Deploying Website

Table of Contents

Introduction

The next step after installing LAMP is to configure your Ubuntu server to have virtual host and download your website files from your GitHub account. A static website would be boring that is why part of this module is to set instruction how to import your MySQL database.

Intended Learning Outcomes

At the end of the module, the students are expected to:

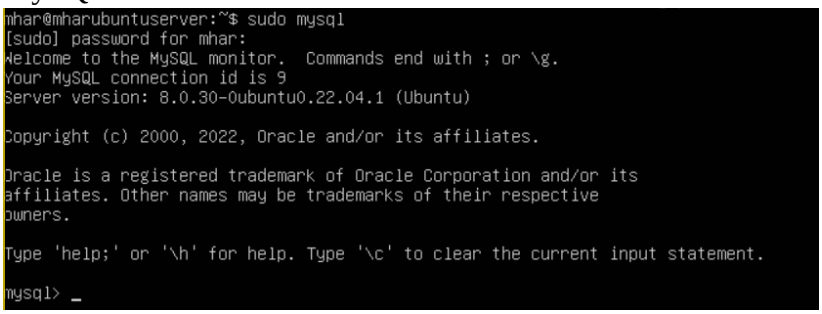
1. Configure or create a new virtual host
2. Clone Github to virtual host directory
3. Import MySQL database and connect website to database

Pre-requisite

A. The following should have been installed prior to doing the succeeding steps:

- Apache

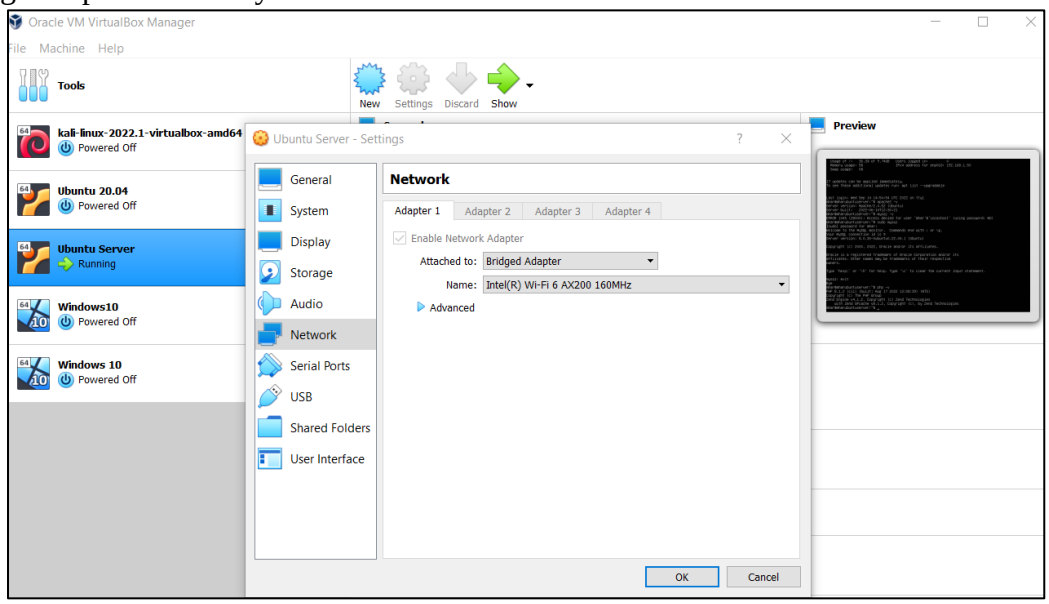
- MySQL



- PHP

B. VM Network adapter should be

- Host only adapter – allow you to connect to host pc
- Bridge adapter – allows you to access internet resources

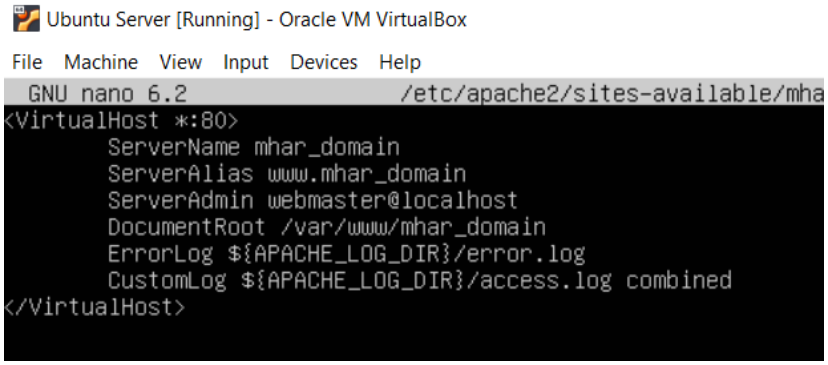


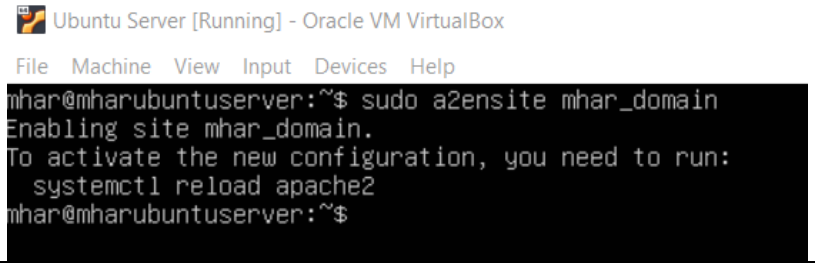
Creating Virtual Host

1.	Create your directory
	`sudo mkdir /var/www/your_domain`
	Ubuntu Server [Running] - Oracle VM VirtualBox
	File Machine View Input Devices Help
	`mhar@mharubuntuserver:~$ sudo mkdir /var/www/mhar_domain`
	`mhar@mharubuntuserver:~$`

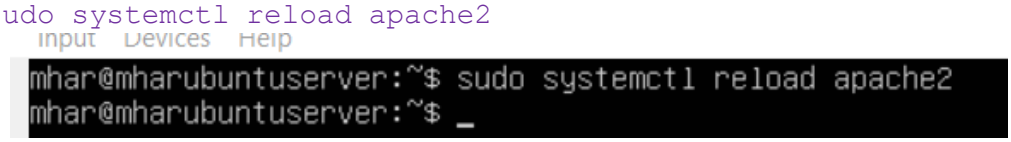
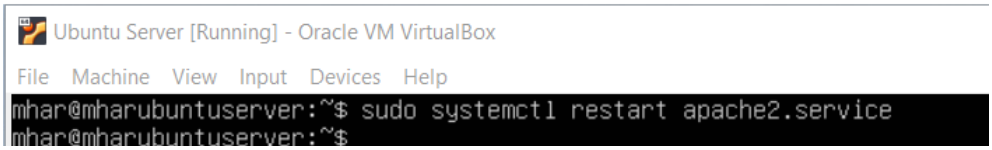
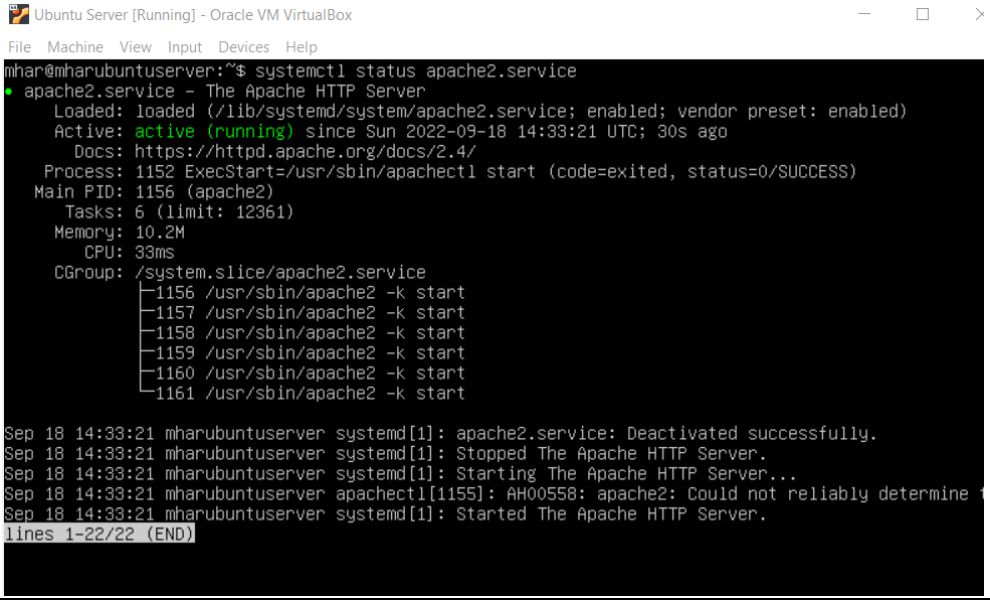
2.	Assign ownership to the folder
	`sudo chown -R $USER:$USER /var/www/your_domain`
	`mhar@mharubuntuserver:~$ sudo chown -R $USER:$USER /var/www/mhar_domain`
	`mhar@mharubuntuserver:~$`

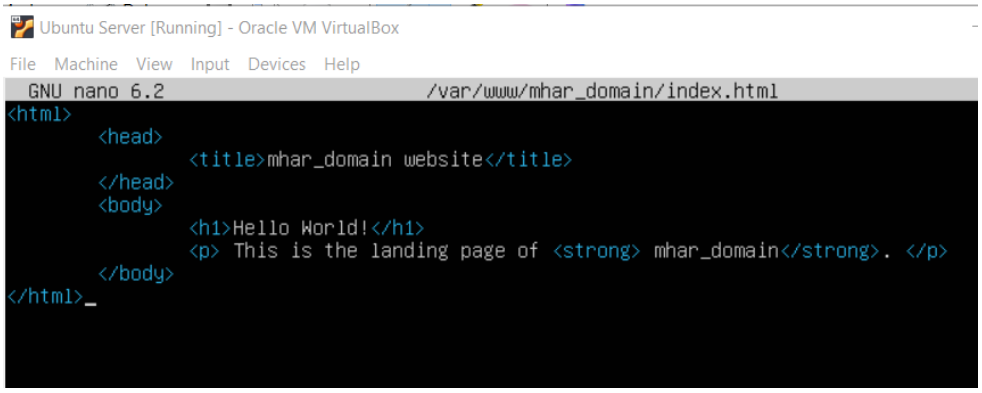
3.	Create a config file in the “site-available” directory.
	`sudo nano /etc/apache2/sites-available/your_domain.conf`
	Ubuntu Server [Running] - Oracle VM VirtualBox
	le Machine View Input Devices Help
	`GNU nano 6.2 /etc/apache2/sites-available/mhar_domain.conf`

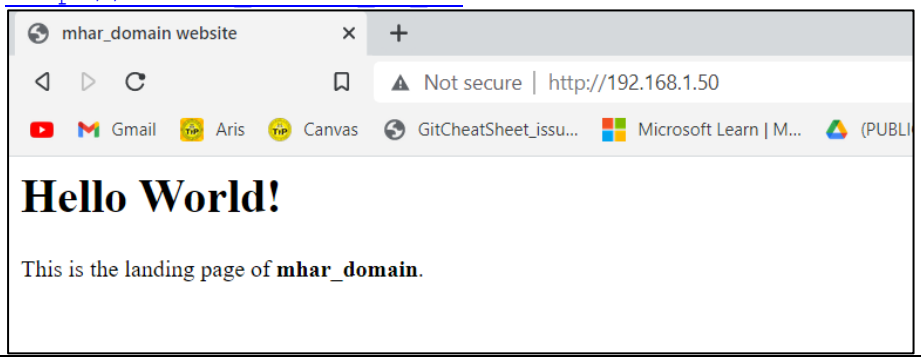
4.	Enter the following configuration
	`<VirtualHost *:80>`
	`ServerName your_domain`
	`ServerAlias www.your_domain`
	`ServerAdmin webmaster@localhost`
	`DocumentRoot /var/www/your_domain`
	`ErrorLog ${APACHE_LOG_DIR}/error.log`
	`CustomLog ${APACHE_LOG_DIR}/access.log combined`
	`</VirtualHost>`
	
	Do not forget to save
	 

5.	Enable the new virtual host `sudo a2ensite your_domain` 

6.	Disable the default website that is shipped with Apache. You can also use the same command by replacing the "000-default" with virtual host you want to disable. `sudo a2dissite 000-default` 

7.	Reload apache for the changes to take effect
	`sudo systemctl reload apache2`
	 
	this line makes the new website active.
	
	Restart apache
	`sudo systemctl restart apache2.service`
	 
	To check if your apache is running, issue the command
	`systemctl status apache2.service`
	 

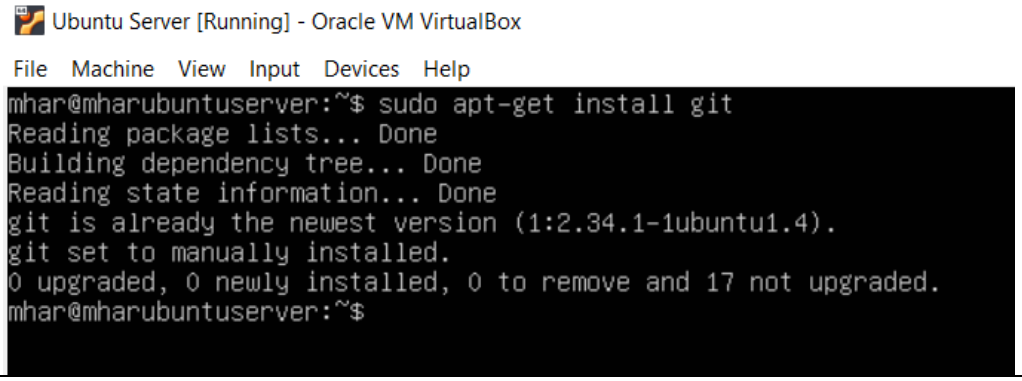
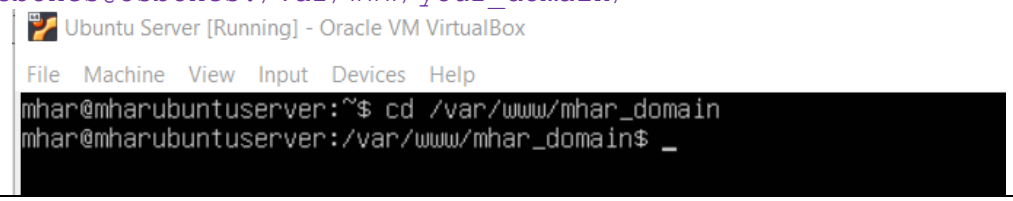
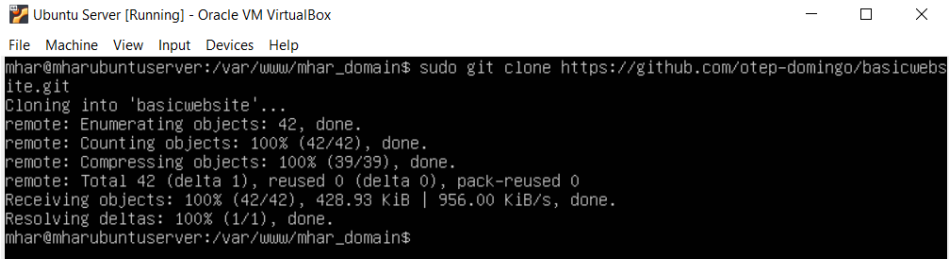
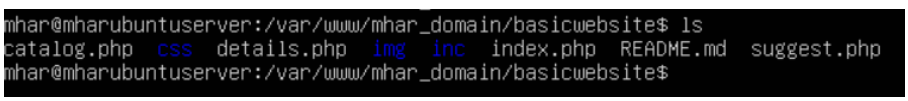
8.	Create a simple page to test your new website.
	`nano /var/www/your_domain/index.html`
	
	enter the following lines or your own html
	`<html>`
	`  <head>`
	`  <title>your_domain website</title>`
	`  </head>`
	`  <body>`
	`  <h1>Hello World!</h1>`
	
	`  <p>This is the landing page of <strong>your_domain</strong>.</p>`
	`  </body>`
	`</html>`
	 

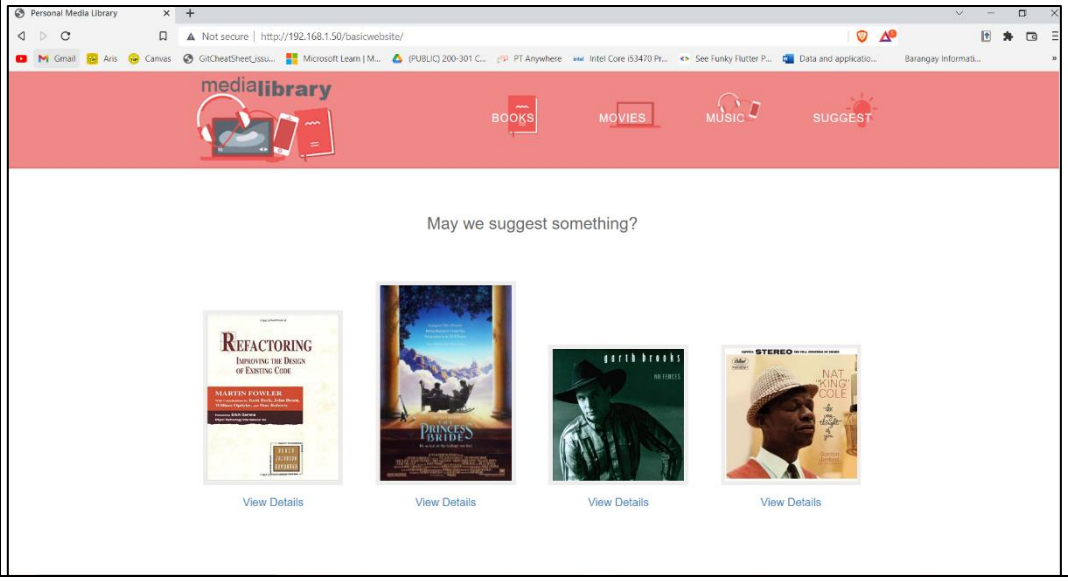
9.	Access now your new website http://server_domain_or_IP 

NOTE	If you cannot access your website using the IP address, follow the procedure how to setup a static IP address.

Downloading your website via GitHub

We will use this method to download our website and save it inside our newly created virtual host directory.

1.	Make sure to install git `sudo apt-get install git` 
2.	Browse to your Virtual host directory `osboxes@osboxes:/var/www/your_domain/` 
3.	Clone you GitHub files `sudo git clone https://github.com/otep-domingo/basicwebsite.git`  Note: I created my GitHub repository to be public temporarily so that we do not have to worry about credentials. Once done with cloning, we can return it back to its private settings.
4.	Check that you can open the “basicwebsite” directory 



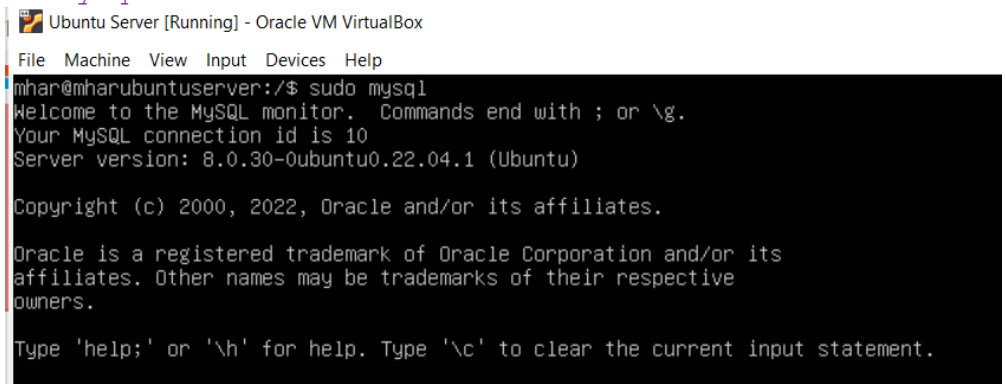
Setup Database

The tutorial uses the ubuntu terminal to setup the database. If you prefer, you can install the phpMyAdmin so that you have a web application where you can manage your database.

1. Change your root privilege

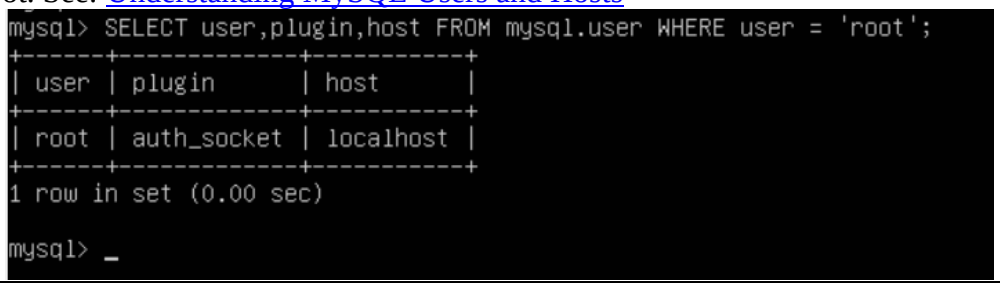
1. Open MySQL prompt from the terminal

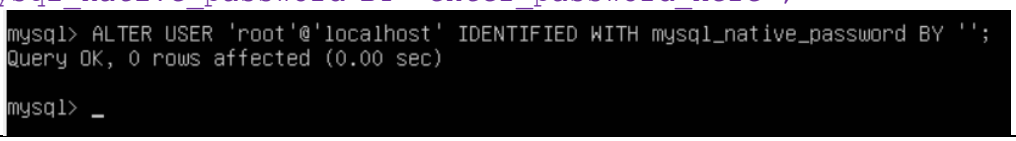
```
sudo mysql
```



if the account is with password:

	`sudo mysql -p -u root`

2.	Run the query
	`SELECT user,plugin,host FROM mysql.user WHERE user = 'root';`
	
	You will see similar output
	
	`+--------+-----------------+--------------+`
	`| user | plugin | host |`
	`+--------+-----------------+--------------+`
	`| root | auth_socket | localhost |`
	`+--------+-----------------+--------------+`
	`1 row in set (0.00 sec)`
	
	Above we can see that the plugin for the root account is set to `auth_socket`. This may also say `caching_sha2_password`. You need to change this to `mysql_native_password`. Also, the host value should be set to `localhost` or `%`. If it's set to anything else, you may not be able to log into phpMyAdmin with root. See: Understanding MySQL Users and Hosts
	 

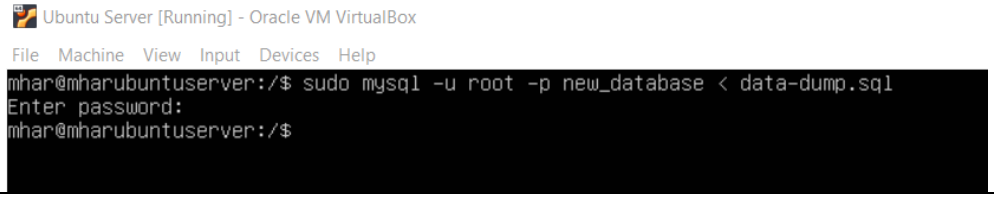
3.	Run the following query to change the plugin value to `mysql_native_password`. Make sure to replace `enter_password_here` with your own. `ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'enter_password_here';` 
4.	Flush privileges. `FLUSH PRIVILEGES;`  1. You should now be able to log into your application connected to MySQL.

2. Import database

The import database is very straightforward.

5.	First create a database. Make sure you do this while inside the mysql prompt. `CREATE DATABASE new_database;` 

6.	Exit the mysql prompt.
	`Exit;`
	 
	Or
	
	`Quit;`

7.	Run the command to import the database sql backup file to your newly created database. `sudo mysql -u username -p new_database < data-dump.sql` 

3. Update PHP database configuration

8.	Navigate to your website folder.
9.	I have the following example setting which is normally done in every PHP application  My database at this moment is “crud_review_2018”
10.	Do not forget to save. Test your application by browsing it via browser.

Honor Pledge:

“I affirm that I have not given or received any unauthorized help on this assignment, and that this work is my own.”

ENRIQUEZ, MHAR VINCENT M.