

7. Livestream

Ce système de surveillance vidéo en direct a été conçu pour permettre une supervision visuelle à distance, bien que pour l'instant, la diffusion soit limitée au réseau local, ce qui nécessite d'être physiquement présent à l'université pour y accéder.

Le système repose sur l'utilisation de FFmpeg, un outil de traitement vidéo, pour capturer et diffuser en direct la vidéo provenant d'un périphérique vidéo, comme une caméra. Le flux vidéo est enregistré localement et peut être consulté en temps réel via un serveur HTTP, accessible depuis un navigateur web. La diffusion en direct est basée sur le protocole HLS (HTTP Live Streaming), couramment utilisé pour le streaming vidéo. Le flux vidéo est découpé en segments courts au format .ts (Transport Stream), qui sont enregistrés en boucle avec un nombre limité de fichiers (ici, trois fichiers). Lorsqu'un nouveau segment est enregistré, le plus ancien est écrasé, garantissant que seuls les segments les plus récents sont conservés. Cette méthode optimise l'utilisation du stockage tout en maintenant un flux vidéo continu. Le fichier .m3u8, quant à lui, est un fichier de playlist utilisé par le protocole HLS. Il contient la liste des segments vidéo .ts disponibles pour la lecture et est mis à jour dynamiquement pour refléter les segments vidéo les plus récents, assurant une lecture fluide et continue du flux vidéo. Cette méthode offre une latence réduite, le flux vidéo étant presque en temps réel, avec un léger délai dû à la durée des segments (environ 10 secondes dans notre cas). Enfin, elle est compatible avec les navigateurs modernes et les appareils mobiles, facilitant l'accès au flux vidéo depuis différents dispositifs. En parallèle de la diffusion en direct de vidéo, le système capture des images à intervalles réguliers, intervalles définis dans un fichier de configuration permettant de les modifier facilement. Ces images serviront par la suite à créer des time-lapses qui montrent l'évolution des plantes sous l'ombrière.

En pratique, les utilisateurs peuvent accéder au flux vidéo en direct via un navigateur web en se connectant au serveur HTTP, à condition d'être sur le même réseau que le serveur puisque ce dernier n'est pas déployé. Cela permet de surveiller l'ombrière en temps réel et de réagir rapidement en cas de problème, comme des dégâts visibles ou des anomalies. Les images capturées à intervalles réguliers sont sauvegardées dans un répertoire spécifique, et leur chemin d'accès est envoyé dans une base de données à l'aide de requêtes SQL. L'envoi des chemins d'accès dans la base de données permet de vérifier rapidement le bon fonctionnement du script de capture d'image, sans avoir besoin de se connecter directement à la machine hébergeant le serveur. Étant donné que les noms des images capturées incluent la date et l'heure de capture, la présence de ces chemins d'accès dans la base de données permet de confirmer que les intervalles de capture respectent bien les paramètres définis. Cela offre un moyen simple et efficace de valider le processus de capture à distance.

Le fonctionnement du système repose sur plusieurs composants clés. Le script Python (*server.py*) orchestre l'ensemble du processus. Le choix d'utiliser un script Python plutôt qu'un script PHP pour la création du Livestream est principalement motivé par la gestion du CORS (Cross-Origin Resource Sharing). L'activation de CORS sur un serveur est essentielle pour autoriser des ressources d'un site web (comme des scripts) à être demandées depuis un domaine différent de celui d'où le site est servi. Bien que plusieurs solutions techniques existaient pour implémenter le CORS, Python a été privilégié en raison de sa simplicité de mise en œuvre côté serveur. Lors du lancement du script Python, la configuration du système est chargée à partir d'un fichier JSON (*config.json*), qui contient des paramètres tels que le périphérique vidéo (*/dev/video0*), le format vidéo (640x480), les répertoires de sortie (*stream* et *images*), et les

intervalles de capture d'images (image_period). Le script Python démarre également les scripts Bash (stream.sh et acquireimg.sh) pour gérer respectivement le streaming vidéo et la capture d'images. Le script stream.sh utilise FFmpeg pour diffuser en direct la vidéo via le protocole HLS, tandis que acquireimg.sh capture les images à partir des segments vidéo .ts et les sauvegarde dans un répertoire spécifique. Enfin, le script Python envoie le chemin de ces images dans la base de données via des commandes SQL pour un suivi structuré.

Le système est configuré pour fonctionner sur des systèmes Linux, comme en témoigne l'utilisation de scripts Bash (stream.sh, acquireimg.sh, et dimw.sh) mais a aussi été porté sur Windows en mettant ces scripts au format Bat. Lors du premier lancement du serveur sur le LattePanda, aucune vidéo n'était enregistrée. Ce problème était lié à l'utilisation de FFmpeg, qui ne parvenait pas à lancer l'enregistrement. Initialement, l'hypothèse retenue pointait une incompatibilité entre le LattePanda et la caméra utilisée pour les tests. Cependant, en utilisant la commande ffplay, la diffusion de la caméra fonctionnait correctement, écartant cette première piste. Une seconde hypothèse a alors été envisagée : le problème pourrait provenir des options utilisées dans la commande FFmpeg. Après plusieurs tests, une solution a été trouvée : pour la première utilisation, il faut créer à l'aide de FFmpeg un fichier .mp4 initial. Une fois cette étape réalisée, la commande pour l'enregistrement de la vidéo pour le Livestream fonctionne sans problème.

Pour garantir un fonctionnement constant sans intervention manuelle, le script dimw.sh a été développé. Ce script résout un problème lié à l'enregistrement en capturant une très courte séquence vidéo au format .mp4, puis en la supprimant immédiatement. Cette opération permet de s'assurer que le périphérique vidéo est correctement initialisé avant de démarrer le streaming, évitant ainsi tout dysfonctionnement ultérieur.

8. Mise en service LattePanda

La mise en service du système sur la LattePanda a été une étape cruciale pour garantir le bon fonctionnement de l'ensemble des composants, incluant la base de données MySQL, le serveur web, l'API météo, et le Livestream. Cette section décrit les étapes clés de cette mise en service, en mettant l'accent sur l'ouverture des ports, l'installation des logiciels nécessaires, la configuration de phpMyAdmin, et l'accès distant à la base de données.

8.1 Ouverture des ports

Pour permettre au système de communiquer avec les autres appareils du réseau, plusieurs ports ont dû être ouverts sur la LattePanda. Le port 3306 a été ouvert pour MySQL, permettant ainsi aux autres ordinateurs du réseau de se connecter à la base de données. Le port 80 a été ouvert pour le serveur web, hébergeant l'interface utilisateur, phpMyAdmin, et l'API météo. Enfin, le port 20 a été ouvert pour des besoins spécifiques, tels que les transferts de fichiers via FTP. Ces ports ont été configurés à l'aide du pare-feu *ufw* sous Linux, avec des commandes spécifiques pour autoriser les connexions entrantes sur ces ports.

```
sudo ufw allow 3306/tcp # Autoriser le port 3306 pour MySQL
```

8.2 Installation des logiciels nécessaires

Plusieurs logiciels ont été installés sur la LattePanda pour assurer le bon fonctionnement du système. MySQL a été installé pour héberger les bases de données « Optiplant » et « Comptes

Utilisateurs », tandis que phpMyAdmin a été configuré pour offrir une interface graphique de gestion de la base de données. Apache a été installé pour servir de serveur web, hébergeant l'interface utilisateur et l'API météo. Python a été utilisé pour exécuter les scripts backend, y compris le serveur Flask pour le Livestream. FFmpeg a été installé pour la capture et le streaming vidéo, et Git a été utilisé pour cloner et mettre à jour les dépôts de code. Ces installations ont été réalisées via des commandes simples sous Linux, garantissant une configuration rapide et efficace.

```
sudo apt install mysql-server phpmyadmin apache2 python3 python3-pip ffmpeg git
```

8.3 Configuration de phpMyAdmin

Une fois phpMyAdmin installé, plusieurs étapes de configuration ont été nécessaires pour le rendre accessible et sécurisé. phpMyAdmin a été configuré pour être accessible via le navigateur à l'adresse <http://192.168.XXX.XXX/phpmyadmin>. Un utilisateur dédié a été créé dans MySQL pour phpMyAdmin, avec des permissions restreintes pour éviter les accès non autorisés. Le fichier de configuration d'Apache a également été modifié pour permettre l'accès à phpMyAdmin depuis le réseau local. Ces modifications ont permis de garantir une interface de gestion de la base de données à la fois accessible et sécurisée.

8.4 Configuration de MySQL pour l'accès distant

Pour permettre à d'autres ordinateurs du réseau d'accéder à la base de données MySQL, plusieurs configurations ont été appliquées. Le fichier de configuration MySQL (/etc/mysql/mysql.conf.d/mysqld.cnf) a été modifié pour écouter sur toutes les interfaces réseau, et non seulement sur *localhost*.

```
bind-address = 0.0.0.0
```

Un utilisateur MySQL avec des permissions d'accès distant a été créé, permettant aux autres ordinateurs du réseau de se connecter à la base de données. Ces modifications ont été suivies d'un redémarrage du service MySQL pour appliquer les changements. Cette configuration a permis une collaboration efficace entre les différents utilisateurs du réseau.

```
CREATE USER 'remote_user'@'adresse_ip_pc' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON Optiplant.* TO 'remote_user'@'adresse_ip_pc';
FLUSH PRIVILEGES ;
```

8.5 Accès à la LattePanda depuis le réseau local

Pour permettre aux autres ordinateurs du réseau d'accéder à la LattePanda, l'adresse IP de la LattePanda a été partagée avec les utilisateurs. Les utilisateurs peuvent accéder à l'interface utilisateur, à phpMyAdmin, et au Livestream via cette adresse IP. Cette configuration a permis une utilisation collaborative du système, avec un accès centralisé à tous les services.

Il est à noter que pendant la phase d'installation du LattePanda, il a fallu utiliser un autre réseau que le réseau filaire de l'université. En effet, le proxy n'est pas configuré pour réaliser ce genre de tâche. De plus, il n'est pas envisageable de se connecter avec un compte personnel sur le proxy, pour des raisons évidentes de protection des données.

Cependant, dans le cadre d'une utilisation normale, il est possible de connecter le server au réseau filaire de l'université dans la mesure où le serveur n'effectue pas de requête sortante, se contentant de répondre aux requêtes entrantes.

8.5.1 Connexion à l'IHM

À la suite de la création d'une connexion spécifique pour l'IHM, définie dans la base de données Optiplant, il a été possible via CODESYS d'ouvrir une connexion sur la base de données à partir de l'IHM. Des requêtes d'insertion sur la base de données et des requêtes de sélection d'informations ont été vérifiées.

8.6 Documentation et support

Pour faciliter la maintenance et l'utilisation du système, une documentation détaillée a été rédigée. Cette documentation inclut les étapes d'installation et de configuration, les commandes utiles pour gérer les services, et les informations de connexion pour la base de données et phpMyAdmin. Cette documentation a été partagée avec les utilisateurs pour leur permettre de résoudre les problèmes courants et de configurer leur environnement si nécessaire.