**Projet ESME – Web crawler et Indexeur**

**Démarrage du projet :**

La première étape du projet est de configurer le crawler qui viendra récupérer les URL des sites traversés à partir d’une source.

Le périmètre de notre crawler web sera les sites d’actualités. Ceci pour une raison, est qu’ils contiennent un grand nombre d’informations sur des sujets divers et variés. Il est tout aussi intéressant de pouvoir trouver le sujet qui nous intéresse mais aussi de pouvoir trouver tous les sites qui en parle pour comparer les informations. Un moteur de recherche est donc tout à fait adapté pour ce sujet.

(Sinon plan B : sites de E-commerce)

Notre site source pour le début du crawling sera : https://www.lemonde.fr

**Technologie utilisée :**

Nous avons choisi d’utiliser Scrapy, un Framework de crawling conçu pour scrapper des sites web à grande échelle.

Le début du projet se fera sur un périmètre restreint (ici, le domaine lemonde.fr) donc d’autres librairie aurait pu convenir comme Beautifulsooup mais, si le projet avance bien, nous pourrions être amener à aller sur d’autres sites que nous ne connaissons pas à l’avance (principe de naviguer entre les liens de chaque page) et le projet pourrait devenir plus important donc il peut être intéressant de commencer directement sur Scrappy.

De plus, c’est une technologie que je n’ai jamais utilisé donc ce sera l’occasion d’apprendre.

**Architecture du projet :**

Une image contenant diagramme, ligne, capture d’écran, Plan

Description générée automatiquement

Figure 1 : Architecture du projet V1

Une image contenant texte, capture d’écran, diagramme, conception

Description générée automatiquement

Figure 2 : Architecture crawler Scrappy

**Premiers rendus :**

Après avoir installé et configurer le projet Scrappy, nous allons chercher les premières données.

En lançant le premier code, nous obtenons une grande liste de liens vers des pages lemonde.fr ainsi que les 200 premiers caractères de leur code html.



Une fois le code arrêté (pour l’instant manuellement) après la récupération de données pendant 23 secondes, Scrappy nous communique les statistiques de la session :

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Ici il y a eu 534 requêtes envoyées avec 516 réponses (200 : indique que la page est récupérée avec succès ; 301 : indique que l’url a été redirigée)

**Intégrer le crawler sur AWS :**

Commençons par configurer AWS. A partir du compte root je vais créer un utilisateur IAM pour ce projet.

Après avoir configuré comme il faut AWS, je copie l’arborescence des fichiers Scrapy contenant mon premier code dans S3.

Il est maintenant temps de configurer une instance EC2 pour pouvoir lancer le code en arrière-plan.

Une fois EC2 configuré et lié avec mon bucket nous pouvons lancer le programme directement dans la console EC2 :

Une image contenant texte, capture d’écran, logiciel, Police

Description générée automatiquement

**Planifier un cron job:**

Un cron job est une tâche planifiée sur EC2.

D’abord, nous configurons un timer sur le programme pour qu’il tourne que 30 secondes pour le moment.

Puis nous configurons le job pour que le crawler s’exécute toutes les heures.

**Stockage et indexation :**

Afin de pouvoir stocker et indexer les données il nous faut utiliser ElasticSearch. Il nous permettra, une fois les données indexées, de l’utiliser comme un moteur de recherche.

Pour des soucis de facilité, notamment de configuration sur Windows, nous paramètrerons ElasticSearch dans un conteneur accessble via Docker.

Une image contenant texte, Police, capture d’écran, ligne

Description générée automatiquement

ElasticSearch est maintenant configuré. Il est accessible sur <https://localhost:9200>

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

ElasticSearch nous servira que plus tard mais au moins il est configuré.

**TODO :**

* **Schéma d’architecture global du système**
* **Zoom sur le crawler et détailler son fonctionnement (logigramme)**
* **A partir du crawler qui tourne en continue sur EC2, stocker le contenu et les URLs dans S3.**