev.mysql.com
Jul 12 14:52:01 ip-172-31-36-88.us-east-2.compute.internal mysql-prepare-db-dir[25745]: Consider joining MariaDB's strong and vibrant community:
Jul 12 14:52:01 ip-172-31-36-88.us-east-2.compute.internal mysql-prepare-db-dir[25745]: https://mariadb.org/get-involved/
Jul 12 14:52:01 ip-172-31-36-88.us-east-2.compute.internal mysqld[25847]: 2019-07-12 14:52:01 0 [Note] /usr/libexec/mysqld (mysqld 10.3.11-MariaDB) starting as process 25847 ...
Jul 12 14:52:01 ip-172-31-36-88.us-east-2.compute.internal mysqld[25847]: 2019-07-12 14:52:01 0 [Warning] Could not increase number of max_open_files to more than 1024 (request: 4183)
Jul 12 14:52:01 ip-172-31-36-88.us-east-2.compute.internal mysqld[25847]: 2019-07-12 14:52:01 0 [Warning] Changed limits: max_open_files: 1024 max_connections: 151 (was 151) table_cache: 421 (was 2000)
Jul 12 14:52:02 ip-172-31-36-88.us-east-2.compute.internal systemd[1]: Started MariaDB 10.3 database server.
[root@ip-172-31-36-88 ~]#
```

Secure your MariaDB database and set root password:

```
# mysql_secure_installation
```

The setup is interactive, so make sure you discuss the options as a group before choosing an answer.



```
root@ip-172-31-36-88 ~]# 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):
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)
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]
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
```

Login to mysql

```
mysql -u root -p
```

Enter the password that you chose for root.

Check the existing databases

```
SHOW DATABASES;
```

```
root@ip-172-31-36-88:~  
MariaDB [(none)]>  
MariaDB [(none)]>  
MariaDB [(none)]>  
MariaDB [(none)]>  
MariaDB [(none)]>  
MariaDB [(none)]> SHOW DATABASES;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| performance_schema |  
+-----+  
3 rows in set (0.000 sec)  
  
MariaDB [(none)]> █
```

Create a new database `wordpress` and give new user `web` access to the `wordpress` database with password `Nust@2019`

```
# mysql -u root -p  
  
mysql> CREATE DATABASE wordpress;  
  
mysql> CREATE USER `admin`@`localhost` IDENTIFIED BY 'pass';  
  
mysql> GRANT ALL ON wordpress.* TO `admin`@`localhost`;  
  
mysql> FLUSH PRIVILEGES;
```

```
mysql> exit
```

```
MariaDB [(none)]> CREATE database wordpress;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> CREATE user 'web'@'localhost' IDENTIFIED BY 'Nust@2019';
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> GRANT ALL ON wordpress.* TO 'web'@'localhost';
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> exit
```

Create a new directory called Downloads in your home directory, and download WordPress;

```
curl https://wordpress.org/latest.tar.gz --output wordpress.tar.gz
```

```
root@ip-172-31-36-88:~/Downloads
[root@ip-172-31-36-88 ~]# ls
anaconda-ks.cfg  original-ks.cfg
[root@ip-172-31-36-88 ~]# mkdir Downloads
[root@ip-172-31-36-88 ~]# cd Downloads/
[root@ip-172-31-36-88 Downloads]# curl https://wordpress.org/latest.tar.gz --output wordpress.tar.gz
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 10.6M  100 10.6M    0     0  3450k      0  0:00:03  0:00:03 --:--:-- 3449k
[root@ip-172-31-36-88 Downloads]# ls
wordpress.tar.gz
[root@ip-172-31-36-88 Downloads]# tar xf wordpress.tar.gz
[root@ip-172-31-36-88 Downloads]# ls
wordpress  wordpress.tar.gz
[root@ip-172-31-36-88 Downloads]#
```

NB: Do not forget to extract!!!

Move the wordpress folder to the /var/www/html/

Change ownership of the wordpress directory to user apache and group. Why are we doing this?
Discuss as a group.

```
chown -R apache:apache /var/www/html/wordpress
```

Connect the website to the database

Make a copy of the wp-config-sample.php to wp-config.php

```
root@ip-172-31-36-88 wordpress)#  
root@ip-172-31-36-88 wordpress)#  
root@ip-172-31-36-88 wordpress)# ls  
index.php  readme.html  wp-admin  wp-comments-post.php  wp-config-sample.php  wp-cron.php  wp-links-opml.php  wp-login.php  wp-settings.php  wp-trackback.php  
license.txt  wp-activate.php  wp-blog-header.php  wp-config.php  wp-content  wp-includes  wp-load.php  wp-mail.php  wp-signup.php  xmlrpc.php  
root@ip-172-31-36-88 wordpress)#
```

Modify the wp-config.php and add the database name, user, and password. Check other parameters as well.

```

/*
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the
 * installation. You don't have to use the web site, you can
 * copy this file to "wp-config.php" and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * MySQL settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://codex.wordpress.org/Editing_wp-config.php
 *
 * @package WordPress
 */

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

/** MySQL database username */
define( 'DB_USER', 'username_here' );

/** MySQL database password */
define( 'DB_PASSWORD', 'password_here' );

/** MySQL 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', '' );

/**#@+
 * 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 log in again.
 *
 * @since 2.6.0
 */
define( 'AUTH_KEY',         'put your unique phrase here' );
define( 'SECURE_AUTH_KEY',  'put your unique phrase here' );
define( 'LOGGED_IN_KEY',    'put your unique phrase here' );
-- INSERT --

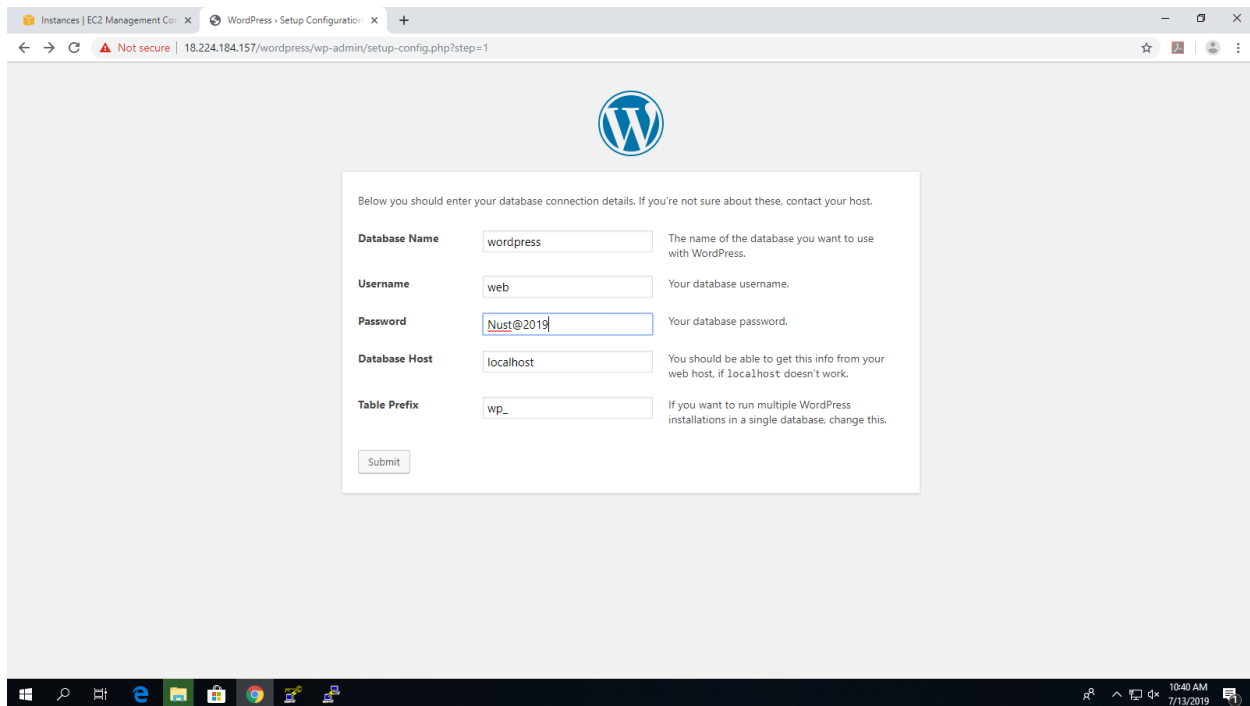
```

Access WordPress installation wizard and perform the actual WordPress installation. Navigate your browser to <http://localhost/wordpress>

NB: replace localhost with your public IP, but if you cannot connect, then troubleshoot!!

Some steps have been skipped, but you should see a WordPress logo, and click on “lets go”. You will see the following page below.

Enter the details that you used in your `wp-config.php` file in order to connect to the database from the web interface.



The screenshot shows a web browser window with the URL `18.224.184.157/wordpress/wp-admin/setup-config.php?step=1`. The page features the WordPress logo at the top center. Below the logo, a text prompt reads: "Below you should enter your database connection details. If you're not sure about these, contact your host." The form contains five input fields with corresponding labels and help text:

- Database Name:** `wordpress`. Help text: "The name of the database you want to use with WordPress."
- Username:** `web`. Help text: "Your database username."
- Password:** `Nust@2019`. Help text: "Your database password."
- Database Host:** `localhost`. Help text: "You should be able to get this info from your web host. If localhost doesn't work."
- Table Prefix:** `wp_`. Help text: "If you want to run multiple WordPress installations in a single database, change this."

A "Submit" button is located at the bottom left of the form.

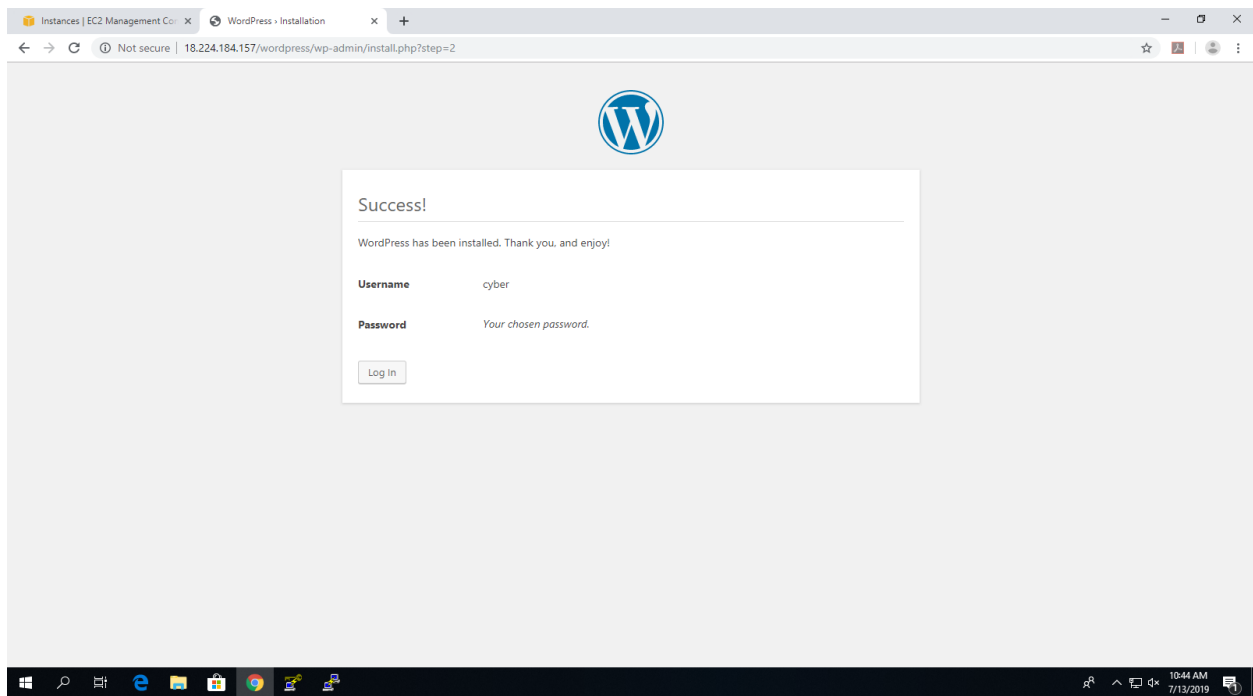
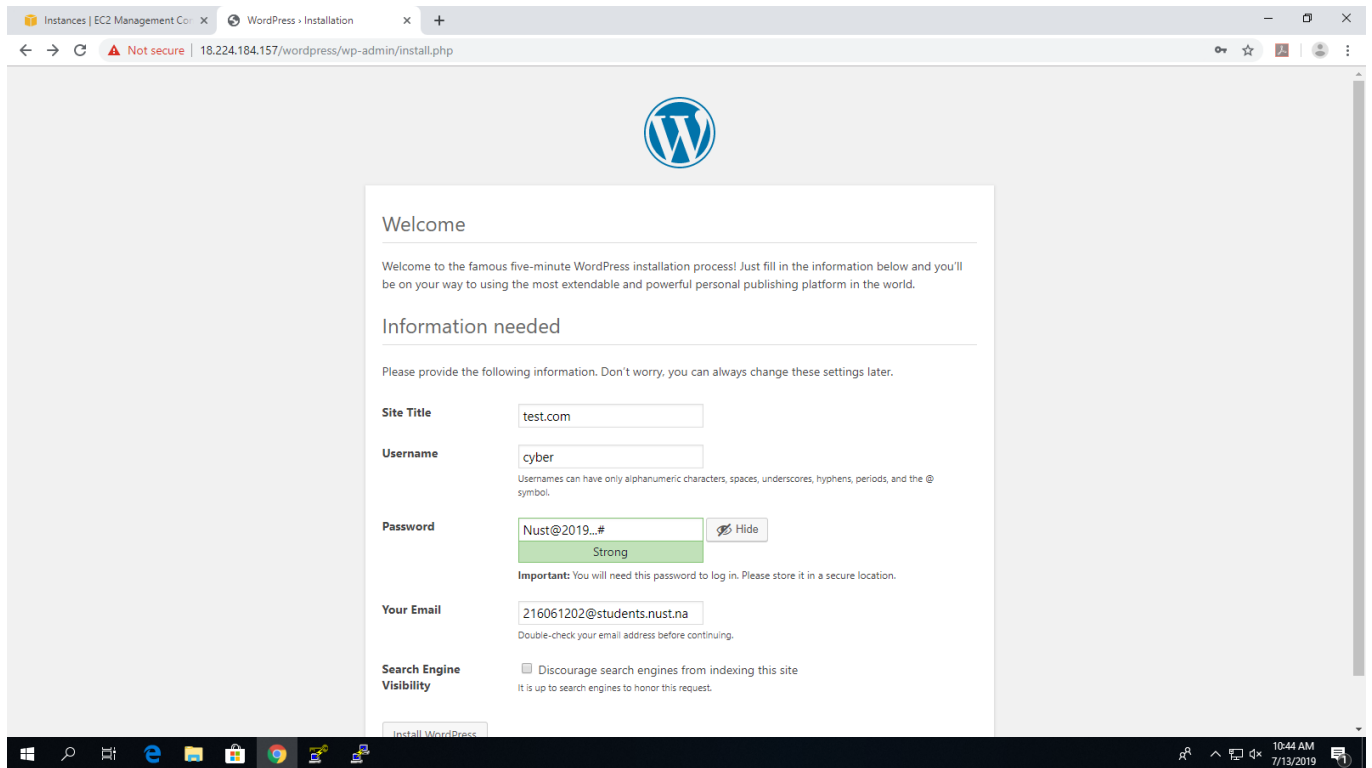
Site details

Site: test.com

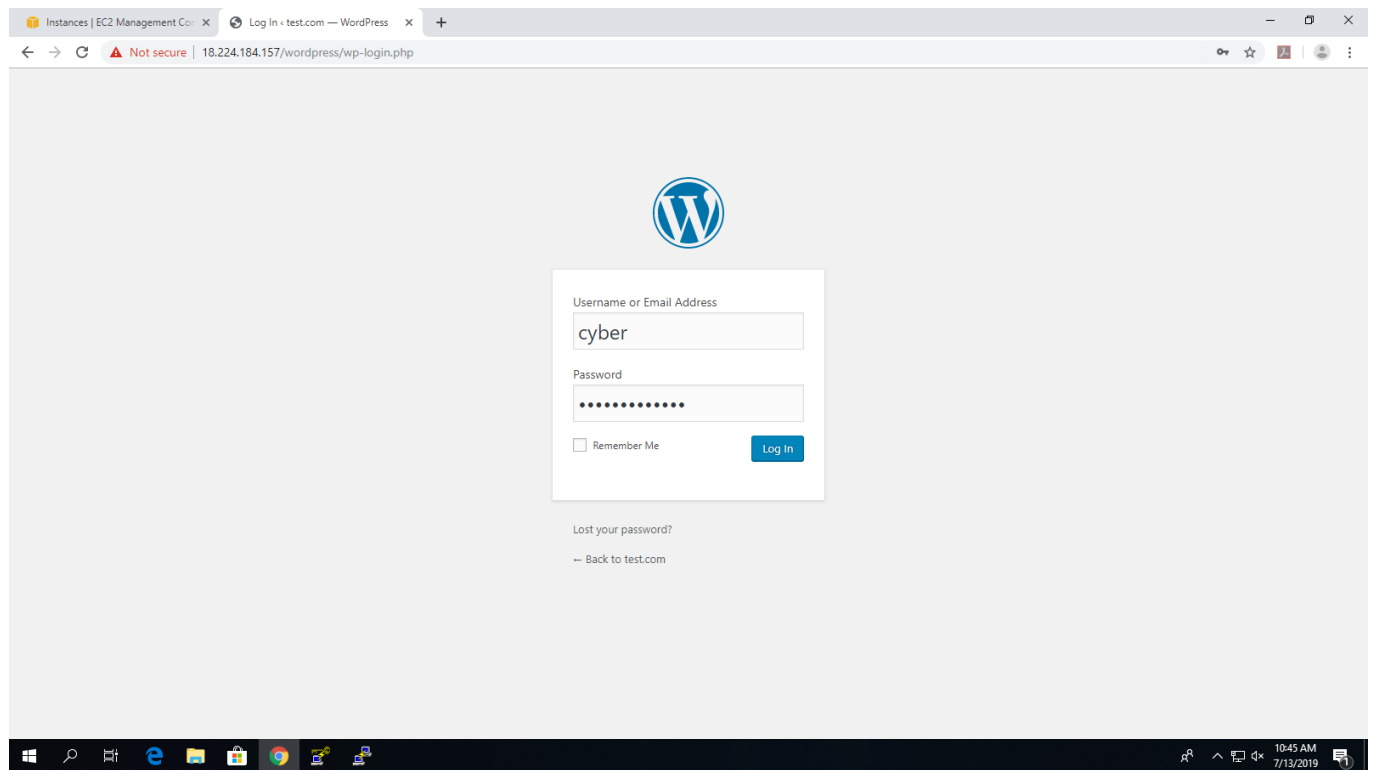
User: cyber

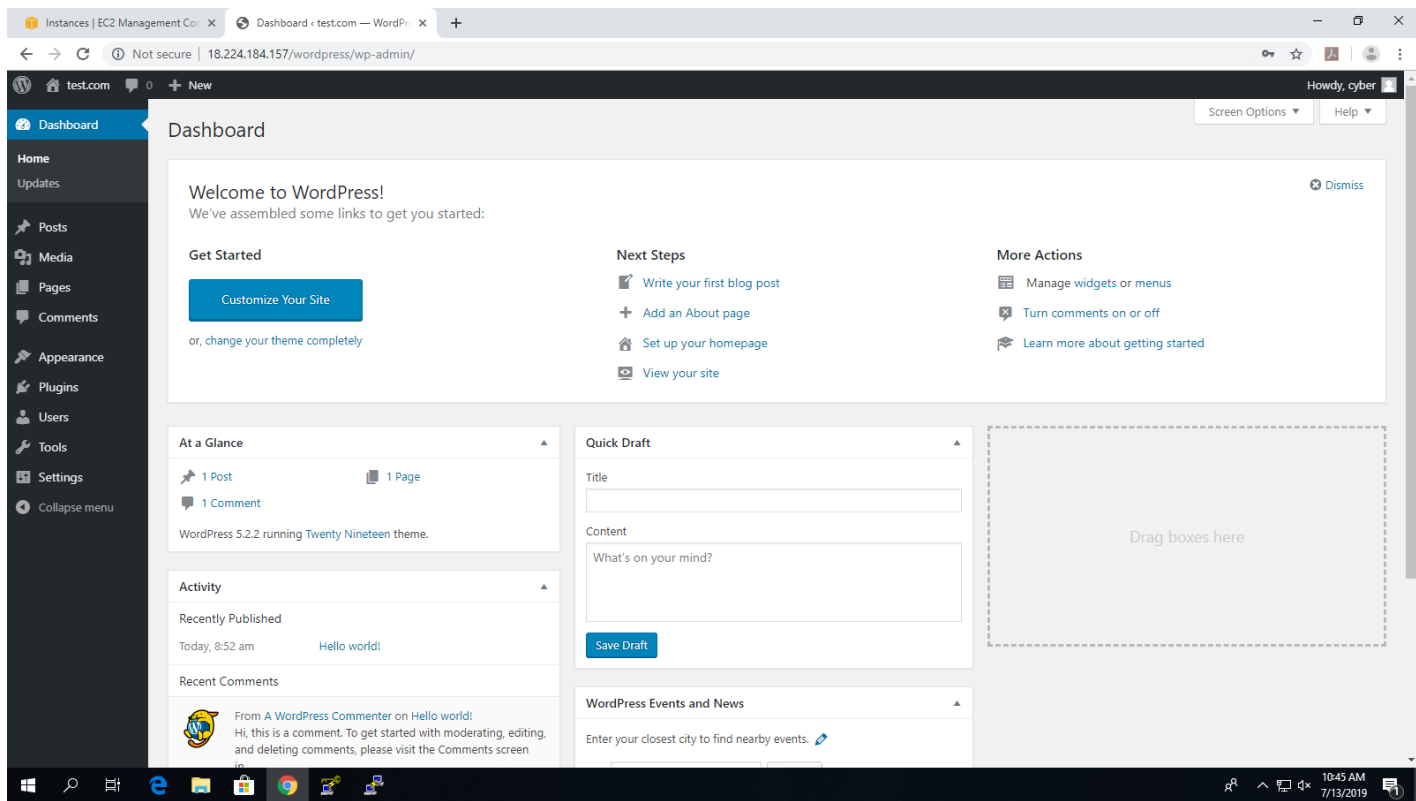
Password: [Nust@2019...#](#)

Email: NUST student email

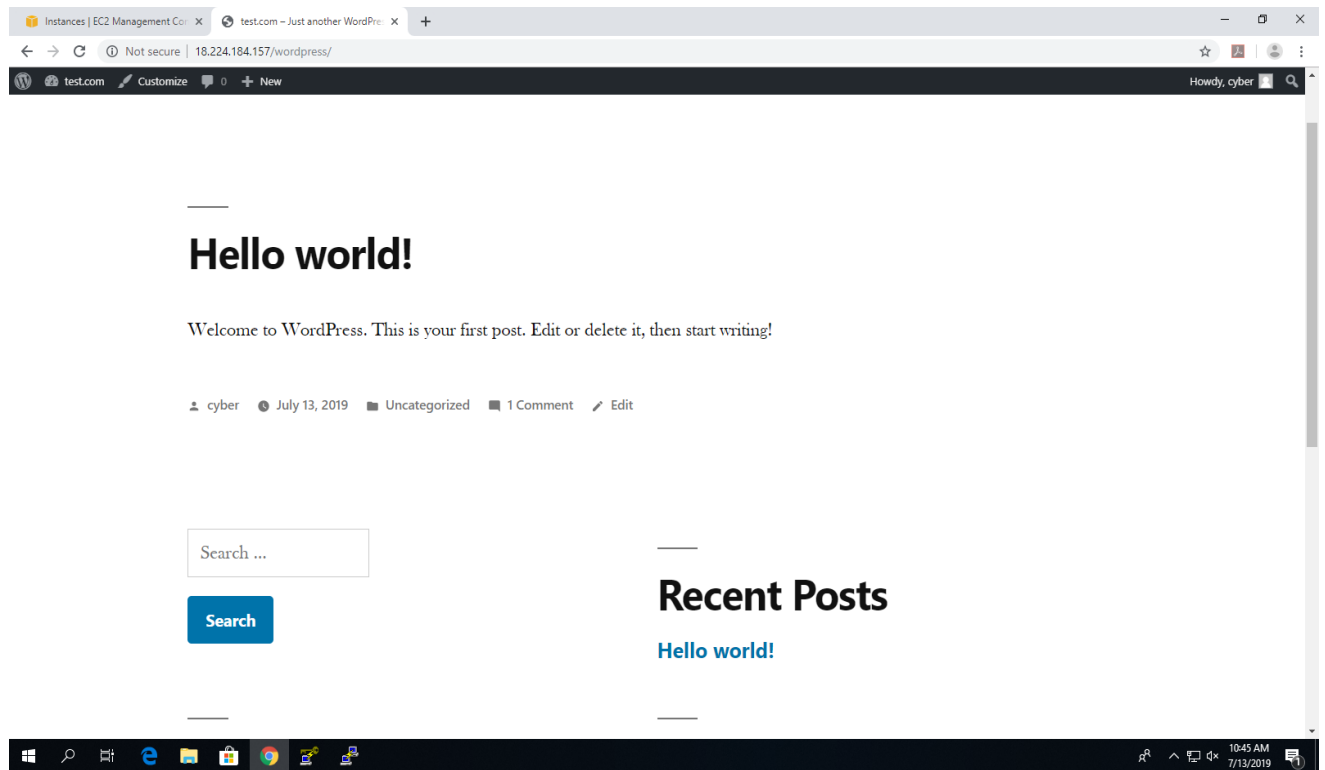


After installation, login to the site using the user created.





Click on test.com to view the home page of the newly created WordPress site



Challenge

- Change the configuration of apache such that your site shows the WordPress site automatically instead of using `http://localhost/wordpress`