

Romain C

Vincent L

Louis B

Thomas H

Soufiane H

**Document d’installation et de configuration**

Table des matières

[Informations machine 3](#_Toc14546510)

[Éléments installés 3](#_Toc14546511)

[Configuration d’apache 3](#_Toc14546512)

[Configuration SSH 3](#_Toc14546513)

[Composer 4](#_Toc14546514)

[Git 4](#_Toc14546515)

[NodeJs 4](#_Toc14546516)

[Expect 4](#_Toc14546517)

[Application Front (React) 4](#_Toc14546518)

[Application back (Laravel) 4](#_Toc14546519)

# Informations machine

L’adresse IP de la machine est la suivante : 10.1.1.1

Les identifiants sur serveur sont les suivants :

* Pour se connecter en utilisateurs sans droits :
* Identifiant : pi
* Mot de passe : P@ssword
* Pour se connecter en root (non recommandé) :
* Identifiant : root
* Mot de passe : 3f3S

# Éléments installés

Plusieurs éléments ont été installer sur le serveur :

* Un serveur Apache afin de faire fonctionner la partie Web ainsi que tous les modules PHP ;
* Un serveur SSH a été installer afin de prendre la main à distance ;
* Composer afin de lier les librairies PHP ;
* Git afin de faire du versionning ;
* NodeJs c’est un micro-serveur qui permettra d’interpréter le front-end ;
* NPM (le gestionnaire de paquets officiel de NodeJs).

# Configuration d’apache

Pour Apache, le port par défaut n’est pas changé, l’application web communiquera par le port 80.

Concernant le dossier de l’application, il se trouve dans le répertoire suivant :

/var/www/html/ProjetCiscoCesi/

Seul www-data possède les droits sur le dossier.

# Configuration SSH

Pour la configuration SSH. Le port SSH est celui par défaut, le port 22  
Depuis l’extérieur, il faut passer par le port 2222

Une personne ne peut pas se connecter directement en utilisateur root.

# Composer

Composer est un logiciel gestionnaire de dépendances libre écrit en PHP. Il permet à ses utilisateurs de déclarer et d'installer les bibliothèques dont le projet principal a besoin.

# Git

Git est installé afin de permettre de synchroniser le dépôt où sont stockés les parties créées par les développeurs avec celui du PI.

# NodeJs

Node.js est une plateforme logicielle libre et événementielle en JavaScript orientée vers les applications réseau qui doivent pouvoir monter en charge.

Il utilise le port 3000

# Expect

Expect est l’outil qui nous permet d’envoyer des commandes sur les équipements Cisco.

# Application Front (React)

Pour faire fonctionner l’application côté front :

* Se rendre à la racine du dossier situé à /var/www/html/projetCesiCisco ;
* Ouvrir un terminal si ce n’est pas déjà fait ;
* Récupérer le dépôt **GIT** ;
* Se rendre à la racine du dossier front situé à /var/www/html/projetCesiCisco/Code/Front ;
* Saisir npm install dans le terminal;
* Une fois l'installation des node\_modules terminés, saisir npm start ;
* Une fois le serveur lancé, se rendre via le navigateur sur localhost:3000 ;
* (Il est nécessaire au préalable d'avoir installé nodejs et d’avoir lancer l’api Laravel).

# Application back (Laravel)

Pour faire fonctionner l’application côté back :

* Se rendre à la racine du dossier situé à /var/www/html/projetCesiCisco ;
* Ouvrir un terminal si ce n’est pas déjà fait ;
* Récupérer le dépôt **GIT** ;
* Se rendre sur /var/www/html/projetCesiCisco/Code/Back ;
* S’assurer d’être en **PHP 7.2 minimum** pour réaliser la suite des actions ;
* Ensuite mettre à jour les packages **Laravel** avec la commande **composer update** ;
* Répondre aux demandes de **COMPOSER** afin d’installer les extensions manquantes de **PHP** ;
* Le module **SSH2** de **PHP** est indispensable pour la connexion au **switch Cisco.**
* Vérifier que le fichier. env est bien présent, sinon, il faudra le créer à partir du fichier .env.example situé dans /var/www/html/projetCesiCisco/Code/Back.
* Un Fichier **.htaccess** sera utile pour la gestion des accès. Il est à placer dans /var/www/html/projetCesiCisco/Code/Back/public . il doit contenir le code suivant :

<IfModule mod\_rewrite.c>

<IfModule mod\_negotiation.c>

Options -MultiViews

</IfModule>

RewriteEngine

# Handle Front Controller...

RewriteCond %{REQUEST\_FILENAME} !-d

RewriteCond %{REQUEST\_FILENAME} !-f

RewriteRule ^ index.php [L ]

# Handle Authorization Header

[R=301,L]

</IfModule>

* Sur un server dédié ne pas oublier de donner les droits d’accès à l’utilisateur www-data :www-data sur tout le répertoire du back ;
* Configuration du VirtualHost d’apache a adapté suivant les configurations :

<VirtualHost \*:80>

ServerName "ProjetCisco.local"

ServerAdmin webmaster@localhost

DocumentRoot /var/www/html/ProjetCiscoCesi/Code/Back/public

ErrorLog ${APACHE\_LOG\_DIR}/error.log

CustomLog ${APACHE\_LOG\_DIR}/access.log combined

<Directory /var/www/html/ProjetCiscoCesi/Code/Back/public>

Options FollowSymLinks MultiViews

Order Allow,Deny

Allow from all

ReWriteEngine On

AllowOverride All

</Directory>

</VirtualHost>

* Si tout est bien installer faire les tests à partir du front.