

comment-installer-suitecrm-sur-debian-12

SuiteCRM is free and open-source CRM (Customer Relationship Management) software written in PHP. It helps you organize and manage your business's marketing, sales, and customer service departments. SuiteCRM is suitable for almost every industry, such as manufacturing, public sectors, technology, finance, education, etc.

In this guide, we'll walk you step-by-step through the installation of SuiteCRM on the Debian 12 server. You will install and run SuiteCRM with the LAMP Stack (Apache2, MariaDB, and PHP) on your Debian machine. You will also secure SuiteCRM with SSL/TLS certificates from Letsencrypt.

Prerequisites

The following requirements are necessary to continue with this guide:

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

Installing Dependencies

SuiteCRM is a CRM software written in PHP with MySQL/MariaDB as the database. In this guide, you will be running SuiteCRM with the LAMP Stack (Apache2, MariaDB, and PHP), and now you will install LAMP Stack packages with additional PHP extensions.

First, update your Debian package index via the apt update command below.

```
sudo apt update
```

```
root@debian12:~#  
root@debian12:~# sudo apt update  
Get:1 http://security.debian.org/debian-security bookworm-security InRelease [48.0 kB]  
Get:2 http://httpredir.debian.org/debian bookworm InRelease [151 kB]  
Get:3 http://security.debian.org/debian-security bookworm-security/non-free-firmware Sources [784 B]  
Get:4 http://security.debian.org/debian-security bookworm-security/main Sources [47.5 kB]  
Get:5 http://security.debian.org/debian-security bookworm-security/main amd64 Packages [63.2 kB]  
Get:6 http://security.debian.org/debian-security bookworm-security/main Translation-en [37.8 kB]  
Get:7 http://security.debian.org/debian-security bookworm-security/non-free-firmware amd64 Packages [680 B]  
Get:8 http://security.debian.org/debian-security bookworm-security/non-free-firmware Translation-en [464 B]  
Get:9 http://httpredir.debian.org/debian bookworm-updates InRelease [52.1 kB]  
Get:10 http://httpredir.debian.org/debian bookworm/main Sources [9,640 kB]  
Get:11 http://httpredir.debian.org/debian bookworm/non-free-firmware Sources [6,156 B]
```

Now install LAMP Stack (Apache2, MariaDB, and PHP) packages with some PHP extensions using the following command.

```
sudo apt install apache2 mariadb-server php php-cli php-mysql php-bcmath php-xml php-zip php-curl php-mbstring php-gd php-tidy php-intl php-cli php-opcache php-soap php-imap php-ldap unzip
```

Type y to proceed with the installation.

```
root@debian12:~#  
root@debian12:~# sudo apt install apache2 mariadb-server php php-cli php-mysql php-bcmath php-xml php-zip php-curl php-mbstring php-gd  
php-tidy php-intl php-cli php-opcache php-soap php-imap php-ldap  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
Note, selecting 'php8.2-opcache' instead of 'php-opcache'  
The following additional packages will be installed:  
apache2-bin apache2-data apache2-utils fontconfig-config fonts-dejavu-core galera-4 gawk libabsl20220623 libaon3  
libapache2-mod-php8.2 libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libavif15 libc-client2007e libcgi-fast-perl  
libcgi-pm-perl libclone-perl libconfig-inifiles-perl libdavid6 libdaxctl1 libdbd-mariadb-perl libdbi-perl libde265-0 libdeflate0  
libencode-locale-perl libfcgi-bin libfcgi-perl libfcgi1ldbl libfontconfig1 libgav1-1 libgd3 libheif1 libhtml-parser-perl  
libhtml-tagset-perl libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl libjpeg62-turbo liblerc4
```

Next, run the following systemctl command to verify the apache2 service. This will ensure that the apache2 service is enabled and running.

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

The output **enabled** confirms that apache2 is enabled, and the output **active (running)** confirms that apache2 is running.

```

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: 16217 (apache2)
    Tasks: 6 (limit: 4642)
   Memory: 21.7M
      CPU: 189ms
   CGroup: /system.slice/apache2.service

```

Verify the mariadb service using the following command.

```

sudo systemctl is-enabled mariadb
sudo systemctl status mariadb

```

Similar to the apache2 service output, you should see the mariadb service is enabled and running.

```

root@debian12:~#
root@debian12:~# sudo systemctl is-enabled mariadb
enabled
root@debian12:~# sudo systemctl status mariadb
● mariadb.service - MariaDB 10.11.3 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; preset: enabled)
   Active: active (running) since
     Docs: man:mariadb(8)
           https://mariadb.com/kb/en/library/systemd/
  Main PID: 14240 (mariabdd)
    Status: "Taking your SQL requests now..."
    Tasks: 15 (limit: 4642)
   Memory: 163.8M
      CPU: 819ms

```

Lastly, verify the PHP version and some enabled extensions by executing the following command.

```

php -v
php -m

```

The following output shows that PHP 8.2 is installed with some enabled modules such as *curl*, *exif*, *fileinfo*, and *gd*.

```

root@debian12:~#
root@debian12:~# php -v
PHP 8.2.7 (cli) (built: Jun 9 2023 19:37:27) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.2.7, Copyright (c) Zend Technologies
    with Zend OPcache v8.2.7, 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
hash
iconv

```

Configuring MariaDB Server

After installing dependencies, you will secure your MariaDB server installation and create a new database and user that will be used for SuiteCRM.

Execute the `mariadb-secure-installation` command below to secure your MariaDB server.

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

During the process, input Y to confirm the configuration or n for No and deny the changes.

- 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.

Now that MariaDB is secured, you will create a new database and user for SuiteCRM.

Log in to the MariaDB server via the `mariadb` client command below. Input your MariaDB root password when prompted.

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

Next, run the following queries to create a new database **suitecrmdb** and the user **suitecrm** with password **password**. Be sure to change the password with your own password.

```
CREATE DATABASE suitecrmdb;
CREATE USER 'suitecrm'@'localhost' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON suitecrmdb.* TO 'suitecrm'@'localhost';
FLUSH PRIVILEGES;
```

```
MariaDB [(none)]> CREATE DATABASE suitecrmdb;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> CREATE USER 'suitecrm'@'localhost' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]>
MariaDB [(none)]> GRANT ALL PRIVILEGES ON suitecrmdb.* TO 'suitecrm'@'localhost';
Query OK, 0 rows affected (0.001 sec)

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

Now run the following query to verify the user **suitecrm**.

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

In the following output, you should see the user **suitecrm** has access to the database **suitecrmdb**.

```
MariaDB [(none)]> SHOW GRANTS FOR 'suitecrm'@'localhost';
+-----+
| Grants for suitecrm@localhost |
+-----+
| GRANT USAGE ON *.* TO `suitecrm`@`localhost` IDENTIFIED BY PASSWORD '*2470C0C06DEE42FD1618BB998' |
| GRANT ALL PRIVILEGES ON `suitecrmdb`.* TO `suitecrm`@`localhost` |
+-----+
2 rows in set (0.000 sec)

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

Type quit to exit from the MariaDB server.

Configuring PHP

Now that the MariaDB server is configured, the next step is to configure your PHP installation. The SuiteCRM required some changes on PHP configurations, such as custom error_reporting, enabling the OPcache module, and also custom upload_max_filesize and post_max_size.

Open the default PHP configuration `/etc/php/8.2/apache2/php.ini` and `/etc/php/8.2/cli/php.ini` using the following nano editor command.

```
sudo nano /etc/php/8.2/apache2/php.ini
sudo nano /etc/php/8.2/cli/php.ini
```

Change the default parameters of some configurations using the following lines. Be sure to adjust the **date.timezone** with your server timezone, and the **memory_limit** with the server memory.

```
date.timezone = Europe/Amsterdam
```

```
max_execution_time = 60
memory_limit = 256M
```

```
upload_max_filesize = 25M
post_max_size = 25M
```

Change the default **error_reporting** parameter like this:

```
error_reporting = E_ALL & ~E_DEPRECATED & ~E_STRICT & ~E_NOTICE & ~E_WARNING
```

Uncomment the **session.save_path** parameter to configure the default session path to `/var/lib/php/sessions`.

```
session.save_path = "/var/lib/php/sessions"
```

Now enable the OPcache extension and adjust some settings like the following:

```
opcache.enable=1
```

```
; The OPcache shared memory storage size.
opcache.memory_consumption=256
```

```
; The maximum number of keys (scripts) in the OPcache hash table.
; Only numbers between 200 and 100000 are allowed.
opcache.max_accelerated_files=20000
```

```
; When disabled, you must reset the OPcache manually or restart the
; webserver for changes to the filesystem to take effect.
opcache.validate_timestamps=0
```

Save and close the file when you're done.

Now run the following systemctl command to restart the apache2 service and apply the changes that you've made.

```
sudo systemctl restart apache2
```

Downloading and Installing SuiteCRM

At this point, the LAMP Stack is now configured for SuiteCRM. Now you will download the SuiteCRM source code and install it via the command line.

Create a new directory `/var/www/suitecrm` that will be used as the target installation for SuiteCRM.

```
mkdir -p /var/www/suitecrm
```

Move to `/var/www/suitecrm` directory and download the SuiteCRM source code using the wget command below. Be sure to check the [SuiteCRM Download page](#) to get the latest version of the source code. At this time, SuiteCRM **8.4** is the latest version.

```
cd /var/www/suitecrm
wget https://suitecrm.com/download/142/suite84/562972/suitecrm-8-4-0.zip
```

Once downloaded, extract the `suitecrm-8-4-0.zip` file using the unzip command below.

```
unzip suitecrm-8-4-0.zip
```

Then run the following command to set up proper permission and ownership of the SuiteCRM source code.

```
find . -type d -not -perm 2755 -exec chmod 2755 {} \;
find . -type f -not -perm 0644 -exec chmod 0644 {} \;
find . ! -user www-data -exec chown www-data:www-data {} \;
```

```
chmod +x bin/console
```

Lastly, execute the binary file `/var/www/suitecrm/bin/console` to install SuiteCRM via the command line. Be sure to change the details of the database name, user, host, target domain name, admin user, and password that will be used for your SuiteCRM installation.

```
sudo -u www-data ./bin/console suitecrm:app:install -u "alice" -p "password" -U "suitecrm" -P "password" -H "127.0.0.1" -N "suitecrmdb" -S "http://suitecrm.hwdomain.io/"
```

Once the installation is finished, the following output will be shown to your terminal:



```
root@debian12:/var/www/suitecrm#
root@debian12:/var/www/suitecrm# sudo -u www-data ./bin/console suitecrm:app:install -u "alice" -p "password" -U "suitecrm" -P "password" -H "127.0.0.1" -N "suitecrmdb" -S "http://suitecrm.hwdomain.io/"

SuiteCRM Silent Install
=====

Running: check-install-lock
step: check-install-lock | status: done
Installer not locked. Proceeding with install
Running: check-db-connection
step: check-db-connection | status: done
DB credentials ok
Running: install-system-checks
step: install-system-checks | status: done
Running: create-config
step: create-config | status: done
Created silent install config: config_si.php
Running: create-env
step: create-env | status: done
Created .env.local
Running: run-legacy-install
step: run-legacy-install | status: done
Legacy install successful

=====
root@debian12:/var/www/suitecrm#
```

Configuring Apache2 Virtual Host for Suite

In the following step, you will create a new Apache2 virtual host configuration that will be used to run SuiteCRM. Before proceeding, ensure that you have a domain name pointed to your server IP address.

Run the `a2enmod` command below to enable Apache2 modules that are required by SuiteCRM.

```
sudo a2enmod rewrite ssl header
```

Then, create a new Apache2 virtual host configuration `/etc/apache2/sites-available/suitecrm.conf` using the following nano editor command.

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

Insert the following configuration and be sure to change the **ServerName** option with your domain name.

```
<VirtualHost *:80>

DocumentRoot /var/www/suitecrm/public
ServerName suitecrm.hwdomain.io

<Directory /var/www/suitecrm/public>
    Options FollowSymLinks
    AllowOverride All
</Directory>

ErrorLog /var/log/apache2/suitecrm-error.log
CustomLog /var/log/apache2/suitecrm-access.log common

</VirtualHost>
```

Save and close the file when you're done.

Now run the following command to activate the virtual host file `suitecrm.conf` and verify your Apache2 configurations.

```
sudo a2ensite suitecrm.conf
sudo apache2ctl configtest
```

If you've proper syntax, you should get the output **Syntax OK**.

```

root@debian12:~#
root@debian12:~# sudo nano /etc/apache2/sites-available/suitecrm.conf
root@debian12:~#
root@debian12:~# sudo a2ensite suitecrm.conf
Enabling site suitecrm.
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 domain name; please
  manually determine the machine hostname by running hostname, and see the hostname(8) manpage
  for instructions on the proper format to use or edit in /etc/hostname.
Syntax OK
root@debian12:~# sudo systemctl restart apache2
root@debian12:~#

```

Next, run the `systemctl` command below to restart the `apache2` service and apply the changes.

```
sudo systemctl restart apache2
```

Lastly, open your web browser and visit the domain name of your SuiteCRM installation, such as <http://suitecrm.hwdomain.io/>. If everything goes well, you should see the login page of SuiteCRM.

Securing SuiteCRM with SSL/TLS from Letsencrypt

With everything configured, you will now secure SuiteCRM with SSL/TLS certificates from Letsencrypt.

Install Certbot and Certbot Apache plugin using the following `apt` install command. Type `y` to proceed with the installation.

```
sudo apt install certbot python3-certbot-apache
```

Now execute the `certbot` command below to generate SSL/TLS certificates. Be sure to change the domain name and email address within the following command.

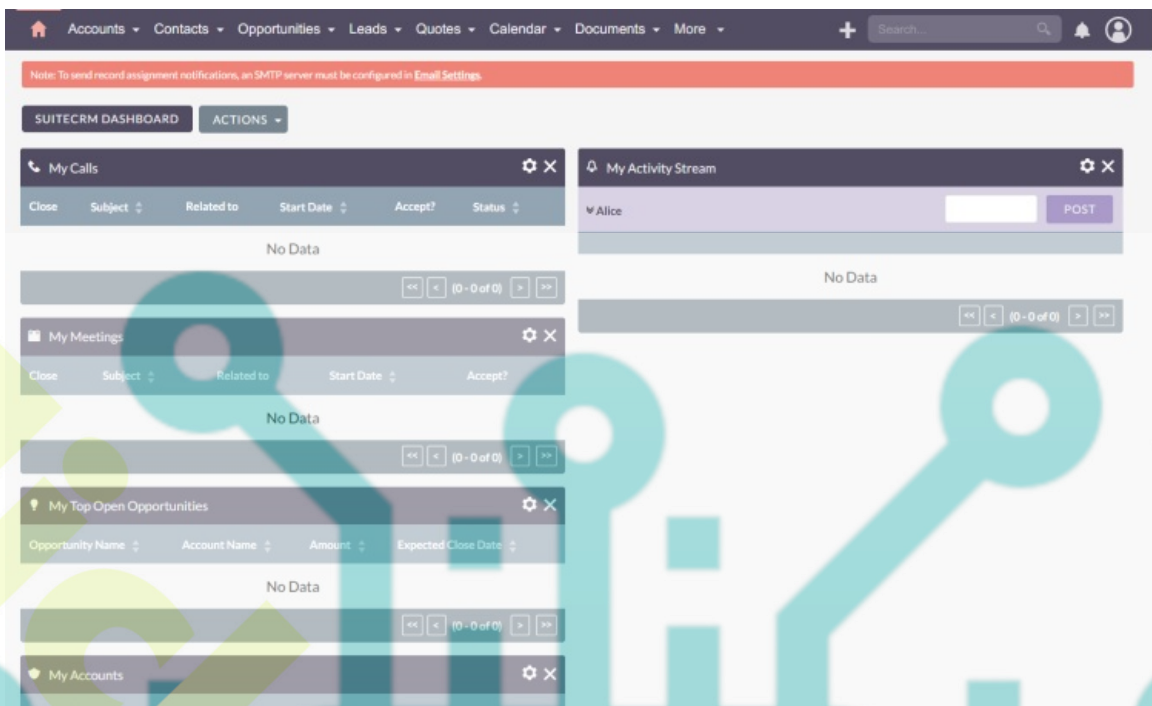
```
sudo certbot --apache --agree-tos --no-eff-email --redirect --hsts --staple-ocsp --email admin@hwdomain.io -d suitecrm.hwdomain.io
```

Once the process is finished, your SSL certificates will be generated at `/etc/letsencrypt/live/suitecrm.hwdomain.io` directory. Also, the SuiteCRM virtual host file `suitecrm.conf` will automatically be configured with HTTPS.

Back to your web browser and visit your SuiteCRM domain name, you should be redirected to the SuiteCRM login page for secure HTTPS connections. Input your admin user and password, then click **Login**.



If everything goes well, you should see the SuiteCRM administration dashboard like the following:



Conclusion

To wrap up, you have now successfully installed SuiteCRM on Debian 12 server with LAMP Stack (Apache2, MariaDB, and PHP). You've also secured the SuiteCRM with SSL/TLS Letsencrypt, which is generated via Certbot. Now you can use SuiteCRM as the main CRM (Customer Relationship Management) application for your business, and you can start by adding the SMTP server to your SuiteCRM installation.
