**Développement d’un système de surveillance et de contrôle de matériel informatique géré par le biais d’une interface Web.**

SAE 51 : Concevoir, réaliser et présenter une solution technique

Rédigé par : Maxime VALLET

**I) Introduction**

**1) Présentation de l’application**

En entreprise, les services hébergés sur les serveurs sont essentiels au bon fonctionnement de celle-ci. Il est donc crucial de posséder un système dont l’objectif est de suivre en continu l’état des serveurs ainsi que du matériel qui achemine les données qui leur sont associées.

L’objectif principal de ce type de système est de détecter, voire d’anticiper les soucis informatiques ou réseau qui pourraient provoquer une interruption du service et ce le plus tôt possible, dans l’optique de les régler afin de diminuer leur impact et d’améliorer le temps de disponibilité.

**2) Interlocuteur et apports**

Afin de dimensionner ce projet, je me suis appuyé sur les conseils de M. Beysson, mon expérience acquise sur la réalisation de précédents projets ainsi que des ressources abordées au cours des années précédentes.

M. Beysson m’a particulièrement conseillé sur la partie interface Web et développement Java. Il m’a notamment donné son avis sur :

- La façon dont il faut réaliser un système d’authentification dans un projet avec les contraintes qui sont nommées par la suite.

- Le chiffrement des données échangées entre un serveur et un client réalisé avec Java.

- La gestion de la latence entre les machines surveillées et l’interface Web (suivi en direct sur la page Web)

- La façon la plus adaptée de récupérer les données sur les machines surveillées.

- Le dimensionnement du projet.

J’ai également utilisé mon expérience de mes projets précédents afin de définir les fonctionnalités implémentées ainsi que les outils utilisés.

Parmi les projets qui m’ont été utiles on retrouve :

- La SAE 52 : j’ai réalisé le backend de ce projet, ce qui m’a permis de m'aiguiller sur la façon dont il faut procéder afin de réaliser un projet similaire. J’ai notamment remarqué que plusieurs points (VM, certificats SSL auto-signés, documentation…) nécessitent un travail plus approfondi afin d’améliorer la coordination entre les membres du groupe et la qualité du projet, ce qui m’a permi de me mieux situer les points à souligner au cours de ce nouveau projet.

- Script de report de configuration et d’utilisation d’un ordinateur (PowerShell) : ce projet m’a permis d’avoir un aperçu de la façon dont on utilise les API windows (WMI) et les contraintes liées à la récupération de ces informations.

Parmi les ressources sur lesquelles je me suis appuyé et qui me serviront à la réalisation de ce projet, on retrouve :

- Les bases de données : R207, R310

- Servlets Java : SAE 23

- Java : R208, R308, R309, R504

- Développement Web : R109, R209

- Bash : R108, R405

- Fonctionnement Git : R409, SAE 52

- Supervision du matériel réseau SNMP : R502

- Gestion de projet : R115

**II) Besoins et contraintes**

**1) Fonctionnalités**

**a) Interface Web**

Interface permettant aux utilisateurs d'utiliser les fonctionnalités suivantes.

**b) Allumage et extinction des machines surveillées**

Control du statut des machines à partir de l’interface web. Cette action nécessitera de posséder les droits sur la machine.

**c) Surveillance de la machine en direct via l’interface Web**

Cette fonctionnalité possédera deux onglets :

* la configuration de la machine suivie possédant ces informations :

| CPU | RAM | Stockage | Réseau | Session |
| --- | --- | --- | --- | --- |
| modèle | quantité | disques | interfaces | version windows |
| nb coeurs/threads | nombre DIMM | taille | vitesse |  |
| fréquence max | vitesse |  |  |  |

* la visualisation en direct plusieur métriques en direct comme :

| CPU | RAM | Stockage | Réseau | Ventilateurs |
| --- | --- | --- | --- | --- |
| charge | utilisation | charge | latence | Vitesse |
| température |  | utilisation | bande passante |  |
| consommation |  | température |  |  |
|  |  | taux d’erreurs |  |  |

Les données envoyées devront subir un chiffrement.

**d) Suivi de problèmes avec le matériel**

Le programme permettra de configurer le seuil d’alerte pour chaque métrique afin de pouvoir avertir un utilisateur que la machine rencontre un problème. Ces alertes se feront par mail, mais il sera aussi possible de les consulter sur l’interface Web.

**e) Détection automatique des machines**

Il devra contenir une fonctionnalité permettant de découvrir toutes les machines qui possèdent les programmes permettant le suivi sur un réseau donné afin de les ajouter à la base de données des machines disponibles.

**f) Optionnel : suivi du matériel réseau**

En option, il pourrait aussi surveiller le matériel réseau de l’infrastructure.

**2) Organisation imaginée**

**a) Schéma**

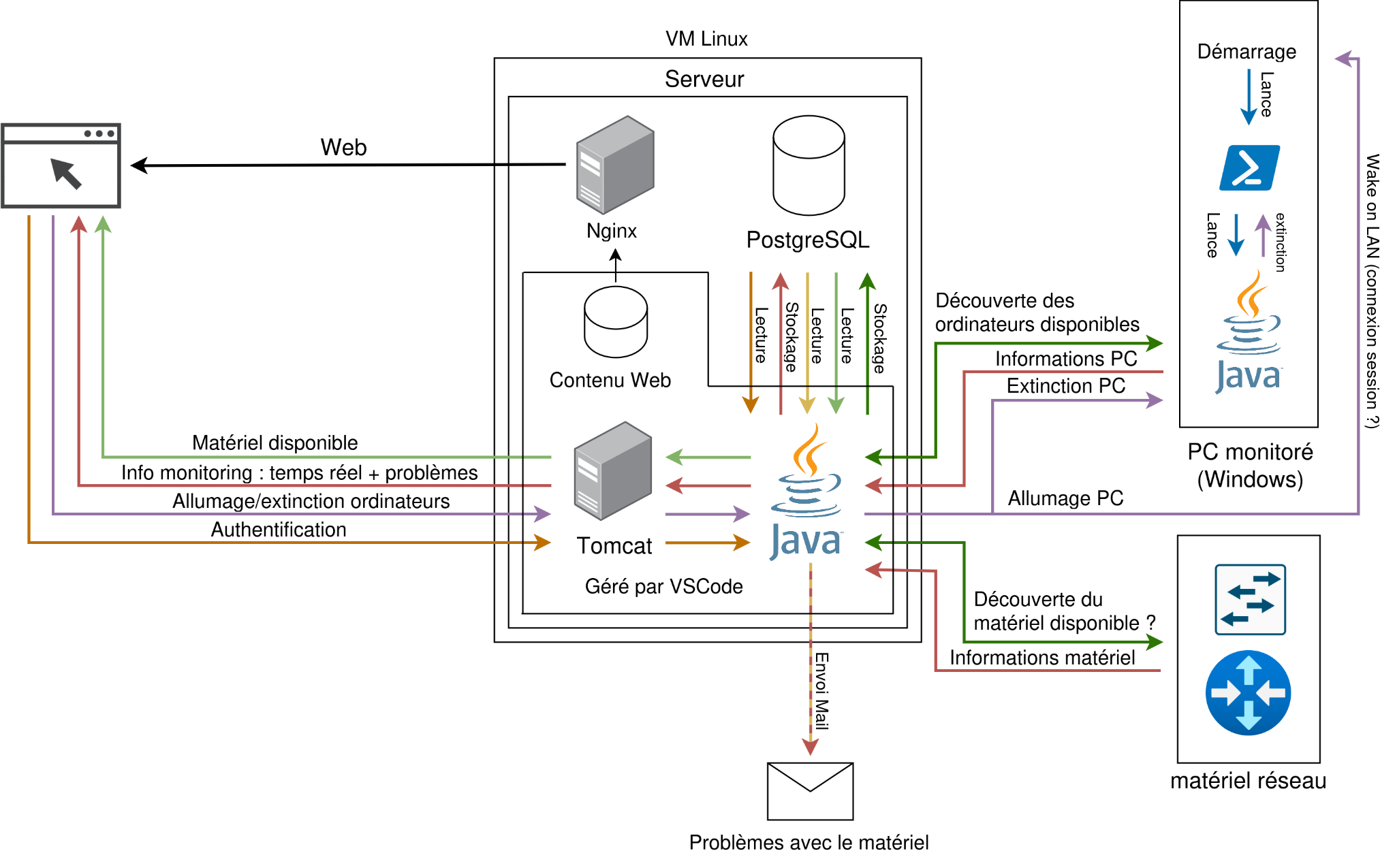


Figure 1 : schéma représentant le fonctionnement simplifié de ce projet

**a) Tomcat + PostgreSQL**

Tomcat a pour rôle de rediriger les requêtes HTTP vers des servlets Java afin d'interagir avec la base de données PostgreSQL, qui elle va stocker toutes les données du projet. J’ai choisi ces logiciels, car ce sont ceux avec lesquels nous avons travaillé dans les ressources enseignées par M. Mery.

**b) Nginx**

Nginx est un serveur Web qui distribue le contenu de l’interface Web (HTML, CSS, JavaScript et ressources annexes).

**c) Java**

Je l’ai choisi, car l’on peut l’utiliser pour faire des servlets et que l’on a travaillé dessus sur de nombreuses ressources (notamment sur les clients/serveurs TCP qui seront utilisés pour l’échange d’informations de suivi).

**d) Powershell**

Powershell a pour objectif de lancer les serveurs Java sur les ordinateurs surveillés (découverte de matériel et suivi). Je l’ai choisi majoritairement car cela permet de ne pas avoir à interagir avec l’ordinateur, mais aussi car il permet d’accéder aux outils fournis par Windows (WMI, CIM, MSInfo32…) ou encore de récupérer des informations auprès des API constructeurs (AMD Ryzen Master Monitoring SDK, Intel Performance Counter Monitor…) pour récupérer des informations plus poussées (température, fréquence…).

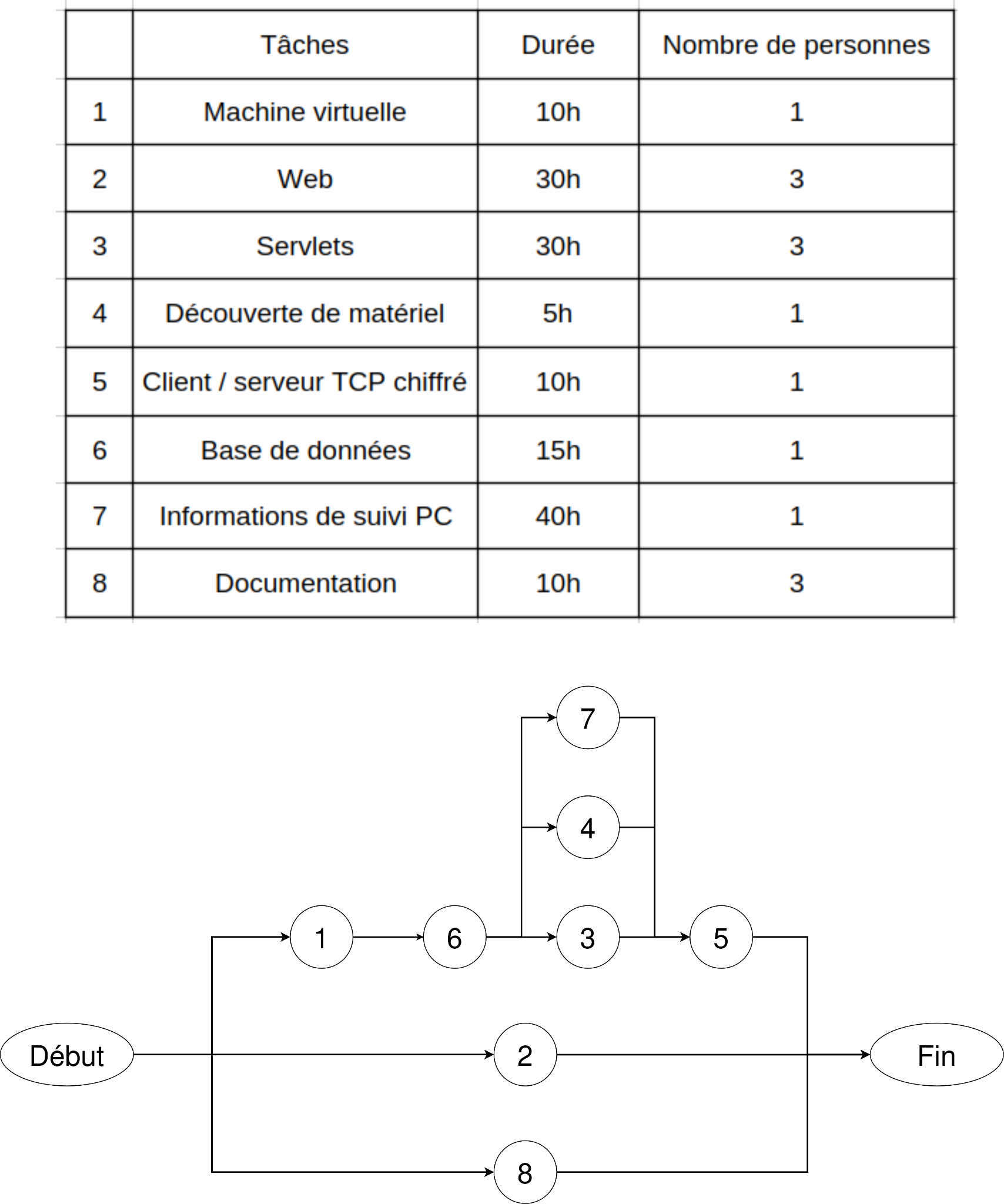
**e) VM**

Le choix d’une VM a été fait pour des contraintes de partage, car il est plutôt long de déployer l'environnement backend et que la résolution de problèmes éventuels peut prendre du temps.

**f) VSCode**

Le choix de n’utiliser que VSCode pour la gestion entière du projet va permettre de réduire la quantité de ressources utilisées en utilisant un seul IDE, mais aussi de synchroniser le projet grâce à l’extension GitHub.

**III) Délais**

****