

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 Contexte
- 2 Architecture du Projet
- 3 Mapping Elasticsearch
- 4 Pipeline d'Ingestion
- 5 Visualisations
- 6 Machine Learning
- 7 Résultats
- 8 Conclusion

Problématique :

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

Objectif :

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

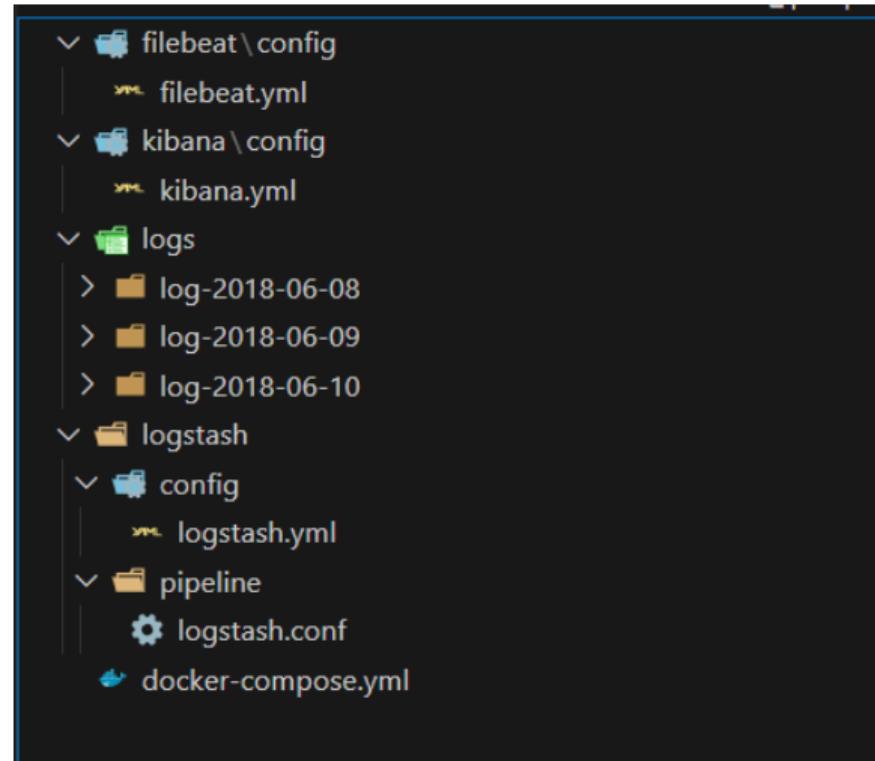
Architecture du Projet

Ingestion :

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

Stockage & Analyse :

- Elasticsearch : Indexation
- Kibana : Visualisation



Stack ELK : Filebeat → Logstash → Elasticsearch → Kibana

Mapping Elasticsearch

The screenshot shows the Elasticsearch Settings interface with the following mapping configuration:

```
JSON Données brutes En-têtes
Enregistrer Copier Tout réduire Tout développer Filtrer le JSON
firefox-logs-2025.11.10:
  aliases: {}
  mappings:
    properties:
      @timestamp:
        type: date
      @version:
        type: text
      fields:
        keyword:
          type: keyword
          ignore_above: 256
      agent:
        properties:
          name:
            type: text
            fields:
              keyword:
                type: keyword
                ignore_above: 256
          type:
            type: text
            fields:
              keyword:
                type: keyword
                ignore_above: 256
          version:
            type: text
            fields:
              keyword:
                type: keyword
                ignore_above: 256
      build_id:
        type: text
        fields:
          keyword:
            type: keyword
            ignore_above: 256
      build_result:
```

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

Pipeline d'Ingestion

The screenshot shows the Elasticsearch Dev Tools interface. At the top, there is a navigation bar with tabs: History, Settings, Variables, and Help. Below the navigation bar, a search bar contains the URL `GET firefox-logs-*/_count`. To the right of the search bar are two icons: a blue play button and a magnifying glass. The main area displays the response from the API call. The response is a JSON object with the following structure:

```
1 {  
2   "count": 70835,  
3   "_shards": {  
4     "total": 1,  
5     "successful": 1,  
6     "skipped": 0,  
7     "failed": 0  
8   }  
9 }
```

Résultat : Documents ingérés avec succès dans Elasticsearch

Dashboard 1 : Overview

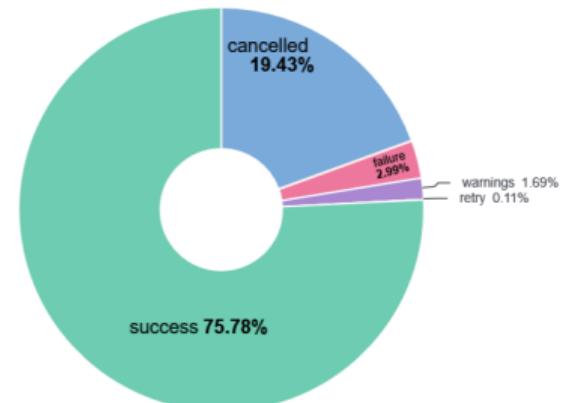
Vue d'ensemble de l'activité des builds

Total Builds

Count of records

70,835

Build Results



Dashboard 1 : Overview (suite)



Top Builders

Top 10 values of builder.keyword

builder.keyword	Count of records
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

Average duration

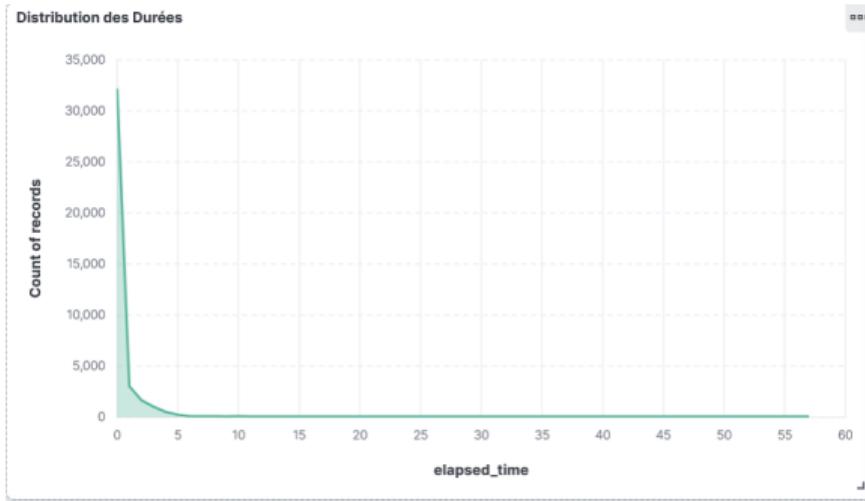
Average of elapsed_time

0.621

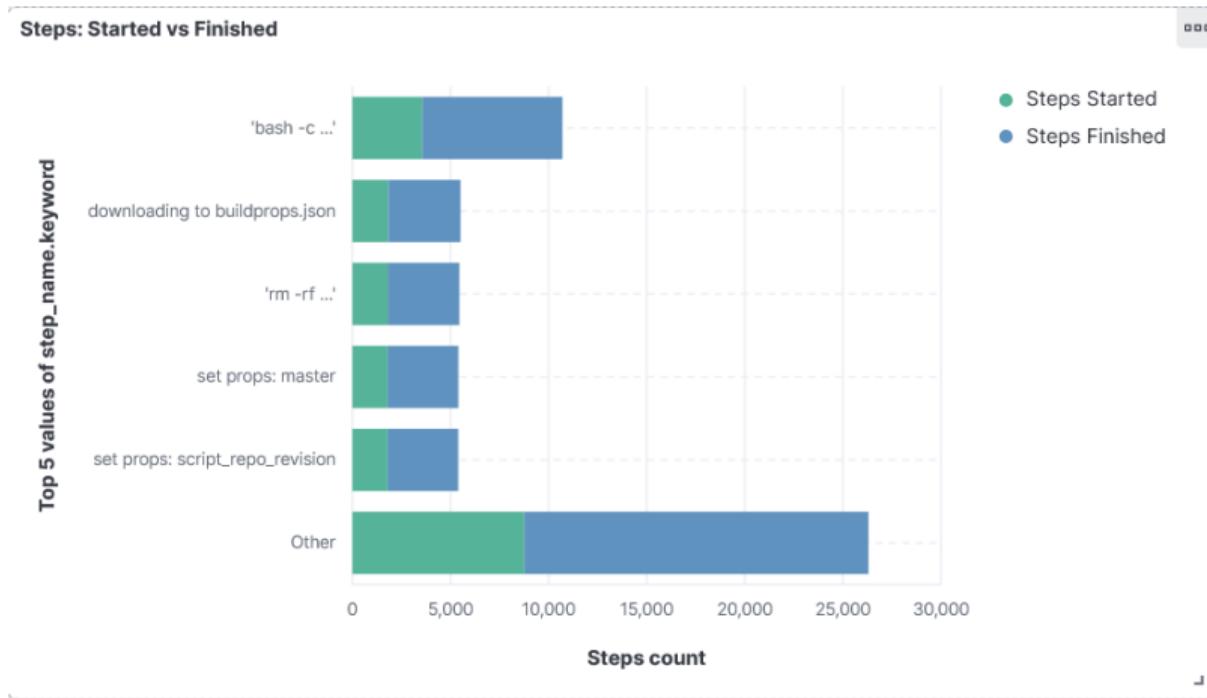
Duration by Step

Top 5 values of step_name.keyword	Average of elapsed_time
'python c:/builds/moz2_slave/tb-c-esr52-w32-d-0\	57
'python c:/builds/moz2_slave/tb-c-esr52-w32-000	56
'make -C ...'	53
'sh /builds/slave/tb-c-esr52-m64-ntly-000000000	52.5
'sh /builds/slave/tb-c-esr52-m64-0000000000000000\	48
Other	0.597

Dashboard 2 : Performance (suite)



Dashboard 2 : Performance (fin)

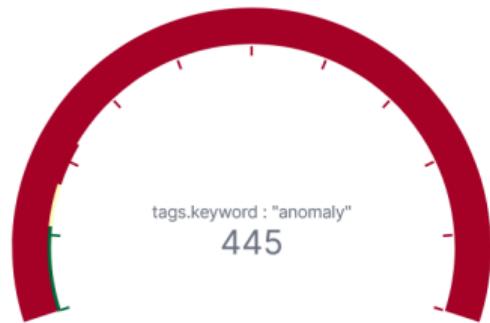


Analyses : Durée moyenne, Distribution, Top steps lents, Évolution

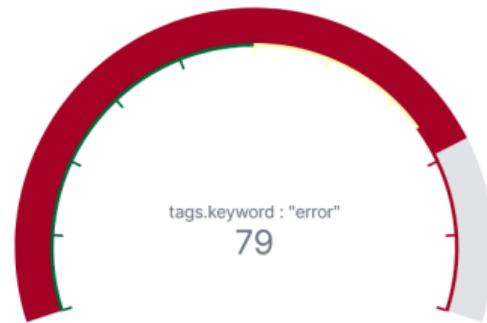
Dashboard 3 : Anomalies

Détection et analyse des anomalies

Records with anomaly tags



Records with error tags



Records with anomaly tags

- 0 - 50
- 50 - 75
- 75 - 100

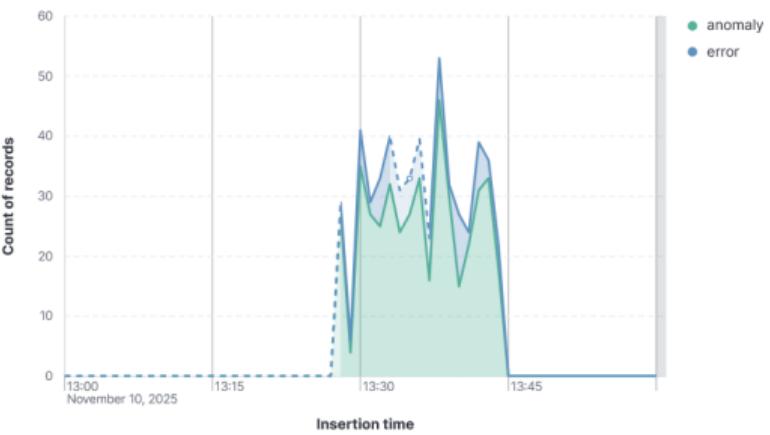
Records with error tags

...

...

Dashboard 3 : Anomalies (suite)

Timeline Errors & Warnings



Top 5 values of message with anomaly tag

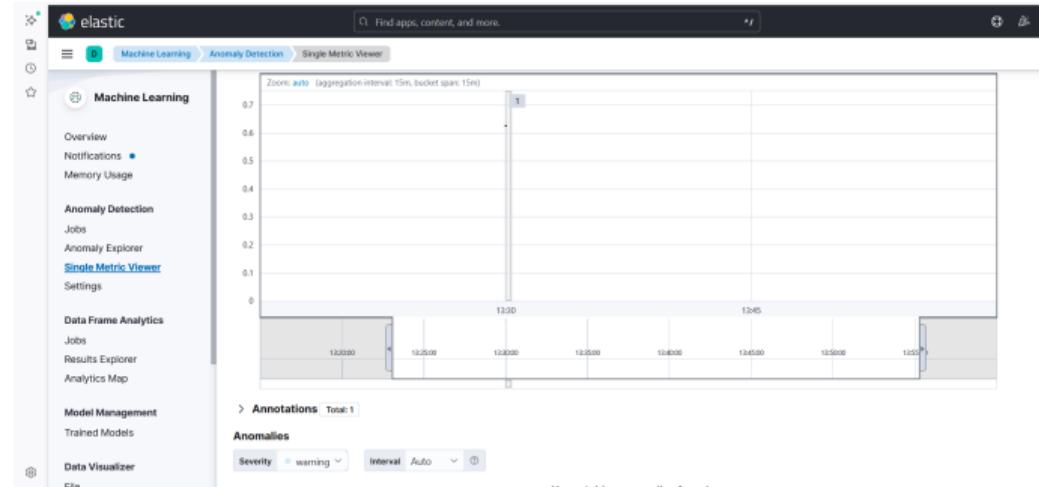
Top 5 values of message with anomaly tag	Records count	Count of records
results: cancelled (6)	357	
results: failure (2)	55	
results: warnings (1)	31	
results: retry (5)	2	

Dashboard 3 : Anomalies (fin)



Métriques clés : Nombre et timeline des erreurs, Top messages d'erreurs, Steps avec durée anormale

Détection d'Anomalies ML



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

Métriques et Insights

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

Conclusion

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 : 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