Install Ticaga

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

```
su demo
```

Let's give our **www** folder the correct permissions, this allows you to create the ticaga folder:

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

Before we continue let's disable **SELinux**:

```
sudo sed -c -i "s/\SELINUX=.*/SELINUX=disabled/" /etc/selinux/config
```

And SELinux rules incase it's playing up (If this fails don't worry, you don't need these rules):

```
semanage fcontext -a -t httpd_sys_rw_content_t "/var/www/ticaga/storage(/.*)?"
semanage fcontext -a -t httpd_sys_rw_content_t "/var/www/ticaga/bootstrap/cache(/.*)?"
restorecon -Rv /var/www/ticaga
sudo setsebool -P httpd_can_network_connect_db 1
```

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
```

Now let's go into the ticaga folder:

```
cd ticaga
```

Let's remove all the default files from Laravel:

```
sudo 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
```

Now let's install Vite, this provides the "node_modules" for Laravel and Ticaga to use:

```
sudo dnf module install nodejs:22 -y
sudo npm install -g npm@11.2.0
sudo npm install
sudo npm install -g vite
```

Let's create the Ticaga .env:

```
sudo nano .env
```

or you can use vi:

```
sudo vi .env
```

Paste in the following:

```
APP_NAME="Laravel"
APP_ENV=production
APP_KEY=
APP_DEBUG=false
APP_URL=http://localhost

APP_LOCALE=en
APP_FALLBACK_LOCALE=en
APP_FAKER_LOCALE=en_US

APP_LIGHT_LOGO=
APP_DARK_LOGO=
LICENSE_KEY=
API_VERSION=v2

APP_MAINTENANCE_DRIVER=file
# APP_MAINTENANCE_STORE=database

PHP_CLI_SERVER_WORKERS=4

BCRYPT_ROUNDS=12

LOG_CHANNEL=stack
LOG_STACK=single
LOG_DEPRECATIONS_CHANNEL=null
LOG_LEVEL=debug

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=ticaga
DB_USERNAME=ticaga_user
DB_PASSWORD=

TICAGA_MAIL=false
# MailSend Emails
MAILERSEND_API_KEY=

# MailGun Emails
MAILGUN_DOMAIN=
MAILGUN_SECRET=
MAILGUN_ENDPOINT=api.eu.mailgun.net

SESSION_DRIVER=database
SESSION_LIFETIME=120
SESSION_ENCRYPT=false
SESSION_PATH=/
SESSION_DOMAIN=null

BROADCAST_CONNECTION=reverb
FILESYSTEM_DISK=local
```

```
QUEUE_CONNECTION=database

CACHE_STORE=database
# CACHE_PREFIX=

MEMCACHED_HOST=127.0.0.1

REDIS_CLIENT=phpredis
REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
REDIS_PORT=6379

MAIL_MAILER=log
MAIL_SCHEME=null
MAIL_HOST=127.0.0.1
MAIL_PORT=2525
MAIL_USERNAME=null
MAIL_PASSWORD=null
MAIL_FROM_ADDRESS="noreply@domain.com"
MAIL_FROM_NAME="${APP_NAME}"

AWS_ACCESS_KEY_ID=
AWS_SECRET_ACCESS_KEY=
AWS_DEFAULT_REGION=us-east-1
AWS_BUCKET=
AWS_USE_PATH_STYLE_ENDPOINT=false

VITE_APP_NAME="${APP_NAME}"

REVERB_APP_ID=
REVERB_APP_KEY=${APP_KEY}
REVERB_APP_SECRET=
REVERB_PORT=8080
REVERB_SCHEME=http
```

You can now edit the main settings:

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

This is your company name.

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

```
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 vendor folder for Ticaga, without this Ticaga won't work. We don't ship the vendor folder as it is huge:

```
composer install
```

Now we need to generate a secure App Key:

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

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

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

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
```

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

Revision #28
Created 4 September 2024 20:56:14 by Michael Dance
Updated 26 June 2025 23:09:45 by Michael Dance