

Synthèse de projet - Système de supervision de VPS

Introduction :

Dans le cadre de mon stage, j'ai pour objectif de concevoir et développer un système de supervision destiné aux étudiants utilisant des VPS. Ce système vise à surveiller l'état des machines virtuelles, détecter les anomalies de performance et alerter les utilisateurs afin de garantir un bon fonctionnement de leur environnement de travail.

Objectifs du projet :

Le but principal est de mettre en place une solution capable de :

- Collecter des informations système (CPU, RAM, stockage, etc.)
- Identifier les dégradations de performance
- Alerter les utilisateurs en cas de problème
- Proposer des actions correctives simples

L'objectif est également de rendre cette solution accessible via une interface web intuitive.

Outils utilisés :

Afin de répondre aux besoins du projet, plusieurs outils complémentaires seront mis en œuvre. Le protocole SNMP (Simple Network Management Protocol) sera utilisé pour la collecte des données système des VPS à distance, permettant d'obtenir des informations essentielles telles que l'utilisation du processeur, de la mémoire ou du stockage. Le langage Python sera employé pour le développement du backend ainsi que pour la création des scripts de monitoring, grâce à sa simplicité et sa richesse de script sur GITHUB. Le framework Flask viendra compléter cet environnement en permettant la mise en place d'une API assurant la communication entre le frontend et les différents scripts. Mais je vais aussi faire une version avec NODE.js pour remplacer la partie avec FLASK se qui vas me permettre de gérer les requêtes du frontend et exécuter des scripts Python.

Par ailleurs, l'interface utilisateur sera développée avec HTML, CSS et JavaScript, afin de concevoir un tableau de bord interactif et le plus ergonomique possible. Enfin, l'ensemble de l'application seras hébergé sur un VPS qu'on m'a fourni. Afin de centraliser les données et de garantir l'accessibilité du système j'ai choisi ces méthodes.

Lien entre les outils :

Le fonctionnement du système repose sur une architecture de type client-serveur dans laquelle chaque composant joue un rôle bien défini. L'interface web, constituant le frontend se qui permet à l'utilisateur de visualiser les informations de supervision et d'interagir avec le système, notamment en activant ou désactivant des scripts ou en en créant de nouveaux.

De son côté, le backend développé avec Flask ou node.js agit comme un intermédiaire entre l'interface utilisateur et les éléments techniques. Il reçoit les requêtes envoyées par le frontend, les traite, puis déclenche les actions nécessaires. Les scripts de monitoring, écrits en Python, sont responsables de la récupération des données système sur les VPS via le protocole SNMP.

Les informations collectées sont ensuite transmises au backend, qui les traite et les renvoie au frontend afin d'être affichées sur le tableau de bord. Ceci permet d'assurer une communication

Synthèse de projet - Système de supervision de VPS

fluide entre les différentes parties du système, garantissant ainsi son bon fonctionnement global. Pour donner un exemple : A la création d'un script sur le frontend le backend doit créer le dossier de création du script pour ensuite le rendre activable ou désactivable.

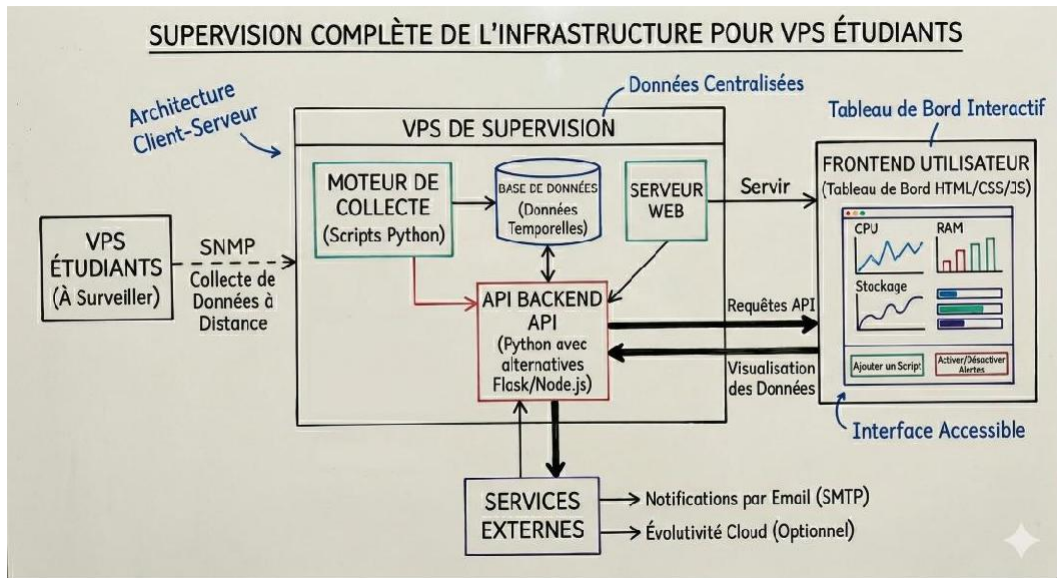


Schéma généré par intelligence artificielle avec l'aide d'un schéma fais sur paint

Conclusion et objectif final :

À terme, ce projet a pour ambition de fournir un outil complet de supervision permettant aux utilisateurs de gérer efficacement leurs VPS de manière plus autonome en fournissant une aide simplifiée à l'administrateur. L'objectif est de proposer une interface à la fois simple, claire et fonctionnelle, tout en intégrant des fonctionnalités avancées de monitoring.

Le système final devra permettre une surveillance en temps réel des ressources, offrir la possibilité de gérer des scripts personnalisés et contribuer à l'amélioration des performances des VPS grâce à la mise en place d'alertes pertinentes. Il s'agira donc d'un outil à la fois pédagogique et pratique, adapté aux besoins des étudiants.

Évolutions et améliorations possibles :

Dans une perspective d'amélioration continue, plusieurs évolutions pourront être envisagées afin d'enrichir les fonctionnalités du système. Il sera notamment possible d'intégrer des mécanismes de notification, comme l'envoi d'emails, afin d'alerter les utilisateurs en cas de problème.

De plus, l'ajout de graphiques avancés offrira une meilleure visualisation de l'évolution des performances dans le temps. Une automatisation plus poussée des actions correctives pourrait également être envisagée afin de rendre le système encore plus autonome. Enfin, un déploiement sur une infrastructure cloud permettrait d'adapter la solution à un plus grand nombre d'utilisateurs et d'améliorer sa robustesse.

Synthèse de projet - Système de supervision de VPS

Apprentissage Critique (AC)	Description (BUT R&T Cybersécurité)	Application dans ton projet
AC1 - Sécuriser un système et un réseau	Identifier les vulnérabilités et mettre en place des mécanismes de protection (authentification, chiffrement, contrôle d'accès)	Sécurisation des échanges SNMP (v3), protection du backend (Flask / Node.js), sécurisation du VPS (firewall, SSH)
AC2 - Superviser et analyser un système	Collecter, analyser et interpréter des données de performance pour détecter des anomalies	Monitoring CPU, RAM, stockage, mise en place d'alertes en cas de surcharge ou panne
AC3 - Automatiser l'administration système	Développer des scripts pour automatiser des tâches de gestion et de maintenance	Scripts Python pour récupérer les données SNMP et analyser les performances
AC4 - Développer une application réseau	Concevoir une application communicante avec une architecture client-serveur	Création d'un dashboard web (HTML/CSS/JS) relié à une API Flask ou Node.js
AC5 - Concevoir une architecture sécurisée	Structurer un système complet en intégrant les contraintes de performance et de sécurité	Mise en place d'une architecture frontend / backend / scripts / VPS supervisés
AC6 - Gérer les flux de données	Assurer la circulation fiable et efficace des données entre les composants d'un système	Communication entre frontend → backend → scripts → VPS via API et SNMP
AC7 - Déployer et maintenir un service	Installer, configurer et maintenir un service en conditions réelles	Hébergement de l'application sur VPS, gestion de la disponibilité du service
AC8 - Améliorer un système (DevSecOps)	Proposer des évolutions en intégrant sécurité, performance et automatisation	Ajout de notifications (email), graphiques, automatisation des actions correctives
AC9 - Assurer la résilience et la continuité	Garantir le bon fonctionnement même en cas de problème	Mise en place d'alertes pour anticiper les pannes et éviter les interruptions
AC10 - Sensibiliser à la cybersécurité	Comprendre les enjeux et appliquer les bonnes pratiques	Prise en compte des risques liés à l'exposition des VPS et des API