

comment installer typo3 cms sur debian 12

TYPO3 is a free and open-source Enterprise-grade content management system. It provides enterprise-level features such as a scalable CMS with multisite support, multilingual installations, strong security implementation, blazingly fast, and can be run anywhere.

Using TYPO3 CMS allows you to build flexible and reliable websites. The TYPO3 CMS is backed a vibrant professional community. By design, the TYPO3 CMS is a pluggable content management system with adaptable and decoupled architecture.

This tutorial shows you how to install the TYPO3 CMS on a Debian 12 server on a LAMP Stack (Apache2, MariaDB, and PHP).

Prerequisites

Before you proceed with this guide, ensure you have:

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

Installing Dependencies

TYPO3 is an open-source content management system written in PHP and supports the MySQL/MariaDB database server. The TYPO3 CMS can be installed via Composer PHP Dependency Manager and can be run with Apache2 or Nginx web server.

In this guide, you will run TYPO3 CMS with the LAMP Stack (Apache2, MariaDB, and PHP) and Composer, which can be installed easily via APT from the official Debian repository. Complete the following steps to install those dependencies for TYPO3 CMS.

First, update and refresh your Debian repository using the command below.

```
sudo apt update
```

```
root@debian12:~#  
root@debian12:~# sudo apt update  
Hit:1 http://httpredir.debian.org/debian bookworm InRelease  
Hit:2 http://security.debian.org/debian-security bookworm-security InRelease  
Hit:3 http://httpredir.debian.org/debian bookworm-updates InRelease  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done
```

Now install package dependencies for TYPO3 CMS using the *apt install* command below. With this command, you will install the LAMP Stack (Apache2, MariaDB, PHP) with additional PHP extensions and the Composer as a PHP dependency manager.

```
sudo apt install apache2 mariadb-server composer php php-common php-mysql libapache2-mod-php php-gd php-curl php-json php-xmlrpc  
php-intl php-bcmath php-zip php-apcu php-mbstring php-fileinfo php-xml php-soap
```

Type yo to confirm and proceed with the installation.

```
root@debian12:~#  
root@debian12:~# sudo apt install apache2 mariadb-server composer php php-common php-mysql libapache2-mod-php php-gd php-curl php-json php-xmlrpc php-intl p  
hp-bcmath php-zip php-apcu php-mbstring php-fileinfo php-xml php-soap -y  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
Note, selecting 'php8.2-common' instead of 'php-fileinfo'  
The following additional packages will be installed:  
apache2-bin apache2-data apache2-utils fontconfig-config fonts-dejavu-core galera-4 gawk git git-man jsonlint libabsl20220623 libaom3  
libapache2-mod-php8.2 libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libavif15 libcgi-fast-perl libcgi-pm-perl libclone-perl  
libconfig-inifiles-perl libdavid1 libdaxctl1 libdbd-mariadb-perl libdbi-perl libde265-0 libdeflate0 libencode-locale-perl liberror-perl libfcgi-bin
```

After dependencies are installed, you must verify each dependency to ensure that the installation is successful.

Execute the following command to ensure the apache2 service is running and enabled.

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

The following output indicates 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: 15951 (apache2)
    Tasks: 6 (limit: 4642)
   Memory: 21.1M
      CPU: 152ms
   CGroup: /system.slice/apache2.service

```

Now verify the mariadb service using the command below.

```

sudo systemctl is-enabled mariadb
sudo systemctl status mariadb

```

Similar to the apache2 service, the output **enabled** confirms that the mariadb is enabled. And the output **active (running)** confirms that the mariadb is 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:mariabdb(8)
           https://mariadb.com/kb/en/library/systemd/
  Main PID: 13981 (mariabdb)
    Status: "Taking your SQL requests now..."
    Tasks: 11 (limit: 4642)
   Memory: 184.6M
      CPU: 663ms
   CGroup: /system.slice/mariadb.service
           └─13981 /usr/sbin/mariabdb

```

Next, verify your PHP version and extensions using the following command.

```

php -v
php -m

```

You should see that PHP 8.2 is installed with some extensions enabled, such as *apcu*, *curl*, *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]
apcu
bcmath
calendar
Core
ctype
curl
date
dom
exif
FFI
fileinfo
filter

```

Lastly, execute the following command to ensure that the COMposer is installed. Then, verify the Composer version.

```

which composer
sudo -u www-data composer -v

```


PHP Version 8.2.7	
System	Linux debian12 6.1.0-9-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.27-1 (2023-05-08) x86_64
Build Date	Jun 9 2023 19:37:27
Build System	Linux
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.2/apache2
Loaded Configuration File	/etc/php/8.2/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/8.2/apache2/conf.d
Additional .ini files parsed	/etc/php/8.2/apache2/conf.d/10-mysqld.ini, /etc/php/8.2/apache2/conf.d/10-openssl.ini, /etc/php/8.2/apache2/conf.d/10-pdo.ini, /etc/php/8.2/apache2/conf.d/15-xml.ini, /etc/php/8.2/apache2/conf.d/20-apcu.ini, /etc/php/8.2/apache2/conf.d/20-bcmath.ini, /etc/php/8.2/apache2/conf.d/20-calendar.ini, /etc/php/8.2/apache2/conf.d/20-ctype.ini, /etc/php/8.2/apache2/conf.d/20-curl.ini, /etc/php/8.2/apache2/conf.d/20-dom.ini, /etc/php/8.2/apache2/conf.d/20-exif.ini, /etc/php/8.2/apache2/conf.d/20-fdi.ini, /etc/php/8.2/apache2/conf.d/20-fileinfo.ini, /etc/php/8.2/apache2/conf.d/20-ftp.ini, /etc/php/8.2/apache2/conf.d/20-gd.ini, /etc/php/8.2/apache2/conf.d/20-gettext.ini, /etc/php/8.2/apache2/conf.d/20-iconv.ini, /etc/php/8.2/apache2/conf.d/20-intl.ini, /etc/php/8.2/apache2/conf.d/20-mbstring.ini, /etc/php/8.2/apache2/conf.d/20-mysqli.ini, /etc/php/8.2/apache2/conf.d/20-pdo_mysql.ini, /etc/php/8.2/apache2/conf.d/20-phar.ini, /etc/php/8.2/apache2/conf.d/20-posix.ini, /etc/php/8.2/apache2/conf.d/20-readline.ini, /etc/php/8.2/apache2/conf.d/20-shmop.ini, /etc/php/8.2/apache2/conf.d/20-simplexml.ini, /etc/php/8.2/apache2/conf.d/20-soap.ini, /etc/php/8.2/apache2/conf.d/20-sockets.ini, /etc/php/8.2/apache2/conf.d/20-sysvmsg.ini, /etc/php/8.2/apache2/conf.d/20-sysvsem.ini, /etc/php/8.2/apache2/conf.d/20-sysvshm.ini, /etc/php/8.2/apache2/conf.d/20-tokenizer.ini, /etc/php/8.2/apache2/conf.d/20-xmlreader.ini, /etc/php/8.2/apache2/conf.d/20-xmlrpc.ini, /etc/php/8.2/apache2/conf.d/20-xmlwriter.ini, /etc/php/8.2/apache2/conf.d/20-xsl.ini, /etc/php/8.2/apache2/conf.d/20-zip.ini
PHP API	20220829
PHP Extension	20220829
Zend Extension	420220829
Zend Extension Build	API420220829.NTS
PHP Extension Build	API20220829.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
Zend Max Execution Timers	disabled
IPv6 Support	enabled
DTrace Support	available, disabled

Configuring MariaDB Server

In the following section, you will secure the MariaDB server via the *mariadb-secure-installation* utility and create a new database and user that will be used by TYPO3 CMS via the *mariadb* client.

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

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

During the process, you will be asked to change some of the default MariaDB configurations. Input **Y** to apply the changes, or **n** for No to reject it.

Below are some of the MariaDB server configurations 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.

After securing the MariaDB server, you will create a new database and user that will be used by TYPO3 CMS. To create those, you must log in to the MariaDB server via the *mariadb* client command.

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

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

Once logged in, run the following queries to create a new database **typo3db** and user **typo3**. Be sure to change the password in the following queries.

```
CREATE DATABASE typo3db;
GRANT ALL PRIVILEGES ON typo3db.* to typo3@localhost IDENTIFIED BY 'typo3password';
FLUSH PRIVILEGES;
```

```
MariaDB [(none)]> CREATE DATABASE typo3db;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON typo3db.* to typo3@localhost IDENTIFIED BY 'typo3password';
Query OK, 0 rows affected (0.001 sec)

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

Next, run the following query to ensure that the user **typo3** can access the database **typo3db**.

```
SHOW GRANTS FOR typo3@localhost;
```

An output below confirms that the MariaDB user **typo3** can access the database **typo3db**.

```
MariaDB [(none)]> SHOW GRANTS FOR typo3@localhost;
+-----+-----+-----+-----+-----+-----+
| Grants for typo3@localhost                                     |
+-----+-----+-----+-----+-----+-----+-----+
| GRANT USAGE ON *.* TO `typo3`@`localhost` IDENTIFIED BY PASSWORD '*74FD13F3D41E97F2CE56EB7E55C4C5A8E9984A94'|
| GRANT ALL PRIVILEGES ON `typo3db`.* TO `typo3`@`localhost`  |
+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.000 sec)

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

Type quit to exit from the MariaDB server.

Downloading TYPO3 CMS via Composer

In this guide, you will install TYPO3 CMS version 12, which can be downloaded via Composer. In the following steps, you will set up the web root directory and download the TYPO3 CMS source, You will also configure proper permission and ownership for the TYPO3 CMS web root directory.

Create a new web root directory `/var/www/typo3` and additional directories `/var/www/{.cache,.config}` using the following command. The directory `/var/www/typo3` will be used as the web root directory for your TYPO3 CMS installation.

```
mkdir -p /var/www/{.cache,.config,typo3}
```

Now run the command below to change the ownership of the `/var/www/typo3` directory to the user `www-data`. Then, enable read and write access for the owner of the `/var/www/typo3` directory.

```
sudo chown -R www-data:www-data /var/www/{.cache,.config,typo3}
sudo chmod u+rw /var/www/typo3
```

Next, go to the `/var/www/typo3` directory and download the TYPO3 CMS source via the `composer` command below. In this example, you will be downloading the TYPO3 CMS 12.

```
cd /var/www/typo3
sudo -u www-data composer create-project typo3/cms-base-distribution:^12 .
```

The following output will be shown during the process:

```
root@debian12:~#
root@debian12:~# mkdir -p /var/www/{.cache,.config,typo3}
root@debian12:~#
root@debian12:~# sudo chown -R www-data:www-data /var/www/{.cache,.config,typo3}
root@debian12:~# sudo chmod u+rw /var/www/typo3
root@debian12:~#
root@debian12:~# cd /var/www/typo3
root@debian12:/var/www/typo3# sudo -u www-data composer create-project typo3/cms-base-distribution:^12 .
Creating a "typo3/cms-base-distribution:^12" project at "."
Info from https://repo.packagist.org: #StandWithUkraine
Installing typo3/cms-base-distribution (v12.4.0)
 - Downloading typo3/cms-base-distribution (v12.4.0)
 - Installing typo3/cms-base-distribution (v12.4.0): Extracting archive
Created project in /var/www/typo3/.
Loading composer repositories with package information
Updating dependencies
Lock file operations: 102 installs, 0 updates, 0 removals
 - Locking bacon/bacon-qr-code (2.0.8)
 - Locking composer/composer (2.6.0)
```

When finished, below is the output you should get:

```

- Installing typo3/cms-info (v12.4.5): Extracting archive
- Installing nikic/php-parser (v4.17.1): Extracting archive
- Installing typo3/cms-install (v12.4.5): Extracting archive
- Installing typo3/cms-reactions (v12.4.5): Extracting archive
- Installing typo3/cms-rte-ckeditor (v12.4.5): Extracting archive
- Installing typo3/cms-seo (v12.4.5): Extracting archive
- Installing typo3/cms-setup (v12.4.5): Extracting archive
- Installing typo3/cms-sys-note (v12.4.5): Extracting archive
- Installing typo3/cms-t3editor (v12.4.5): Extracting archive
- Installing typo3/cms-t3template (v12.4.5): Extracting archive
- Installing typo3/cms-viewpage (v12.4.5): Extracting archive
- Installing typo3/cms-webhooks (v12.4.5): Extracting archive
8 package suggestions were added by new dependencies, use 'composer suggest' to see details.
Generating autoload files
Generating class alias map file
45 packages you are using are looking for funding.
Use the 'composer fund' command to find out more!
No security vulnerability advisories found
root@debian12:/var/www/typo3#

```

After the TYPO3 CMS source code is downloaded, execute the `ls` command below to list the source code.

```
ls
```

You should see the list of TYPO3 CMS source code like the following:

```

root@debian12:/var/www/typo3#
root@debian12:/var/www/typo3# ls
composer.json  composer.lock  LICENSE  public  README.md  vendor
root@debian12:/var/www/typo3#
root@debian12:/var/www/typo3#

```

Setting Up Apache2 Virtual Host

In the following section, you will create a new Apache2 virtual host configuration that will be used to run the TYPO3 CMS. So before going further, ensure that you have a domain name pointed to your server IP address.

First, enter the following command to enable modules `rewrite` and `headers`.

```
sudo a2enmod rewrite headers
```

Now create a new virtual host configuration `/etc/apache2/sites-available/typo3.conf` using the following `nano` editor command.

```
sudo nano /etc/apache2/sites-available/typo3.conf
```

Insert the following configuration and be sure to change the domain name of your TYPO3 CMS installation.

```

<VirtualHost *:80>
    ServerAdmin admin@hwdomain.io
    DocumentRoot /var/www/typo3/public
    ServerName hwdomain.io

    <Directory /var/www/typo3/public/>
        Options FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    <Directory /var/www/typo3/public/>
        RewriteEngine on
        RewriteBase /
        RewriteCond %{REQUEST_FILENAME} !-f
        RewriteRule ^(.*) index.php [PT,L]
    </Directory>
</VirtualHost>

```

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

Next, run the command below to activate the new virtual host file `typo3.conf` and verify your Apache2 syntax.

```
sudo a2ensite typo3.conf
sudo apachectl configtest
```

When you've proper Apache2 syntax, you should get an output **Syntax OK**.

```
root@debian12:~#
root@debian12:~# sudo a2enmod rewrite headers
Enabling module rewrite.
Enabling module headers.
To activate the new configuration, you need to run:
  systemctl restart apache2
root@debian12:~#
root@debian12:~# sudo nano /etc/apache2/sites-available/typo3.conf
root@debian12:~#
root@debian12:~# sudo a2ensite typo3.conf
Enabling site typo3.
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
s this message
Syntax OK
root@debian12:~#
```

Lastly, enter the command below to restart the apache2 service and apply the changes that you've made.

```
sudo systemctl restart apache2
```

Securing TYPO3 CMS with Certbot

After configuring the virtual host, you will secure your TYPO3 CMS installation by enabling secure HTTPS with SSL/TLS certificates from Letsencrypt. You will be using Certbot and Certbot Apache plugin for generating Letsencrypt certificates.

Install Certbot and Certbot Apache plugin using the following apt command.

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

Type y to proceed with the installation.

```
root@debian12:~#
root@debian12:~# sudo apt install certbot python3-certbot-apache
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  augeas-lenses libaugeas0 python3-acme python3-augeas python3-certbot py
  python3-distro python3-icu python3-josepy python3-openssl python3-parsed
Suggested packages:
  augeas-doc python-certbot-doc python-certbot-nginx augeas-tools python-
  python3-cryptography-vectors python-openssl-doc python3-openssl-dbg
The following NEW packages will be installed:
  augeas-lenses certbot libaugeas0 python3-acme python3-augeas python3-cer
  python3-configobj python3-cryptography python3-distro python3-icu python
0 upgraded, 18 newly installed, 0 to remove and 34 not upgraded.
Need to get 2,675 kB of archives.
After this operation, 11.3 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

Once installation is finished, run the certbot command below to generate new SSL/TLS certificates for your TYPO3 CMS installation. Be sure to change the domain name and the email address with your information.

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

Your SSL/TLS certificates will be available at `/etc/letsencrypt/live/domain.com` directory. And the virtual host file `typo3.conf` will automatically configured with HTTPS, which is done via the Certbot Apache plugin.

Installing TYPO3 CMS via Web Installer

First, run the following command to create a new file `/var/www/typo3/public/FIRST_INSTALL`. The TYPO3 web installer will not be available unless the file `/var/www/typo3/public/FIRST_INSTALL` is available.

```
sudo -u www-data touch /var/www/typo3/public/FIRST_INSTALL
```

Now open your web browser and visit your TYPO3 CMS domain name, such as <https://hwdomain.io/>. You will be redirected to the TYPO3 web installer.

Ensure your system environment is ready, then click **No problems detected, and continue with the installation**.



Input details of MariaDB database user, password, and host, then click **Continue**.



Installing TYPO3 CMS

Step 2 of 5 complete.

40%

Select database

You will need to create a database user with the appropriate privileges to access your database.

Connection [MySQL] Manually configured MySQL TCP/IP connection	
Username typo3	Password ●●●●●●●●
Host 127.0.0.1	Port 3306
Continue	

Select the option **Use existing empty database** and select the database **typo3db**. Then, click **Continue** again.



Installing TYPO3 CMS

Step 3 of 5 complete.



Select a database

Use an existing empty database:
Select a database.

Create a new database:
Specify a name for your TYPO3 database.

Continue

Next, input the new admin user, password, and email address for your TYPO3 CMS installation. Click **Continue** to confirm.



Installing TYPO3 CMS

Step 4 of 5 complete.

80%

Create Administrative User & Specify Site Name

Use this account to log into the backend of your site.

The password you provide for this account is also used to access the **Install Tool**.

On this page you can also set a name for your new website.

Username

alice

Password

••••••••



Email address

alice@hwdomain.io

Site name

New TYPO3 site

Continue

Once the installation is finished, you should get the following page.



Installing TYPO3 CMS

Step 5 of 5 complete.

100%

Installation Complete

The Installation Wizard can create an empty page at the root of your website to help get you started. If you would like to use this option select "Create empty starting page".

- Create empty starting page
- Take me straight to the backend

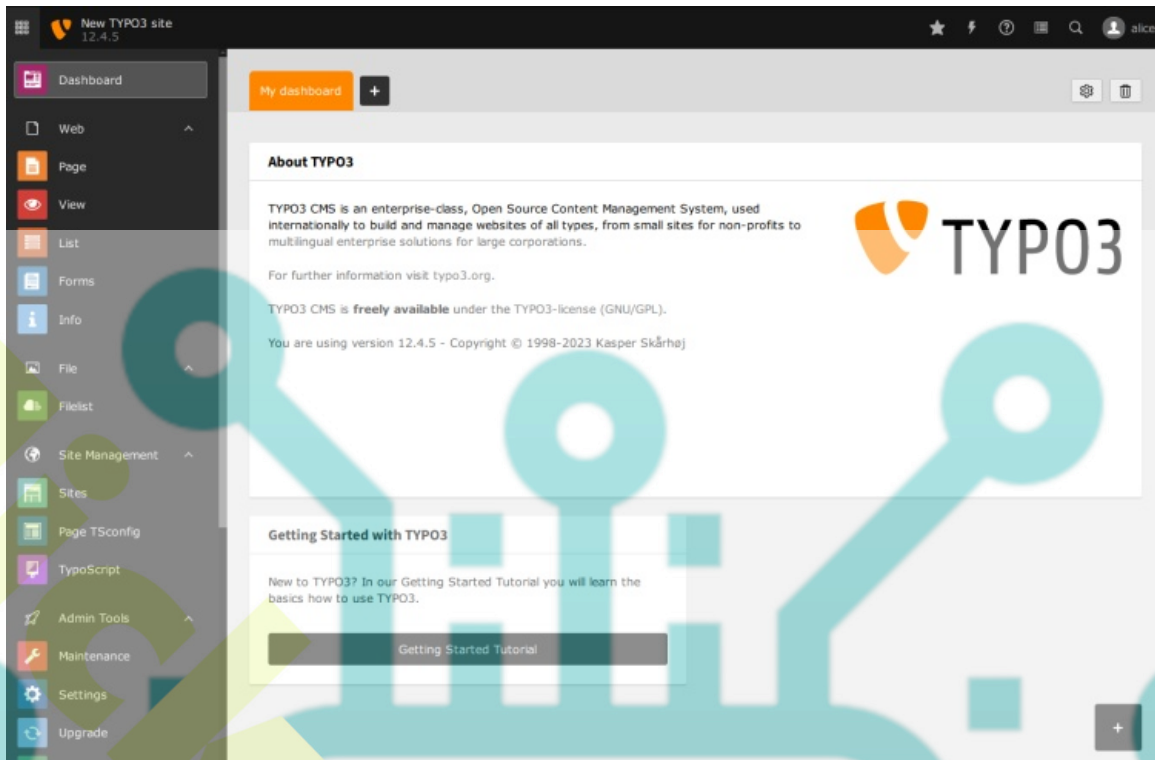
Open the TYPO3 Backend

Click the **Open the TYPO3 Backend** to access the TYPO3 administration dashboard.

You should get the TYPO3 login page. Input your username and password, then click **Login**.

The screenshot shows the TYPO3 login page. At the top is the TYPO3 logo. Below it is a username input field containing 'alice'. Underneath is a password input field with a toggle for visibility. A prominent orange 'Login' button is centered below the password field. Below the button is a link for 'Forgot your password?'. At the bottom, there is a link for 'More about TYPO3' and the TYPO3 logo again.

When login is successful, you should see the TYPO3 administration dashboard like the following:



On the top right menu, click the information menu and you should get detailed information about your TYPO3 CMS installation. In the following screenshot, you should see that TYPO3 CMS **12** is installed with PHP **8.2**, the MariaDB server **10.11**, and the Apache2 web server **2.4**.

A screenshot of a 'System Information' modal window. It provides a short system overview and lists the following details:

TYPO3 Version	12.4.5
Webserver	Apache/2.4.57 (Debian)
PHP Version	8.2.7
Database (Default)	MySQL 10.11.3-MariaDB-1
Application Context	Production
Composer mode	Enabled
Operating System	Linux 6.1.0-9-amd64

At the bottom, there is a green checkmark icon and the text: 'Your system is fully operational. Have a nice day.'

Conclusion

In conclusion, you have successfully installed TYPO3 CMS on the Debian 12 server with the LAMP Stack (Apache2, MariaDB, and PHP) and Composer PHP package manager. You've also secured your TYPO3 CMS installation with SSL/TLS certificates generated from Letsencrypt using Certbot and Certbot Apache plugin. You can now use TYPO3 CMS to manage your content, record management, set up additional language for multilingual sites, and many more.