Comment installer Prestashop sur Debian-12

Prestashop is a free open-source e-commerce solution written in PHP. It allows you to host and create online shops yourself and grow your online business. Prestashop is a fully customisable and feature-rich e-commerce solution that allows you to create comprehensive e-commerce websites. Prestashop is now used by more than 250,000 online shops around the world and is available in 65 languages.

In this tutorial, I will guide you step by step through the installation of PrestaShop on a Debian 12 server. You install PrestaShop with the LAMP stack (Apache2, MariaDB and PHP) and then secure your PrestaShop installation with SL/TLS certificates via Letsencrypt.

Prerequisites

Before you start, please have the following ready:

- A Debian 12 server.
- A non-root user with administrator privileges.
- A domain name pointed to the server IP address.

Installing Dependencies

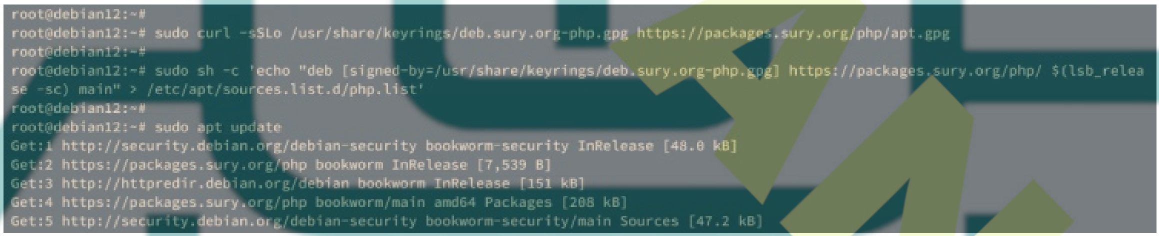
PrestaShop is an open-source platform for creating an online store written in PHP with MariaDB/MySQL as the database. In the following step, you will install dependencies for PrestaShop, which includes LAMP Stack (Apache2, MariaDB, and PHP) packages with some additional PHP extensions. You will install specifically PHP 8.1 via a third-party repository because the current PrestaShop still does not yet support PHP 8.2.

First, run the following command to add a third-party PHP repository and GPG key to your Debian system.

```
sudo curl -sSLo /usr/share/keyrings/deb.sury.org-php.gpg https://packages.sury.org/php/apt.gpg
sudo sh -c 'echo "deb [signed-by=/usr/share/keyrings/deb.sury.org-php.gpg] https://packages.sury.org/php/ $(lsb_release -sc) main" >
/etc/apt/sources.list.d/php.list'
```

Update and refresh your Debian package index by executing the following apt command.

```
sudo apt update
```

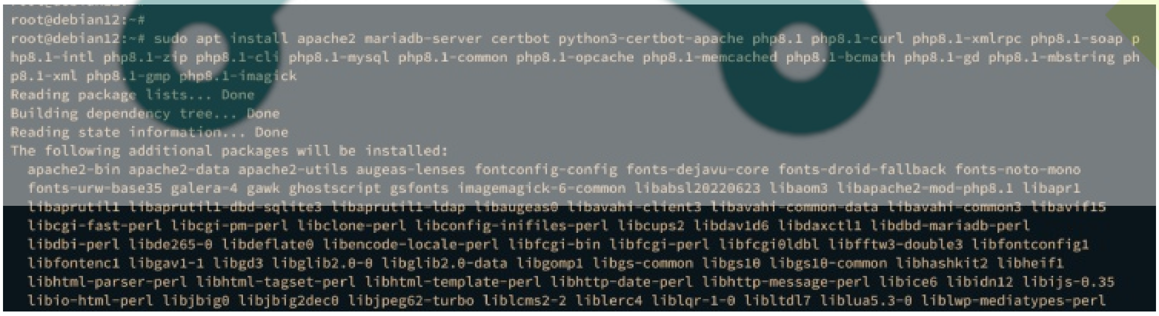


```
root@debian12:~#
root@debian12:~# sudo curl -sSLo /usr/share/keyrings/deb.sury.org-php.gpg https://packages.sury.org/php/apt.gpg
root@debian12:~#
root@debian12:~# sudo sh -c 'echo "deb [signed-by=/usr/share/keyrings/deb.sury.org-php.gpg] https://packages.sury.org/php/ $(lsb_relea
se -sc) main" > /etc/apt/sources.list.d/php.list'
root@debian12:~#
root@debian12:~# sudo apt update
Get:1 http://security.debian.org/debian-security bookworm-security InRelease [48.0 kB]
Get:2 https://packages.sury.org/php bookworm InRelease [7,539 B]
Get:3 http://httpredir.debian.org/debian bookworm InRelease [151 kB]
Get:4 https://packages.sury.org/php bookworm/main amd64 Packages [208 kB]
Get:5 http://security.debian.org/debian-security bookworm-security/main Sources [47.2 kB]
```

Once the repository is updated, install dependencies for PrestaShop using the following command. This will install *LAMP Stack* (Apache2, MariaDB, PHP 8.1) packages, *Certbot*, *Certbot Apache plugin*, and *additional PHP extensions* that are required by PrestaShop.

```
sudo apt install apache2 mariadb-server certbot python3-certbot-apache php8.1 php8.1-curl php8.1-xmlrpc php8.1-soap php8.1-intl
php8.1-zip php8.1-cli php8.1-mysql php8.1-common php8.1-opcache php8.1-memcached php8.1-bcmath php8.1-gd php8.1-mbstring php8.1-xml
php8.1-gmp php8.1-imagick unzip
```

Type y to proceed with the installation.



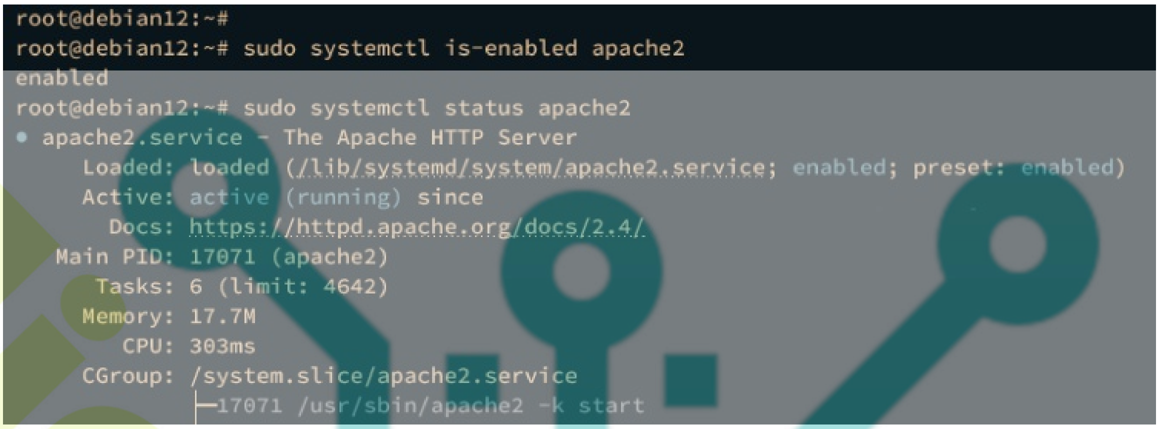
```
root@debian12:~#
root@debian12:~# sudo apt install apache2 mariadb-server certbot python3-certbot-apache php8.1 php8.1-curl php8.1-xmlrpc php8.1-soap p
hp8.1-intl php8.1-zip php8.1-cli php8.1-mysql php8.1-common php8.1-opcache php8.1-memcached php8.1-bcmath php8.1-gd php8.1-mbstring ph
p8.1-xml php8.1-gmp php8.1-imagick
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils augeas-lenses fontconfig-config fonts-dejavu-core fonts-droid-fallback fonts-noto-mono
  fonts-urw-base35 galera-4 gawk ghostscript gsfonts imagemagick-6-common libabsl20220623 libaom3 libapache2-mod-php8.1 libapr1
  libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libaugeas0 libavahi-client3 libavahi-common-data libavahi-common3 libavif15
  libcgi-fast-perl libcgi-pm-perl libclone-perl libconfig-inifiles-perl libcups2 libdav1d6 libdaxctl1 libdbd-mariadb-perl
  libdbi-perl libde265-0 libdeflate0 libencode-locale-perl libfcgi-bin libfcgi-perl libfcgi0ldbl libfftw3-double3 libfontconfig1
  libfontenc1 libgav1-1 libgd3 libglib2.0-0 libglib2.0-data libgomp1 libgs-common libgs10 libgs10-common libhashkit2 libheif1
  libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl libhttp-date-perl libhttp-message-perl libice6 libidn12 libijs-0.35
  libio-html-perl libjbig0 libjbig2dec0 libjpeg62-turbo liblcms2-2 liblerc4 liblqr-1-0 libltdl7 liblua5.3-0 liblwp-mediatypes-perl
```

After dependencies are installed, verify each dependency using the following command.

Verify the **apache2** service to ensure that the service is running by executing the systemctl command below.

```
sudo systemctl is-enabled apache2
sudo systemctl status apache2
```

The following result will be shown, which confirms that apache2 service is running and enabled.

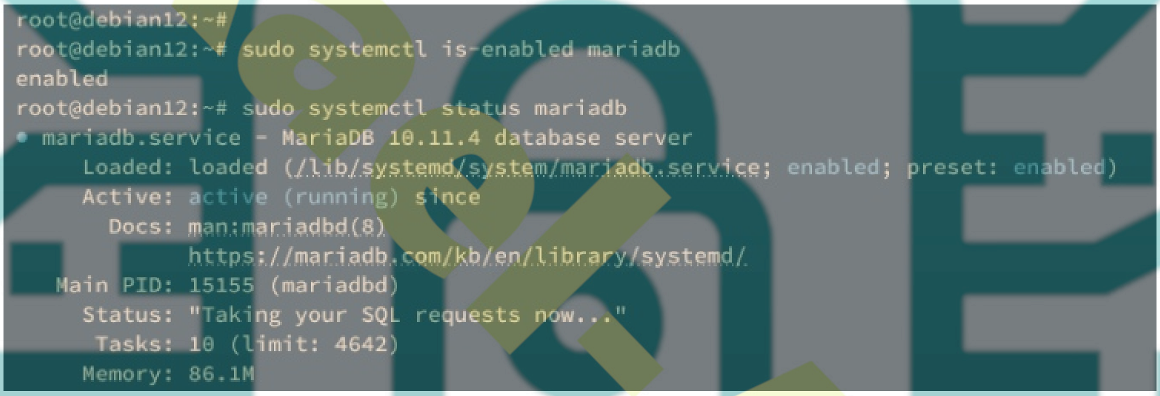


```
root@debian12:~#
root@debian12:~# sudo systemctl is-enabled apache2
enabled
root@debian12:~# sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 17071 (apache2)
      Tasks: 6 (limit: 4642)
     Memory: 17.7M
        CPU: 303ms
     CGroup: /system.slice/apache2.service
             ├─17071 /usr/sbin/apache2 -k start
```

Now verify the **mariadb** service using the following command.

```
sudo systemctl is-enabled mariadb
sudo systemctl status mariadb
```

The displayed output below confirms that the mariadb service is running and enabled.

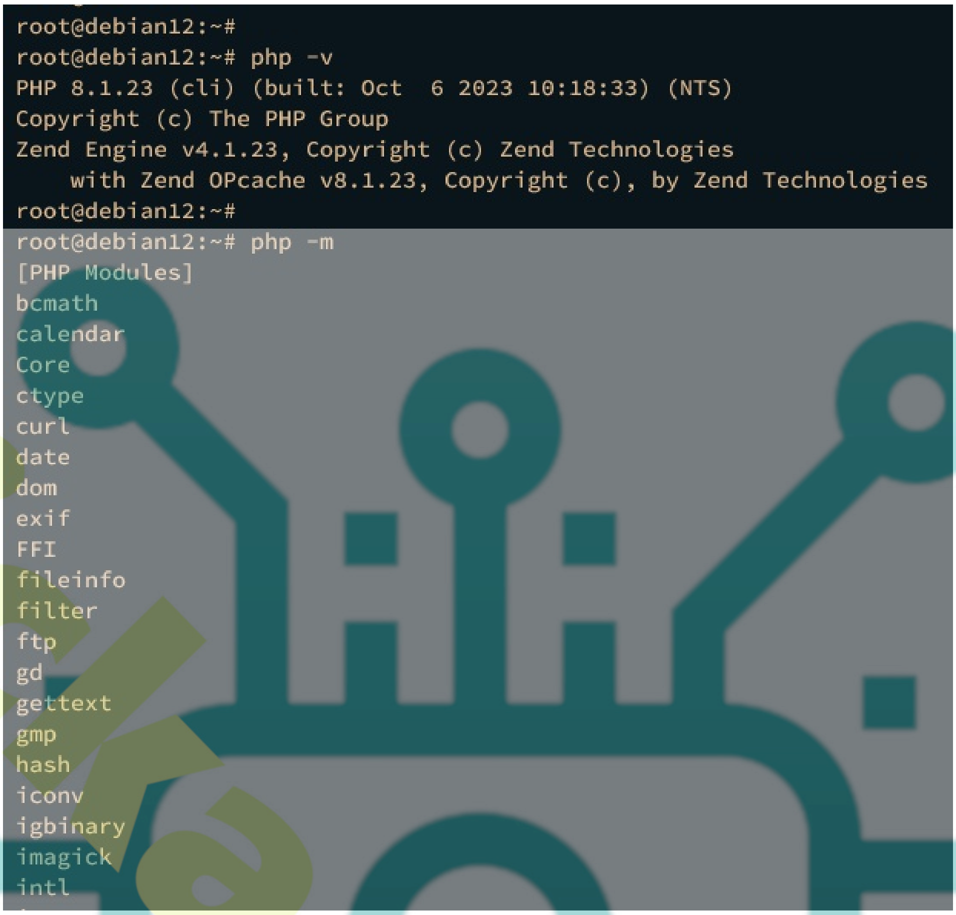


```
root@debian12:~#
root@debian12:~# sudo systemctl is-enabled mariadb
enabled
root@debian12:~# sudo systemctl status mariadb
● mariadb.service - MariaDB 10.11.4 database server
     Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; preset: enabled)
     Active: active (running) since
       Docs: man:mariadbd(8)
             https://mariadb.com/kb/en/library/systemd/
   Main PID: 15155 (mariadbd)
     Status: "Taking your SQL requests now..."
      Tasks: 10 (limit: 4642)
     Memory: 86.1M
```

Lastly, run the following command to verify your PHP version and list enabled extensions.

```
php -v
php -m
```

You should see that **PHP 8.1** is installed on your Debian machine with the list extensions enabled.



```
root@debian12:~#
root@debian12:~# php -v
PHP 8.1.23 (cli) (built: Oct  6 2023 10:18:33) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.1.23, Copyright (c) Zend Technologies
    with Zend OPcache v8.1.23, Copyright (c), by Zend Technologies
root@debian12:~#
root@debian12:~# php -m
[PHP Modules]
bcmath
calendar
Core
ctype
curl
date
dom
exif
FFI
fileinfo
filter
ftp
gd
gettext
gmp
hash
iconv
igbinary
imagick
intl
```

Configuring MariaDB Server

Now that you've installed dependencies for PrestaShop, the next step is to secure your MariaDB Server installation via the *mariadb-secure-installation* utility. Then, you will need to create a new database and user for PrestaShop.

Execute the *mariadb-secure-installation* command below to secure your MariaDB Server installation.

```
sudo mariadb-secure-installation
```

Input **Y** to confirm and apply the new configuration, or **n** for No, and reject it. Below of MariaDB Server configurations that you will be asked for:

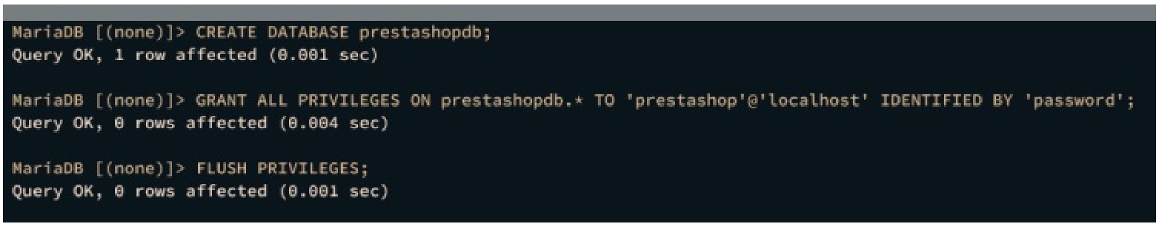
- Switch to unix_socket authentication?. Input n and press ENTER. The default MariaDB root user is already protected. optionally, you can also enable it by typing y for yes.
- Change the root password?. Input y to confirm and set up your new MariaDB root password.
- Remove anonymous user?. Input y to confirm.
- Disallow root login remotely? Input y to confirm. Only local connection will be allowed if you are using the MariaDB root user.
- Remove test database and access to it?. Input y to confirm and remove the default database 'test'.
- Lastly, input y again to reload all tables privileges on your MariaDB server and apply new changes.

Next, run the *mariadb* client command below to log in to the MariaDB Server. Input your MariaDB root password when prompted.

```
sudo mariadb -u root -p
```

Once logged in, execute the following queries to create a new database **prestashopdb**, a user **prestashop**, with the password is **password**. The new MariaDB database and user here will be the user for PrestaShop installation.

```
CREATE DATABASE prestashopdb;
GRANT ALL PRIVILEGES ON prestashopdb.* TO 'prestashop'@'localhost' IDENTIFIED BY 'password';
FLUSH PRIVILEGES;
```

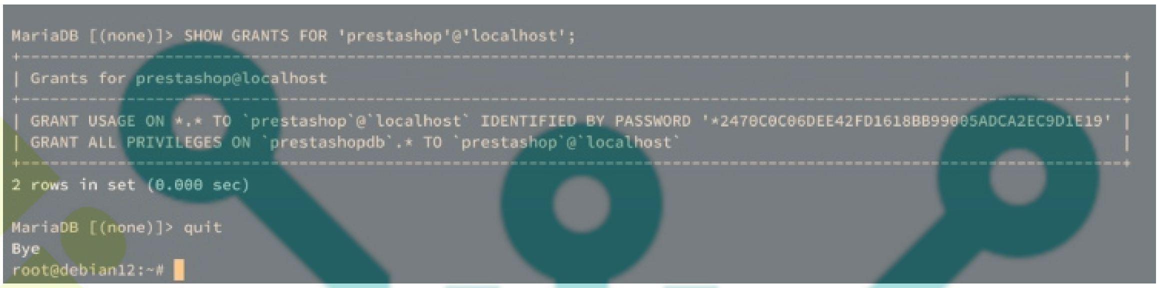


```
MariaDB [(none)]> CREATE DATABASE prestashopdb;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON prestashopdb.* TO 'prestashop'@'localhost' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.004 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.001 sec)
```

Next, run the following query to verify the MariaDB user **prestashop**.

```
SHOW GRANTS FOR 'prestashop'@'localhost';
```

The displayed output below reveals that the MariaDB user **prestashop** is allowed to access the database **prestashopdb**.



```
MariaDB [(none)]> SHOW GRANTS FOR 'prestashop'@'localhost';
+------------------------------------------------------------------------------------------------------------+
| Grants for prestashop@localhost                                                                              |
+------------------------------------------------------------------------------------------------------------+
| GRANT USAGE ON *.* TO `prestashop`@`localhost` IDENTIFIED BY PASSWORD '*2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19' |
| GRANT ALL PRIVILEGES ON `prestashopdb`.* TO `prestashop`@`localhost`                                         |
+------------------------------------------------------------------------------------------------------------+
2 rows in set (0.000 sec)

MariaDB [(none)]> quit
Bye
root@debian12:~#
```

Lastly, type quit to exit from the MariaDB Server.

Configuring PHP

After configuring MariaDB Server, you will configure your PHP installation by modifying the php.ini file. Then, you will also verify your environment to ensure that everything related to PHP configuration is met with PrestaShop requirements.

Open the default PHP configuration */etc/php/8.1/apache2/php.ini* using the following nano editor command.

```
sudo nano /etc/php/8.1/apache2/php.ini
```

Change the configuration below and be sure to adjust the *date.timezone* and *memory_limit* parameters with your server environment.

```
date.timezone = Europe/Amsterdam
max_execution_time = 130
memory_limit = 256M
allow_url_fopen = On
allow_url_include = Off
post_max_size = 128M
upload_max_filesize = 128M
max_input_vars = 5000
```

Save the file and exit the editor when you're done.

Next, run the following systemctl command to restart the *apache2* service and apply the changes.

```
sudo systemctl restart apache2
```

Now that you've configured PHP, you will verify that your server environment is ready for PrestaShop installation. This can be done by using the environment checker provided by PrestaShop.

Move to the */var/www/html* directory and download the environment check script via the wget command below. You will see the file **v1.1.tar.gz** on your current working directory.

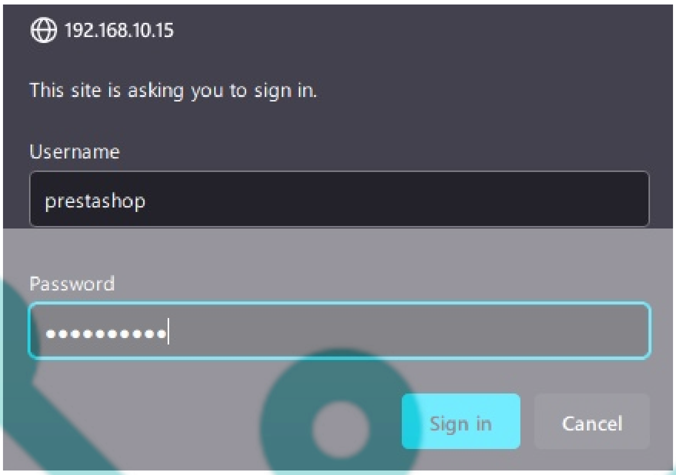
```
cd /var/www/html
wget https://github.com/PrestaShop/php-ps-info/archive/refs/tags/v1.1.tar.gz
```

Extract the file **v1.1.tar.gz** and rename the extracted directory to **check-ps**.

```
tar -xf v1.1.tar.gz
mv php-ps-info-1.1 check-ps
```

Lastly, open your web browser and visit the server IP address followed by the path **check-ps** to access the PrestaShop environment checker script, such as http://192.168.10.15/check-ps/phppsinfo.php.

Input the default username prestashop and password prestashop, then click **Sign in**.



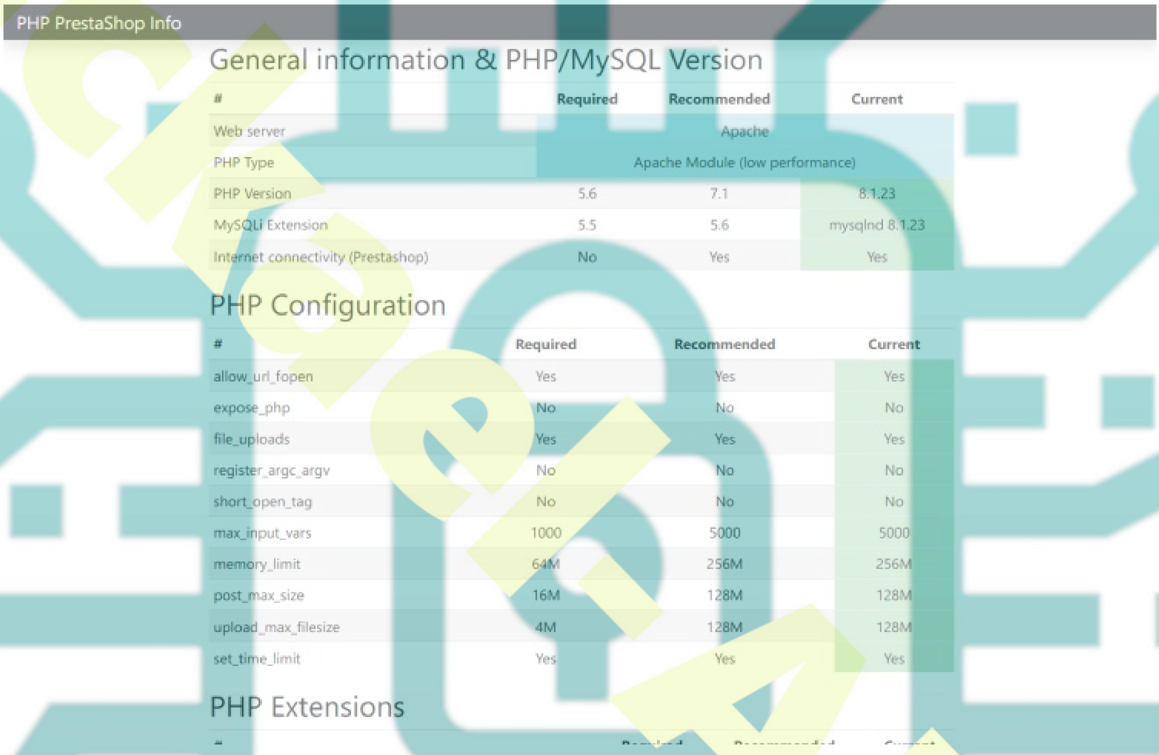
192.168.10.15

This site is asking you to sign in.

Username: prestashop

Password: ••••••••••

Ensure that your dependencies, PHP configurations, and PHP extensions are met with the PrestaShop requirements. If not, you can adjust your settings based on the recommendation on the page.



PHP PrestaShop Info

General information & PHP/MySQL Version

#	Required	Recommended	Current
Web server	Apache		
PHP Type	Apache Module (low performance)		
PHP Version	5.6	7.1	8.1.23
MySQLi Extension	5.5	5.6	mysqlnd 8.1.23
Internet connectivity (Prestashop)	No	Yes	Yes

PHP Configuration

#	Required	Recommended	Current
allow_url_fopen	Yes	Yes	Yes
expose_php	No	No	No
file_uploads	Yes	Yes	Yes
register_argc_argv	No	No	No
short_open_tag	No	No	No
max_input_vars	1000	5000	5000
memory_limit	64M	256M	256M
post_max_size	16M	128M	128M
upload_max_filesize	4M	128M	128M
set_time_limit	Yes	Yes	Yes

PHP Extensions

#	Required	Recommended	Current

Downloading Prestashop

Now that the MariaDB Server and PHP are configured, the next step is to download the PrestaShop source code and configure the installation directory with proper permission and ownership.

Move your working directory to */var/www/* and download the PrestaShop source code via the wget command below. Be sure to check the PrestaShop release page to get the latest version of PrestaShop.

```
cd /var/www/
wget https://github.com/PrestaShop/PrestaShop/releases/download/8.1.2/prestashop_8.1.2.zip
```

Now extract the PrestaShop source code via the unzip command below. You should see the PrestaShop source within the zip file **prestashop.zip**.

```
unzip prestashop_8.1.2.zip
```

Extract the file **prestashop.zip** using the unzip command below to the target directory */var/www/prestashop*. The PrestaShop source code will be extracted to the **prestashop** directory, so your Prestashop installation directory should be **/var/www/prestashop**.

```
unzip prestashop.zip -d /var/www/prestashop
```

Lastly, run the following command to set up the proper ownership and permission for the PrestaShop installation directory */var/www/prestashop*. The ownership should use **www-data** with the permission **u+rw**, which means the owner can read and write to the target directory.

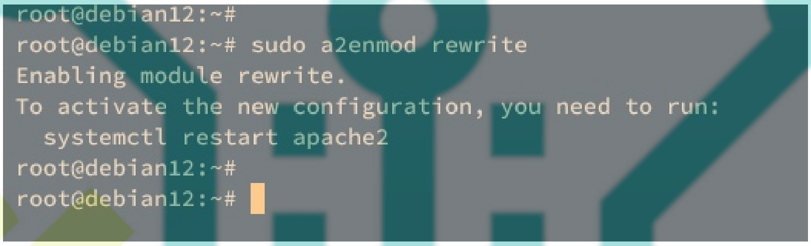
```
sudo chown -R www-data:www-data /var/www/prestashop
sudo chmod u+rw /var/www/prestashop
```

Configuring Apache2 Virtual Host

In the following step, you will create a new Apache2 virtual host configuration that will be used to run prestaShop. So before that, ensure that you've your domain name pointed to a server IP address.

To start, run the following command to enable the **rewrite** module for Apache.

```
sudo a2enmod rewrite
```



```
root@debian12:~#
root@debian12:~# sudo a2enmod rewrite
Enabling module rewrite.
To activate the new configuration, you need to run:
  systemctl restart apache2
root@debian12:~#
root@debian12:~# 
```

Then, create a new virtual host configuration */etc/apache2/sites-available/prestashop.conf* using the nano editor command.

```
sudo nano /etc/apache2/sites-available/prestashop.conf
```

Insert the following configuration and be sure to change the domain name within the **ServerName** parameter.

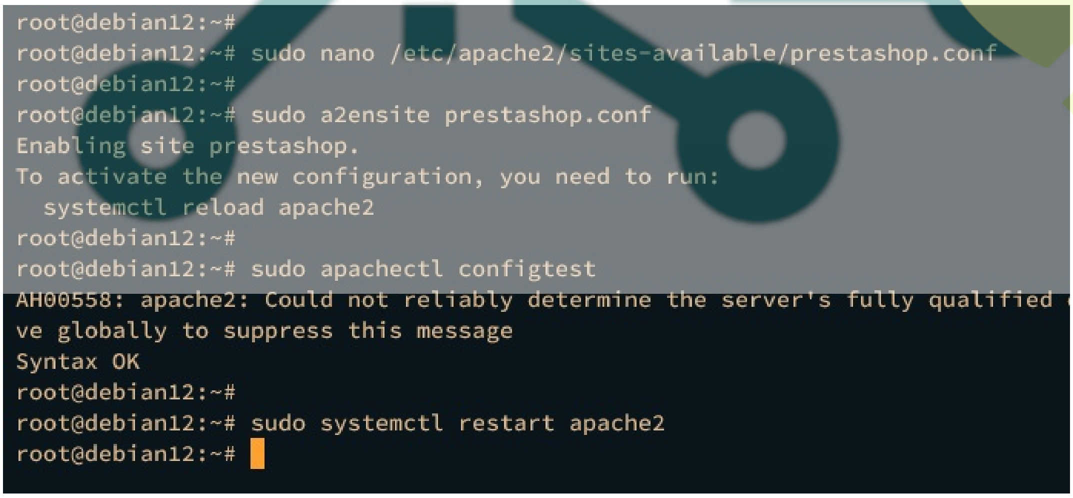
```
<VirtualHost *:80>

    ServerAdmin admin@hwdomain.io
    DocumentRoot /var/www/prestashop
    ServerName hwdomain.io

    <Directory /var/www/prestashop>
        # enable the .htaccess rewrites
        AllowOverride All
        Options +Indexes
        Require all granted

        # Disable back office token
        # SetEnv _TOKEN_ disabled
    </Directory>

    ErrorLog /var/log/apache2/prestashop.error.log
    CustomLog /var/log/apache2/prestashop.access.log combined
</VirtualHost>
```

Save and close the file when you're done.

Next, run the following command to activate the virtual host file *prestashop.conf* and verify your Apache configuration.

```
sudo a2ensite prestashop.conf
sudo apachectl configtest
```

If you've proper Apache configuration, the output **Syntax OK** will be displayed.



```
root@debian12:~#
root@debian12:~# sudo nano /etc/apache2/sites-available/prestashop.conf
root@debian12:~#
root@debian12:~# sudo a2ensite prestashop.conf
Enabling site prestashop.
To activate the new configuration, you need to run:
  systemctl reload apache2
root@debian12:~#
root@debian12:~# sudo apachectl configtest
AH00558: apache2: Could not reliably determine the server's fully qualified
ve globally to suppress this message
Syntax OK
root@debian12:~#
root@debian12:~# sudo systemctl restart apache2
root@debian12:~# 
```

Now run the systemctl command below to restart the *apache2* service and apply the changes.

```
sudo systemctl restart apache2
```

Lastly, generate the SSL/TLS certificates for your PrestaShop domain name by executing the following certbot command. Be sure to change the email address and the domain name with your information.

```
sudo certbot --apache --agree-tos --no-eff-email  --redirect --hsts --staple-ocsp --email alice@hwdomain.io -d hwdomain.io
```

Once the process is finished, your virtual host file **prestashop.conf** will automatically configured with HTTPS and your SSL/TLS certificates will be available at */etc/letsencrypt/live/hwdomain.io* directory.

Installing PrestaShop via Command Line

At this point, you can now finish up your PrestaShop installation via web UI or command line. In this example, you'll finish your PrestaShop installation via the command line.

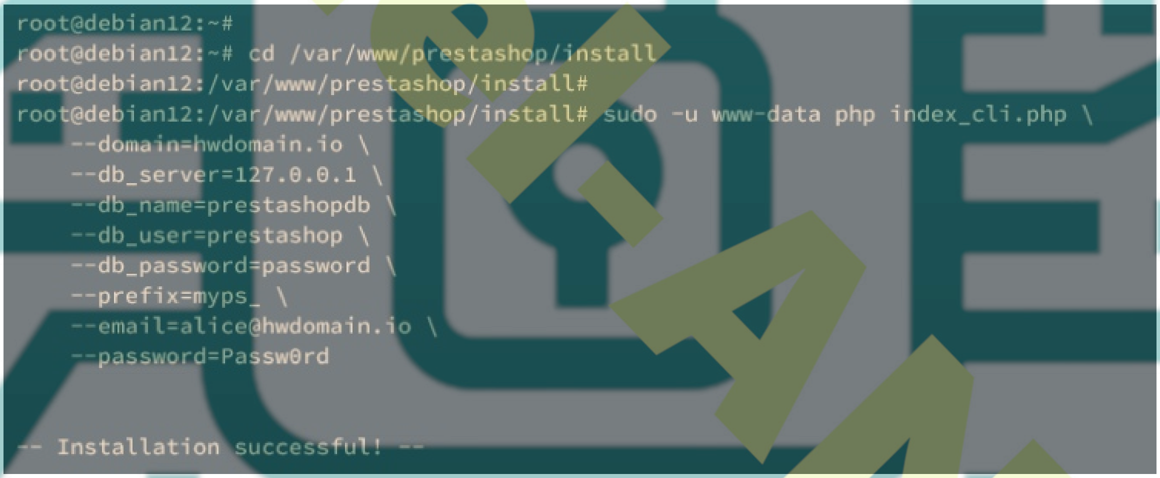
Go to the PrestaShop installation directory */var/www/prestashop/install*.

```
cd /var/www/prestashop/install
```

Run the **index_cli.php** script like the following to start PrestaShop installation via the command line. Be sure to change the details of a domain name, MariaDB database host, name, username, and password. Also, be sure to change the email address and password that will be used as admin for PrestaShop.

```
sudo -u www-data php index_cli.php --domain=hwdomain.io --db_server=127.0.0.1 --db_name=prestashopdb --db_user=prestashop --
db_password=password --prefix=myps_ --email=alice@hwdomain.io --password=Passw0rd
```

Once the installation is finished, the output "**Installation successful**" will be shown like the following.



```
root@debian12:~#
root@debian12:~# cd /var/www/prestashop/install
root@debian12:/var/www/prestashop/install#
root@debian12:/var/www/prestashop/install# sudo -u www-data php index_cli.php \
    --domain=hwdomain.io \
    --db_server=127.0.0.1 \
    --db_name=prestashopdb \
    --db_user=prestashop \
    --db_password=password \
    --prefix=myps_ \
    --email=alice@hwdomain.io \
    --password=Passw0rd


-- Installation successful! --
```

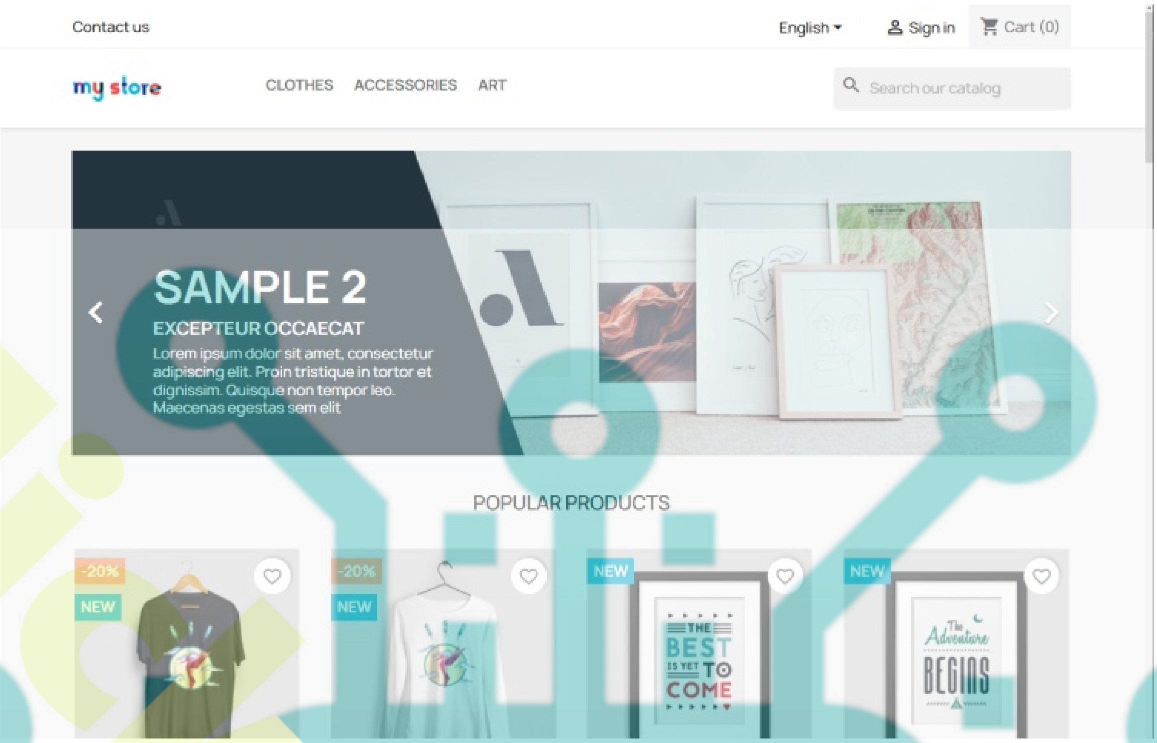
Next, run the following command to set up proper permission for some PrestaShop directories. This will allow the **www-data** as the owner to read and write to those target directories.

```
sudo chmod u+rw /var/www/prestashop/var/cache
sudo chmod u+rw /var/www/prestashop/var/logs
sudo chmod u+rw /var/www/prestashop/img
sudo chmod u+rw /var/www/prestashop/mails
sudo chmod u+rw /var/www/prestashop/modules
sudo chmod u+rw /var/www/prestashop/translations
sudo chmod u+rw /var/www/prestashop/upload
sudo chmod u+rw /var/www/prestashop/download
sudo chmod u+rw /var/www/prestashop/app/config
sudo chmod u+rw /var/www/prestashop/app/Resources/translations
```

Now run the command below to remove the */var/www/prestashop/install* directory and secure your PrestaShop installation.

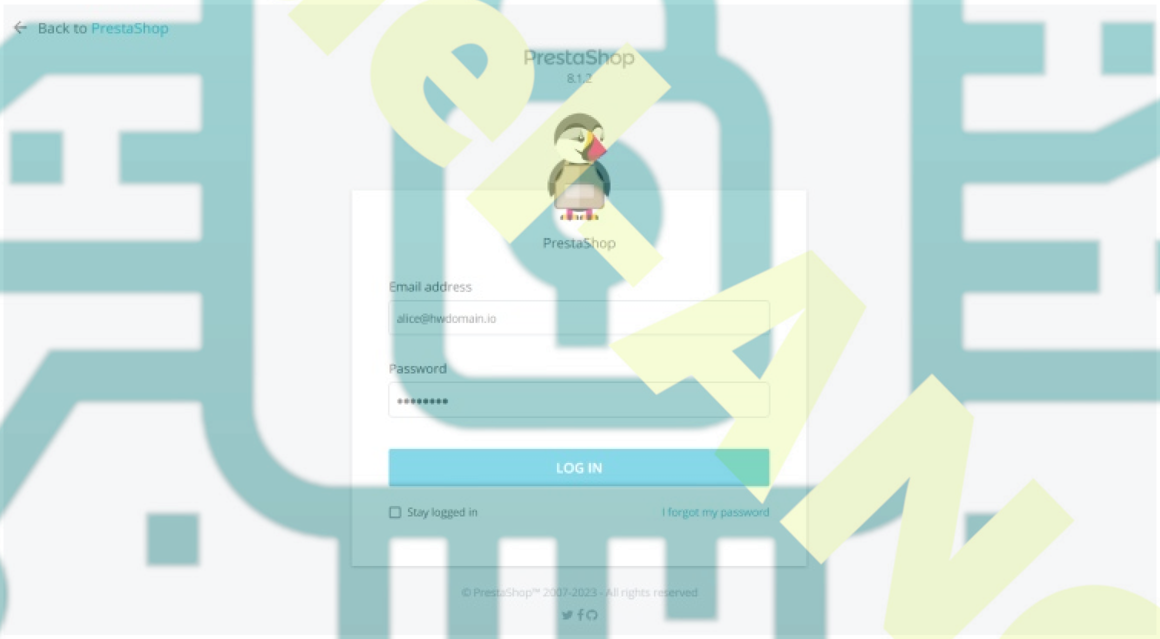
```
sudo rm -rf /var/www/prestashop/install
```

Then, open your web browser and visit your PrestaShop domain name, such as http://hwdomain.io/. Once the installation is successful, you should be redirected to a secure HTTPS connection, and you should see the default homepage of your PrestaShop installation.


Contact us
English
Sign in
Cart (0)
my store
CLOTHES
ACCESSORIES
ART
Search our catalog
SAMPLE 2
EXCEPTEUR OCCAECAT
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin tristique in tortor et dignissim. Quisque non tempor leo. Maecenas egestas sem elit
POPULAR PRODUCTS
-20%
NEW
-20%
NEW
NEW
NEW

Next, visit the PrestaShop domain name followed by the admin path, such as https://hwdomain.io/admin/. If everything goes well, you should be redirected to the PrestaShop login page. Also, you can see the default URL path for PrestaShop has changed to **admin051rgjztgtkvgcie6dp**.

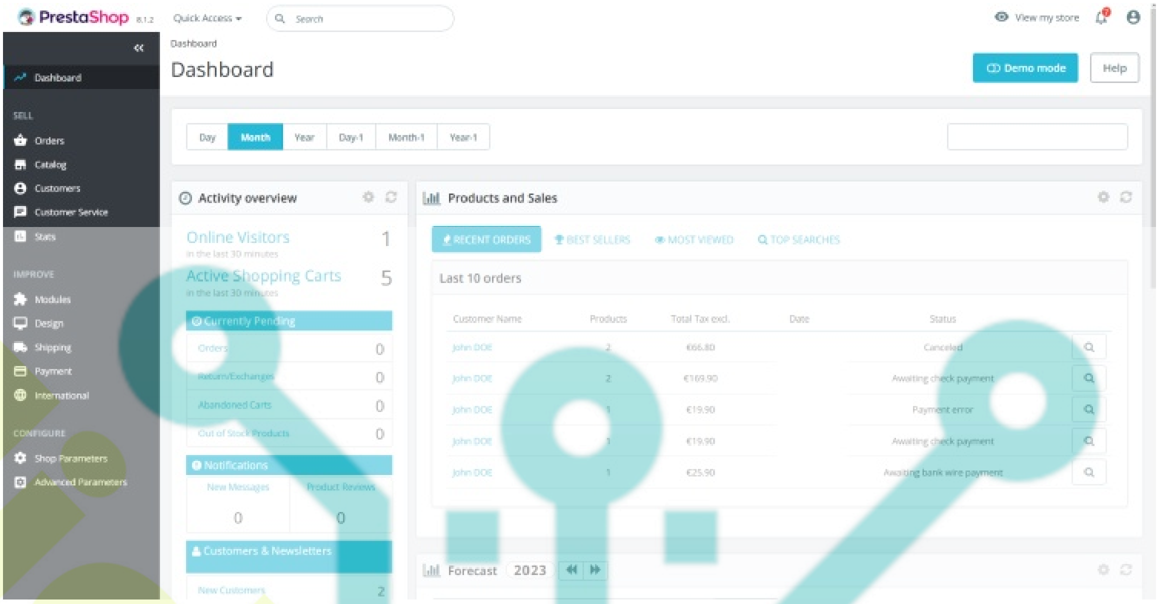
Input your admin user and password, then click **Login**.



Email address: alice@hwdomain.io

Password: ••••••••

If you have a proper admin user and password, you should be presented with the PrestaShop administration dashboard.



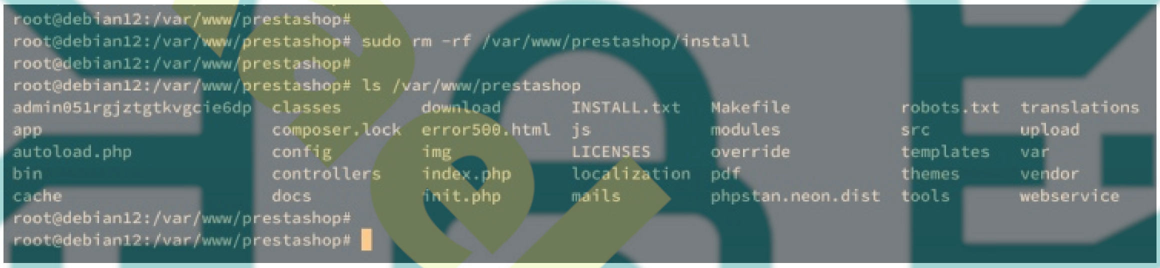
Last 10 orders

Customer Name	Products	Total Tax excl.	Date	Status
John DOE	2	€66.80		Canceled
John DOE	2	€169.90		Awaiting check payment
John DOE	1	€19.90		Payment error
John DOE	1	€19.90		Awaiting check payment
John DOE	1	€25.90		Awaiting bank wire payment

Another consideration, you can run the following command to check the PrestaShop login admin.

```
ls /var/www/prestashop
```

The output below shows that the PrestaShop admin page is available at **admin051rgjztgtkvgcie6dp**.



```
root@debian12:/var/www/prestashop#
root@debian12:/var/www/prestashop# sudo rm -rf /var/www/prestashop/install
root@debian12:/var/www/prestashop#
root@debian12:/var/www/prestashop# ls /var/www/prestashop
admin051rgjztgtkvgcie6dp  classes        download        INSTALL.txt   Makefile            robots.txt  translations
app                       composer.lock  error500.html   js            modules             src         upload
autoload.php              config         img             LICENSES      override            templates   var
bin                       controllers    index.php       localization  pdf                 themes      vendor
cache                     docs           init.php        mails         phpstan.neon.dist   tools       webservice
root@debian12:/var/www/prestashop#
root@debian12:/var/www/prestashop# 
```

Conclusion

To conclude, you've now successfully installed PrestaShop on Debian 12 step-by-step. You've installed PrestaShop 8 with LAMP Stack (Apache2, MariaDB, and PHP) and secured PrestaShop installation with SSL/TLS certificates. Furthermore, you've also learned how to start PrestaShop installation via the command line. Now you can add new themes and additional extensions to extend your PrestaShop installation.