



Rapport de stage

Mise en place d'une solution de ticketing auprès
des communes

Réalisé par :
SOULIER Rémi

Sous la supervision de :
WANLIN Eric

Année universitaire 2023-2024

Remerciements

Avant de commencer ce rapport, je tiens à remercier mon maître de stage M. WANLIN Eric, le directeur du service informatique M. GRACIA Jean-Philippe, les autres membres du service informatique M. BAILLY Thomas et M. LE GOURRIEREC Dominique pour m'avoir permis de réaliser mon stage, de m'accorder une confiance très importante pour déployer un outil important pour la collectivité et de m'avoir guidé tout au long du stage.

Je remercie également l'ensemble des agents de la communauté de communes du Grand Pic Saint-Loup de m'avoir accueilli et intégré au sein de la structure.

Je remercie également M. TROMBETTONI Gilles d'avoir tutoré ce stage et de m'avoir rendu visite pour s'intéresser à ma mission.

Sommaire

Remerciements.....	1
Glossaire.....	4
Résumé.....	6
Présentation de l'entreprise.....	8
Communauté de Communes du Grand Pic St Loup.....	8
Service informatique.....	9
Déploiement d'une solution de ticketing à destination des communes.....	10
Introduction.....	11
Cahier des charges.....	12
Recherche des pré-requis.....	13
Directives sur les outils à utiliser.....	13
Pré-requis de l'application GLPI.....	13
Installation de la machine virtuelle.....	14
Configuration des ressources de la machine virtuelle.....	14
Paramétrage des connexions réseau.....	15
Partitionnement du stockage de la machine virtuelle.....	16
Installation des pré-requis.....	16
Installation d'Apache2.....	16
Installation et configuration de PHP.....	17
Installation de MariaDB.....	17
Clonage du dépôt de GLPI.....	17
Attribution du nom de domaine.....	18
Installation du certificat SSL et configuration du VHost Apache.....	18
Installation de GLPI.....	18
Premières réflexions sur la configuration de GLPI pour le support informatique.....	19
Passage à des comptes nominatifs.....	21
Création des comptes nominatifs.....	21
Echanges avec la police rurale.....	24
Définition des profils de la police rurale et séparation des entités.....	26
Rédaction des guides utilisateurs.....	27
Conclusion du projet 1.....	27
Plateforme de collecte des informations du Plan InterCommunal de Sauvegarde.....	28
Introduction.....	29
Analyse du fichier initial.....	30
Cahier des charges.....	31
Liste des cas d'utilisation et rédaction des user stories.....	31
Configuration du serveur web.....	34
Création du modèle entité association et du script sql.....	34
Structure du site web.....	36
Fonctionnalités du site.....	37

Fonctionnalités générales.....	37
Connexion.....	37
Menu de navigation.....	38
Fonctionnalités à destination des communes.....	38
Liste des équipements.....	39
Informations de l'utilisateur.....	40
Fonctionnalités à destination de l'administrateur.....	41
Gestion des intitulés.....	41
Export de la base de données en CSV.....	42
Conclusion du projet 2.....	43
Conclusion.....	43

Glossaire

GLPI (Gestionnaire Libre de Parc Informatique) : Un logiciel open-source de gestion de parc informatique et de helpdesk.

DMZ (Demilitarized Zone) : Une zone de réseau physique ou logique qui contient et expose les services externes d'une organisation à un réseau non fiable, généralement l'Internet. La DMZ ajoute une couche de sécurité supplémentaire au réseau local (LAN) d'une organisation, car elle limite l'accès direct entre les systèmes internes et externes.

MariaDB : Un système de gestion de bases de données relationnelles, dérivé de MySQL.

Pare-feu (Firewall) : Un dispositif de sécurité réseau qui surveille et contrôle le trafic réseau entrant et sortant en fonction de règles de sécurité prédéfinies.

iptables : Un utilitaire en ligne de commande utilisé pour configurer les règles du pare-feu sur les systèmes Linux.

SSH (Secure Shell) : Un protocole réseau cryptographique pour les opérations sécurisées sur des réseaux non sécurisés.

Plug-in Fields : Un module complémentaire pour GLPI permettant d'ajouter des champs de texte dans divers menus de l'application.

Plug-in Data Injection : Un module complémentaire pour GLPI permettant d'insérer un grand nombre de données via un fichier .csv.

Plug-in FormCreator : Un module complémentaire pour GLPI permettant de créer des formulaires personnalisés pour la création de tickets.

Session : Une manière de stocker des informations (sous forme de variables) à travers plusieurs pages d'un site web pour un utilisateur donné.

MVC (Modèle-Vue-Contrôleur) : Un modèle architectural utilisé pour le développement d'applications web, séparant les données (Modèle), l'interface utilisateur (Vue), et la logique de contrôle (Contrôleur).

User Story : Une description simple et concise d'une fonctionnalité du point de vue de l'utilisateur final.

Script SQL : Un ensemble d'instructions SQL utilisées pour interagir avec une base de données.

Faker : Une bibliothèque Python utilisée pour générer des données factices.

CSV (Comma-Separated Values) : Un format de fichier utilisé pour stocker des données tabulaires sous forme de texte, où chaque ligne représente un enregistrement et chaque champ est séparé par une virgule.

SSL (Secure Sockets Layer) : Un protocole de sécurité standard pour établir un lien crypté entre un serveur web et un navigateur.

htaccess : Un fichier de configuration utilisé par le serveur web Apache pour définir des règles spécifiques de configuration pour un répertoire.

Netiwan : Le fournisseur d'accès internet de l'entreprise.

Registrar : Une organisation ou une société qui gère la réservation de noms de domaine sur Internet.

Plan Intercommunal de Sauvegarde : Un plan visant à répertorier les équipements de toutes les communes membres d'une intercommunalité afin de pouvoir les mobiliser en cas de crise.

Entité : Dans GLPI, une entité est une division logique utilisée pour séparer les différents services ou départements au sein de l'application.

Compte générique : Un compte utilisateur partagé par plusieurs personnes, souvent utilisé pour des accès communs ou temporaires.

Compte nominatif : Un compte utilisateur individuel, attribué à une seule personne, garantissant une meilleure sécurité et traçabilité.

Formulaire Google : Un outil de Google permettant de créer des formulaires en ligne pour collecter des informations.

Session de navigateur : Une session de navigateur est une période pendant laquelle un utilisateur interagit avec une application web, souvent identifiée par un cookie de session.

Salage et hachage : Techniques de sécurité utilisées pour protéger les mots de passe en ajoutant une valeur aléatoire (sel) avant de les chiffrer (hachage).

Résumé

Durant mon stage à la communauté de communes du Grand Pic Saint Loup, j'ai travaillé sur deux projets principaux.

Le premier projet concernait le déploiement et la configuration de l'application GLPI pour la gestion des demandes d'assistances. J'ai configuré l'application GLPI, créé des comptes utilisateurs, mis en place des formulaires personnalisés pour les demandes de la police rurale et défini les rôles des membres du service. Enfin, j'ai rédigé des guides utilisateurs pour faciliter l'utilisation de l'application et une notice d'installation de l'application.

Le second projet concernait le développement d'un site web pour le Plan Intercommunal de Sauvegarde, permettant aux communes de renseigner leurs équipements. J'ai analysé le fichier Excel initial et créé un modèle de base de données. Des user stories ont été définies pour les différentes fonctionnalités, et un script SQL a été généré pour la base de données. Le site permet aux utilisateurs de gérer les articles, contacts, et lieux de remisage, avec des fonctionnalités d'ajout, de modification, et de suppression. J'ai structuré le site web en utilisant l'architecture MVC et implémenté des fonctionnalités de connexion, de gestion des contacts et des entrepôts, ainsi que des fonctionnalités spécifiques pour les administrateurs.

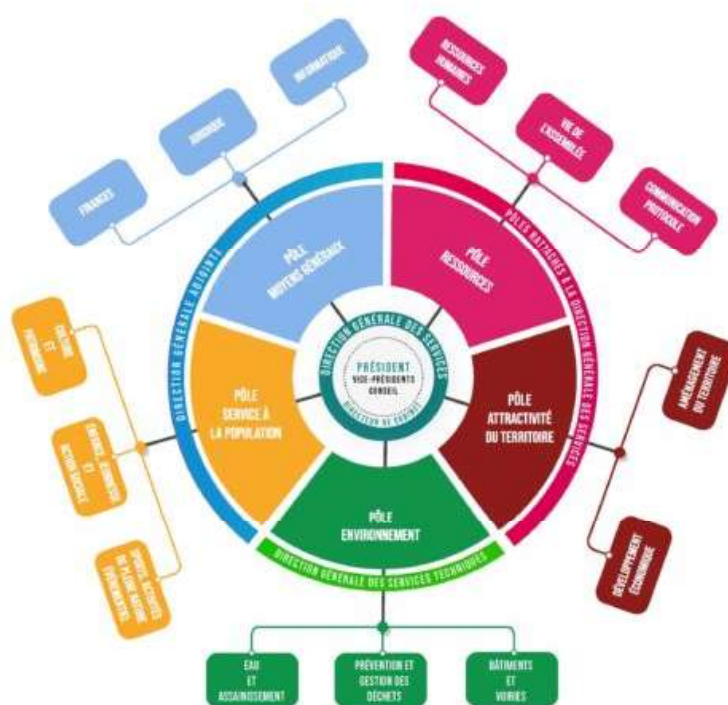
Présentation de l'entreprise

Communauté de Communes du Grand Pic St Loup

La Communauté de Communes du Grand Pic Saint Loup est une collectivité territoriale située au nord de Montpellier, dans le département de l'Hérault. Elle a été créée en 2010 suite à la fusion de trois communautés de communes. Son nom provient du sommet montagneux emblématique au centre de son territoire : le Pic Saint-Loup.

Cette communauté regroupe 36 communes et compte plus de 50 000 habitants. Elle est dirigée par le conseil communautaire composé du président de la communauté, élu parmi les maires des différentes communes, 14 vice-présidents en charge des commissions et 67 conseillers. Le conseil communautaire a pour rôle de décider des actions à engager sur proposition des commissions, voter les délibérations, le budget et régler les affaires qui sont de la compétence de la communauté de communes. En plus de ce conseil, les agents sont sous la direction de la directrice générale des services.

La communauté de communes est composée de 14 services qui opèrent dans différents secteurs d'activités tels que les ressources humaines, la culture et le patrimoine, ou encore la gestion des déchets. Ces services sont répartis dans 5 pôles : le pôle ressources, attractivité du territoire, environnement, service à la population et les moyens généraux.



Ces services sont répartis dans 11 sites sur le territoire de la communauté de communes. Parmi eux se trouve l'hôtel de la communauté, le siège de la collectivité, le centre technique intercommunal ou encore la halle du verre, un musée géré par la communauté de communes.

Service informatique

Le Service informatique de la communauté de communes est rattaché au pôle des moyens généraux. Il est composé de 4 membres, un directeur et 3 techniciens, et est situé au sein de l'hôtel de la communauté. Son rôle consiste à gérer le parc informatique de tous les services (postes de travail, serveurs, lignes téléphoniques ,etc ...) et d'apporter une assistance aux agents qui rencontrent des problèmes informatiques

Projet 1

-

Déploiement d'une solution de ticketing à destination des communes

Introduction

De nos jours, les mairies ont un usage quasi quotidien de l'informatique, cependant, de nombreuses communes n'ont pas les moyens ni l'utilité d'engager un technicien pour régler les problèmes informatiques du quotidien comme un poste qui ne démarre pas ou encore qui n'a pas accès à internet. Seulement, les agents de mairie n'ont pas toujours les compétences pour régler ces problèmes qui peuvent être très handicapants et ont donc souvent recours à des prestataires externes .

Lors des dernières élections, 14 commissions ont été mises en place par les élus de la communauté de communes du Grand Pic st Loup. Ces commissions sont définies en fonction des projets et des actions de la collectivité. Parmi ces 14 commissions se trouve la commission « Fonds de concours et mutualisation », qui consiste à mutualiser des achats, travaux, services pour plusieurs communes en même temps (ex : acheter 30 défibrillateurs pour avoir des réductions au lieu de les acheter un par un).

Lors d'une commission de mutualisation, la mise en place d'une période de test concernant le support informatique de premier niveau pour les communes a été actée. Pour ce test, huit communes de taille et d'infrastructures variées ont été sélectionnées (de 1 à 2 postes à plus de 50 pour la plus importante).

Pour fournir ce support, la DSI souhaite déployer un logiciel de ticketing afin de centraliser les demandes de tickets, gérer l'historique des suivis et extraire des statistiques. Actuellement, ce support est assuré par mail ou téléphone et un logiciel de prise en main à distance a été déployé et mis à disposition de ces communes.

En parallèle, la Communauté de communes dispose d'une unité de police rurale. Cette dernière est sollicitée par les communes pour diverses interventions (déchets sauvages, constructions illégales, ...). Il a donc également été décidé de centraliser ces demandes via cette application.

Je vais donc vous présenter, dans un premier temps, le cahier des charges de ce projet, puis, dans un second temps, le rapport technique de ce projet, et terminer par les différents manuels d'utilisation.

Cahier des charges

A mon arrivée, le cahier des charges n'avait pas été défini. Je l'ai donc rédigé après les différentes réunions auxquelles j'ai participé. Je l'ai divisé en trois parties : les besoins généraux , les besoins du support informatique puis ceux de la police rurale

besoins généraux

- les tickets doivent appartenir à des catégories définies
- Les utilisateurs doivent être notifiés par mail à chaque action sur leur ticket.
- les techniciens peuvent ajouter une réponse d'un ticket à la base de connaissances si le problème est récurrent

Support informatique

- Créer des comptes nominatifs pour les agents des communes.
- Le champ description des tickets doit être obligatoire pour avoir des informations détaillées et traiter plus efficacement les tickets.
- les techniciens ne peuvent voir que les tickets liés au support informatique

police rurale

- Créer des comptes génériques pour les communes
- Rendre obligatoires les champs de texte de la personne à contacter et de l'emplacement.
- créer deux profils différents :
 - Responsable qui voit tous les tickets en lien avec la police rurale et qui peut les attribuer aux agents.
 - agent qui peut voir seulement les tickets qui lui sont attribués et peut les résoudre et les archiver

Recherche des pré-requis

Pour réaliser ce projet, les techniciens du service informatique m'ont demandé d'utiliser le logiciel de gestion de parc informatique GLPI. Ils ont choisi ce logiciel car il possède toutes les fonctionnalités nécessaires à la réalisation du projet. De plus, c'est un logiciel qu'ils ont l'habitude d'utiliser car une version de GLPI est déjà déployée en interne.

Pour déployer ce logiciel, il existe certains pré-requis . En plus de ces pré-requis, mon tuteur de stage m'a donné quelques directives sur les outils à utiliser

Directives sur les outils à utiliser

Lors des premières journées de stage, j'ai demandé des informations sur la méthode à suivre à l'équipe du service informatique . Deux directives principales en sont ressorties :

- La machine virtuelle hébergée sur le serveur interne doit tourner avec un système d'exploitation Ubuntu LTS.
- la machine virtuelle doit être configurée en DMZ sur le réseau

Pré-requis de l'application GLPI

Pour pouvoir utiliser efficacement GLPI, il est nécessaire de respecter les prérequis définis par Teclib, la société à l'origine de GLPI. Les prérequis sont donc les suivants :

- La version de PHP installée sur la machine virtuelle doit être comprise entre la version 7.4 et 8.2
- GLPI est compatible avec 4 serveurs Web différents : Apache 2 (ou plus récent) , Nginx , Lighttpd et Microsoft IIS
- Pour la base de données, GLPI supporte 2 SGBD : MySQL (v5.7 minimum) et MariaDB (v10.2 minimum)

Installation de la machine virtuelle

Pour héberger l'application, j'ai dû créer une machine virtuelle sur le serveur interne. L'infrastructure de la CCGPSL est composé de 3 nœuds hyperconvergés gérée par un hyperviseur VMware. Une infrastructure de PRA est située sur un site secondaire, permettant une reprise d'activité en cas de rupture de service.

Les détails techniques de l'installation de la machine virtuelle et de l'application sont indiqués dans l'annexe 5.

Configuration des ressources de la machine virtuelle

Après avoir défini les prérequis, j'ai pu démarrer l'installation de la machine virtuelle. Je me suis connecté à l'application VMware sur le serveur interne de l'entreprise et j'ai configuré les ressources auxquelles avait accès la machine virtuelle.

Edit Settings

VM-REMI

×

Virtual Hardware

VM Options

ADD NEW DEVICE

> CPU	2	▼	ⓘ
> Memory	4	▼	GB ▼
> New Hard disk *	50	GB ▼	
> SCSI controller 0	LSI Logic Parallel		
> Network adapter 1	21DMZ ▼	<input checked="" type="checkbox"/> Connect...	
> CD/DVD drive 1	Datastore ISO File ▼	<input checked="" type="checkbox"/> Connect...	
> Video card	Specify custom settings ▼		
> Security Devices	Not Configured		
VMCI device			
SATA controller 0	AHCI		
> Other	Additional Hardware		

CANCEL

OK

Fig 1

J'ai ensuite choisi mon système d'exploitation. Dans un premier temps, nous avons choisi la version 24.04 LTS d'Ubuntu sortie en avril 2024, cependant, du fait de sa sortie très récente, la seule version de PHP disponible était la version 8.3.4, et comme vu un peu plus tôt, GLPI ne supporte que jusqu'à la version 8.2. Nous avons donc finalement décidé d'utiliser la version 22.04 LTS qui validait tous les prérequis de GLPI.

Paramétrage des connexions réseau

Une fois la machine virtuelle créée, il a fallu configurer le système d'exploitation. J'ai donc créé le compte utilisateur et défini la langue et la disposition du clavier. J'ai ensuite configuré les paramètres réseau de la machine, je lui ai attribué une adresse ip en DMZ (192.168.21.X sur le réseau de l'entreprise) afin qu'elle soit accessible en local mais également depuis l'extérieur du réseau privé. J'ai également défini les serveurs DNS locaux et le nom de domaine du réseau

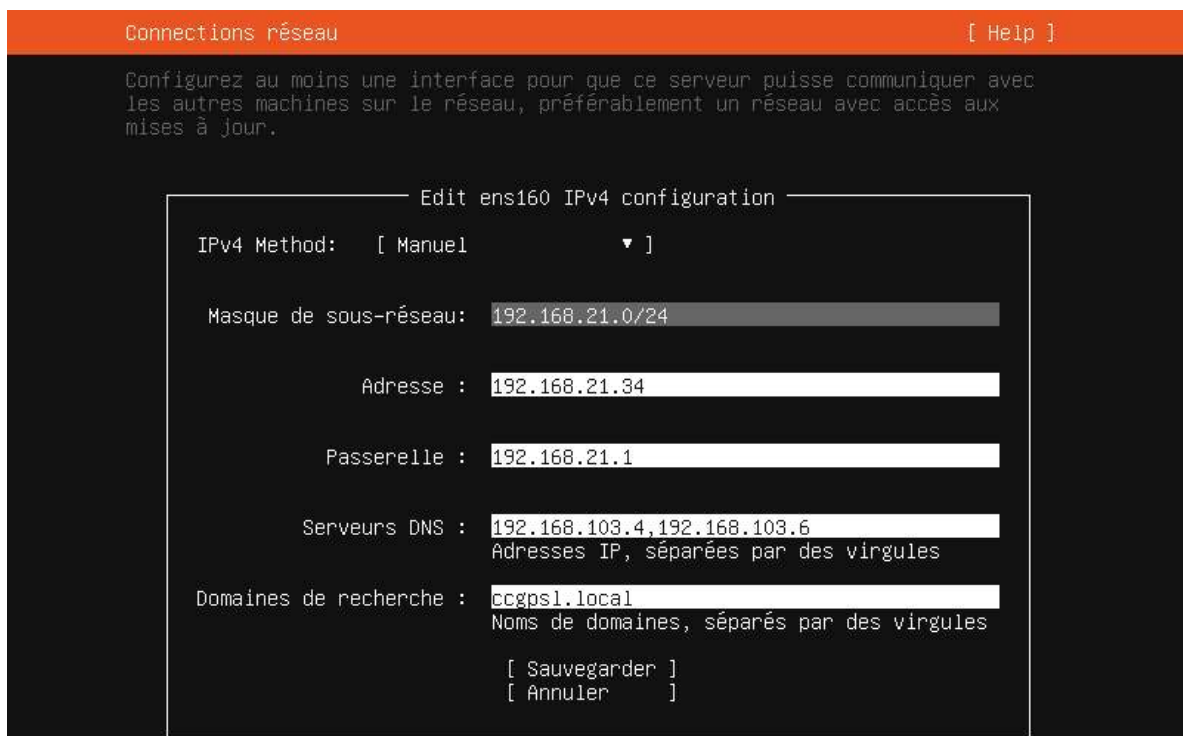


Fig 2

Partitionnement du stockage de la machine virtuelle

J'ai ensuite défini l'espace que les différentes partitions du disque virtuel allaient utiliser. J'ai donc attribué 512 mo pour le système de démarrage /boot et le reste du disque automatiquement pour les autres partitions soit 49,5 go

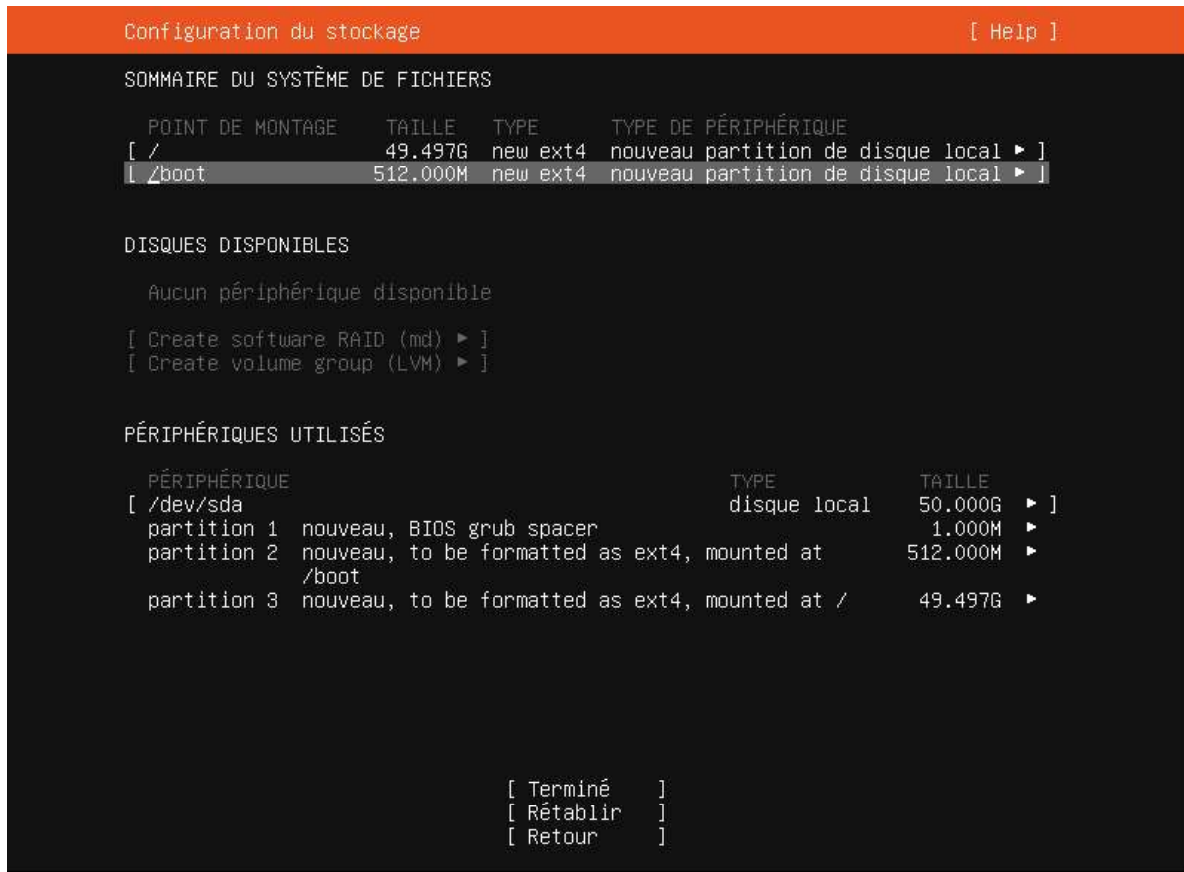


Fig 3

J'ai ensuite installé le serveur openSSH afin de pouvoir configurer la machine virtuelle avec une console via une connexion SSH

Installation des pré-requis

Une fois le système d'exploitation configuré, nous avons installé les différents prérequis.

Installation d'Apache2

Après avoir vérifié que le serveur soit bien à jour, j'ai installé et activé Apache.

Installation et configuration de PHP

J'ai installé PHP dans sa version 8.2.

je l'ai ensuite configuré selon les besoins de glpi.

Installation de MariaDB

J'ai ensuite installé le serveur mariaDB et créé la base de données. J'ai choisi d'installer mariaDB car c'est une solution open source

Clonage du dépôt de GLPI

Pour installer GLPI, il faut cloner le dépôt GitHub de la version qui nous intéresse (ici 10.0.2)

Pour cela, j'ai récupéré l'archive du dépôt, créé un répertoire à la racine du serveur web et placé l'archive décompressée dans ce dossier.

Configuration du pare feu

Pour que mon application soit accessible depuis internet, j'ai dû configurer des nouvelles règles sur le pare-feu mis en place par Netiwan, le fournisseur d'accès internet de l'entreprise . Pour pouvoir lier une adresse publique à mon adresse privée, j'ai choisi une adresse non utilisée dans un fichier excel interne qui liste toutes les adresses. Ensuite, sur l'interface du pare-feu, j'ai créé 2 règles pour rediriger les requêtes sur les ports 80 pour les requêtes HTTP et les ports 443 pour les requêtes HTTPS vers mon adresse locale

J'ai également configuré le pare-feu de la machine virtuelle avec le module iptables . J'ai donc ouvert les ports suivants :

- 22 pour autoriser les connexions en SSH depuis des appareils sur le réseau local.
- 80 pour autoriser les connexions via des requêtes HTTP
- 443 pour autoriser les connexions via des requêtes HTTPS

Attribution du nom de domaine

Afin de faciliter l'accès du site aux utilisateurs, il était nécessaire de lier l'adresse publique à un nom de domaine. J'ai donc discuté avec l'équipe du service informatique d'un nom de domaine adapté, en sachant qu'il devait se terminer par ccgpsl.fr. On a donc décidé d'utiliser le nom de domaine support.ccgpsl.fr.

Une fois le nom de domaine défini, nous avons fait une demande chez NFrance, un prestataire de la communauté de communes qui a un rôle de registrar, pour déposer le nom de domaine et le déployer par propagation sur tous les DNS dans le monde.

Installation du certificat SSL et configuration du VHost Apache

Pour pouvoir utiliser les requêtes HTTPs qui correspondent au protocole SSL, il faut utiliser un certificat SSL pour crypter les données. Dans cette optique, le service informatique possède un certificat wildcard qui peut être utilisé pour sécuriser tous les sites web avec pour nom de domaine *.ccgpsl.fr .

J'ai donc placé le fichier du certificat (.crt) et de la clé (.key) dans le répertoire /etc/ssl/gmpi/ de ma machine virtuelle

J'ai ensuite créé un fichier de configuration du Virtualhost d'Apache et renseigné l'ensemble des informations de mon site, puis créé un lien symbolique de mon fichier de configuration vers le répertoire des sites actifs pour qu'il soit pris en compte.

Ensuite, pour faciliter l'accès à l'application, j'ai écrit un script PHP à la racine du serveur web pour réécrire l'URL et accéder directement à l'application dans le dossier /gmpi.

```
<?php
header("Location: gmpi/");
die();
?>
```

Installation de GLPI

Pour pouvoir installer GLPI, il faut donner les droits en écriture à l'utilisateur www-data (le serveur web) pour modifier les fichiers lors de l'installation du logiciel

J'ai donc suivi le programme d'installation de l'application où j'ai indiqué quelle base de données utiliser . Une fois l'installation terminée, 4 comptes utilisateurs ont été créés automatiquement :

- un super admin
- un technicien
- un utilisateur lambda
- un compte qui peut seulement poster des tickets

j'ai donc supprimé les comptes "technicien" et "utilisateurs" et modifié le mot de passe du compte super admin pour éviter qu'une personne malveillante puisse se connecter avec les identifiants par défaut.

J'ai également supprimé le programme d'installation pour éviter de réinstaller le logiciel par erreur

Premières réflexions sur la configuration de GLPI pour le support informatique

Après avoir installé la machine virtuelle et l'application GLPI, j'ai discuté de la façon de configurer le logiciel avec mon tuteur . Lors de cette discussion , il m'a expliqué que le logiciel devait fonctionner de la manière suivante :

- Les huit communes possèdent un compte qui leur est propre.
- Tous les utilisateurs de la commune ont accès au compte générique.
- lors qu'un utilisateur envoie un ticket, il doit indiquer son identité et un moyen de le contacter
- le technicien contacte l'utilisateur par mail/téléphone pour régler son problème et ferme ensuite le ticket sur GLPI

J'ai donc créé les comptes des communes , renseigné les intitulés des catégories de tickets, créé les comptes des techniciens .

J'ai ensuite cherché un moyen d'ajouter des champs de texte au formulaire de base pour les tickets . Sur le forum de GLPI , j'ai appris l'existence du plug-in Fields. C'est un plug-in qui permet d'ajouter des champs de texte dans de nombreux menus de GLPI

J'ai donc installé le plug-in et crée 4 champs de textes que j'ai attaché au formulaire de création de tickets :

- Nom
- Prénom
- Téléphone
- Adresse mail

une fois les paramètres enregistrés, les nouveaux champs étaient accessibles dans la création du ticket



The image shows a screenshot of the GLPI ticket creation form. It features four new text input fields stacked vertically, each with a label to its left. The labels are 'Nom de l'utilisateur *', 'Prénom de l'utilisateur *', 'Téléphone *', and 'Email'. The asterisks indicate that these fields are required. The input fields are empty and have a light gray border. Below the form, there is a horizontal line.

Passage à des comptes nominatifs

Le 15 avril, j'ai été convié à une réunion entre le service informatique de la communauté de communes, la mairie de Saint-Clément-de-Rivière et la société SI Cloud, un prestataire en charge de la mise en place et de la gestion du parc informatique de la mairie, afin de définir le rayon d'action des différentes parties au sujet du support informatique. Cette réunion m'a permis de comprendre l'ampleur de mon projet. Sur le chemin du retour, j'ai discuté avec mon maître de stage des problématiques auxquelles j'avais pensé au sujet des comptes génériques. En effet, j'ai relevé certains points qui posaient problème avec cette façon de fonctionner.

- La confidentialité : si le compte était générique, un utilisateur aurait la possibilité de consulter les tickets de ses collègues et d'avoir connaissance de problèmes qui ne le concernent pas.
- Le mot de passe devrait être connu de tous les utilisateurs, ce qui poserait un problème s'il venait à changer.
- Un utilisateur pourrait créer des tickets en usurpant l'identité d'un autre
- Un employé qui aurait quitté l'entreprise ou aurait été licencié aurait encore accès à l'application et pourrait faire des demandes d'assistance inappropriées.

Au vu de ces problèmes, on a discuté avec le directeur du service et décidé de créer des comptes nominatifs pour chaque utilisateur des communes.

Création des comptes nominatifs

Pour créer les comptes nominatifs, j'ai dans un premier temps cherché s'il existait une façon de créer des comptes en masse via l'injection d'un fichier .csv . J'ai alors trouvé un plug-in qui correspondait parfaitement à mes besoins : le plug-in data injection . Ce plug-in permet d'insérer un grand nombre de données via un fichier .csv . Je l'ai donc configuré avec l'entête suivant afin de bien compléter les profils :

identifiant; nom de famille; prénom; téléphone; mot de passe; email; lieu; catégorie
--

J'ai également envoyé un fichier .xlsx (pour éviter les problèmes d'encodage à l'ouverture) aux communes de l'échantillon en leur demandant de me donner le nom, prénom, téléphone et mail de tous les utilisateurs de leur commune :

```
nom de famille;prénom;téléphone;mail
```

J'ai ensuite écrit 2 scripts en python :

- un script qui m'a généré des fichiers factices pour faire des tests

```
from faker import Faker`
import Faker
import csv

def genererFichierCommune(nomCommune):
    with open("fichiersCommunesTest/fichier" + nomCommune + '.csv', 'w',
newline='', encoding='utf-8') as csvfile:
        writer = csv.writer(csvfile, delimiter=';')
        writer.writerow(['nom', 'prénom', 'téléphone', 'email'])
        for i in range(3):
            firstname = fake.first_name()
            lastname = fake.last_name()
            phonenumber = fake.phone_number().replace(" ",
""").replace("+33", "0").replace("(0)", "").replace(")", "")
            writer.writerow([firstname, lastname, phonenumber, firstname
+ "." + lastname + "@yopmail.com"])

fake = Faker('fr_FR')
listeCommunes = ["Claret", "CausseDeLaSelle", "MasDeLondres",
"SaintClementDeRiviere", "SaintJeanDeCuculles",
"Valflaunes", "LeTriadou", "NotreDameDeLondres"]

for commune in listeCommunes:
    genererFichierCommune(commune)
```

- un script qui a généré le fichier d'insertion

```
import csv
import os
from faker import Faker
def miseEnFormeListeAgents(file):
    with open(file, newline='', encoding='utf-8') as csvfile:
        file = file[18:]
        agents = csv.reader(csvfile, delimiter=';')
        entete = True
        for agent in agents:
            if entete:
                entete = False
                continue
            else:
                id = agent[1][0] + agent[0].replace(" ",
""").replace("-", "")
                nom = agent[0]
                prenom = agent[1]
                telephone = agent[2].replace(" ", "")
                mdp = Faker().password(length=12, special_chars=True,
digits=True, upper_case=True, lower_case=True)
                email = agent[3]
                lieu = file[7:-4]
                categorie = "Commune-" + lieu
                for ID in ids:
                    if id == ID:
                        id = id[:1]+agent[1][1]+id[1:]
                        break
                ligne = id.lower() + ';' + nom + ';' + prenom + ';' +
telephone + ';' + mdp + ';' + email + ';' + lieu + ';' + categorie + '\n'
                ids.append(id)
                fichierInsertion.write(ligne.encode())

ids = []
with open('fichierInsertionReel.csv', 'wb') as fichierInsertion:
    ligneEntete = 'identifiant;nom de famille;prénom;téléphone;mot de
passe;email;lieu;catégorie\n'
    fichierInsertion.write(ligneEntete.encode())
    for fichier in os.listdir('fichiers communes'):
        miseEnFormeListeAgents('fichiers communes/' + fichier)
```

Une fois tous les fichiers réceptionnés, je les ai insérés dans GLPI avec le plug-in

Echanges avec la police rurale

Lors du passage d'un agent de la police rurale pour un autre sujet, j'ai pu lui poser des questions par rapport à leur fonctionnement actuel des demandes de police rurale. Il m'a donc expliqué qu'ils utilisaient un formulaire Google et qu'ensuite, la responsable du service leur attribuait les missions demandées par les communes.

J'ai donc essayé de modifier le formulaire de création de tickets avec le plug-in forms. Cependant, il était impossible d'adapter les champs en fonction de la catégorie du ticket. J'ai donc cherché une autre solution et je me suis penché sur le plug-in formCreator qui me permettait de créer plusieurs formulaires (un pour chaque catégorie des incidents informatiques et des demandes de police rurale) j'ai donc pu créer des formulaires de tickets personnalisés en fonction des catégories

Je lui ai ensuite demandé quels utilisateurs étaient habilités à faire des demandes auprès de leur service. Les personnes habilitées sont :

- les Maires
- certains adjoints de mairie
- les secrétaires
- les directeurs généraux

Au vu du peu de personnel habilité à faire des demandes, nous avons décidé de créer des comptes génériques pour les communes afin que toutes les demandes auprès de la commune soient regroupées dans un même compte.

Pour créer ces comptes génériques, j'ai créé un fichier d'insertion .csv

Identifiant; nom commun; téléphone; mot de passe; email; catégorie
--

J'ai ensuite écrit un nouveau script python pour alimenter le fichier d'insertion

```
import re
from faker import Faker
def ajouter_espace_avant_majuscules(s):
    return re.sub(r"(<|^)([A-Z])", r" \1", s)

communes = [ ... ]

telephones = [ ... ]

emails = [ ... ]

with open('fichierInsertionCompteGénérique.csv', 'w', encoding="utf-8") as
fichierInsertion:
    ligneEntete = 'identifiant;nom commune;téléphone;mot de passe;email;catégorie\n'
    fichierInsertion.write(ligneEntete)
    for commune in communes:
        id = commune.replace(" ", "").replace("-", "")
        nom = ajouter_espace_avant_majuscules(commune.replace("-", ""))
        telephone = telephones[communes.index(commune)].replace(" ", ".")
        mdp = Faker().password(length=8, special_chars=True, digits=True,
upper_case=True, lower_case=True)
        email = emails[communes.index(commune)]
        categorie = "Compte-Générique-PoliceRurale"
        ligne = id.lower() + ';' + nom + ';' + telephone + ';' + mdp + ';' + email + ';'
+ categorie + '\n'
        fichierInsertion.write(ligne)
```

Définition des profils de la police rurale et séparation des entités

Nous avons également défini les rôles des membres du service de police rurale :

- Responsable : le rôle du responsable est de réceptionner les tickets entrants. Lorsqu'un ticket arrive, il vérifie les disponibilités des agents et attribue la mission à l'agent dont les compétences sont les plus adaptées pour mener à bien la mission. Il doit donc pouvoir voir tous les tickets à destination de la police rurale et pouvoir les attribuer à un agent précis
- Agent: le rôle de l'agent est d'effectuer sur le terrain les missions que son responsable lui a attribuées. Il doit voir seulement les tickets qui lui sont attribués. Lorsqu'il estime avoir complété sa mission, il doit proposer une solution à l'utilisateur qui va clore ou non le ticket.

Avec les différents besoins qui m'ont été énoncés, j'ai décidé d'utiliser la fonctionnalité qui permet de créer des entités et j'ai donc créé 2 entités pour cloisonner ces 2 services. J'ai assigné tous les comptes utilisateurs à des entités pour leur empêcher d'accéder aux services qui ne leur sont pas dédiés

	entité Informatique	entité Police Rurale
Super Admin	V	V
Technicien Informatique	V	X
Responsable Police	X	V
"Police agent"	X	V
Compte Utilisateur Nominatif	V	X
"Compte Générique Commune	X	V

Rédaction des guides utilisateurs

Une fois l'application configurée , j'ai rédigé quatre guides pour les utilisateurs :

- Guide du demandeur informatique (annexe 1)
- Guide du demandeur de la police rurale (annexe 2)
- Guide Agent de Police Rurale (Annexe 3)
- Guide responsable police rurale (annexe 4)

Ces guides ont pour but d'accompagner les utilisateurs afin qu'ils utilisent l'application sans difficulté . J'y explique par exemple comment accéder au site et se connecter à l'application , comment ouvrir un ticket ou encore comment proposer une solution . Je les ai donc rédigés en suivant le modèle des autres guides de l'entreprise puis les ai transmis au service communication afin de les adapter à la charte graphique de l'établissement

Conclusion du projet 1

J'ai donc réussi, au fil du stage, à améliorer la configuration de l'application pour rendre un produit complet qui répond à tous les besoins qui ont été énoncés par les différents utilisateurs. Grâce à cette mission, j'ai pu développer les compétences suivantes :

- Administrer des systèmes informatiques communicants complexes , notamment avec l'installation de la machine virtuelle et son déploiement sur le réseau, l'application des règles sur le firewall, la mise en place d'un certificat SSL et les demandes liés au registrar pour enregistrer le nom de domaine
- Gérer des données de l'information, par exemple, en récupérant les données des utilisateurs des communes et en les mettant en forme pour pouvoir les exploiter.