1   Installation Manual Server

1.1   Primary configuration setup

1.1.1   Prerequisites

- VPS
- Domain name
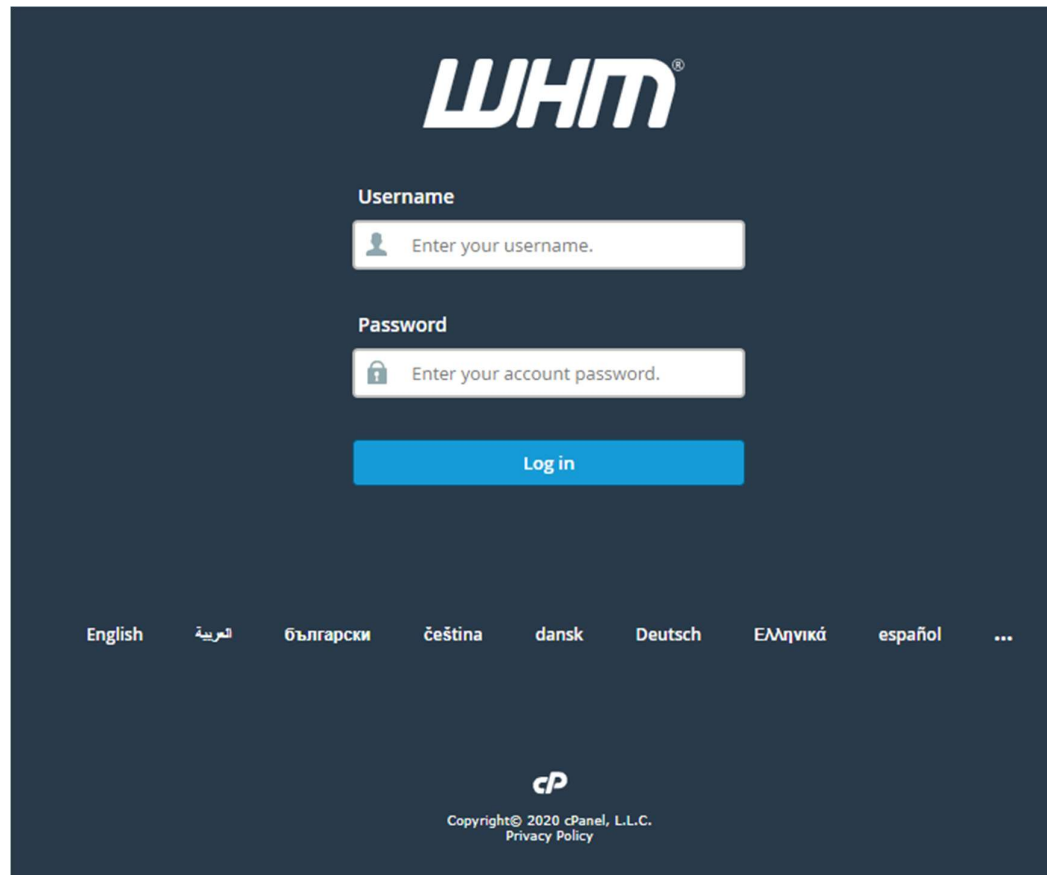- Basic server knowledge

For this project we used a VPS with cPanel, but this is not required. Domain name is also not required but otherwise your URL depends on the providers URL and might not be as accessible/clear for the end user.

1.1.2   Configure cPanel

To configure cPanel you first must create an account using WHM, you can access WHM by using your website (which was obtained from the server provider) and add :2087.

So https://yoursite.com:2087

Log-in using the credentials obtained from the provider.

Create a user for your new domain by clicking on "Create a New Account".

Enter information in the required fields

Now you can use your cPanel by surfing to:

https://yoursite.com:2083

and entering the credentials of your newly created account

1.1.3   Configure subdomains

When you are logged in in cPanel you can create new subdomains by clicking on the subdomain tab.
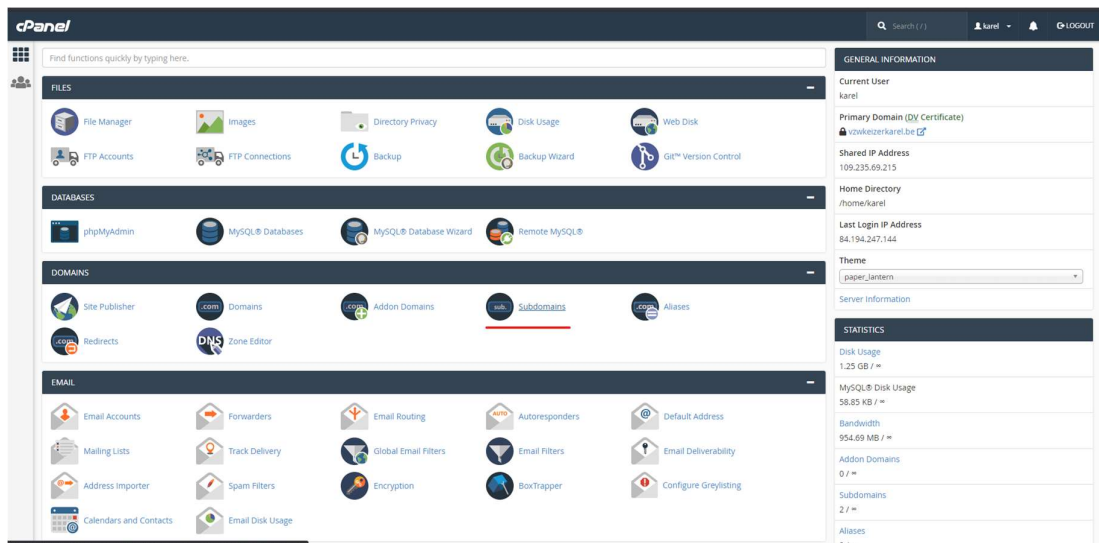
Enter the required information.
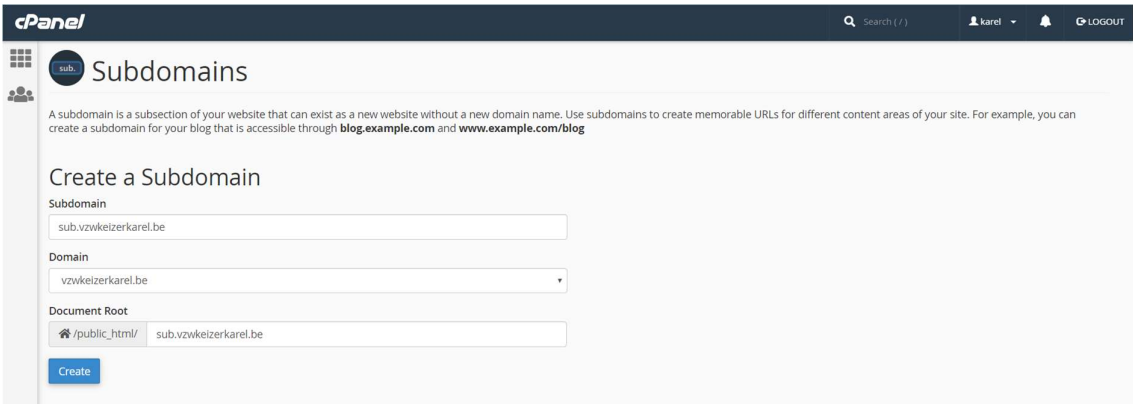
1.1.4   Enable HTTPS

To enable HTTPS, go to domains in cPanel and select Force HTTPS Redirect.

1.1.5   Disable access to website files

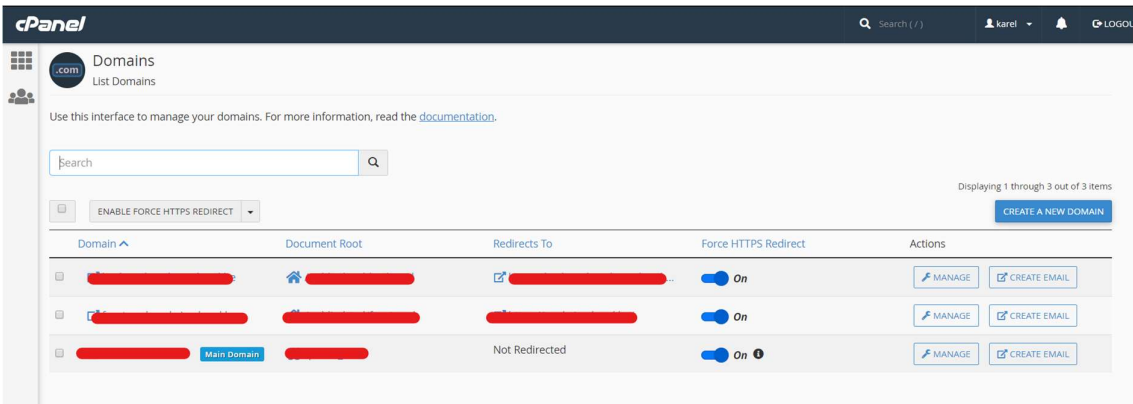To disable users from accessing your file tree you only must add the following line to your .htaccess file which is in the home directory of your domain (mostly public_html folder).

Options -Indexes

1.2   Back-end configuration

If you pull the newest version of the back-end from the git repo (by clicking download on the GitHub page) you will have to do the following steps.

1.2.1   Rename the public folder

Rename the *public* folder to *public_html*

1.2.2   Modify Laravel files

Open *public_html/index.php* and add some code to overwrite the default public path

```php
1    $app = require_once __DIR__.'/../bootstrap/app.php';
2    // overwrite the default public path with the path of the current folder
3    $app->bind('path.public', function() {
4        return __DIR__;
5    });
```

Open *bootstrap/app.php* and fix the path for Artisan scripts

```php
1    $app = new Illuminate\Foundation\Application(
2        $_ENV['APP_BASE_PATH'] ?? dirname(__DIR__)
3    );
4    // fix path for Artisan scripts
5    $app->bind('path.public', function() {
6        return base_path().'/public_html';
7    });
```

1.2.3   Modify Laravel mix

Open *webpack.mix.js*

- Set the public path to *public_html*
- Change the output path for **mix.js()** to *js* and for **mix.sass()** to *css*

```js
1    mix.setPublicPath('public_html');
2
3    mix.js('resources/js/app.js', 'js')
4        .sass('resources/sass/app.scss', 'css');
```

1.3   Put your website online

1.3.1   Recompile your assets for production

The **assets** *app.css* and *app.js* are not production ready and **should be minified/optimized** using the command

```
npm run prod..
```

This command was done in a fully configured php-storm environment.

1.3.2   Upload the files to your hosting provider

- Upload all folders, except *.git*, *.idea*, *database*, *node_modules* and *tests*
- Upload only the file *.env*

1.3.3   Disable debug mode

It's best practice to **disable debug mode on a live webserver**

Open the **REMOTE** (not the local!) *.env* file and set **APP_DEBUG=false**

1.3.4   Export/import the database

Create a **new database on your hosting**

- Open your **remote phpMyAdmin**
- **"Import"** (= the tab next to "Export") *yourdatabase.sql* (in your newly created database on your remote phpMyAdmin)
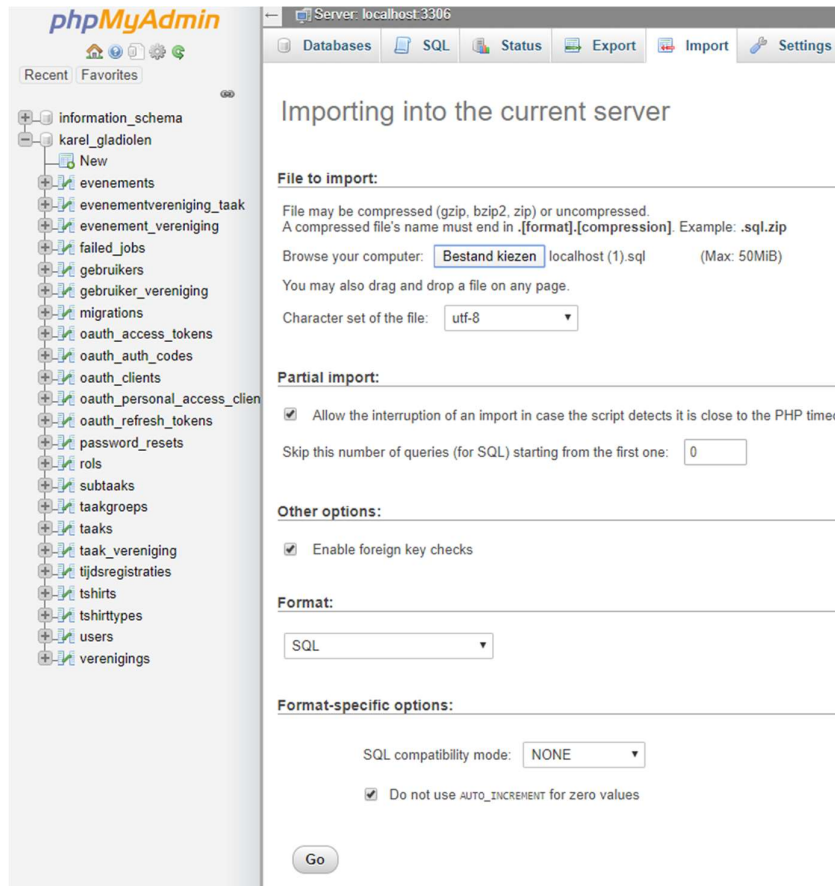
Open the **REMOTE *.env*** file and **update all DB_... variables**

Don't forget to enter:

INSERT INTO `oauth_clients` (`id`, `user_id`, `name`, `secret`, `redirect`, `personal_access_client`, `password_client`, `revoked`, `created_at`, `updated_at`) VALUES(1, 1, 'test', 'JoAymaKKIzKJEriHMCQcMjsOBxNdpmWZBSUpbelY', 'http://localhost', 1, 0, 0, '2020-02-23 15:08:13', '2020-02-23 15:08:13');

in the database using php-myadmin to enable the login module