

# Getting Started - Ubuntu

- [Requirements](#)
- [Update the server and install Apache](#)
- [Install PHP](#)
- [Install Database](#)
- [Virtual Host](#)
- [Let's Encrypt SSL](#)
- [loncube loaders](#)
- [Add new user for SSH](#)
- [Install Composer](#)
- [Install Ticaga](#)
- [Create an Agent](#)

# Requirements

Ticaga requires the latest version of PHP, this is because we pride ourselves on being up-to date with security updates, functions and cool new features.

Minimum requirements:

- PHP: **8.4**
- MYSQL: **8.0**
- Ioncube loaders: **13.3.1**
- Composer
- Laravel 11.x
- Lets Encrypt for SSL

Environment:

We recommend Ubuntu 22.04 on a Virtual Private server with at least 2GB RAM.

# Update the server and install Apache

So let's update the server and ensure everything is ready for us to continue.

```
sudo apt update -y && sudo apt upgrade -y
```

Now let's install Apache:

```
sudo apt install apache2 -y
```

Let's start and enable Apache on boot:

```
sudo systemctl start apache2 && sudo systemctl enable apache2
```

# Install PHP

Let's install PHP:

Firstly we need to get the Ondrej repo ready for 8.4:

```
sudo add-apt-repository ppa:ondrej/php
sudo add-apt-repository ppa:ondrej/apache2
sudo sed -i 's/oracular/noble/g' /etc/apt/sources.list.d/ondrej-ubuntu-php-oracular.sources
sudo sed -i 's/oracular/noble/g' /etc/apt/sources.list.d/ondrej-ubuntu-apache2-oracular.sources
sudo apt update
```

Now we can install the PHP and functions we require:

```
sudo apt install php8.4 php8.4-cli -y
sudo apt install php8.4-
{bz2,curl,mbstring,intl,fpm,gd,mysqlnd,opcache,zip,intl,common,bcmath,readline,apcu,xml,dom,mcrypt} -y
sudo apt install php-cli php-mailparse -y
```

# Install Database

To use Ticaga we need a database, we recommend MYSQL, to install MySQL just run the following:

```
sudo apt install mysql-server -y
```

Now we need to start and enable the MySQL service just like Apache:

```
sudo systemctl start mysql && sudo systemctl enable mysql
```

Now you can run the MySQL installation set-up:

```
sudo mysql_secure_installation
```

This loads the installation for MySQL, [find out more and the questions here](#).

Time to create the database and user for Ticaga:

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

Enter your root password, or the new password you created in the last step.

```
CREATE DATABASE ticaga;  
CREATE USER 'ticaga_user'@'localhost' IDENTIFIED BY 'p@s3w0r$2024!';  
GRANT ALL PRIVILEGES ON ticaga.* TO 'ticaga_user'@'localhost';  
FLUSH PRIVILEGES;  
exit;
```

**ticaga** = MySQL Database Name

**ticaga\_user** = MySQL Username

**p@s3w0r\$2024!** = MySQL Password (Please **don't use** this)

**ticaga.\*** = Database name and wildcard. (You **need** the **.\***)

**ticaga\_user** = MySQL Username

Congratulations you've now got the database ready.

# Virtual Host

Now we need to create the virtual host file to show Ticaga to the world.

Let's create the ticaga configuration file, in our example we're installing on <https://ubuntu.ticaga.com> for testing:

```
sudo nano /etc/apache2/sites-available/ubuntu.ticaga.com.conf
```

Paste and edit the following for your needs:

```
<VirtualHost *:80>
    ServerName ubuntu.ticaga.com
    DocumentRoot /var/www/ticaga/public
    <Directory /var/www/ticaga/public>
        Options FollowSymlinks
        AllowOverride All
        Require all granted
        <Files .env>
            Order allow,deny
            Deny from all
        </Files>
    </Directory>
    ErrorLog ${APACHE_LOG_DIR}/ubuntu.ticaga.com_error.log
    CustomLog ${APACHE_LOG_DIR}/ubuntu.ticaga.com.log combined
</VirtualHost>
```

You **need** to keep the **/public** folder because if you don't you'll **expose** all the Laravel files in the Ticaga installation.

Now we need to activate our new config:

```
sudo a2ensite ubuntu.ticaga.com.conf
```

And disable the default one:

```
sudo a2dissite 000-default.conf
```

Now restart Apache:

```
systemctl reload apache2
```

# Let's Encrypt SSL

Now we need to install Let's Encrypt, this is a way to get a free SSL Certificate for our Ticaga installation.

This allows you to use **https://ubuntu.ticaga.com** instead of <http://ubuntu.ticaga.com>.

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

If you're using **Nginx** you will need to change **--apache** to **--nginx**. We recommend Apache as it's easier to maintain.



# Ioncube loaders

Firstly we need to install unzip:

```
sudo apt-get install zip unzip -y
```

We'll need that for Ticaga later too.

Let's grab the Ioncube files:

```
cd /tmp;  
wget -N https://downloads.ioncube.com/loader_downloads/ioncube_loaders_lin_x86-64.zip  
unzip ioncube_loaders_lin_x86-64.zip
```

Let's move it to PHP 8.4 loaders to the correct location:

We need to find out where the loader files are stored on Ubuntu, so let's run:

```
php -i | grep extension_dir
```

ours outputs: **/usr/lib/php/20240924**

Now let's move the files:

```
sudo cp /tmp/ioncube/ioncube_loader_lin_8.4.so /usr/lib/php/20240924
```

Now we need to load them in the PHP.ini:

```
sudo nano /etc/php/8.4/apache2/php.ini
```

You'll also need to edit the CLI.ini as-well eg:

```
sudo nano /etc/php/8.4/cli/php.ini
```

Else you won't see the ioncube loaders working.

Paste this under the [PHP] section:

```
zend_extension = /usr/lib/php/20240924/ioncube_loader_lin_8.4.so
```



# Add new user for SSH

Laravel doesn't like it when you run it in root, this is for security purposes. We agree with them it's safer, however you can use root.

```
adduser ticaga
```

This creates a user called "ticaga". You should have created a password whilst you added a user.

```
usermod -aG sudo ticaga
```

This allows you to add the user "ticaga" to the sudo group, **this allows you to use sudo.**

# Install Composer

Let's install Composer, this allows us to install all the amazing vendor files to use Ticaga:

```
curl -sS https://getcomposer.org/installer -o /tmp/composer-setup.php
```

Let's run the installer:

```
sudo php /tmp/composer-setup.php --install-dir=/usr/local/bin --filename=composer
```

# Install Ticaga

Let's log into the new user we created earlier:

```
su ticaga
```

Let's give our **www** folder the correct permissions:

```
sudo chown -R $USER /var/www/
```

---

Let's go to the www folder:

```
cd /var/www/
```

Let's create the Laravel project which we will use for Ticaga:

```
composer create-project --prefer-dist laravel/laravel ticaga
```

Provide permissions to the ticaga folder:

```
sudo chown -R $USER /var/www/ticaga
```

Now let's go into the ticaga folder:

```
cd ticaga
```

Let's remove all the default files from Laravel:

```
rm -rf ./* ./env ./env.example ./composer.lock
```

**THIS IS AN IMPORTANT STEP AND DANGEROUS, ENSURE THE ./ IS BEFORE THE \*, ELSE YOU WILL DELETE THE WHOLE SERVER!**

If you are worried about breaking your server, please contact our support team or open a sales ticket if you have a trial license.

---

Let's get the latest version of Ticaga:

```
wget -N https://ticaga.com/latest.zip
```

Unzip the files:

```
unzip latest.zip
```

Let's install NodeJS for Ticaga:

```
sudo apt install nodejs -y  
sudo apt install npm -y
```

Now we've done everything we need, it's time to edit the .env file for Ticaga.

```
sudo mv .env.example .env
```

and let's edit the newly created .env:

```
sudo nano .env
```

or you can use vi to edit:

```
sudo vi .env
```

You can now edit the main settings:

```
APP_NAME="Ticaga Demo"
```

This is your company name.

```
APP_URL=https://demo.ticaga.com
```

This is your URL to your Ticaga, sub-domain or main domain if you prefer.

```
DB_CONNECTION=mysql  
DB_HOST=127.0.0.1  
DB_PORT=3306  
DB_DATABASE=ticaga  
DB_USERNAME=ticaga_user  
DB_PASSWORD=p@s3w0r$2024!
```

These are the database name, username and password you created when doing the [Database](#).

Let's provide the permissions for the bootstrap and storage folders:

```
sudo chmod -R ugo+rw bootstrap storage
```

Now we need to run the composer command to create the vendors folder for Ticaga, without this Ticaga won't work. We don't ship the vendors folder as it is huge:

```
composer install
```

Now we need to generate a secure App Key:

```
sudo php artisan key:generate
```

Let's sort the storage link out so you can upload avatars:

```
sudo php artisan storage:link
```

Let's create the database tables for Ticaga and seed the database with content pre-filled for Ticaga:

```
sudo php artisan migrate --seed
```

---

Now let's install Vite (this ensures nothing goes wrong during set-up):

```
sudo npm install  
sudo npm install -g vite
```

---

If you see this:

```
added 153 packages, and audited 154 packages in 8s
```

```
37 packages are looking for funding
```

```
run `npm fund` for details
```

```
6 vulnerabilities (3 moderate, 3 high)
```

```
To address all issues, run:
```

```
npm audit fix
```

```
Run `npm audit` for details.
```

Run the following:

```
sudo npm audit fix
```

Next upgrade NPM run:

```
sudo npm update
```

---

Publish Livewire Assets:

```
php artisan livewire:publish --assets
```

Need to replace the BladeCompiler for Laravel for Ioncube to work:

```
sudo mv ./hot_fix/vendor/laravel/framework/src/Illuminate/View/Compilers/BladeCompiler.php  
./vendor/laravel/framework/src/Illuminate/View/Compilers/BladeCompiler.php
```

Lets now build Ticaga:

```
sudo npm run build
```

Clear Ticaga's caches:

```
php artisan optimize:clear
```

Reboot the server to kick Ioncube into action:

```
sudo reboot
```

Log back into the server and go to Ticaga:

```
cd /var/www/ticaga
```

Log into the username:

```
su ticaga
```

Load up Ticaga:

```
php artisan up
```

That's it now you have a fully working Ticaga.

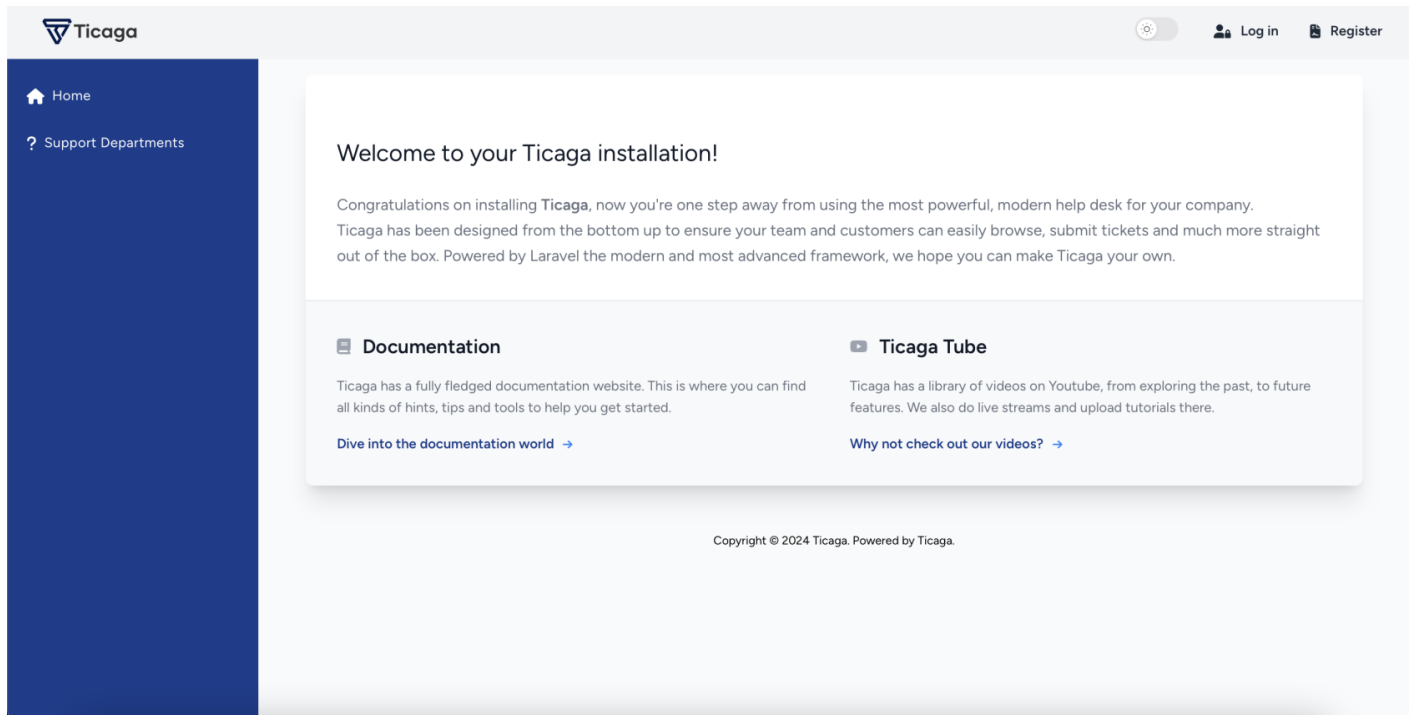




# Create an Agent


Now you've got your Ticaga up and running go to your installation:

eg: **https://demo.ticaga.com**



Now click on **Register**

Fill in the form, click **Register**.

Ticaga

Name

Company

Email

Mobile Number

Password

Confirm Password

☐ I agree to the [Terms of Service](#) and [Privacy Policy](#)

[Already registered?](#)

REGISTER

**Voila!** You're the super admin of your Ticaga installation.