

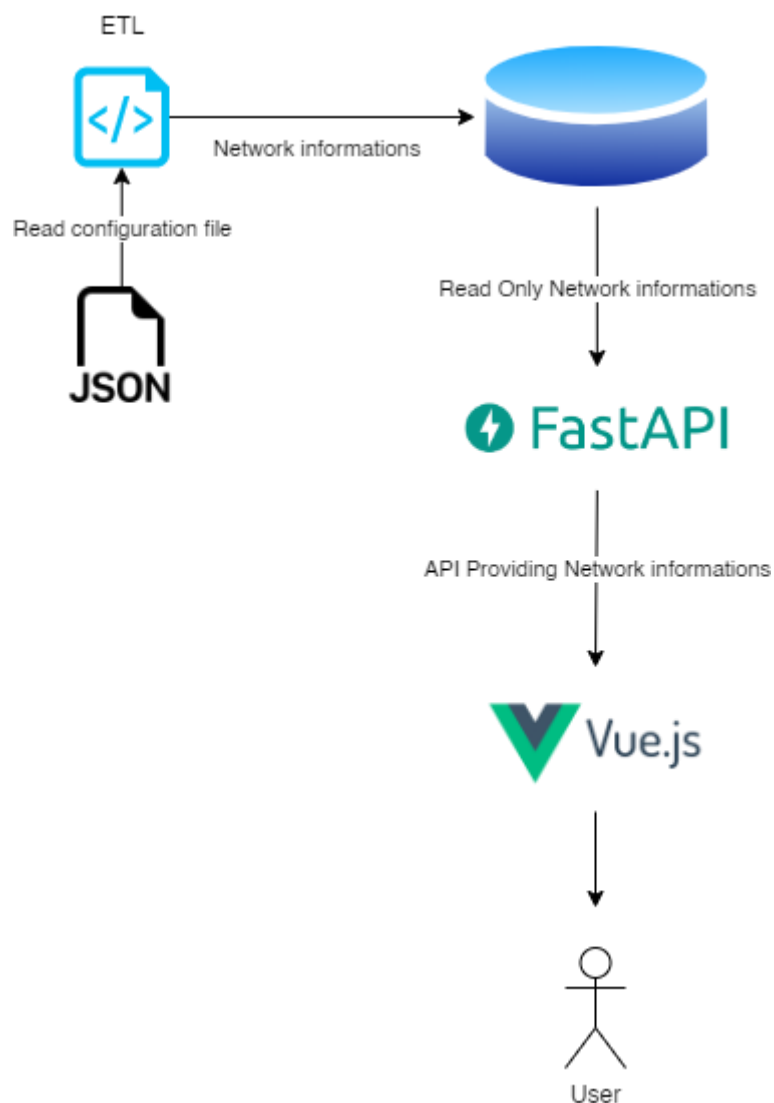
Rattrapage ICC : Architecture Microservices

Technologies imposées :

- API : FastAPI
- ETL : Python
- FrontEnd : VueJS
- Database : Sqlite

Dans ce sujet, nous allons créer une application permettant de monitorer son réseau en testant via un « ping » la latence depuis la machine hôte vers des machines distantes dont les caractéristiques seront fournies en dans un json de configuration.

L'architecture se découpe comme suit :



JSON : Cette partie là est un JSON de configuration, il devra être formaté de cette façon :

```

{
  "monitoring_delay": 5, // Le délai entre deux pings sur les entités
  "entities": [ // Les entités à monitorer
    {
      "ip": "192.168.1.1",
      "domain": "sfr.router",
      "description": "Home router"
    },
    {
      "ip": "192.168.1.61",
      "domain": "plex",
      "description": "Plex media server"
    },
    {
      "ip": "192.168.1.95",
      "domain": "sfr.tv",
      "description": "SFR TV"
    },
    {
      "ip": "192.168.1.94",
      "domain": "nvidia.shield",
      "description": "Nvidia Shield Pro for TV"
    },
    {
      "ip": "8.8.8.8",
      "domain": "dns.google.public",
      "description": "Public DNS from Google"
    },
    {
      "ip": "213.136.12.232",
      "domain": "riot.games",
      "description": "Public server from Riot Games"
    }
  ]
}

```

ETL : Cette partie est un script capable de lire la configuration. En fonction de cette configuration il sera capable de réaliser un ping sur chaque entité décrite. De plus il devra réaliser un ping toutes les N secondes, N étant définie par la clé « monitoring_delay » dans le json de configuration. Pour chaque ping, pour chaque entité il devra écrire dans une table en base de données Sqlite (nativement supportée par Python)

Base de données : Cette partie doit stocker chaque entrée de ping pour chaque entité. Une entrée est composée de la façon suivante :

- Timestamp (Date à laquelle le « ping » est effectué)
- IP (IP de l'entité vers laquelle le ping est effectué)

- Domain (Nom de domaine de l'entité vers lequel le ping est effectué)
- Description (Description de l'entité vers laquelle le ping est effectué)
- Ping (Latence associé à la requête « ping » entre la machine hôte et la machine distante, en ms)
- PacketLoss (Booléen, définit si la machine distante est atteignable ou non en regardant si le « ping » échoue)

FastAPI : Cette API doit être capable d'envoyer les données vers le FrontEnd. Le développeur est responsable du design de cette API et ses choix impacteront directement le design du FrontEnd.

VueJS : Cette application FrontEnd doit être capable de recevoir et afficher les données depuis la base de données. Plusieurs graphiques sont attendus :

Un graphique courbe 2D qui va représenter le « ping » vs le temps :

- Une courbe pour chaque entité
- Une couleur pour chaque entité
- En abscisse le timestamp, en ordonnée la valeur du « ping »

Un graphique en barre :

- Une barre par entité
- Une couleur pour chaque entité
- La hauteur de chaque barre est définie par le nombre de requêtes « ping » échouée (Le nombre de fois où on a du PacketLoss depuis le démarrage de l'application)

Le développeur est responsable du choix de la librairie graphique.

Livrable attendu :

- Un lien vers le Github du projet sera exigé.
- Le projet devra être public sur Github.
- Tout retard sera pénalisé de 4 points par jour.
- Un script de lancement Bash permettra d'orchestrer le lancement de l'application et EGALEMENT la fermeture de l'application (kill de tous les process nécessaire au fonctionnement de l'application)
- Toute initiative prise en dehors du cadre de ce sujet sera fortement valorisée (Responsivité de l'application, Charte graphique élégante, ...)

Date de rendu : 12/06/2022 23:59