

How to Install LAMP Stack on CentOS 8/RHEL 8

This tutorial is going to show you how to install LAMP stack on CentOS 8 and RHEL 8.

What's LAMP Stack?

A software stack is a set of software tools bundled together. LAMP stands for **L**inux, **A**pache, **M**ariaDB/**M**ysql and **P**HP, all of which are open source. It is the most common software stack that powers dynamic websites and web applications. Linux is the operating system; Apache is the web server; MariaDB/MySQL is the database server and PHP is the server-side scripting language responsible for generating dynamic web pages.

Prerequisites

You can download and install RHEL 8 by following the tutorial below.

- [How to Download and Install RHEL 8 for Free](#)

If you are looking for a VPS (Virtual Private Server), then you can register an account at Vultr via [my referral link](#) to get \$50 free credit for use over 30 days.

This tutorial uses root account to manage administration tasks. To switch to root, run the following command and enter root password.

```
su -
```

Step 1: Update Software Packages

Before we install the LAMP stack, it's a good idea to run the following command to update repository and software packages.

```
dnf update
```

Step 2: Install Apache Web Server on CentOS 8/RHEL 8

Enter the following command to install Apache Web server. The `httpd-tools` package will install some useful utilities like Apache HTTP server benchmarking tool (ab).

```
dnf install httpd httpd-tools
```

After it's installed, we can start Apache with this command:

```
systemctl start httpd
```

Enable Apache to auto start at system boot time by running the following command.

```
systemctl enable httpd
```

Now check its status.

```
systemctl status httpd
```

Output:

```
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Sat 2019-10-12 06:43:15 UTC; 14s ago
     Docs: man:httpd.service(8)
 Main PID: 14515 (httpd)
   Status: "Running, listening on: port 80"
    Tasks: 213 (limit: 5092)
  Memory: 24.8M
   CGroup: /system.slice/httpd.service
           └─14515 /usr/sbin/httpd -DFOREGROUND
           └─14516 /usr/sbin/httpd -DFOREGROUND
           └─14517 /usr/sbin/httpd -DFOREGROUND
           └─14518 /usr/sbin/httpd -DFOREGROUND
```

```
l-14519 /usr/sbin/httpd -DFOREGROUND
```

“**Enabled**” indicates that auto start at boot time is enabled and we can see that Apache is running.

Hint: If the above command doesn't immediately quit after running. You need to press “**q**” to make it quit.

Check Apache version.

```
httpd -v
```

Output:

```
Server version: Apache/2.4.37 (centos)
Server built: Oct 7 2019 21:42:02
```

To test if Apache web server is running properly, we can create an `index.html` file under the default document root (`/var/www/html/`) with the following command.

```
echo "Welcome to this site!" > /var/www/html/index.html
```

If you are installing LAMP on your local CentOS 8/RHEL 8 computer, then type `127.0.0.1` or `localhost` in the browser address bar. You should see the welcome message, which means Apache Web server is running properly.



By default, CentOS 8/RHEL 8 forbids public access to port 80. To allow other computers to access the web page, we need to open port 80 in firewalld, the dynamic firewall manager on RHEL/CentOS. Run the following command to open port 80.

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

If you want to enable HTTPS on Apache later, then you also need to open port 443.

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

The `--permanent` option will make this firewall rule persistent across system reboots. Next, reload

the firewall daemon for the change to take effect.

```
systemctl reload firewalld
```

Now the Apache web page is accessible publicly.

We need to make user `apache` as the owner of web directory. By default it's owned by the root user.

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

By default, Apache uses the system hostname as its global `ServerName`. If the system hostname can't be resolved in DNS, then you will probably see the following error after running `sudo apachectl configtest` command.

```
AH00558: apache2: Could not reliably determine the server's fully qualified domain name,
using 127.0.0.1. Set the 'ServerName' directive globally to suppress this message
```

To solve this problem, we can set a global `ServerName` in Apache. Install the Nano command-line text editor and use it to create a new configuration file.

```
sudo dnf install nano
```

```
sudo nano /etc/httpd/conf.d/servername.conf
```

Add the following line in this file.

```
ServerName localhost
```

Save and close the file. To save a file in Nano text editor, press `Ctrl+O`, then press Enter to confirm. To exit, press `Ctrl+X`. Reload Apache for the change to take effect.

```
sudo systemctl reload httpd
```

Now if you run the `sudo apachectl configtest` command again, you won't see the above error message.

Step 3: Install MariaDB Database Server on CentOS 8/RHEL 8

MariaDB is a drop-in replacement for MySQL. It is developed by former members of MySQL team who are concerned that Oracle might turn MySQL into a closed-source product. Enter the following command to install MariaDB on CentOS 8/RHEL 8.

```
dnf install mariadb-server mariadb -y
```

After it's installed, we need to start it.

```
systemctl start mariadb
```

Enable auto start at system boot time.

```
systemctl enable mariadb
```

Check status:

```
systemctl status mariadb
```

output:

```
● mariadb.service - MariaDB 10.3 database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; vendor preset: disabled)
   Active: active (running) since Sat 2019-10-12 09:02:53 UTC; 33s ago
     Docs: man:mysqld(8)
           https://mariadb.com/kb/en/library/systemd/
 Main PID: 18608 (mysqld)
   Status: "Taking your SQL requests now..."
    Tasks: 30 (limit: 5092)
  Memory: 77.0M
   CGroup: /system.slice/mariadb.service
           └─18608 /usr/libexec/mysqld --basedir=/usr
```

“**Enabled**” indicates that auto start at boot time is enabled and we can see that MariaDB server is running. Now we need to run the security script.

```
mysql_secure_installation
```

When it asks you to enter MariaDB root password, press Enter key as the root password isn't set yet. Then enter `|y|` to set the root password for MariaDB server.

```
File Edit View Search Terminal Help
[root@rhel8 ~]# 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): Press 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] y Enter y to set a new MariaDB root password
New password: █
```

Next, you can press Enter to answer all remaining questions, which will remove anonymous user, disable remote root login and remove test database. This step is a basic requirement for MariaDB database security. (Note that the letter `[Y]` is capitalized, which means it's the default answer.)

```
Remove anonymous users? [Y/n] Press Enter
... 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] Press 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] Press 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] Press 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@rhel8 ~]# █
```

Now you can run the following command and enter MariaDB root password to log into MariaDB

shell.

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

```
File Edit View Search Terminal Help
[root@rhel8 ~]# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 18
Server version: 10.3.10-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> exit;
Bye
[root@rhel8 ~]# █
```

To exit, run

```
exit;
```

Step 4: Install PHP on CentOS 8/RHEL 8

Install PHP and some common modules using the following command.

```
dnf install php php-fpm php-mysqlnd php-opcache php-gd php-xml php-mbstring -y
```

Apache web server on CentOS 8/RHEL 8 by default uses PHP-FPM instead of mod_php to run PHP code, so in the above command we also installed `php-fpm`. After it's installed, we need to start it.

```
systemctl start php-fpm
```

Enable auto start at system boot time.

```
systemctl enable php-fpm
```

Check status:

```
systemctl status php-fpm
```

output:

```
● php-fpm.service - The PHP FastCGI Process Manager
   Loaded: loaded (/usr/lib/systemd/system/php-fpm.service; enabled; vendor preset: disabled)
   Active: active (running) since Sat 2019-10-12 09:54:37 UTC; 3s ago
 Main PID: 19755 (php-fpm)
  Status: "Ready to handle connections"
   Tasks: 6 (limit: 5092)
  Memory: 24.5M
  CGroup: /system.slice/php-fpm.service
          └─19755 php-fpm: master process (/etc/php-fpm.conf)
             └─19757 php-fpm: pool www
                └─19758 php-fpm: pool www
                   └─19759 php-fpm: pool www
                      └─19760 php-fpm: pool www
                         └─19761 php-fpm: pool www
```

“**Enabled**” indicates that auto start at boot time is enabled and we can see that PHP-FPM is running. The `php-fpm` package installs a `php.conf` file in `/etc/httpd/conf.d/` directory, so we need to restart Apache web server, in order to run PHP code.

```
systemctl restart httpd
```

We also need to run the following command to tell SELinux to allow Apache to execute PHP code via PHP-FPM.

```
setsebool -P httpd_execmem 1
```

Step 5: Test PHP

To test PHP-FPM with Apache Web server, we need to create a `info.php` file in the document root directory.

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

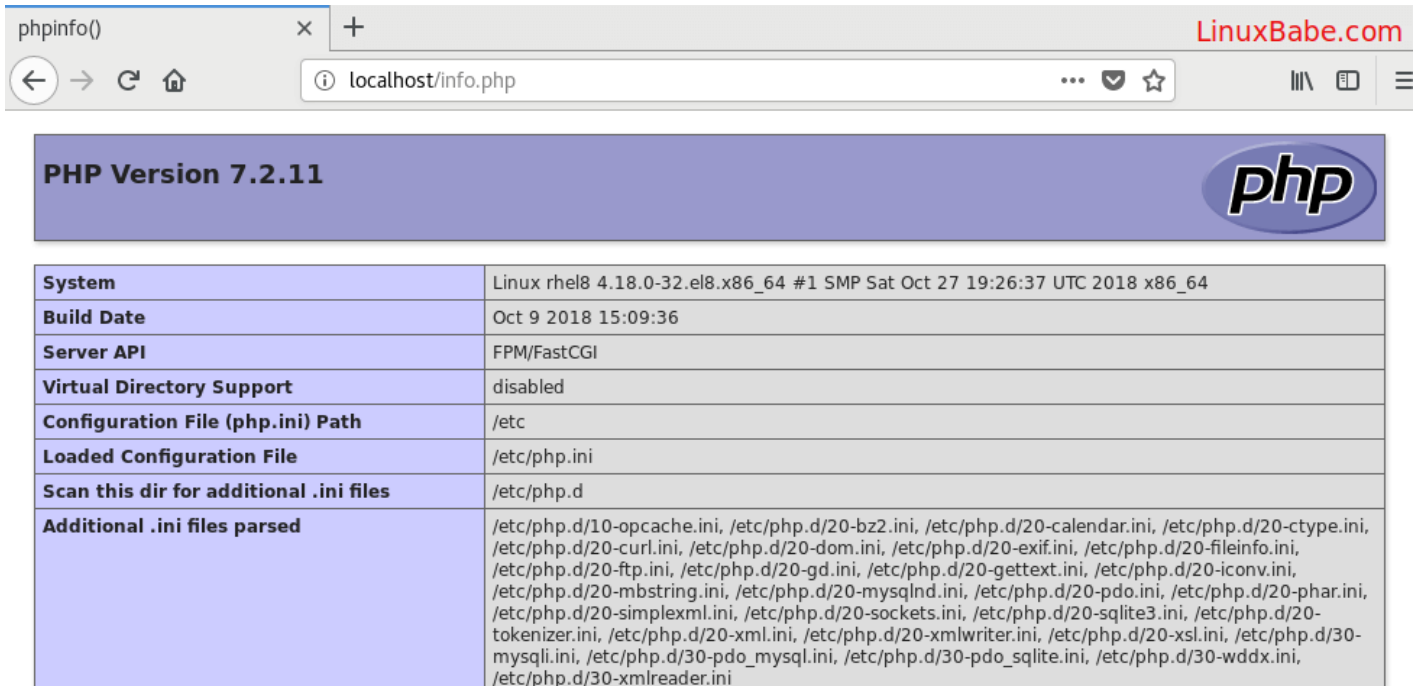
Paste the following PHP code into the file.

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

Save and close the file. If you installed LAMP stack on a local CentOS 8/RHEL 8 server, type in `127.0.0.1/info.php` or `localhost/info.php` in the browser address bar. You should see your

server's PHP information. This means PHP scripts can run properly with Apache web server.

If RHEL 8/CentOS is running on a remote server, then enter `server-ip-address/info.php` in browser address bar. Replace `server-ip-address` with your actual IP address.



The screenshot shows a web browser window with the address bar containing `localhost/info.php`. The page title is `phpinfo()`. The page content displays the PHP version **7.2.11** and the PHP logo. Below the header is a table with the following information:

System	Linux rhel8 4.18.0-32.el8.x86_64 #1 SMP Sat Oct 27 19:26:37 UTC 2018 x86_64
Build Date	Oct 9 2018 15:09:36
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/10-opcache.ini, /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-dom.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-mbstring.ini, /etc/php.d/20-mysqlnd.ini, /etc/php.d/20-pdo.ini, /etc/php.d/20-phar.ini, /etc/php.d/20-simplexml.ini, /etc/php.d/20-sockets.ini, /etc/php.d/20-sqlite3.ini, /etc/php.d/20-tokenizer.ini, /etc/php.d/20-xml.ini, /etc/php.d/20-xmlwriter.ini, /etc/php.d/20-xsl.ini, /etc/php.d/30-mysqli.ini, /etc/php.d/30-pdo_mysql.ini, /etc/php.d/30-pdo_sqlite.ini, /etc/php.d/30-wddx.ini, /etc/php.d/30-xmlreader.ini

If the browser fails to display the PHP info but prompt you to download the **info.php** file, simply restart Apache and PHP-FPM.

```
sudo systemctl restart httpd php-fpm
```

Then you should be able to see the PHP info in the web browser.

Apache Automatic Restart

If for any reason your Apache process is killed, you need to run the following command to restart it.

```
sudo systemctl restart httpd
```

Instead of manually typing this command, we can make Apache automatically restart by editing the `httpd.service` systemd service unit. To override the default systemd service configuration, we create a separate directory.

```
sudo mkdir -p /etc/systemd/system/httpd.service.d/
```

Then create a file under this directory.

```
sudo nano /etc/systemd/system/httpd.service.d/restart.conf
```

Add the following lines in the file, which will make Apache automatically restart 5 seconds after a failure is detected.

```
[Service]
Restart=always
RestartSec=5s
```

Save and close the file. Then reload systemd.

```
sudo systemctl daemon-reload
```

To check if this would work, kill Apache with:

```
sudo pkill httpd
```

Then check Apache status. You will find Apache automatically restarted.

```
systemctl status httpd
```

Allow Apache to Make Outgoing Network Connections

By default, SELinux forbids Apache to make outgoing network connections. If Apache needs to make requests to an outside network service, then run the following command to allow this action.

```
setsebool -P httpd_can_network_connect on
```

Revision #1

Created 11 July 2021 19:02:43 by Admin

Updated 11 July 2021 19:03:19 by Admin