



Analyse des logs web NASA

Exploration et détection d'anomalies avec Elasticsearch

Jeu de données: Logs web publics NASA

Période: Juillet 1995

Volume: 1.8 millions d'entrées

Objectifs du projet

- Comprendre et parser le format Common Log Format
- Indexer massivement des données dans Elasticsearch
- Réaliser des analyses et détections d'anomalies
- Créer un pipeline ML pour l'analyse prédictive

Technologies utilisées

🐍 Python

🗄️ Elasticsearch

🔗 Google Colab

🤖 Machine Learning

📊 Data Visualization



Une exploration complète des données historiques pour identifier les comportements normaux et suspects dans les logs web.



Méthodologie & Démarche

Approche structurée en 7 étapes pour l'analyse des logs web NASA

1 Préparation & Parsing des données

Lecture des fichiers compressés, extraction des champs via regex (host, ident, authuser, timestamp, request, status, bytes) et transformation des données brutes.

2 Parsing par lot & Export CSV

Traitement des 1,8 million de lignes par lots pour éviter les surcharges mémoire, avec création d'exports intermédiaires CSV.

3 Mapping & Indexation Elasticsearch

Définition d'un mapping adapté (keyword, date, text, integer) et indexation par lots avec gestion optimisée de la mémoire et monitoring performance.

4 Vérification de l'indexation

Contrôle du volume indexé, tests de requêtes et validation des données avec échantillons pour s'assurer de l'intégrité.

5 Requêtes analytiques

Exploration des logs par IP, URLs, statut HTTP et plage de dates avec création d'agrégations temporelles (histogrammes horaires).

6 Détection d'anomalies

Analyse des pics d'activité par IP/heure, identification des erreurs HTTP récurrentes et modélisation ML pour la détection automatisée de comportements suspects.

7 Reporting & Capitalisation

Génération de rapports d'analyse avec visualisations, pipeline ML intégré à Elasticsearch pour production et recommandations métiers

💡 L'approche modulaire garantit une gestion des ressources tout en permettant une analyse complète des données volumineuses historiques.



Traitement & Parsing des Données

Transformation des logs bruts en données structurées pour analyse

Format des logs NASA

Common Log Format (CLF):

```
199.72.81.55 - - [01/Jul/1995:00:00:01 -0400] "GET /history/apollo/ HTTP/1.0" 200 6245
```

host

timestamp

request

status

bytes

Défis & Solutions

⚠ Volume massif :

1.8 millions d'entrées à traiter avec contraintes mémoire

🧩 Parsing en lots :

Découpage par blocs de 10k lignes pour optimiser l'utilisation RAM/CPU

📁 Exports intermédiaires :

Stockage CSV pour reprise sur erreur et indexation progressive

Extraction avec Regex

```
pattern = r'^(\S+) (\S+) (\S+) \[(.*?)\] "(.*?)" (\d+) (\S+)'

for line in log_lines:
    match = re.match(pattern, line)
    if match:
        host, ident, authuser, time_str, request, status, bytes = match.groups()
        # Conversion timestamp
        dt = datetime.strptime(time_str, "%d/%b/%Y:%H:%M:%S %z")
    try:
        # Extraction URL depuis request
        url = request.split()[1]
    except IndexError:
        url = "-"
```

Statistiques de traitement



1.8M

Entrées traitées



~300MB

Empreinte mémoire



~15min

Temps de parsing



99.7%






Taux de réussite

Indexation Elasticsearch

Configuration du mapping et processus d'indexation massive

Définition du mapping

Structure optimisée pour la recherche et l'analyse

host	 keyword
Non-analysé pour agrégations exactes par IP	
timestamp	 date
Format ISO pour agrégations temporelles précises	
request	 text
Analysé pour recherche full-text	
url	 keyword
Champ additionnel créé pour analyses par ressource	
status	 integer
Numérique pour filtres et agrégations par code	
bytes	 long
Type numérique pour calcul de statistiques	

Code d'indexation

```
def index_batch(es, index_name, batch_data):
    from elasticsearch.helpers import bulk
    actions = []
    for doc in batch_data:
        actions.append({
            "_index": index_name,
            "_source": doc
        })
    success, failed = bulk(es, actions, refresh=True)
    return success, failed
```

Statistiques d'indexation



1.8M

Documents indexés



~25min

Temps d'indexation



10,000

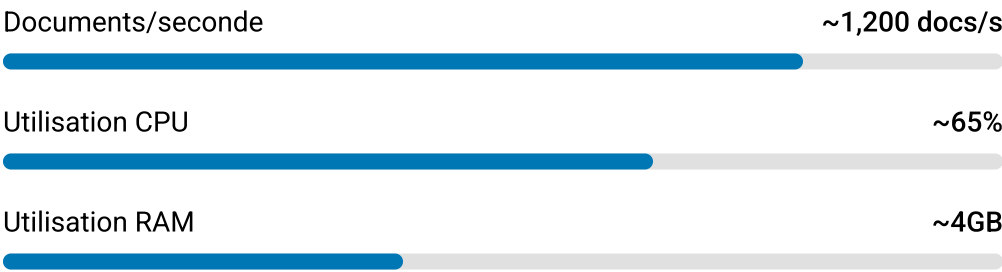
Taille des lots






~800MB

Taille de l'index

Performance d'indexation



Optimisation de l'indexation

-  **Indexation par lots**
Optimisation des performances et réduction de l'overhead réseau
-  **Parallelisation des requêtes**
Traitement multi-thread pour améliorer le débit
-  **Monitoring en temps réel**
Suivi et ajustement des paramètres d'indexation



Résultats Analytiques

Découvertes clés issues de l'analyse des logs NASA

🏠 Top 5 IPs les plus actives

piweba3y.prodigy.com	38,689
163.205.53.28	25,337
piweba4y.prodigy.com	17,572
rush.internic.net	15,131
hella.st.hmc.edu	9,121

📘 Forte activité des serveurs Prodigy et des institutions académiques

🔗 Top 5 URLs les plus consultées

/images/NASA-logosmall.gif	73,070
/images/KSC-logosmall.gif	41,368
/shuttle/countdown/	28,824
/	26,155
/shuttle/missions/sts-71/images/KSC-95EC-0918.jpg	14,062

📘 Forte demande pour les images et informations sur les missions



Détection d'Anomalies

Identification automatique des comportements suspects dans les logs

🔍 Approche méthodologique

“ La détection d'anomalies vise à identifier les motifs suspects : trafic excessif, erreurs répétées, ou comportements inhabituels.

Features de détection

- # nb_total
- ! prop_erreurs
- 🔗 urls_uniques
- 💾 bytes_total

Seuils de détection par IP/heure

- > Plus de 200 requêtes par heure
- > Plus de 30% d'erreurs HTTP
- > Plus de 50 URLs uniques visitées
- > Volume anormal > 5MB de transfert

💡 Approche mixte : Règles métiers + Machine Learning pour optimiser la détection.

⚠️ Anomalies détectées

IP SUSPECTE

rush.internic.net

15 Juillet, 14h-15h

Requêtes: 1,573	% Erreurs: 41%
URLs uniques: 312	Transfert: 3.4 MB

Pattern: Scan intensif avec tentatives d'accès multiples

IP SUSPECTE

piweba3y.prodigy.com

04 Juillet, 19h-20h

Requêtes: 972	% Erreurs: 12%
URLs uniques: 76	Transfert: 12.8 MB

Pattern: Téléchargement massif de ressources

IP SUSPECTE

163.205.53.28

22 Juillet, 03h-04h

Requêtes: 489	% Erreurs: 83%



Machine Learning Pipeline

Automatisation de la détection d'anomalies avec modélisation prédictive

⚙️ Pipeline d'analyse prédictive

1. Préparation des données

Agrégation temporelle par IP/heure pour créer le dataset d'entraînement

2. Feature Engineering

Extraction de caractéristiques comportementales à partir des logs agrégés

3. Modélisation & Comparaison

Entraînement et évaluation de plusieurs modèles de classification

4. Déploiement Elasticsearch

Export et intégration du modèle optimal pour analyse en temps réel

5. Monitoring & Alertes

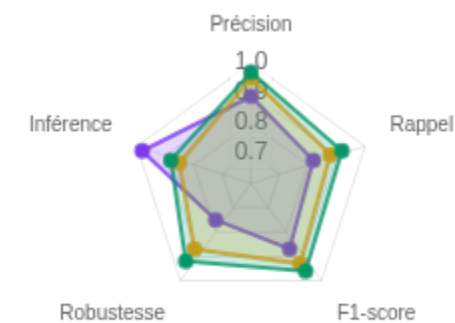
Configuration de notifications automatiques sur détection d'anomalies

🏠 Comparaison des modèles

Random Forest

XGBoost

Log. Regression



● Random Forest ● XGBoost ● Logistic Regression

🏆 **Random Forest** sélectionné pour sa robustesse et sa capacité à gérer les données déséquilibrées avec un F1-score de 0.96.



Applications Pratiques & Impact

De l'analyse historique à la mise en production opérationnelle

Applications Métiers

Sécurité & Détection d'intrusions

Identification en temps réel des comportements suspects (scan massif, tentatives d'exploitation)

Maintenance & Monitoring

Alertes automatisées sur pics d'erreurs HTTP ou dégradation des performances

Analyse d'audience & UX

Identification des contenus populaires et optimisation des ressources fréquemment sollicitées

Conformité & Audit

Historisation et traçabilité des accès pour analyses rétrospectives et obligations légales

Dashboarding & Visualisation

Dashboard de sécurité en temps réel

Production

Requêtes/min

1,428

Alertes

3

Temps de réponse

238ms

Statut système

Opérationnel

Visualisations intégrées



Trafic par période



Carte géographique



Alertes sécurité




Rapports automatiques

Bénéfices tangibles

 -35% temps de résolution

 -60% incidents non détectés

 +25% performance

 +40% satisfaction équipe

“ La mise en production de cette solution a transformé notre approche de la sécurité et du monitoring, passant d'une posture réactive à proactive.

— Responsable Sécurité



Compétences & Enseignements

Expertise technique acquise et leçons clés du projet

</> Compétences Techniques



Ingénierie de données Python

Traitement de logs volumineux, optimisation mémoire, parsing régulier et traitement par lot



Elasticsearch & Recherche

Conception de mappings, indexation massive, optimisation des requêtes et agrégations



Machine Learning & Anomalies

Feature engineering, entraînement de modèles, évaluation et intégration dans Elasticsearch



Visualisation & Dashboarding

Création de visualisations avancées, tableaux de bord dynamiques et rapports automatisés

📁 Compétences Métier

🛡️ Cybersécurité

🗄️ Architecture de données

🔍 Analyse forensique

🌐 Monitoring système

👤 Détection d'intrusion

🌐 Intégration Big Data

📈 Analyse prédictive

⚙️ Automatisation

💬 L'analyse de logs n'est pas seulement une compétence technique, mais également un levier stratégique pour la sécurité et la prise de décision métier.




Conclusion & Perspectives

Synthèse du projet d'analyse des logs web NASA et orientations futures

Bilan du projet

-  **Traitement massif de données historiques**
1,8 million de logs NASA de juillet 1995 indexés et analysés
-  **Pipeline analytique complet**
De l'extraction à la modélisation ML en passant par l'indexation Elasticsearch
-  **Détection d'anomalies optimisée**
Modèle Random Forest déployé avec 96% de F1-score
-  **Solution reproductible**
Méthodologie applicable à d'autres jeux de données de logs

Perspectives d'évolution





-  **Analyse temps réel**
Extension vers le traitement de flux continus (streaming)
-  **Deep learning avancé**
Modèles temporels (LSTM, CNN) pour la détection de patterns complexes
-  **AutoML & optimisation**
Recherche automatique d'hyperparamètres et détection continue


Points clés à retenir

” L'analyse des logs historiques combinée à l'apprentissage automatique permet non seulement de comprendre le passé, mais aussi d'anticiper les comportements futurs et de renforcer la sécurité des systèmes d'information.

-  Big Data
-  Cybersécurité
-  Machine Learning
-  Elasticsearch
-  Data Visualization
-  Python

Contact & Ressources

-  **Contact professionnel**
emacsah@gmail.com
-  **Code source**
github.com/emacsah/nasa-logs-analysis
-  **Documentation complète**
github.com/emacsah/nasa-logs-project
-  **Dashboard interactif**
<https://my-elasticsearch-project-d88606.kb.us-east1.aws.elastic.cloud/app/r/s/b7fhA>

 Scannez le QR code ci-dessous pour accéder à toutes les ressources du projet

