

Rapport intermédiaire projet Supervision

Bacchetta - Perraud

Introduction :

Dans le cadre de notre parcours en Master 2 Réseaux et Télécoms, nous avons entrepris un projet de développement d'un système de supervision. Ce rapport intermédiaire vise à présenter les technologies prévues pour ce projet, à faire un état d'avancement des réalisations à ce jour, à évaluer notre situation par rapport au planning initial, et à fournir quelques schémas illustratifs.

Module de configuration :

Nous avons choisi d'utiliser une base de données **PostgreSQL** pour la gestion des informations de configuration. Dans cette base de données, nous allons mettre en place la fonctionnalité permettant d'ajouter des machines en spécifiant des détails tels que la description, le type de matériel, l'adresse IP, les intervalles de surveillance et le type d'information à surveiller (comme le nombre d'octets ou les erreurs).

De plus, nous allons implémenter la possibilité de supprimer des machines et de stocker les journaux dans la base de données. Pour faciliter le déploiement et la configuration de notre système, nous avons utilisé **Docker** pour regrouper plusieurs applications, y compris la base de données, la page web, la supervision et les scripts. Cela nous permet d'automatiser le lancement de ces applications et de les configurer de manière préétablie.

Module de surveillance :

Nous avons choisi d'utiliser le protocole **SNMP** (Simple Network Management Protocol) pour la surveillance en temps réel de nos équipements.

Pour cela, nous allons utiliser la bibliothèque **PySNMP** en Python, qui nous permet d'interroger périodiquement les équipements et de récupérer les informations pertinentes.

Nous allons également développer un script automatisé de demande SNMP, qui nous permet de stocker les informations récupérées dans notre base de données pour une analyse et affichage ultérieur.

Module de Log :

Nous prévoyons de développer une page web qui affichera les résultats du module de surveillance sous forme de graphiques.

Pour cela, nous utiliserons les frameworks **Flask** et **Plotly**, ainsi que les langages **HTML**, **JSON**, **PHP** et **JavaScript**.

Cette page web proposera également une interface d'accueil avec une authentification des comptes.

Nous créerons un compte administrateur avec des droits étendus et des comptes lambdas avec des droits en lecture seule. La page web affichera une liste des machines réelles qui seront surveillées. Les fonctionnalités disponibles incluront l'ajout, la modification et la suppression de machines, ainsi que l'accès aux journaux de chaque machine. De plus, nous mettrons en place un système de couleurs pour indiquer l'état de chaque machine.

USE CASES

Module de Configuration

Ajouter / Supprimer un Élément

****Acteurs **** Administrateur

****Scénario ****

L'administrateur sélectionne le type de matériel.

L'administrateur entre les informations requises pour le matériel.

Le système valide les informations et ajoute / supprime le matériel à la liste des éléments

Modifier un Élément :

****Acteurs **: **** Administrateur

****Scénario **: ****

L'administrateur sélectionne un élément à modifier.

L'administrateur modifie les informations.

Le système valide et met à jour les informations.

Module de Surveillance

Surveiller un Élément

****Acteurs **: **** Système, Équipement Réseau

****Scénario **: ****

Le système envoie une requête SNMP à l'équipement réseau.

L'équipement réseau répond aux informations requises.

Le système affiche les informations de manière textuelle et graphique.

Notifier une Défaillance :

****Scénario **: ****

Le système détecte une défaillance.

Le système envoie une notification à l'administrateur.

L'administrateur consulte les détails de la défaillance.

Module de Log

Logger un Évènement :

****Acteurs :**** Système

****Scénario :****

Un événement (information, erreur) se produit.

Le système enregistre l'événement dans le fichier ou la base de données de log.

Consulter les Logs :

****Acteurs :**** Administrateur

****Scénario :****

L'administrateur ouvre l'interface de log.

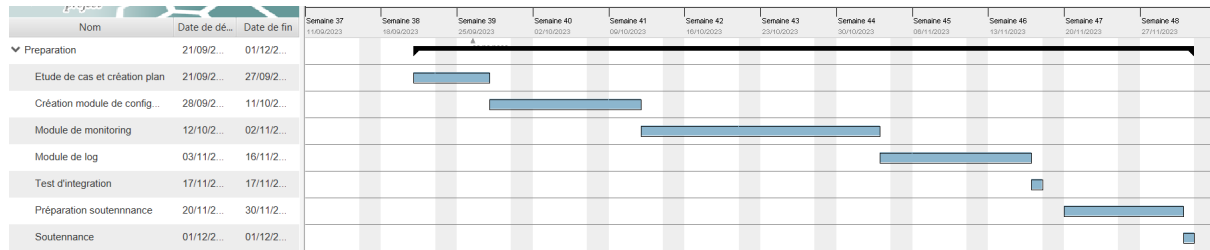
L'administrateur consulte les événements enregistrés.

Voici un schéma de l'utilisation supposée du système de supervision



ÉTAT ACTUEL DU PROJET

Un point sur l'état d'avancement du projet :



Si l'on suit le diagramme de Gantt, nous sommes en bonne piste pour finir le projet :

Le module de configuration est terminé :

- Docker est en place et les conteneurs sont fonctionnels
- Les machines virtuelles sont déployées sur Azure

Les module de supervision et de log sont en développement, ce qui nous rend en avance sur le planning

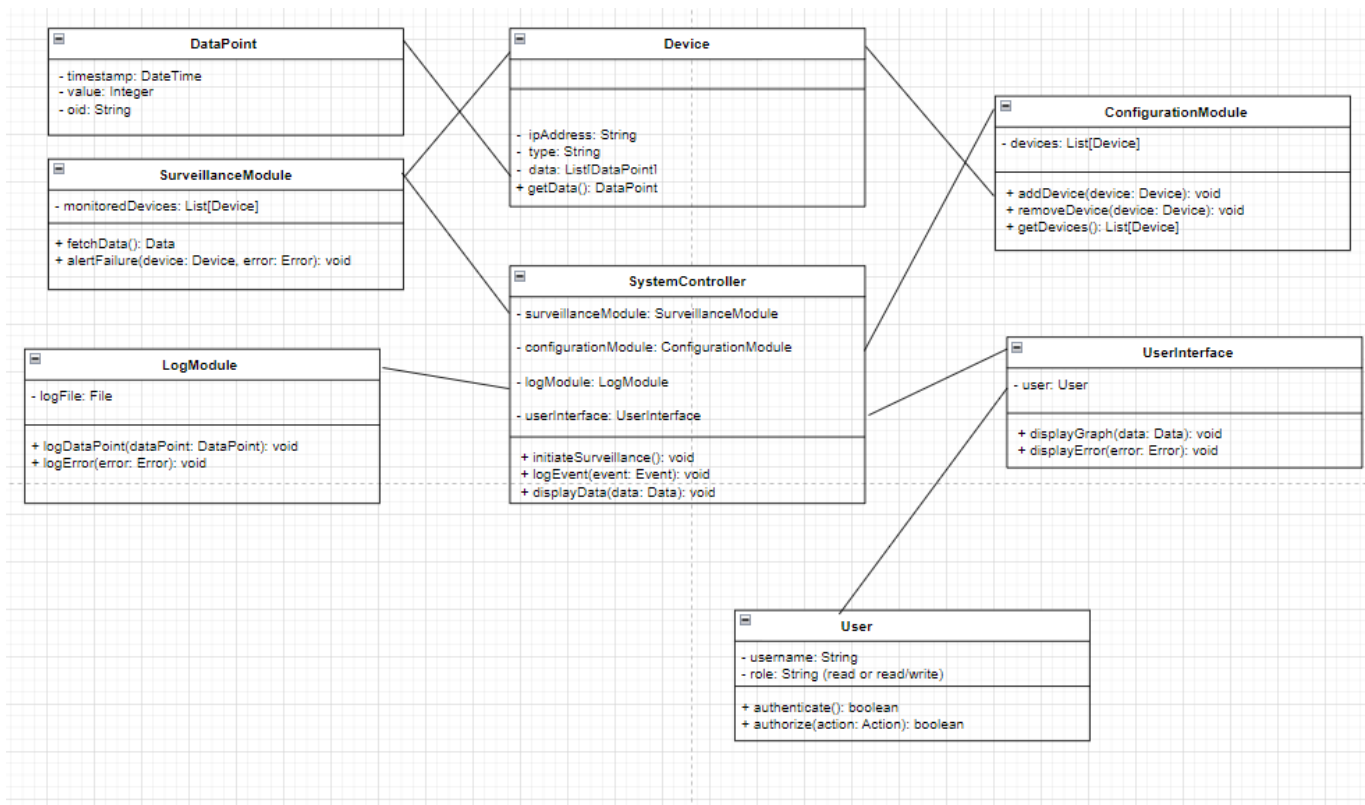
- Le site web est lancé et il est possible de lancer des requêtes snmps de base
- Une liste de machines, d'OIDs et d'actions sont codées en dur sur le site pour faire les tests de fonctionnement

Ce qu'il nous reste à faire :

- La partie front-end : affichage des résultats et appels des fonctions d'administration présentes (ajout de serveurs, modification pour un serveur, retrait de serveur), graphiques
- Gestion de l'authentification sur les clients graphiques et séparation des accès entre simple utilisateur et administrateur.
- Gestion des alertes
- Populer la BDD → Lier la BDD au site Web via PHP

Diagramme de classes

Voici notre diagramme de classe, vous trouverez une représentation visuelle des classes principales de notre système, leurs attributs, leurs méthodes ainsi que les relations entre elles. L'objectif de ce diagramme est de fournir une vue d'ensemble claire et concise de la conception de notre application.



<https://github.com/Bachett/PRJSUPBACCPERR/>