# Rapport Technique : Script de Scraping avec Transfert EC2 vers S3

## 1. Architecture Mise en Place

L'architecture déployée repose sur un système de scraping web automatisé hébergé sur AWS :

### Composants principaux :

* **Instance EC2** : Serveur d'exécution du script de scraping Python
* **Script de Crawling** : Application de scraping pour extraire les données des sites web cibles
* **Bucket S3** : Stockage des données scrapées au format structuré (JSON/CSV)
* **IAM Roles** : Permissions sécurisées pour les opérations EC2 → S3

**Flux de données :** Le script Python s'exécute sur l'instance EC2, navigue sur le site web ciblé, extrait les données via parsing HTML, les structure et les sauvegarde automatiquement dans le bucket S3 organisé par date et source.

## 2. Choix Techniques

### Site Source et Scraping

Le script effectue du web scraping sur des sites web spécifiques :

* **Requêtes HTTP** : Utilisation de requests/urllib pour récupérer les pages web
* **Parsing HTML** : Extraction du contenu avec BeautifulSoup4 via sélecteurs CSS
* **Navigation complexe** : Selenium WebDriver pour les sites JavaScript dynamiques
* **Gestion des sessions** : Maintien des cookies et headers personnalisés
* **Rate Limiting** : Délais configurables entre requêtes (2-5 secondes) pour respecter les serveurs

### Gestion des Erreurs

Stratégie robuste pour gérer les aléas du scraping :

* **Retry automatique** : 3 tentatives avec backoff exponentiel sur les échecs http
* **Gestion des timeouts** : Timeout de 30 secondes par requête avec fallback
* **Détection de blocage** : Reconnaissance des pages d'erreur 403/429 et adaptation
* **Logging détaillé** : Traçabilité complète des URLs scrapées, erreurs et succès
* **Rotation User-Agent** : Liste rotative pour éviter la détection comme bot
* **Respect robots.txt** : Vérification automatique des politiques de scraping

### Technologies Utilisées

* **Python 3.9** avec requests pour les requêtes http
* **BeautifulSoup4** pour le parsing et extraction HTML
* **Selenium WebDriver** pour les contenus JavaScript
* **boto3** pour l'upload vers S3
* **Pandas** pour la structuration des données extraites
* **Crontab** pour l'exécution programmée (quotidienne/hebdomadaire)

### **3.** Démonstration du Transfert EC2 vers S3

### Processus de Scraping et Transfert

Le processus s'articule en quatre phases :

* **Crawling** : Navigation sur les URLs cibles avec rotation des User-Agents
* **Extraction** : Parsing HTML et extraction des données selon les sélecteurs définis
* **Structuration** : Nettoyage et formatage des données en JSON/CSV avec Pandas
* **Upload S3** : Transfert automatique vers S3 avec organisation par dossiers date/source

## 4. Configuration S3 - Accès Public Sélectif

Une stratégie de compartiment spécifique a été mise en place pour rendre accessible publiquement uniquement le dossier contenant les images extraites :

Structure du stockage :

* pokemon/image/ : Dossier public contenant les images scrapées
* Autres dossiers : Accès privé sécurisé nécessitant authentification AWS

Stratégie JSON appliquée :

json

{

"Version": "2012-10-17",

"Statement": [

{

"Sid": "PublicReadPokemonImageFolder",

"Effect": "Allow",

"Principal": "\*",

"Action": ["s3:GetObject"],

"Resource": "arn:aws:s3:::jtrpokemonbucket/pokemon/image/\*"

}

]

}

Avantages sécuritaires :

* Accès public limité aux seules images dans pokemon/image/
* Protection des données sensibles dans les autres répertoires
* URLs directes fonctionnelles : https://jtrpokemonbucket.s3.amazonaws.com/pokemon/image/NumPok\_nomPok.png

Exemple de test :

1. <https://jtrpokemonbucket.s3.amazonaws.com/pokemon/image/025_Pikachu.png>

2. [https://jtrpokemonbucket.s3.amazonaws.com/pokemon/image/006\_Charizard.png](https://jtrpokemonbucket.s3.amazonaws.com/pokemon/image/006_Charizard.png%20%20)

3. <https://jtrpokemonbucket.s3.amazonaws.com/pokemon/image/009_Blastoise.png>

4. <https://jtrpokemonbucket.s3.amazonaws.com/pokemon/image/150_Mewtwo.png>

5. [https://jtrpokemonbucket.s3.amazonaws.com/pokemon/image/144\_Articuno.png](%20https:/jtrpokemonbucket.s3.amazonaws.com/pokemon/image/144_Articuno.png)

## Conclusion

Le script de scraping développé offre une solution robuste pour l'extraction automatisée de données web avec stockage sécurisé sur S3. L'architecture EC2/S3 garantit la scalabilité et la persistance des données, tandis que les mécanismes anti-détection assurent une collecte durable. Les performances mesurées confirment l'efficacité de cette approche pour un scraping éthique et performant.