**Name of Title:** Learning Nginx

**Video Name:** The LEMP stack: Installing MariaDB

**Estimated Length:**

**Author Name:** Michael Jenkins

**Chapter\_Section\_Video:**

**Video Objective:**

At the end of this video the learner will be able to install and configure mysql on a Ubuntu VM.

**Script:**

In this lesson we’ll complete our LEMP stack by installing MariaDB. MariaDB is an open source fork of the popular MySQL database. It uses the same interface and commands as MySQL so it works as a drop in replacement. You can learn more about MariaDB at MariaDB dot org.

EDITOR: Fly in a link to https://mariadb.org/

I’m connected to our development server as the root user. Nginx and PHP have already been installed and configured. Now let’s install MariaDB packages.

We’ll be installing the database server along with a client that will allow us to easily create and manipulate databases.

We’ll do this with:

CTRL+L

apt install -y mariadb-server mariadb-client

Once the installation is complete, we can confirm all is well by checking the status of the server with systemctl:

CTRL+L

systemctl status mysqld.service --no-pager

One thing to note here is that even though we installed the MariaDB server, it still uses the name “mysqld”. It also uses mysql for the client which we can check with the --version switch:

CTRL+L

mysql --version

mysql Ver 15.1 ...

Now that we have our MariaDB client and server installed, let’s secure the installation.

SLIDE: Securing MariaDB

Set a root password

Disable remote connections for the root account

Remove anonymous accounts

\*Required for production systems\*

The initial installation for the database server includes some defaults that we need to change to make things more secure. For example, we need to set a root password, disable remote connections for root, and remove anonymous accounts.

These steps may not be necessary for a development, but they are definitely required for databases used in production. Getting into the habit of securing your database installation is a good thing. It's also quick, easy, and can keep you from getting hacked!

BACK TO TERMNAL

All we have to do is use a tool called mysql secure installation comes with MariaDB. Running this tool starts a wizard that walks us through the process of making sure everything is secure.

CTRL+L

mysql\_secure\_installation

At our root prompt we type mysql\_secure\_installation. And the first thing we see is this reminder that indeed, running all parts of the secure installation script is recommended for all servers being used in production.

The first question asks for the current password. We didn’t set one during install so we can just press enter here.

The next question asks if we want to set a root password and of course we do so press enter again to select the default of Yes.

Choose a password that’s easy to remember but difficult for others to guess.

123

123

Next up, we need to remove anonymous users. This forces all users to have an account set up before they can connect. We’ll hit enter to apply that.

Now we remove remote access for the root account. This means no one can connect to our server as root. They’ll have to be logged on to the server and *then* they can connect with the root account and password.

Next, we’ll remove the test database and apply the changes we’ve made so far.

See how easy it was to make our MySQL installation secure!?

Now we can get to the fun part of setting up the database that we’ll use for our LEMP demonstration.

First we’ll connect to the database as the root user with the mysql client.

CTRL+L

Mysql -u root -p

We’ll enter mysql dash U root and then dash P. the dash U passes the user we want to connect as and the -p tells the mysql client to prompt us for the password.

Once we’re logged in, we can start creating our demo database with a few SQL commands:

First we’ll create the database but only if it does not exist..which we know it doesn't! Then, we’ll create a user we’ll call admin. And then we’ll give the admin user all privileges on the appointment database using the password admin.

CTRL+L

create database if not exists appointments;

create user if not exists 'admin';

grant all on appointments.\* to 'admin'@'localhost' identified by 'admin';

Now let’s disconnect from the root user and log in with the admin user. To log out, type exit.

CTRL+D

Ah you see that?! MariaDB is so nice it even says “Bye” when we log out. :D

Now that we have an account set up, we can use the mysql command again; this time with admin and the password admin:

Mysql -u admin -p

Once we’re logged in, we can check on the database we created with a few SQL commands:

CTRL+L  
 Show databases;

use appointments;

Show tables;

exit

First we’ll just list the databases with the show databases command; then we can select the appointments database with the use command. And finally we can take a look inside the appointments database with the show tables command.

We just created this database and haven’t entered any data so of course, the DB should be empty.

Now that we have MariaDB installed, we have all the components of our LEMP stack.

**Conclusion:**

Type your conclusion statement here.

**Script and Media:**

Break the script up into parts and align it with any media (slides, web, CLI, etc.)

| **Part** | **Script** | **Media** |
| --- | --- | --- |
|  |  |  |

**Exercise Files:**

**Basement:**

Let’s start our MariaDB installation by booting up the VM for this lesson. If you’re following along with the exercise files, you can use the Vagrantfile for this chapter. It will boot the VM and install nginx, PHP, and the demo site.

If you're not using the exercise files, you can follow along with a VM running Ubuntu 18.04 LTS. You’ll need root access and you’ll need to install nginx and PHP.

vagrant up

OK I’m logged into the development VM. All of the commands we’ll run need elevated permissions so let’s go ahead and become the root user:

CTRL+L  
 sudo su -

Now we can run apt update to update the package repositories:

CTRL+L

apt update