

Analyse et Visualisation des Logs Firefox

Utilisation de la Stack ELK

OUHNI Kamal MONCIF Mouad

ENSIAS

10 novembre 2025

Encadrant : Noureddine Kerzazi

Plan de la Présentation

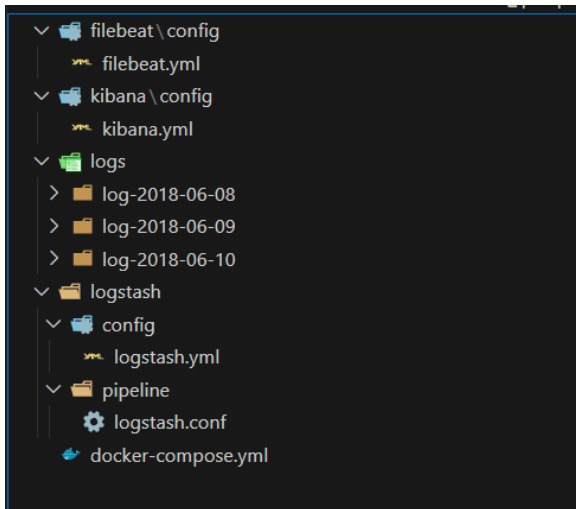
- 1 Introduction
- 2 Architecture
- 3 Ingestion des Données
- 4 Visualisations
- 5 Machine Learning
- 6 Résultats
- 7 Conclusion

Problématique :

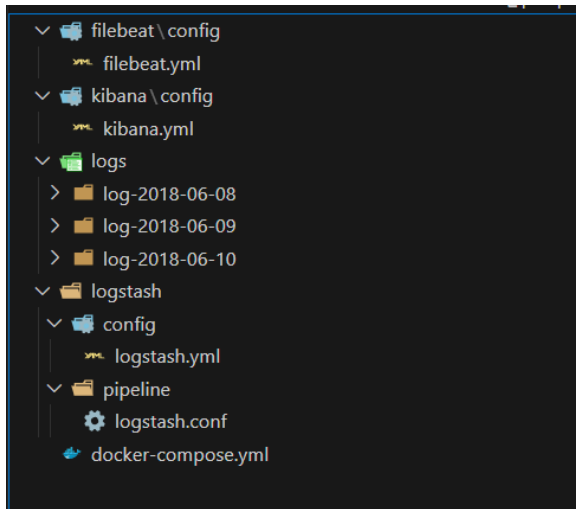
- Analyse de logs Firefox
- Volume important de données
- Détection d'anomalies

Objectif :

- Visualisation en temps réel
- Monitoring des builds
- Détection automatique



Architecture du Projet



Stack ELK : Filebeat → Logstash → Elasticsearch → Kibana

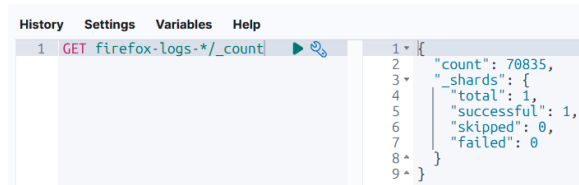
Composants

Ingestion :

- Filebeat : Collecte des logs
- Logstash : Parsing et transformation

Stockage & Analyse :

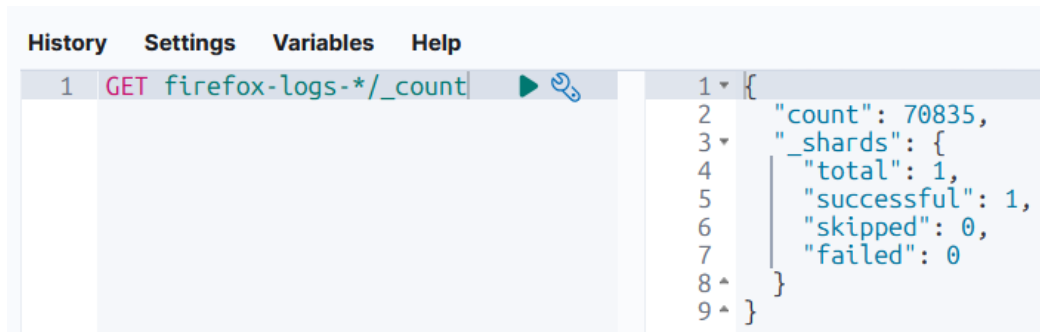
- Elasticsearch : Indexation
- Kibana : Visualisation



The screenshot shows a Kibana console interface with a light blue header containing 'History', 'Settings', 'Variables', and 'Help'. The main area is divided into two panes. The left pane shows a single entry: '1 GET firefox-logs-*/_count'. The right pane shows the JSON response for this query, with line numbers 1 through 9 on the left. The response is a JSON object with a 'count' field and a '_shards' object containing 'total', 'successful', 'skipped', and 'failed' counts.

```
1 GET firefox-logs-*/_count
1 {
2   "count": 70835,
3   "_shards": {
4     "total": 1,
5     "successful": 1,
6     "skipped": 0,
7     "failed": 0
8   }
9 }
```

Pipeline d'Ingestion

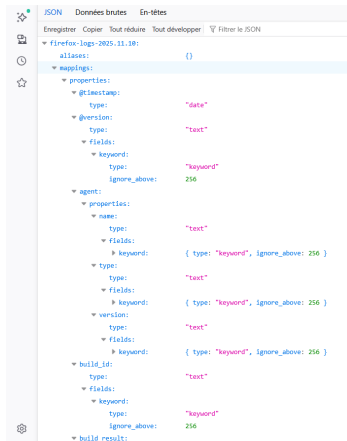


The screenshot shows a Kibana console interface with a light blue header containing tabs for 'History', 'Settings', 'Variables', and 'Help'. Below the header, a single entry in the history table shows a 'GET' request to the endpoint 'firefox-logs-*/_count'. To the right of the endpoint is a green play button icon and a blue link icon. The response is displayed on the right side of the console, showing a JSON object with a 'count' of 70835 and a '_shards' object containing 'total': 1, 'successful': 1, 'skipped': 0, and 'failed': 0. The response is formatted with line numbers 1 through 9 on the left.

```
1 GET firefox-logs-*/_count
2 {
3   "count": 70835,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   }
}
```

Résultat : Documents ingérés avec succès dans Elasticsearch

Mapping Elasticsearch



- Index pattern : firefox-log-*
- Types de champs optimisés
- Configuration des aggregations

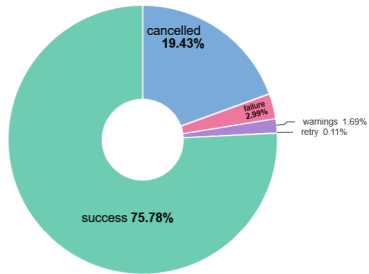
Vue d'ensemble de l'activité des builds

Total Builds

Count of records

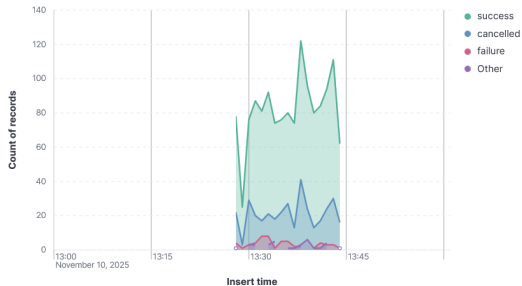
70,835

Build Results



Dashboard 1 : Overview (suite)

Builds Over Time



Top Builders

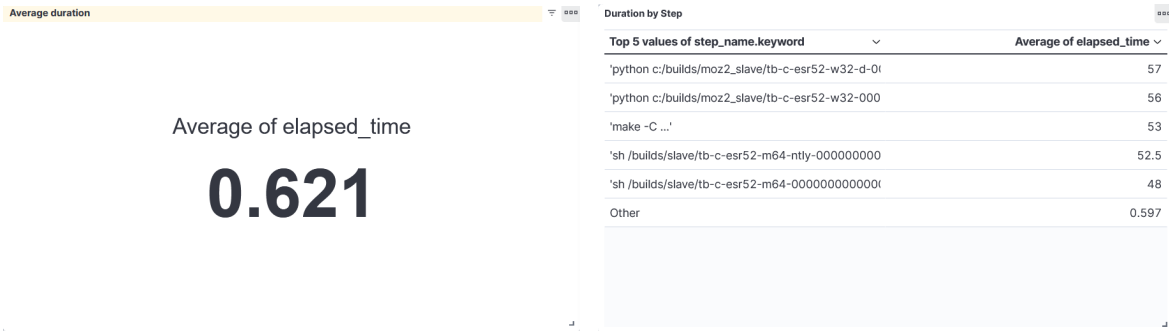
Top 10 values of builder.keyword	Count of records
comm-esr52_win7_ix_test-mozmill	4
comm-esr52_win7_ix_test-xpcshell	4
comm-esr52_xp_ix_test-mozmill	4
comm-esr52_xp_ix_test-xpcshell	4
comm-esr52_yosemite_r7_test-mozmill	4
comm-esr52_yosemite_r7_test-xpcshell	4
mozilla-esr52-macosx64	4
mozilla-esr52-macosx64-debug	4
mozilla-esr52-win64-debug	4
Other	1,795

Dashboard 1 : Overview (fin)

Activity par Slave	
Top 5 values of slave_name.keyword	Count of records
t-xp32-ix-006	35
t-xp32-ix-009	33
t-xp32-ix-010	33
t-xp32-ix-005	32
t-xp32-ix-007	32
Other	1,672

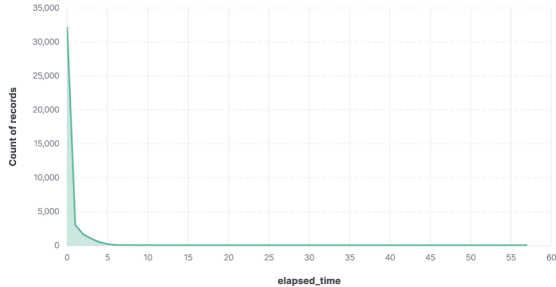
Métriques clés : Total builds, Taux succès, Distribution, Timeline, Top builders

Analyse des performances et durées d'exécution



Dashboard 2 : Performance (suite)

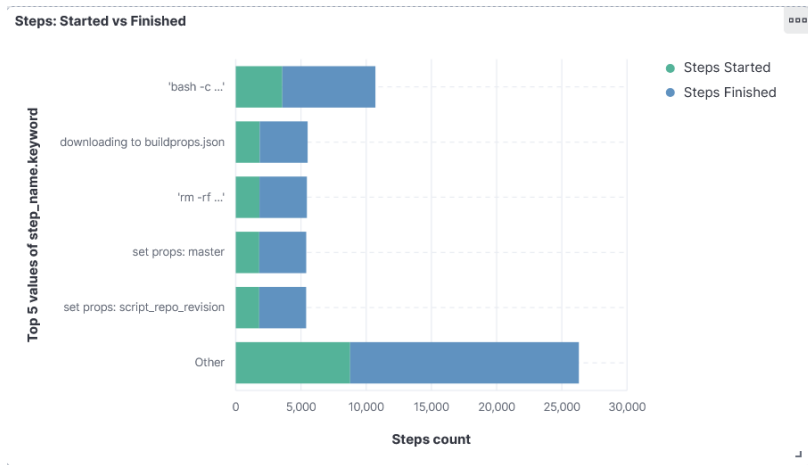
Distribution des Durées



Évolution Durée Moyenne des Steps

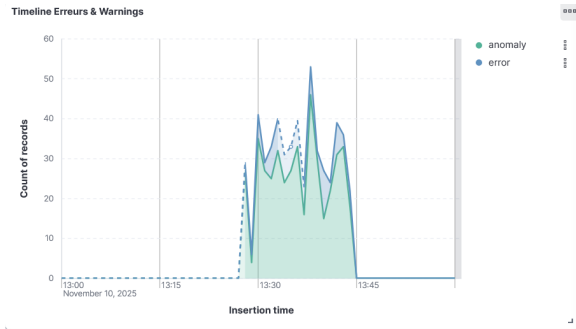


Dashboard 2 : Performance (fin)



Analyses : Durée moyenne, Distribution, Top steps lents, Évolution

Détection et analyse des anomalies



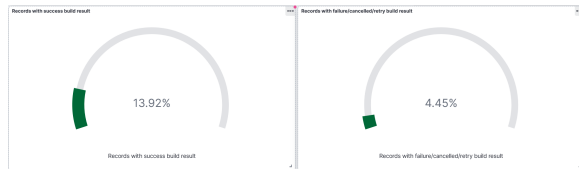
Dashboard 3 : Anomalies (suite)

Top 5 values of message with anomaly tag

Top 5 values of message with anomaly tag: ▾

Records count ▾ Count of records ▾

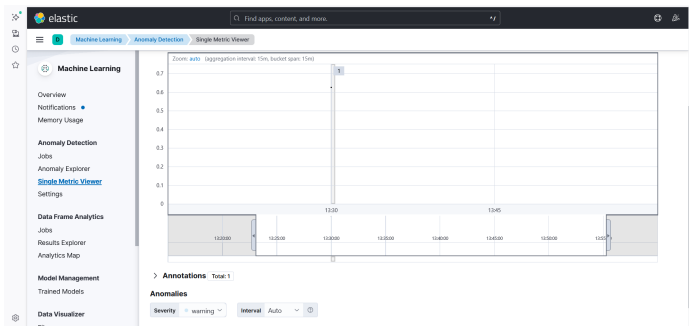
results: cancelled (6)	357
results: failure (2)	55
results: warnings (1)	31
results: retry (5)	2



6 visualisations pour détecter :

- Nombre et timeline des erreurs
- Top messages d'erreurs
- Builds en échec
- Steps avec durée anormale
- Niveau d'anomalies global

Détection d'Anomalies ML



- **Job** : Détection durée anormale
- **Fonction** : `mean(elapsed_time)`
- **Bucket span** : 15 minutes
- **Résultat** : Détection automatique temps réel

Métriques Obtenues :

- Nombre total de builds
- Taux de succès global
- Durées moyennes
- Distribution erreurs

Insights Opérationnels :

- Steps à optimiser
- Patterns d'erreurs
- Variations performance
- Anomalies critiques

Solution complète pour monitoring et analyse des builds Firefox

Stack ELK 8.11.0 :

- Elasticsearch
- Logstash
- Kibana
- Filebeat

Infrastructure :

- Docker
- Docker Compose
- Machine Learning
- Allocation : 2.6 GB

3 Dashboards — 16 Visualisations — 1 Job ML

Réalisations :

- Pipeline d'ingestion automatisé
- Parsing et enrichissement des logs
- Indexation optimisée dans Elasticsearch
- 3 dashboards interactifs avec 16 visualisations
- Détection d'anomalies avec Machine Learning

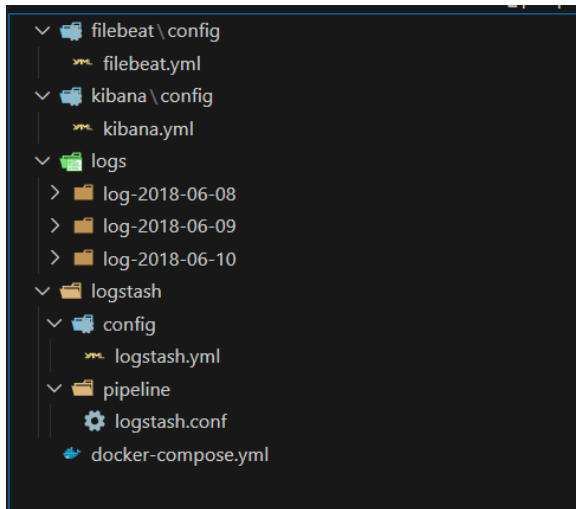
Perspectives :

- Extension à d'autres sources de logs
- Alertes automatiques avancées
- Analyses prédictives
- Intégration CI/CD

Merci pour votre attention

Questions ?

Annexe : Architecture Détaillée



Annexe : Configuration

Filebeat :

- Surveillance : logs/log-*/
- Multi-line parsing
- Output : Logstash

Logstash :

- Filtres Grok
- Enrichissement avec tags
- Output : Elasticsearch

Elasticsearch :

- Index pattern : firefox-log-*
- Mapping personnalisé
- Optimisé pour aggregations