



Ecole Supérieure Privée des Technologies et de l'Ingénierie

SPECIALITE : SSIR-2D

RAPPORT DE PROJET CLOUD

Implémentation d'une architecture ELK

Réalisée par : Wissem nasri

Encadré par : mourad melliti

Année Universitaire : 2023 – 2024

Table de matière

1. Introduction	1
1.1. Contexte et Objectifs	1
2. Présentation de l'Architecture ELK	1
2.1. Description des Composants ELK (Elasticsearch, Logstash, Kibana, Metricbeat, Filebeat, Setup Certificates)	1
2.2. Avantages et Limitations de l'Architecture ELK	2
2.2.1. Avantages :	2
2.2.2. Limitations :	2
3. Analyse des Besoins et Objectifs	2
3.1. Identification des Besoins Techniques et Fonctionnels	2
3.2. Spécifications Non-Fonctionnelles (Sécurité, Performances, Évolutivité)	3
3.3. Enjeux et Contraintes du Projet	3
4. Conception de l'Architecture	3
4.1. Schéma d'Architecture ELK	3
5. Mise en Œuvre de l'Architecture ELK	4
5.1. Préparation de l'Environnement (Docker et Docker Compose)	4
5.2. Déploiement des services	5
5.2.1. Déploiement de Setup Certificates	6
5.2.2. Déploiement de Elasticsearch	6
5.2.3. Déploiement de Kibana	6
5.2.4. Déploiement de Logstash, Filebeat et Metricbeat	6
6. Tests et Validation de l'Architecture	7

Table des figure

Figure 1 : Schéma d'Architecture ELK	4
Figure 2:installation docker & docker-compose	5
Figure 3:fichier de configuration des services	5
Figure 4:déploiement des services	5
Figure 5:vérification du lancement des services.....	6
Figure 6:: certificats générer par Setup certificates	7
Figure 7:fonctionnement du metricbeat	7
Figure 8:teste de fonctionnement de filebeat.....	8
Figure 9:este de fonctionnement de logstash	9

1. Introduction

1.1. Contexte et Objectifs

Les systèmes modernes génèrent de grandes quantités de logs, contenant des informations précieuses pour le suivi, la sécurité, et l'optimisation des performances. Gérer ces données efficacement est un défi majeur pour les administrateurs et les équipes de sécurité. Dans ce contexte, l'architecture ELK (Elasticsearch, Logstash, Kibana) est une solution largement utilisée pour centraliser, traiter et visualiser ces données de manière rapide et accessible.

L'objectif de ce rapport est de documenter la mise en œuvre d'une architecture ELK, en partant de l'analyse des besoins jusqu'à l'évaluation des résultats, afin de faciliter l'analyse des logs et la détection d'anomalies.

2. Présentation de l'Architecture ELK

2.1. Description des Composants ELK (Elasticsearch, Logstash, Kibana, Metricbeat, Filebeat, Setup Certificates)

L'architecture ELK repose sur trois composants principaux : Elasticsearch, Logstash, et Kibana. Elle est souvent étendue avec d'autres outils comme Metricbeat et Filebeat pour la collecte de données et l'analyse.

- **Elasticsearch** : Un moteur de recherche et d'analyse distribué utilisé pour stocker et indexer les données de logs. Il permet de rechercher rapidement des informations spécifiques dans de grandes quantités de données.
- **Logstash** : Un pipeline de traitement de données qui collecte, filtre et envoie les logs vers Elasticsearch. Il est hautement configurable et peut traiter les données issues de plusieurs sources.
- **Kibana** : Un outil de visualisation qui permet d'afficher les données stockées dans Elasticsearch sous forme de graphiques, tableaux de bord, et alertes. Kibana rend les données plus accessibles pour une analyse en temps réel.
- **Metricbeat** : Un outil léger qui collecte les métriques des systèmes et services (CPU, mémoire, réseau, etc.) et les envoie à Elasticsearch pour surveillance et analyse.
- **Filebeat** : Un collecteur léger qui envoie les logs des fichiers directement vers Elasticsearch ou Logstash, facilitant l'ingestion des données de journaux sans surcharge du système.

- **Setup Certificates** : La configuration des certificats SSL/TLS pour sécuriser les échanges de données entre les composants de l'architecture ELK, garantissant ainsi la confidentialité et l'intégrité des informations transmises.

2.2. Avantages et Limitations de l'Architecture ELK

2.2.1. Avantages :

- **Centralisation** : Facilite la collecte et la gestion des logs de différentes sources en un seul endroit.
- **Analyse en Temps Réel** : Avec Kibana, les utilisateurs peuvent surveiller et analyser les données en temps réel.
- **Scalabilité** : Elasticsearch est conçu pour gérer des volumes importants de données et peut être étendu pour répondre aux besoins croissants.

2.2.2. Limitations :

- **Consommation de Ressources** : Les composants ELK peuvent être gourmands en mémoire et CPU, surtout avec de gros volumes de données.
- **Configuration Complexe** : La configuration et le déploiement d'ELK peuvent nécessiter des compétences techniques et du temps, surtout pour des fonctionnalités avancées comme la sécurité.
- **Sécurisation** : Nécessite des configurations supplémentaires, comme les certificats SSL/TLS, pour assurer la sécurité des données en transit.

3. Analyse des Besoins et Objectifs

3.1. Identification des Besoins Techniques et Fonctionnels

Dans cette partie, nous identifions les besoins techniques et fonctionnels que l'architecture ELK doit satisfaire pour réussir. Ces besoins incluent :

- **Collecte de Logs** : Le système doit pouvoir collecter des logs de différents serveurs, applications et dispositifs réseau.
- **Analyse des Données** : Une capacité d'analyse avancée pour extraire des informations pertinentes à partir des logs collectés.
- **Visualisation** : Les utilisateurs doivent pouvoir visualiser les données sous forme de tableaux de bord interactifs et graphiques dans Kibana.
- **Alertes et Notifications** : L'architecture doit être capable de générer des alertes en fonction d'événements ou d'anomalies détectées dans les données.

3.2. Spécifications Non-Fonctionnelles (Sécurité, Performances, Évolutivité)

Les spécifications non-fonctionnelles de l'architecture ELK comprennent des exigences telles que :

- **Sécurité** : Le système doit assurer la protection des données, en particulier via des protocoles sécurisés comme SSL/TLS pour le chiffrement des communications.
- **Performances** : La capacité de traiter et d'analyser un grand volume de logs en temps réel tout en maintenant des temps de réponse optimaux.
- **Évolutivité** : L'architecture doit être capable de s'adapter à une augmentation du volume de données collectées et analysées, et de s'étendre en fonction des besoins futurs.

3.3. Enjeux et Contraintes du Projet

Plusieurs enjeux et contraintes doivent être pris en compte pour assurer la réussite du projet, notamment :

- **Volume de Données** : La gestion efficace d'un grand nombre de logs en temps réel, sans nuire aux performances du système.
- **Sécurité** : Garantir l'intégrité des données collectées et le respect des meilleures pratiques de sécurité.
- **Scalabilité** : S'assurer que l'architecture puisse évoluer pour s'adapter à la croissance future des données ou du nombre d'utilisateurs.

4. Conception de l'Architecture

4.1. Schéma d'Architecture ELK

L'architecture ELK a été déployée sur une machine unique fonctionnant sous **RedHat 9.3**, avec l'utilisation de **Docker** et **Docker Compose** pour orchestrer les différents services. Les composants suivants ont été installés et configurés sur cette machine :

- **Elasticsearch** : Pour l'indexation, le stockage et la recherche des données.
- **Kibana** : Pour la visualisation et l'analyse des données indexées par Elasticsearch.
- **Logstash** : Pour la collecte, la transformation et l'envoi des logs vers Elasticsearch.
- **Filebeat** : Pour la collecte des logs à partir des systèmes cibles et leur envoi à Logstash ou directement à Elasticsearch.

- **Metricbeat** : Pour la collecte des métriques systèmes (CPU, mémoire, réseau, etc.) et leur envoi vers Elasticsearch ou Logstash.
- **Setup Certificates**: Pour sécuriser les communications entre tous les composants ELK (Elasticsearch, Logstash, Kibana, et les Beats) grâce à des certificats.

Le schéma d'architecture montre comment ces composants communiquent entre eux, avec **Filebeat** et **Metricbeat** collectant les logs et métriques des systèmes distants, **Logstash** traitant et envoyant ces données vers **Elasticsearch** pour l'indexation, et **Kibana** permettant de visualiser les résultats. Le **setup certificates** est utilisé pour sécuriser ces communications entre les services à travers des canaux chiffrés.

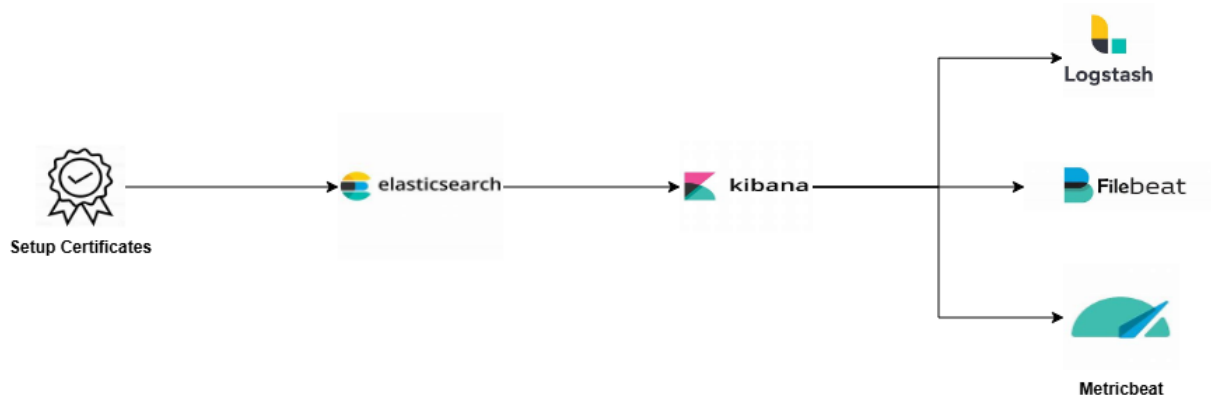


Figure 1 : Schéma d'Architecture ELK

5. Mise en Œuvre de l'Architecture ELK

5.1. Préparation de l'Environnement (Docker et Docker Compose)

La mise en œuvre de l'architecture ELK repose sur une machine virtuelle RedHat 9.3 où Docker et Docker Compose sont utilisés pour déployer les différents services de l'architecture. Voici les étapes principales de la préparation de l'environnement :

- **Installation de Docker** : Docker a été installé sur la machine virtuelle pour permettre l'exécution des conteneurs des différents services de l'architecture ELK (Elasticsearch, Logstash, Kibana, Filebeat, Metricbeat).
- **Installation de Docker Compose** : Docker Compose est utilisé pour simplifier l'orchestration des services et leur interconnexion. Il permet de définir tous les services nécessaires dans un seul fichier de configuration (docker-compose.yml), facilitant ainsi leur gestion et déploiement.

```
[root@localhost elk]# docker --version
Docker version 27.3.1, build cel2230
[root@localhost elk]# docker-compose --version
Docker Compose version v2.30.1
[root@localhost elk]#
```

Figure 2: installation docker & docker-compose

Un fichier docker-compose.yml a été préparé pour définir et orchestrer l'ensemble des services (setup certificates, Elasticsearch, Kibana, Logstash, metricbeat, filebeat)

```
[root@localhost elk]# ls
docker-compose.yml  filebeat.yml  logstash_ingest_data  logstash_ingest_data
filebeat_ingest_data  kibana.yml  logstash.conf  metricbeat.yml
```

Figure 3: fichier de configuration des services

5.2. Déploiement des services

Le déploiement de l'architecture ELK a été effectué sur une machine virtuelle RedHat 9.3, utilisant Docker et Docker Compose pour orchestrer les différents services nécessaires. Ce processus a impliqué le déploiement successif des composants de l'architecture, en commençant par la configuration des certificats SSL/TLS pour sécuriser les communications entre les services. Ensuite, **Elasticsearch**, **Kibana**, **Logstash**, **Filebeat**, et **Metricbeat** ont été déployés, chacun étant configuré pour fonctionner de manière sécurisée et efficace au sein de l'architecture ELK.

```
[root@localhost elk]# docker-compose up --build -d
WARN[0000] /home/nasri/elk/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion
[+] Running 7/7
 ✓ Network elastic          Created           0.5s
 ✓ Container es-setup-1     Healthy         2.4s
 ✓ Container es-es01-1     Healthy        46.6s
 ✓ Container es-filebeat01-1 Started        45.9s
 ✓ Container es-kibana-1   Healthy        98.5s
 ✓ Container es-metricbeat01-1 Started       100.6s
 ✓ Container es-logstash01-1 Started       100.6s
[root@localhost elk]#
```

Figure 4: déploiement des services

5.2.1. Déploiement de Setup Certificates

Le service **setup certificates** a été déployé pour générer et fournir des certificats nécessaires à la sécurisation des communications entre les différents services de l'architecture ELK. Ces certificats permettent à Elasticsearch, Kibana, Logstash, Metricbeat et Filebeat de communiquer de manière sécurisée.

5.2.2. Déploiement de Elasticsearch

Grâce aux certificats fournis par le service **setup certificates**, Elasticsearch a été déployé et configuré pour accepter les connexions sécurisées. Le service a été configuré pour indexer et stocker les données provenant des autres composants.

5.2.3. Déploiement de Kibana

Kibana a été configuré pour se connecter à Elasticsearch de manière sécurisée, en utilisant les certificats fournis. Il permet ainsi de visualiser les données collectées par Elasticsearch dans des dashboards interactifs.

```
[root@localhost elk]# docker ps
```

CONTAINER ID	IMAGE	PORTS	COMMAND	NAMES	CREATED	STATUS
e7b4da5840b2	docker.elastic.co/logstash/logstash:8.8.2	5044/tcp, 9600/tcp	"/usr/local/bin/dock...	es-logstash01-1	35 minutes ago	Up 34
57d9ced66667	docker.elastic.co/beats/metricbeat:8.8.2		"/usr/bin/tini -- /u..."	es-metricbeat01-1	35 minutes ago	Up 34
d31b2a76ff15	docker.elastic.co/beats/filebeat:8.8.2		"/usr/bin/tini -- /u..."	es-filebeat01-1	35 minutes ago	Up 34
5d13fe956b69	docker.elastic.co/kibana/kibana:8.8.2		"/bin/tini -- /usr/l..."	es-kibana-1	35 minutes ago	Up 34
aabbf8fac1e4	docker.elastic.co/elasticsearch/elasticsearch:8.8.2	0.0.0.0:5601->5601/tcp, ::5601->5601/tcp	"/bin/tini -- /usr/l..."	es-es01-1	35 minutes ago	Up 35

```
[root@localhost elk]#
```

Figure 5: vérification du lancement des services

5.2.4. Déploiement de Logstash, Filebeat et Metricbeat

Logstash, Filebeat et Metricbeat ont été déployés pour collecter et envoyer les données vers Elasticsearch, avec une communication sécurisée via les certificats fournis par **setup certificates**. Ces services assurent la collecte des logs et des métriques, qui sont ensuite traités et stockés dans Elasticsearch.

6. Tests et Validation de l'Architecture

Pour garantir le bon fonctionnement et la fiabilité de l'architecture ELK déployée, une série de tests a été effectuée.

```
[root@localhost data]# ls
ca ca.zip certs.zip es01 fleet-server instances.yml kibana
```

Figure 6:: certificats générés par Setup certificates

Dernière partie réalisée : ajout d'un fleetserver APM elastic agent et une app de test :

```
[root@localhost elk]# docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS
566f50f11a95   docker.elastic.co/beats/metricbeat:8.8.2   "/usr/bin/tini -- /u..."   3 minutes ago   Up Ab...
a4068f8852fa   docker.elastic.co/kibana/kibana:8.8.2     "/bin/tini -- /usr/l..."   About an hour ago   Up 2 t...
18bccfd5a1b4   docker.elastic.co/elasticsearch/elasticsearch:8.8.2   "/bin/tini -- /usr/l..."   About an hour ago   Up 3 t...
```

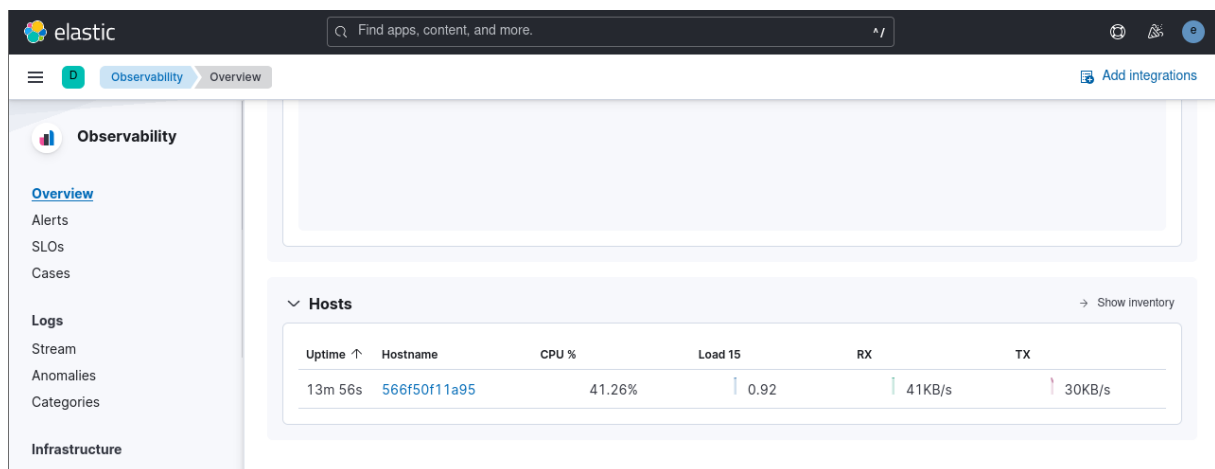
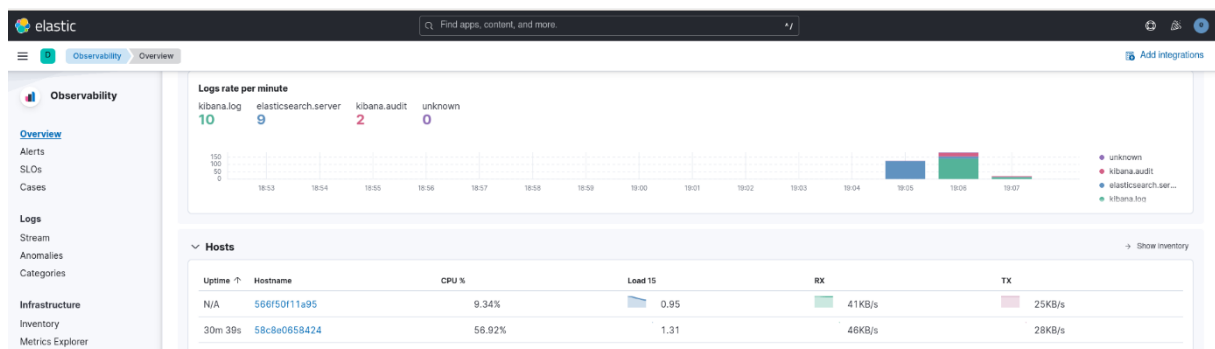


Figure 7: fonctionnement du metricbeat



```
[root@localhost elk]# cp /var/log/cron filebeat_ingest_data/cron.log
[root@localhost elk]# nano filebeat_ingest_data/cron.log
```

```
Nov  6 23:01:01 localhost systemd[1001]: Normal exit (0 jobs ran)
Nov  6 23:01:01 localhost CROND[18605]: (root) CMDEND (run-parts /etc/cron.hourly project-elk)
```

```
Nov  6 23:01:01 localhost CROND[18605]: (root) CMDEND (run-parts /etc/cron.hourly project-elk)
```

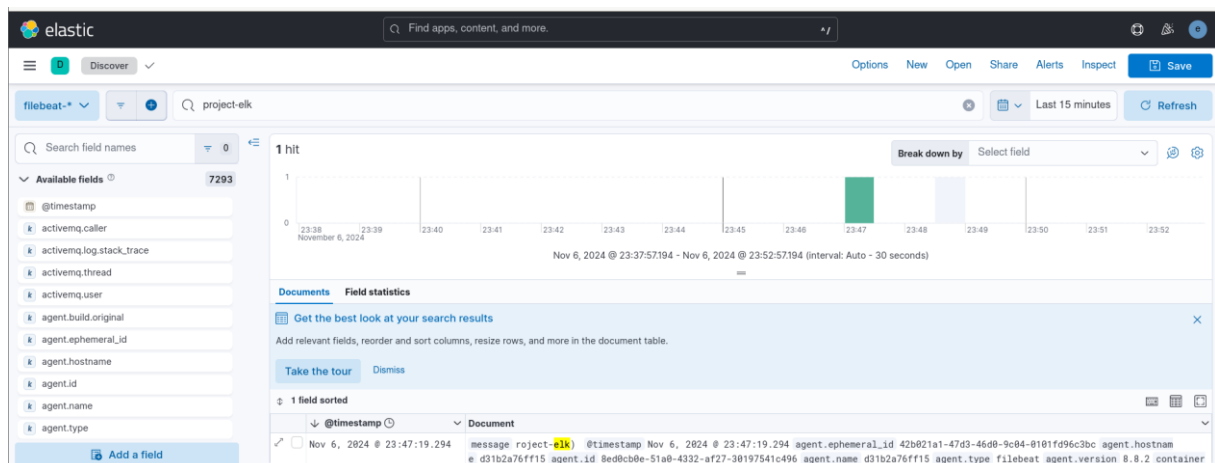


Figure 8: teste de fonctionnement de filebeat

```
GNU nano 5.6.1                                logstash_ingest_data/cron.log
Nov  3 15:27:24 localhost crond[1407]: (CRON) STARTUP (1.5.7)

Nov  6 23:01:01 localhost CROND[18605]: (root) CMDEND (run-parts /etc/cron.hourly project-elk)
Nov  6 23:01:01 localhost CROND[18605]: (root) CMDEND (run-parts /etc/cron.hourly nasri-wissem)
```

The screenshot shows the Elastic UI interface for Index Management. The page displays a table of indices with columns: Name, Health, Status, Primaries, Replicas, Docs count, Storage size, and Data stream. The table shows two indices: 'logstash-2024.11.03' and 'logstash-2024.11.06'. Both indices have a 'yellow' health status and are 'open'. The 'logstash-2024.11.03' index has 1 primary, 1 replica, 5 docs, and a storage size of 24.13kb. The 'logstash-2024.11.06' index has 1 primary, 1 replica, 74 docs, and a storage size of 87.61kb.

Name	Health	Status	Primaries	Replicas	Docs count	Storage size	Data stream
logstash-2024.11.03	yellow	open	1	1	5	24.13kb	
logstash-2024.11.06	yellow	open	1	1	74	87.61kb	

```
1 GET /logstash-2024.11.06/_search

{
  "value": 5,
  "relation": "eq",
  "max_score": 1,
  "hits": [
    {
      "_index": "logstash-2024.11.06",
      "_id": "TSDMASMBnZldJTtgIj5A",
      "_score": 1,
      "_source": {
        "message": "Nov 7 00:01:02 localhost CROND[44900]: (root) CMDEND (run\n-parts /etc/cron.hourly wissem-nasr)",
        "event": {
          "original": "Nov 7 00:01:02 localhost CROND[44900]: (root) CMDEND (run\n-parts /etc/cron.hourly wissem-nasr)",
          "log": {
            "file": {
              "path": "/usr/share/logstash/ingest_data/cron.log"
            },
            "@version": "1",
            "host": {
              "name": "06542c4b565d"
            },
            "@timestamp": "2024-11-06T23:25:17.813263792"
          }
        }
      }
    }
  ]
}
```

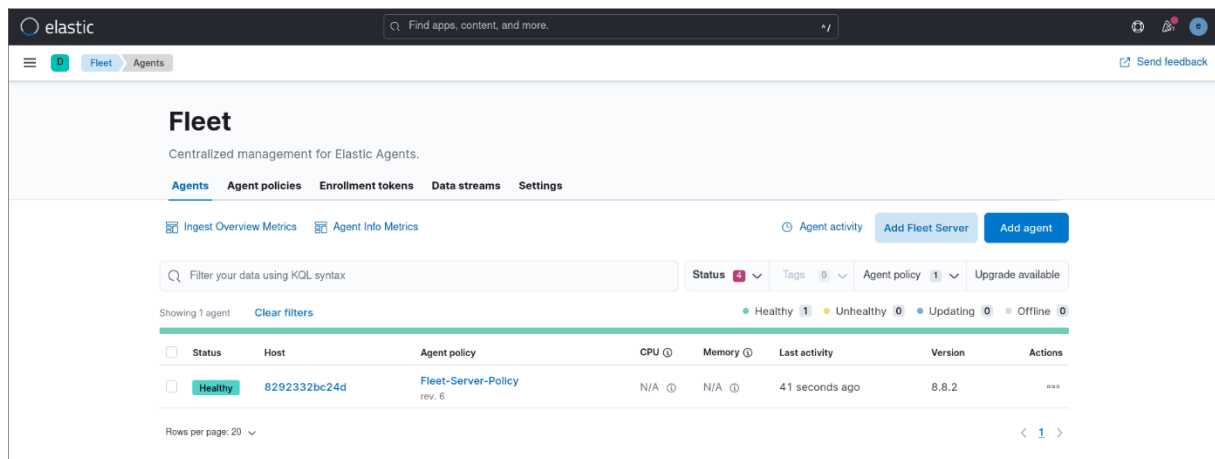
```
1 GET /logstash-2024.11.06/_search

{
  "name": "06542c4b565d",
  "@timestamp": "2024-11-06T23:25:17.811207498Z",
  "_index": "logstash-2024.11.06",
  "_id": "SyDMASMBnZldJTtgIj5A",
  "_score": 1,
  "_source": {
    "message": "Nov 3 15:27:24 localhost crond[1407]: (CRON) STARTUP (1.5.7)",
    "event": {
      "original": "Nov 3 15:27:24 localhost crond[1407]: (CRON) STARTUP (1.5.7)",
      "log": {
        "file": {
          "path": "/usr/share/logstash/ingest_data/cron.log"
        },
        "@version": "1",
        "host": {
          "name": "06542c4b565d"
        },
        "@timestamp": "2024-11-06T23:25:17.810210779Z"
      }
    }
  }
}
```

```
1 GET /logstash-2024.11.06/_search

{
  "_index": "logstash-2024.11.06",
  "_id": "TCDMASMBnZldJTtgIj5A",
  "_score": 1,
  "_source": {
    "message": "Nov 6 23:01:01 localhost CROND[18605]: (root) CMDEND (run\n-parts /etc/cron.hourly project-elk)",
    "event": {
      "original": "Nov 6 23:01:01 localhost CROND[18605]: (root) CMDEND (run\n-parts /etc/cron.hourly project-elk)",
      "log": {
        "file": {
          "path": "/usr/share/logstash/ingest_data/cron.log"
        },
        "@version": "1",
        "host": {
          "name": "06542c4b565d"
        },
        "@timestamp": "2024-11-06T23:25:17.810210779Z"
      }
    }
  }
}
```

Figure 9:este de fonctionnement de logstash



```
[root@localhost elk]# docker ps --format "table {{.ID}}\t{{.Names}}\t{{.Status}}"
CONTAINER ID      NAMES                STATUS
176e8f8ef00e      es-metricbeat01-1    Up 37 minutes
6b0307635ddc      es-logstash01-1      Up 37 minutes
8292332bc24d      es-fleet-server-1     Up 37 minutes
34695f0801c3      es-kibana-1           Up 29 minutes (healthy)
4b9ff3a7acab      es-filebeat01-1      Up 39 minutes
08e699d3e124      es-es01-1            Up 39 minutes (healthy)
[root@localhost elk]# docker cp es-es01-1:/usr/share/elasticsearch/config/certs/ca/ca.crt ./
Successfully copied 3.07kB to /home/nasri/elk/.
[root@localhost elk]# ls
ca.crt                filebeat_ingest_data kibana.yml            logstash.conf         metricbeat.yml
docker-compose.yml    filebeat.yml          log                   logstash_ingest_data
[root@localhost elk]#

[root@localhost elk]# openssl x509 -fingerprint -sha256 -noout -in ./ca.crt | awk -F"=" '{ print $2 }' | sed s://g
B87FEA90B847B9EE79BD0D7C93559A54B07C3968B8F27EF02A5F1730D9EF45C
[root@localhost elk]#

[root@localhost elk]# docker cp es-es01-1:/usr/share/elasticsearch/config/certs/ca/ca.crt ./
Successfully copied 3.07kB to /home/nasri/elk/.
[root@localhost elk]# ls
ca.crt                filebeat_ingest_data kibana.yml            logstash.conf         metricbeat.yml
docker-compose.yml    filebeat.yml          log                   logstash_ingest_data
[root@localhost elk]# cat ca.crt
-----BEGIN CERTIFICATE-----
MIIDSTCCAjGgAwIBAgIUbZsuRwmVvhH8XpYwtt7h45YS1pEwDQYJKoZIhvcNAQEL
BQAwNDEyMDA1UEAxMhbnVhbnRlc3R5YyBDZXJ0aWZpY2F0ZSB1b29sIEF1dG9nZW5l
cmF0ZWQgQ0EwHhcNMjQxMTA3MTU0NTA3WWhcNMjQxMTA3MTU0NTA3WjA0MTIwMAYD
VQDEYlFbGZdGljIENlcjZmZmljYXRlIFRvb2wgQXV0b2dldmVvYXRlZCB0TCC
ASiWdQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBALafPK6MG6BVlTcAmd2fUhcA
b+EMVMQTi02qDtAkVLA7c9qaYxzG2CIDuVJ/JuLdJoHVKiDqLS/Uoo2XbVYUdmcz
EN464Nw+KfUAhZAM9XRYU7BDmv7HkhpZN9V1haw7paEyKI1kgL6DRFn8mM+qn23a
4e7qrOhbi84VqWvNDTQBFguxe80HATFJbWYgwXFG6g7HSp+igbtEvX4LTxdUcb5z
Iw6V3y/ojcyXVsi62k6TuvKpKGfMrAws/WefK86fvRtPBWG5CJgYKDP1Ax/1rbTR
oa0LmASUDTNkYI6yx8F33myHHinxXHuB+LkCcCic6HGwPflPrdB/KLCJ7uebEwUC
AwEAAaNTMFEwHQYDVIR00BBYEFKUXsMSvaexXN+8UF8sJwz9iH6SjMB8GA1UdIwQY
MBaAFKUXsMSvaexXN+8UF8sJwz9iH6SjMA8GA1UdEwEB/wQFMAMBAf8wDQYJKoZI
hvcNAQELBQADggEBAI/BnzWT8EEllrx1MjdVuqXXY1LC8Tf4tXP2BiP4fcj2FV0a
00xRCMHhmbpWpmZZIkpHZZRjaL1monyepNl0/f4dh3dSSlAbAljTSZQSn9P0h3yP
5gmjszcBV+jCFwPW5f7LaE+LGfsB/yમેય80Pz/wF2Kg7iUYHgGYItcf/00Z6ojkl
bo4dC7ek/r9iV9BsexV+79lx8A9mD+kJ+u1c+gcDdCrAUDV53EstCkpc9WRCvHAM
H07YT4FpxqzXkFwRGueb4UR6BRnPE4FKUhB3BQRtWA1cMzLYo82J3LQz0ecErH/G
2dea3eiRbdc3UibYGLU6WFAOU6QfEWfD3T0xG2I=
-----END CERTIFICATE-----
```

elastic

Find apps, content, and more.

^/

e

D

Fleet

Settings

Send feedback

Add Fleet Server

Outputs

Specify where agents will send data.

Name	Type
default	Elasticsearch

Add output

Agent Binary Download

Specify where the agents will download their binary and where it will be overwritten.

Name	Host
Elastic Artifacts	https://artifacts.elastic.co

Edit output

Add another URL

Elasticsearch CA trusted fingerprint (optional)

ID7C93559A54B07C3968B8F27EF02A5F1730D9EF45C

Proxy

Select proxy

Advanced YAML configuration

-----BEGIN CERTIFICATE-----
D040C/EK/T91V9BSEXYT/91X8A9MDTKJTUIC
+gcDdCrAUDV53EstCkpc9WRCvHAM
H07YT4FpxqzXkFwRGueb4UR6BRnPE4FKUhB3BQRtWA1cMz
1Yo82J3LQz0ecErH/G
2dea3eiRbdc3UibYGLU6WfFa0U6QfEWfD3T0xG2I=
-----END CERTIFICATE-----

Cancel

Save and apply settings

elastic

Find apps, content, and more.

e

D

Fleet

Agents

Send feedback

Fleet

Centralized management for Elastic Agents.

Agents

Agent policies

Enrollment tokens

Data streams

Settings

Ingest Overview Metrics

Agent Info Metrics

Agent activity

Add Fleet Server

Add agent

Filter your data using KQL syntax

Status

Tags

Agent policy

Upgrade available

Showing 1 agent

Clear filters

Healthy

Unhealthy

Updating

Offline

	Status	Host	Agent policy	CPU	Memory	Last activity	Version	Actions
	Healthy	8292332bc24d	Fleet-Server-Policy rev. 7	11.81 %	573 MB	1 minute ago	8.8.2	...

Rows per page: 20

elastic

Find apps, content, and more.

e

D

Fleet

Agent policies

Fleet-Server-Policy

Send feedback

View all agent policies

Revision 7

Integrations 4

Agents 1 agent

Last updated on Nov 22, 2024

Actions

Integrations

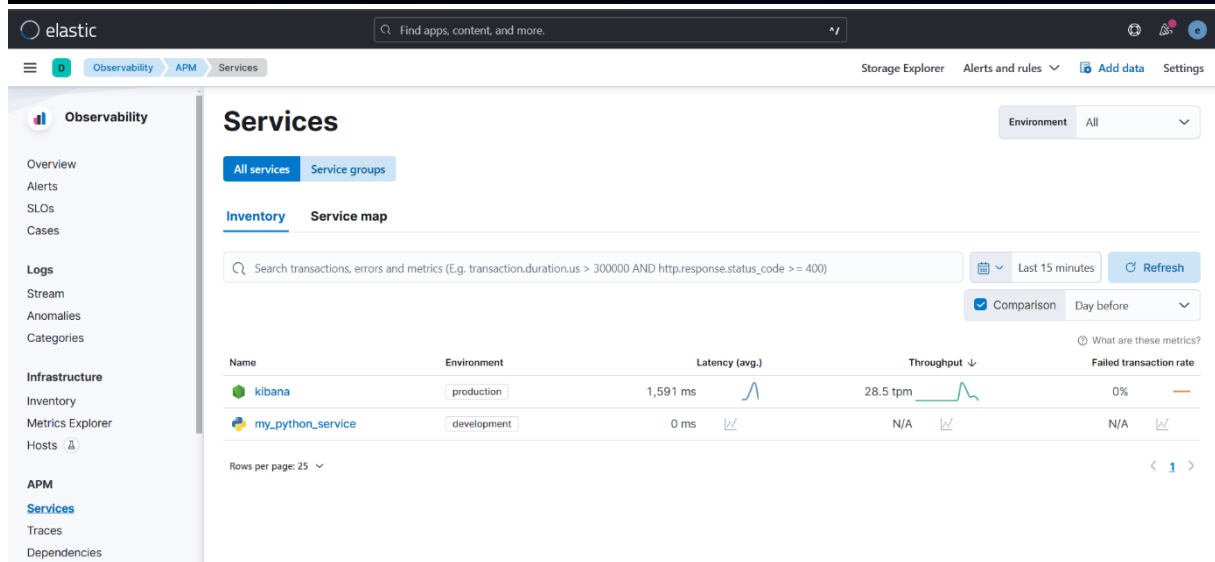
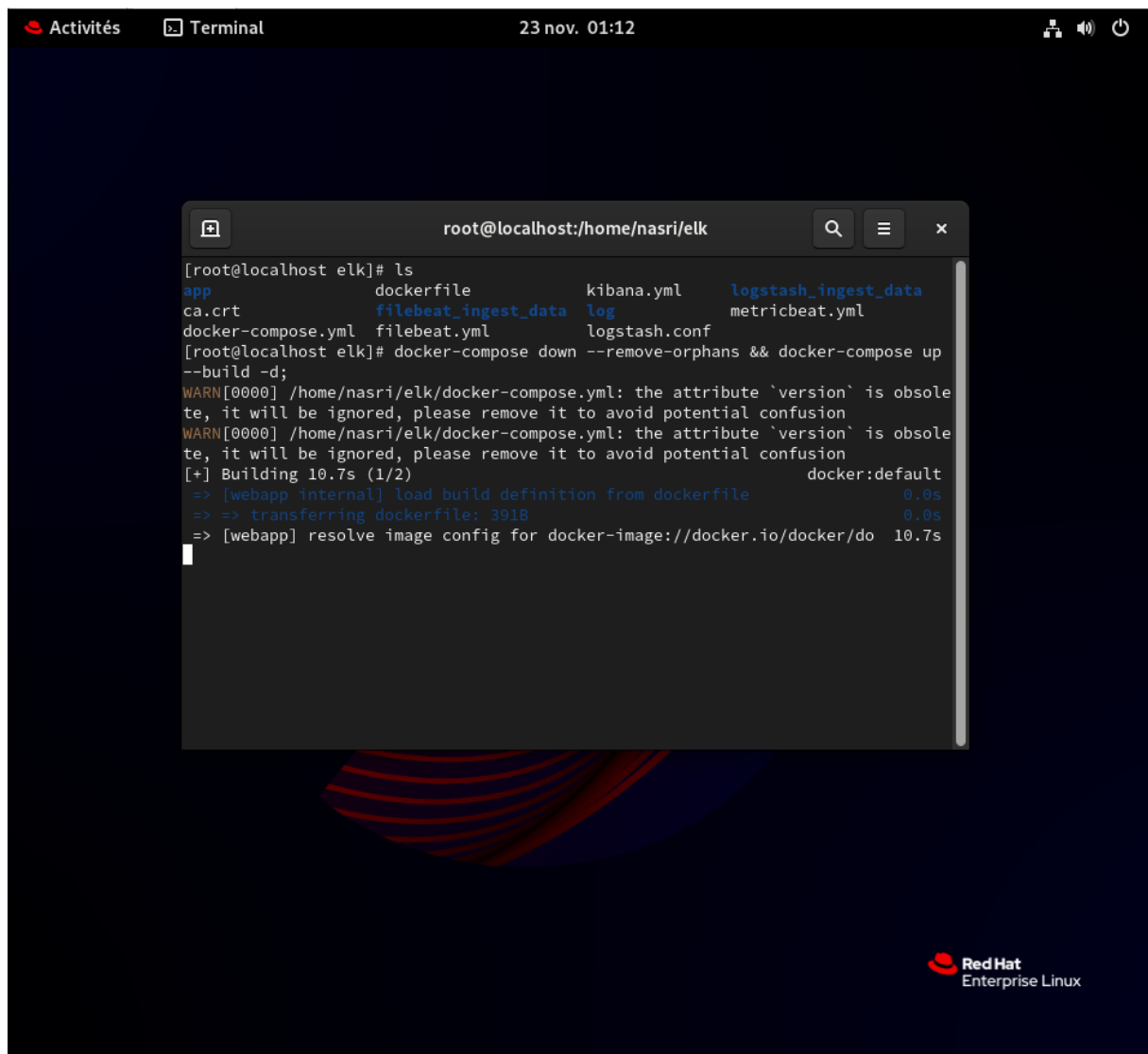
Settings

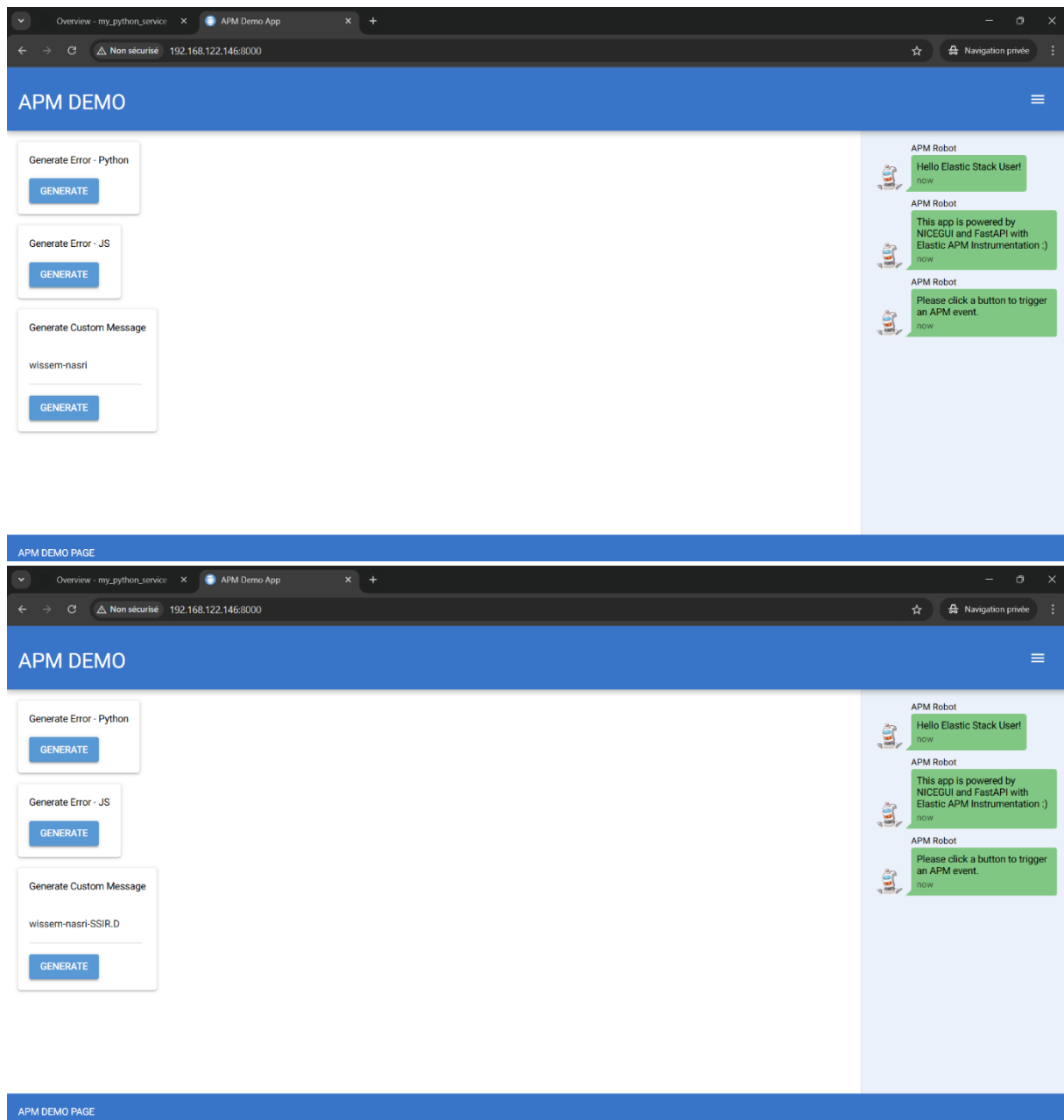
Search...

Namespace

Add Integration

Name	Integration	Namespace	Actions
apm-1	Elastic APM v8.8.2	default	...
elastic_agent-1	Elastic Agent v1.10.0	default	...
fleet_server-1	Fleet Server v1.3.1	default	...
system-1	System v1.39.0	default	...





Overview - my_python_service

APM Demo App

Non sécuriséhttps://192.168.122.146:5601/app/apm/services/my_python_service/overview?comparisonEnabled=true&environment=ENVIRONMENT_ALL&query=&latencyAggregatio...Navigation privée

elasticFind apps, content, and more.

ObservabilityAPMServicesmy_python_serviceOverviewStorage ExplorerAlerts and rulesAdd dataSettings

Observability

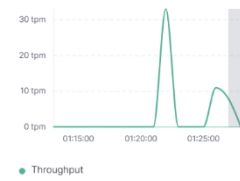
OverviewAlertsSLOsCases

LogsStreamAnomaliesCategories

InfrastructureInventoryMetrics ExplorerHosts

APMServicesTracesDependencies

Throughput



30 tpm
20 tpm
10 tpm
0 tpm

01:15:0001:20:0001:25:00

ThroughputDay before


Transactions

View transactions

Name	Latency (avg.)	Throughput	Failed transaction rate	Impact
GET /nicegu/1.3.2/static	254 ms	1.6 tpm	0%	
GET /	2,236 ms	0.1 tpm	0%	
GET /custom_message/	156 ms	0.6 tpm	0%	
GET /nicegu/1.3.2/component...	270 ms	0.2 tpm	0%	
GET /error	22 ms	0.9 tpm	0%	

< 1 >

Failed transaction rate



100%
50%
0%

01:15:0001:20:0001:25:00

Errors

View errors

Type	Name	Last seen	Occurrences
ZeroDivisionError	ZeroDivisionError: division...	2 minutes ago	14 occ.
TimeoutError	TimeoutError: JavaScript ...	2 minutes ago	12 occ.
	Custom Message: wissem...	2 minutes ago	5 occ.
	Custom Message: wissem...	a minute ago	5 occ.

0183f3b7c7d89ae69e121b2cd7ced33APM Demo App

Non sécuriséhttps://192.168.122.146:5601/app/apm/services/my_python_service/errors/0183f3b7c7d89ae69e121b2cd7ced33?query=&rangeFrom=now-15m&rangeTo=now&enviro...Navigation privée

elasticFind apps, content, and more.

ObservabilityAPMServicesmy_python_serviceErrors0183f3b7c7d89ae69e121b2cd7ced33Storage ExplorerAlerts and rulesAdd dataSettings

Observability

OverviewAlertsSLOsCases

LogsStreamAnomaliesCategories

InfrastructureInventoryMetrics ExplorerHosts

APMServicesTracesDependencies

Error occurrences0Day before0

Error sample< 1 of 5 >View 5 occurrences in Discover

2 minutes ago | GET http://localhost:8000/custom_message/wissem-nasri | Other | GET /custom_message/{message} | development

Log message

Custom Message: wissem-nasri

Exception message

N/A

Culprit

__main__ - <module>

Log stack traceException stack traceMetadata

