

Cahier des charges - Application de collecte d'informations sur les machines d'infrastructure

Introduction

Le présent cahier des charges décrit les spécifications du projet visant à développer une application Python dédiée à la collecte d'informations sur les machines d'une infrastructure, incluant les réseaux, serveurs, machines virtuelles, etc. Ce projet sera réalisé en suivant les méthodes agiles pour assurer un développement itératif et collaboratif.

Objectifs

L'application doit permettre la collecte détaillée d'informations sur les machines au sein d'une infrastructure, en mettant l'accent sur la sécurité (cybersécurité) et l'efficacité opérationnelle (DevOps).

Fonctionnalités

Socle de l'application

1. Interface Utilisateur (UI) :

- Tableau de bord intuitif pour visualiser les informations collectées.
- Affichage clair des statistiques clés.
- Framework **Flask** privilégié

2. Collecte d'Informations :

- Mise en place d'un système robuste pour collecter des informations sur les ressources système, configurations, etc.

3. Compatibilité :

- Prise en charge de différents systèmes d'exploitation UNIX.
- Facilité d'intégration avec d'autres outils de supervision.

Fonctionnalités Cybersécurité (options pour parcours Cyber)

1. Audit de Sécurité :

- Collecte d'informations spécifiques pour effectuer un audit de sécurité.

2. Détection d'Anomalies :

- Fonctionnalité permettant de détecter des anomalies dans les comportements des machines.

3. Analyse de Vulnérabilités :

- Collecte d'informations pour identifier les vulnérabilités potentielles.

4. Surveillance des Certificats SSL/TLS :

- Informations sur les certificats SSL/TLS pour assurer une communication sécurisée.

5. Gestion des Politiques de Sécurité :

- Collecte d'informations pour évaluer la conformité aux politiques de sécurité établies.

6. Journal d'Audit :

- Enregistrement d'un journal d'audit pour une traçabilité complète des activités.

Fonctionnalités DevOps (options pour parcours DevCloud)

1. Collecte de Logs d'Application :

- Collecte d'informations liées aux logs d'application pour faciliter le débogage.

2. Intégration Continue/Déploiement Continu (CI/CD) :

- Informations nécessaires pour soutenir des pratiques CI/CD.

3. Surveillance des Performances :

- Collecte d'informations sur les performances des machines.

4. Gestion des Ressources Système :

- Informations relatives à l'utilisation des ressources système.

Méthodologie de Développement

Le développement de l'application suivra les méthodes agiles, avec des itérations courtes, des revues régulières et une flexibilité pour l'ajout de fonctionnalités émergentes.

La durée de l'itération sera définie par l'équipe suivant ses capacité de travail (**4 membres dédiés au projet**) et d'une charge linéaire pour l'accomplissement du projet.

Livrables Attendus

Les livrables techniques devront être déposés dans un dépôt Git (code, tests, documentation) accessible au product owner.

1. Product backlog (**muable**).
2. Sprint backlog (**2 itérations minimum, découpage en tâches et estimations**).
3. Code source documenté.
4. Manuel d'utilisation.
5. Rapport de tests.
6. Présentation détaillée du projet.
7. Liste des user stories réalisées en fin de sprint (**% de complétion de l'objectif initial**).

Plateforme de test

L'entreprise ne dispose pas d'infrastructure dédiée pour ce projet. Dans le cadre de cette réalisation, le projet sera considéré comme une version **MVP (minimum valuable product)** et sera donc fonctionnel

sur un environnement restreint. L'équipe en charge de la réalisation définira et réalisera les scénarios de démonstration en se basant sur des machines virtuelles. Une architecture "client-serveur" sera mise en place nécessitant au minimum **2 VMs**.

Calendrier

Le projet sera divisé en deux itérations minimum, avec des revues à la fin de chaque itération pour évaluer les progrès et ajuster les priorités.

La préparation des itérations doit être réalisée en accord avec le product owner (FR Menguy).

L'équipe en charge du projet devra **communiquer de manière régulière** avec le product owner sur l'avancement des itérations.

- Début du projet : **mardi 12 décembre 2023**
- Démonstration itération 1 : Le product owner ne pourra pas être présent à celle-ci. `Un rapport de fin d'itération devra être produit et livré par mail`.
- Démonstration itération 2 : **mardi 23 décembre 2023**
- Démonstration des itérations suivantes à planifier suivant le rythme des itérations.

NB: le product owner sera présent sur site :

- 2h le mardi 12 décembre, + 2h (options cyber)
- 2h le jeudi 14 (option devops)
- 4h le jeudi 21 (option cyber)
- 4h le vendredi 22 (option devops)
- 2x 4h le mardi 23

Des temps de **présence obligatoire** pour l'équipe sont également ajouter dans vos agendas afin d'assurer une continuité de mission agile (volume horaire : **16 heures**).

Critères d'évaluation

- Maturité de l'application de la méthode de gestion de projet
- Qualité de code
- Utilisation efficiente de Git
- Collaboration sur le projet (ex: développement des tests en parallèle du développement fonctionnel), horodatage des commits Git pris en compte dans l'évaluation de ce critère
- Utilisation des séances d'autonomie pour avancer sur le projet par groupe (ex: partage de connaissances en binôme, planification, estimation des tâches ...)
- Utiliser un outil type **Trello** ou **WeKan** pour le suivi d'avancement du projet. Donner accès au PO à cet espace.
- Ensemble fonctionnel (% d'avancement par rapport à l'estimation initiale)
- Démonstration de l'application