EEE 4482 Server Installation and Programming

Worksheet 7 – Apache Installation with LAMP

Objective: To familiarize the steps in installing Apache, PHP and MySQL

Tools: Windows PC

Software: Oracle VM Virtual Box version 6.1.12
CentOS 7

Topics covered:

- Installation of Apache, PHP and MySQL

Component list:
None

Note: Use the virtual machine after Worksheet 01.

Install Apache, PHP And MySQL On CentOS 7

This worksheet shows how you can install an Apache2 webserver on a CentOS 7.0 platform with PHP support (mod_php) and MySQL support. LAMP is short for **L**inux, **A**pache, **M**ySQL, **P**HP7.4.

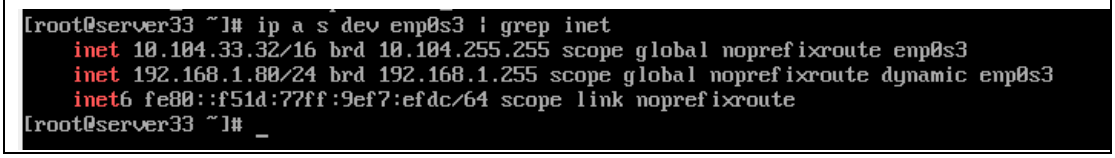
1 Preliminary Note

In this worksheet, I use the hostname *server1.example.com* with the IP address *192.168.0.100*. These settings might differ for you. You must replace them where appropriate. Also, you have to login as root to perform all tasks in this worksheet.

For example, to check your IP, use command

[root@server33 ~]# ip a s dev enp0s3 | grep inet

My computer has the following response. The first line IP 10.104.33.32 is added in Worksheet01. The second line shows the real IP. Therefore, in this case, the IP is 192.168.1.80.



I will add the EPEL (Extra Packages for Enterprise Linux) repo here to install the latest phpMyAdmin (part 7) as follows:

```
rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY*
yum -y install epel-release
```

Note: Extra Packages for Enterprise Linux (or EPEL) is a Fedora Special Interest Group that creates, maintains, and manages a high quality set of additional packages for Enterprise Linux, including, but not limited to, Red Hat Enterprise Linux (RHEL), CentOS and Scientific Linux (SL), Oracle Linux (OL).

2 Installing MySQL / MariaDB

MariaDB is a MySQL fork of the original MySQL developer Monty Widenius. MariaDB is compatible with MySQL and I've chosen to use MariaDB here instead of MySQL. To install MySQL, we install MariaDB like this:

```
yum -y install mariadb-server mariadb
```

Then we create the system startup links for MySQL (so that MySQL starts automatically whenever the system boots) and start the MySQL server:

```
systemctl start mariadb.service
systemctl enable mariadb.service
```

Set passwords for the MySQL root account:

```
mysql_secure_installation
```

```
[root@server1 ~]# mysql_secure_installation
/usr/bin/mysql_secure_installation: line 379: find_mysql_client:
command not found

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): <--ENTER
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] ***<-- Press Y + ENTER***
New password: ***<--yourmariadbpassword (netlab123) + ENTER***
Re-enter new password: ***<--yourmariadbpassword (netlab123) + ENTER***
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 + ENTER***
... Success!

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

Disallow root login remotely? [Y/n] ***<-- Y + ENTER***
... 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 + ENTER***
- 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 + ENTER***

... 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!

[root@server1 ~]#

3 Installing Apache2

CentOS 7 ships with apache 2.4. Apache2 is directly available as a CentOS 7.0 package. Therefore we can install it like this:

```
yum -y install httpd
```

If apache is not installed, the response looks like

```
(4/5): httpd-tools-2.4.6-93.el7.centos.x86_64.rpm                 |  92 kB  00:00:00
(5/5): httpd-2.4.6-93.el7.centos.x86_64.rpm                       | 2.7 MB  00:00:00
--------------------------------------------------------------------------------
Total                                              3.4 MB/s | 3.0 MB  00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : apr-1.4.8-5.el7.x86_64                                       1/5
  Installing : apr-util-1.5.2-6.el7.x86_64                                  2/5
  Installing : httpd-tools-2.4.6-93.el7.centos.x86_64                       3/5
  Installing : mailcap-2.1.41-2.el7.noarch                                  4/5
  Installing : httpd-2.4.6-93.el7.centos.x86_64                             5/5
  Verifying  : apr-1.4.8-5.el7.x86_64                                       1/5
  Verifying  : httpd-tools-2.4.6-93.el7.centos.x86_64                       2/5
  Verifying  : mailcap-2.1.41-2.el7.noarch                                  3/5
  Verifying  : httpd-2.4.6-93.el7.centos.x86_64                             4/5
  Verifying  : apr-util-1.5.2-6.el7.x86_64                                  5/5

Installed:
  httpd.x86_64 0:2.4.6-93.el7.centos

Dependency Installed:
  apr.x86_64 0:1.4.8-5.el7                      apr-util.x86_64 0:1.5.2-6.el7
  httpd-tools.x86_64 0:2.4.6-93.el7.centos      mailcap.noarch 0:2.1.41-2.el7

Complete!
```

If apache is installed, the response looks like

[root@server1 ~]# yum install httpd

Loaded plugins: fastestmirror, langpacks

Loading mirror speeds from cached hostfile

** base: ftp.plusline.de*

```
 * extras: mirror.23media.de
 * updates: mirror.23media.de
Package httpd-2.4.6-17.el7.centos.1.x86_64 already installed and
latest version Nothing to do
[root@server1 ~]#
```

By default, apache will be installed. If it is not installed, please install it as shown above.

Now configure your system to start Apache at boot time...

```
systemctl start httpd.service

systemctl enable httpd.service
```

In CentOS 7.0, uses Firewall-cmd. I will customize it to allow external access to port *80* (http) and *443* (https).

```
firewall-cmd --permanent --zone=public --add-service=http
firewall-cmd --permanent --zone=public --add-service=https
firewall-cmd --reload
```

Open a browser and enter "http://192.168.0.100" in the address bar. You should see the Apache2 placeholder page: (**Note: you need to use your IP instead of *192.168.0.100***) *(To check your IP, type command "ip a | grep inet")*

4 Installing PHP7.4

We can install PHP7 with modules as follows:

```
yum install epel-release yum-utils -y

yum install http://rpms.remirepo.net/enterprise/remi-release-7.rpm -y

yum-config-manager --enable remi-php74

yum install php php-common php-mcrypt php-cli php-gd php-curl php-mysql -y
```

We must restart Apache afterwards:

```
systemctl restart httpd.service
```

5 Testing PHP7.4 / Getting Details About Your PHP7.4 Installation

The document root of the default website is /var/www/html. We will now create a small PHP file (info.php) in that directory and call it in a browser. The file will display lots of useful details about our PHP installation, such as the installed PHP version.

```
vi /var/www/html/info.php
```

```
<?php  phpinfo(); ?>
```

Reminder in using vi editor

[root@server33 ~]#vi /var/www/html/info.php

Press <i> to enter input mode.

Start typing

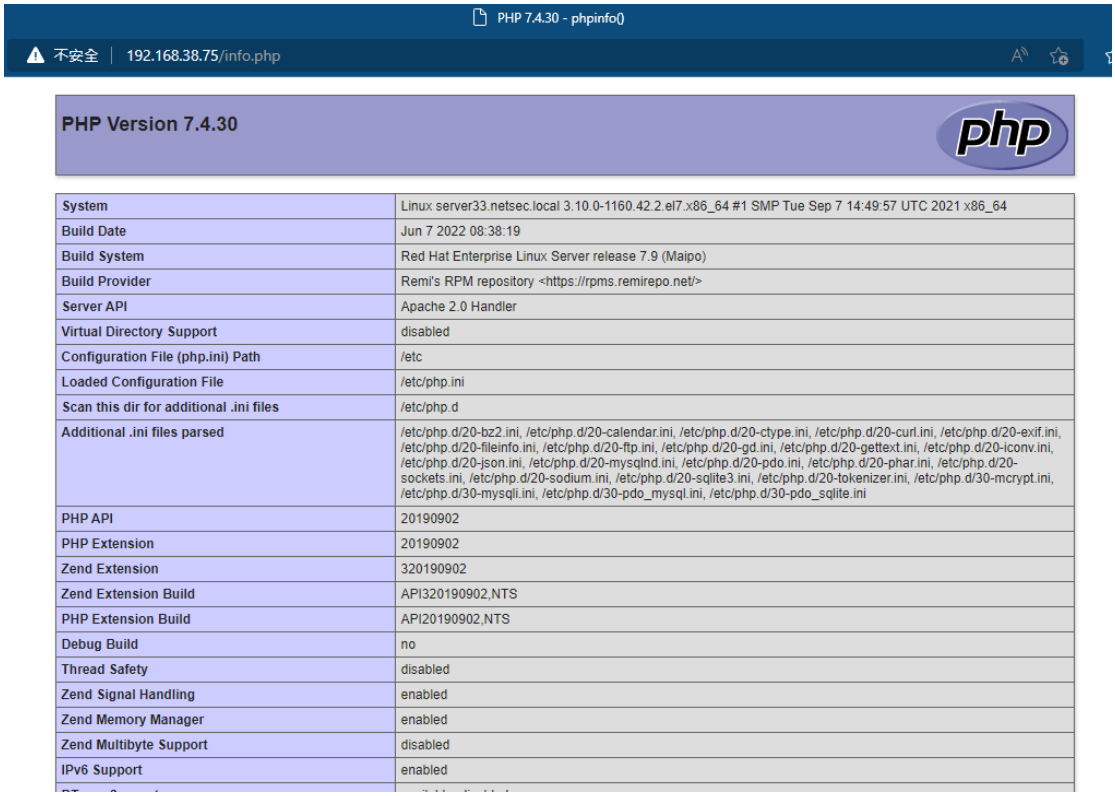
```
<?php phpinfo(); ?>

Press <Esc> and type ":w!" to write file.

Press <Esc> and type ":q" to quit vi editor.
```

Now we call that file in a browser (e.g. http://192.168.0.100/info.php):

(**Note: you need to use your IP instead of *192.168.0.100***)



System	Linux server33.netsec.local 3.10.0-1160.42.2.el7.x86_64 #1 SMP Tue Sep 7 14:49:57 UTC 2021 x86_64
Build Date	Jun 7 2022 08:38:19
Build System	Red Hat Enterprise Linux Server release 7.9 (Maipo)
Build Provider	Remi's RPM repository <https://rpms.remirepo.net/>
Server API	Apache 2.0 Handler
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-curl.ini, /etc/php.d/20-exif.ini, /etc/php.d/20-fileinfo.ini, /etc/php.d/20-ftp.ini, /etc/php.d/20-gd.ini, /etc/php.d/20-gettext.ini, /etc/php.d/20-iconv.ini, /etc/php.d/20-json.ini, /etc/php.d/20-mysqlnd.ini, /etc/php.d/20-pdo.ini, /etc/php.d/20-phar.ini, /etc/php.d/20-sockets.ini, /etc/php.d/20-sodium.ini, /etc/php.d/20-sqlite3.ini, /etc/php.d/20-tokenizer.ini, /etc/php.d/30-mcrypt.ini, /etc/php.d/30-mysqli.ini, /etc/php.d/30-pdo_mysql.ini, /etc/php.d/30-pdo_sqlite.ini
PHP API	20190902
PHP Extension	20190902
Zend Extension	320190902
Zend Extension Build	API320190902,NTS
PHP Extension Build	API20190902,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled

PHP7 is working now, and it's working through the *`Apache 2.0`* Handler, as shown in the *`Server API line`*. If you scroll further down, you will see all modules that are already enabled in PHP7. MySQL is not listed there which means we don't have MySQL support in PHP7 yet. [Note: This is your first PHP page.]

6 Getting MySQL Support In PHP7.4

To get MySQL support in PHP, we can install the *`php-mysql`* package. It's a good idea to install some other PHP7 modules as well as you might need them for your applications.

You can search for available PHP7 modules like this:

```
yum search php
```

Pick the ones you need and install them like this:

```
yum -y install php-mysql
```

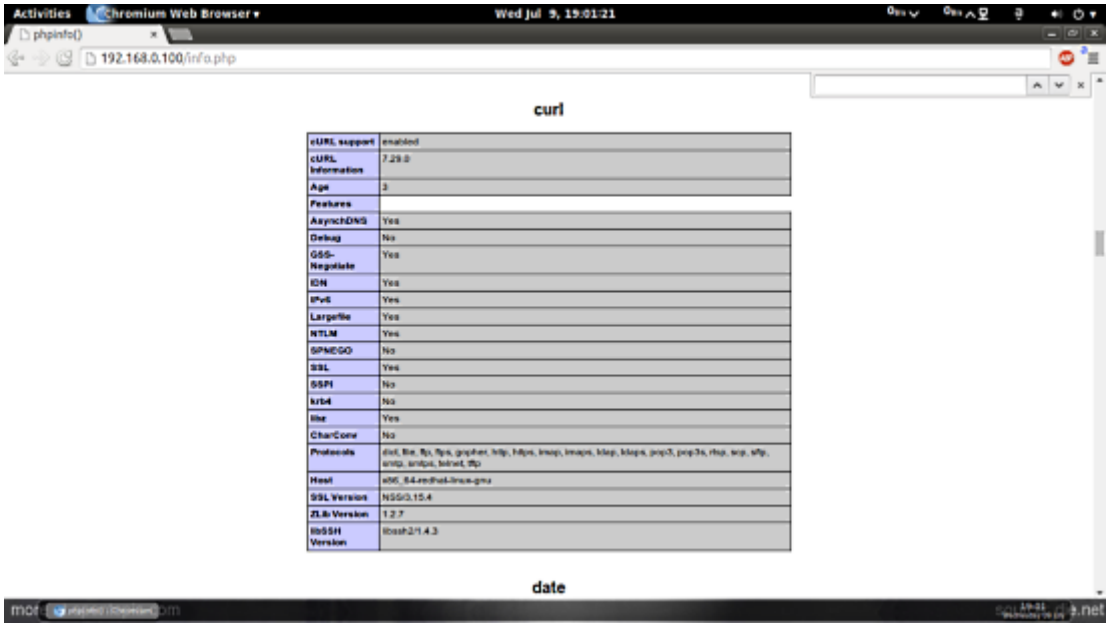
In the next step I will install some common PHP modules that are required by CMS Systems like Wordpress, Joomla and Drupal:

```
yum -y install php-gd php-ldap php-odbc php-pear php-xml php-xmlrpc php-mbstring php-snmp php-soap curl curl-devel
```

Now restart Apache2:

```
systemctl restart httpd.service
```

Now reload `http://192.168.0.100/info.php` in your browser and scroll down to the modules section again. You should now find lots of new modules like crul and mysql etc there.:



7 phpMyAdmin installation

phpMyAdmin is a web interface through which you can manage your MySQL databases. phpMyAdmin can now be installed as follows:

```
yum -y install phpMyAdmin
```

Now we configure phpMyAdmin. We change the Apache configuration so that phpMyAdmin allows connections not just from localhost:

```
vi /etc/httpd/conf.d/phpMyAdmin.conf
```

```
[...]
Alias /phpMyAdmin /usr/share/phpMyAdmin

Alias /phpmyadmin /usr/share/phpMyAdmin


<Directory /usr/share/phpMyAdmin/>

   <IfModule mod_authz_core.c>

     # Apache 2.4

     <RequireAny>

#add the following 1 line

       Require all granted

       Require ip 127.0.0.1

       Require ip ::1

     </RequireAny>

   </IfModule>

   <IfModule !mod_authz_core.c>

     # Apache 2.2
```

```
#change the following 2 lines

     Order Allow,Deny

     Allow from All

     Allow from 127.0.0.1

     Allow from ::1

   </IfModule>

</Directory>

 [...]
```

Next we change the authentication in phpMyAdmin from *cookie* to *http*:

```
vi /etc/phpMyAdmin/config.inc.php
```

```
[...]

$cfg['Servers'][$i]['auth_type']     = 'http';    // Authentication metho
d (config, http or cookie based)?

[...]
```

Search the above line and change 'cookie' to 'http".

Restart Apache:

```
systemctl restart  httpd.service
```

Afterwards, you can access phpMyAdmin under http://192.168.0.100/phpmyadmin/ .

Use user and password of MySQL/Mariadb (ie. root and netlab123) when prompt.

8 Links

Apache: http://httpd.apache.org/

PHP: http://www.php.net/

MySQL: http://www.mysql.com/

CentOS: http://www.centos.org/

phpMyAdmin: http://www.phpmyadmin.net/

Reference: https://www.howtoforge.com/