TP - Monitoring des logs d'une application Spring Boot avec Filebeat, Logstash, Elasticsearch et Kibana

Réalisé par **Mr. Iyanou Eraste AKANDE**, Ingénieur des données Telecom à Synaptique Maghreb, Ingénieur Certifié Elasticsearch.

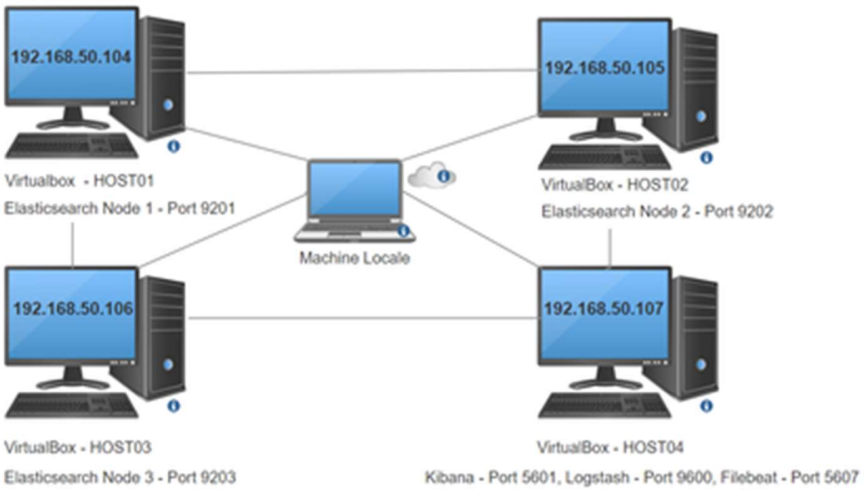
eraste.akande@gmail.com

Etape 1 : Introduction à la pile ELK

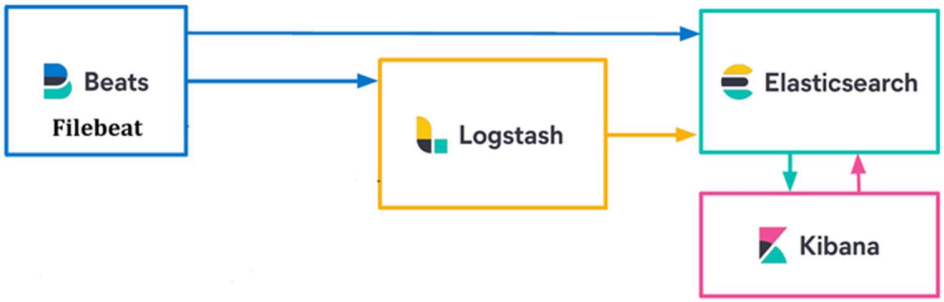
Présentation des composantes de la pile ELK, le rôle de chaque composante ainsi que les différents cas d'utilisation → (PPT Introduction to Elastic Stack).

Etape 2 : Architecture

Les machines virtuelles



Les applications



Etape 3 : Installation et configuration des machines virtuelles

- Téléchargez et installez la version de Virtual Box compatible à votre système d'exploitation sur le site https://www.virtualbox.org/wiki/Downloads.
- Dans certains cas pour les utilisateurs Windows, vous devriez installer au préalable Visual Studio C++ Redistributable à partir du site https://learn.microsoft.com/en-us/cpp/windows/latest-supported-vc-redist?view=msvc-170.
- Téléchargez et installez la version de Vagrant compatible à votre système d'exploitation à partir du site https://developer.hashicorp.com/vagrant/install?product_intent=vagrant.
- Téléchargez le fichier nommé Vagrantfile depuis ce répertoire GitHub https://github.com/iyanou/elastic-training-lab.
- Placez le fichier Vagrantfile dans un dossier et exécutez ce qui suit depuis cet emplacement

```
cd $vagrant_file_repository
vagrant up
```

- Se connecter sur chaque machine et vérifier qu'on arrive a faire un ping vers les autres adresses IP

```
ping -c 10 HOST01
ping -c 10 HOST02
ping -c 10 HOST03
ping -c 10 HOST04
```

Etape 4 : Installation et configuration des applications

1. Les nœuds Elasticsearch

HOST01

- Connexion SSH au 192.168.50.104
- Créer un utilisateur et l'ajouter au groupe `sudo`. Supposons que votre utilisateur s'appelle `training`.

```
sudo adduser training
sudo passwd training
sudo usermod -aG sudo training
su - training
```

Il ne faut pas exécuter Elasticsearch avec le user root pour des raisons de sécurité.

- Création du dossier d'installation

```
sudo mkdir -p /opt/training
```

- Téléchargement de Elasticsearch 8.13.0

```
cd /opt/training
sudo wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-
8.13.0-linux-x86_64.tar.gz
```

- Décompression de Elasticsearch 8.13.0

```
sudo tar -xzf elasticsearch-8.13.0-linux-x86_64.tar.gz
sudo rm elasticsearch-8.13.0-linux-x86_64.tar.gz
```

- Modifier le fichier de configuration `elasticsearch.yml`.

```
cd /opt/training/elasticsearch-8.13.0
sudo rm config/elasticsearch.yml
sudo nano config/elasticsearch.yml

#elasticsearch.yml
cluster.name: es-training
node.name: node01
network.host: 192.168.50.104
http.port: 9201
transport.port: 9301
discovery.seed_hosts: ["192.168.50.104:9301", "192.168.50.105:9302",
"192.168.50.106:9303"]
cluster.initial_master_nodes: ["node01", "node02", "node03"]
#Enable SECURITY
xpack.security.enabled: true
xpack.security.authc.api_key.enabled: true
#Secure HTTP
xpack.security.http.ssl.enabled: true
xpack.security.http.ssl.key: certificates/node01/node01.key
xpack.security.http.ssl.certificate: certificates/node01/node01.crt
xpack.security.http.ssl.certificate_authorities: certificates/ca/ca.crt
#Secure TRANSPORT
```

```
xpack.security.transport.ssl.enabled: true
xpack.security.transport.ssl.key: certificates/node01/node01.key
xpack.security.transport.ssl.certificate: certificates/node01/node01.crt
xpack.security.transport.ssl.certificate_authorities: certificates/ca/ca.crt
xpack.security.transport.ssl.verification_mode: certificate
```

- Mettre à jour le paramètre `vm.max_map_count`

```
sudo grep vm.max_map_count /etc/sysctl.conf
grep 'vm.max_map_count=262144' /etc/sysctl.conf || sudo echo
'vm.max_map_count=262144' | sudo tee -a /etc/sysctl.conf
sudo sysctl -p
```

- Générer les certificats de connexion elasticsearch en utilisant le fichier `instances.yml`

```
cd /opt/training/elasticsearch-8.13.0
sudo mkdir config/certificates
sudo nano config/certificates/instances.yml

#instances.yml
instances:
  - name: node01
    ip:
      - 192.168.50.104
  - name: node02
    ip:
      - 192.168.50.105
  - name: node03
    ip:
      - 192.168.50.106
  - name: kibana
    ip:
      - 192.168.50.107

sudo bin/elasticsearch-certutil ca --silent --pem -out
config/certificates/ca.zip
sudo apt install unzip
sudo unzip -o config/certificates/ca.zip -d config/certificates
sudo bin/elasticsearch-certutil cert --silent --pem -out
config/certificates/bundle.zip --in config/certificates/instances.yml --ca-
cert config/certificates/ca/ca.crt --ca-key config/certificates/ca/ca.key
sudo unzip -o config/certificates/bundle.zip -d config/certificates
sudo rm config/certificates/bundle.zip
sudo rm config/certificates/ca.zip
```

- Autoriser les ports sur le pare-feu

```
sudo ufw enable
sudo ufw allow 9201/tcp
sudo ufw allow 9301/tcp
sudo ufw status
```

- Donner les permissions à l'utilisateur sur le dossier elasticsearch

```
sudo chown -R training:training /opt/training/elasticsearch-8.13.0
```

- Démarrer le nœud Elasticsearch et vérifier les logs

```
cd /opt/training/elasticsearch-8.13.0
./bin/elasticsearch
```

Pour arrêter Elasticsearch, exécutez `ctrl+c`. Donc c'est mieux de le démarrer comme un daemon.

- Démarrer le nœud Elasticsearch comme un daemon et vérifier les logs

```
./bin/elasticsearch -d -p pid
```

Pour arrêter Elasticsearch, exécutez `pkill -F pid`. Mais il faut noter que le daemon va s'arrêter si le host s'éteint et ne se relancera pas automatiquement au redémarrage.

HOST02

- Connexion SSH au 192.168.50.105
- Créer un utilisateur et l'ajouter au groupe `sudo`. Supposons que votre utilisateur s'appelle `training`.

```
sudo adduser training
sudo passwd training
sudo usermod -aG sudo training
su - training
```

Il ne faut pas exécuter Elasticsearch avec le user root pour des raisons de sécurité.

- Création du dossier d'installation

```
sudo mkdir -p /opt/training
```

- Téléchargement de Elasticsearch 8.13.0

```
cd /opt/training
sudo wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-
8.13.0-linux-x86_64.tar.gz
```

- Décompression de Elasticsearch 8.13.0

```
sudo tar -xzf elasticsearch-8.13.0-linux-x86_64.tar.gz
sudo rm elasticsearch-8.13.0-linux-x86_64.tar.gz
```

- Modifier le fichier de configuration `elasticsearch.yml`.

```
cd /opt/training/elasticsearch-8.13.0
sudo rm config/elasticsearch.yml
```

```
sudo nano config/elasticsearch.yml

#elasticsearch.yml
cluster.name: es-training
node.name: node02
network.host: 192.168.50.105
http.port: 9202
transport.port: 9302
discovery.seed_hosts: ["192.168.50.104:9301", "192.168.50.105:9302",
"192.168.50.106:9303"]
cluster.initial_master_nodes: ["node01", "node02", "node03"]
#Enable SECURITY
xpack.security.enabled: true
xpack.security.authc.api_key.enabled: true
#Secure HTTP
xpack.security.http.ssl.enabled: true
xpack.security.http.ssl.key: certificates/node02/node02.key
xpack.security.http.ssl.certificate: certificates/node02/node02.crt
xpack.security.http.ssl.certificate_authorities: certificates/ca/ca.crt
#Secure TRANSPORT
xpack.security.transport.ssl.enabled: true
xpack.security.transport.ssl.key: certificates/node02/node02.key
xpack.security.transport.ssl.certificate: certificates/node02/node02.crt
xpack.security.transport.ssl.certificate_authorities: certificates/ca/ca.crt
xpack.security.transport.ssl.verification_mode: certificate
```

- Mettre a jour le paramètre `vm.max_map_count`

```
sudo grep vm.max_map_count /etc/sysctl.conf
grep 'vm.max_map_count=262144' /etc/sysctl.conf || sudo echo
'vm.max_map_count=262144' | sudo tee -a /etc/sysctl.conf
sudo sysctl -p
```

- Copier les certificats depuis le HOST01

```
cd /opt/training/elasticsearch-8.13.0
sudo mkdir -p config/certificates/ca
sudo mkdir -p config/certificates/node02
sudo nano config/certificates/ca/ca.crt
sudo nano config/certificates/node02/node02.crt
sudo nano config/certificates/node02/node02.key
```

- Autoriser les ports sur le pare-feu

```
sudo ufw enable
sudo ufw allow 9202/tcp
sudo ufw allow 9302/tcp
sudo ufw status
```

- Donner les permissions à l'utilisateur sur le dossier elasticsearch

```
sudo chown -R training:training /opt/training/elasticsearch-8.13.0
```

- Démarrer le nœud Elasticsearch et vérifier les logs

```
cd /opt/training/elasticsearch-8.13.0
./bin/elasticsearch
```

Pour arrêter Elasticsearch, exécutez `ctrl+c`. Donc c'est mieux de le démarrer comme un daemon.

- Démarrer le nœud Elasticsearch comme un daemon et vérifier les logs

```
./bin/elasticsearch -d -p pid
```

Pour arrêter Elasticsearch, exécutez `pkill -F pid`. Mais il faut noter que le daemon va s'arrêter si le host s'éteint et ne se relancera pas automatiquement au redémarrage.

HOST03

- Connexion SSH au 192.168.50.106
- Créer un utilisateur et l'ajouter au groupe `sudo`. Supposons que votre utilisateur s'appelle `training`.

```
sudo adduser training
sudo passwd training
sudo usermod -aG sudo training
su - training
```

Il ne faut pas exécuter Elasticsearch avec le user root pour des raisons de sécurité.

- Création du dossier d'installation

```
sudo mkdir -p /opt/training
```

- Téléchargement de Elasticsearch 8.13.0

```
cd /opt/training
sudo wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-
8.13.0-linux-x86_64.tar.gz
```

- Décompression de Elasticsearch 8.13.0

```
sudo tar -xzf elasticsearch-8.13.0-linux-x86_64.tar.gz
sudo rm elasticsearch-8.13.0-linux-x86_64.tar.gz
```

- Modifier le fichier de configuration `elasticsearch.yml`.

```
cd /opt/training/elasticsearch-8.13.0
sudo rm config/elasticsearch.yml
sudo nano config/elasticsearch.yml
```

```
#elasticsearch.yml
cluster.name: es-training
node.name: node03
network.host: 192.168.50.106
http.port: 9203
transport.port: 9303
discovery.seed_hosts: ["192.168.50.104:9301", "192.168.50.105:9302",
"192.168.50.106:9303"]
cluster.initial_master_nodes: ["node01", "node02", "node03"]
#Enable SECURITY
xpack.security.enabled: true
xpack.security.authc.api_key.enabled: true
#Secure HTTP
xpack.security.http.ssl.enabled: true
xpack.security.http.ssl.key: certificates/node03/node03.key
xpack.security.http.ssl.certificate: certificates/node03/node03.crt
xpack.security.http.ssl.certificate_authorities: certificates/ca/ca.crt
#Secure TRANSPORT
xpack.security.transport.ssl.enabled: true
xpack.security.transport.ssl.key: certificates/node03/node03.key
xpack.security.transport.ssl.certificate: certificates/node03/node03.crt
xpack.security.transport.ssl.certificate_authorities: certificates/ca/ca.crt
xpack.security.transport.ssl.verification_mode: certificate
```

- Mettre à jour le paramètre `vm.max_map_count`

```
sudo grep vm.max_map_count /etc/sysctl.conf
grep 'vm.max_map_count=262144' /etc/sysctl.conf || sudo echo
'vm.max_map_count=262144' | sudo tee -a /etc/sysctl.conf
sudo sysctl -p
```

- Copier les certificats depuis le HOST01

```
cd /opt/training/elasticsearch-8.13.0
sudo mkdir -p config/certificates/ca
sudo mkdir -p config/certificates/node03
sudo nano config/certificates/ca/ca.crt
sudo nano config/certificates/node03/node03.crt
sudo nano config/certificates/node03/node03.key
```

- Autoriser les ports sur le pare-feu

```
sudo ufw enable
sudo ufw allow 9203/tcp
sudo ufw allow 9303/tcp
sudo ufw status
```

- Donner les permissions à l'utilisateur sur le dossier elasticsearch

```
sudo chown -R training:training /opt/training/elasticsearch-8.13.0
```

- Démarrer le nœud Elasticsearch et vérifier les logs

```
cd /opt/training/elasticsearch-8.13.0
./bin/elasticsearch
```

Pour arrêter Elasticsearch, exécutez `ctrl+c`. Donc c'est mieux de le démarrer comme un daemon.

- Démarrer le nœud Elasticsearch comme un daemon et vérifier les logs

```
./bin/elasticsearch -d -p pid
```

Pour arrêter Elasticsearch, exécutez `pkill -F pid`. Mais il faut noter que le daemon va s'arrêter si la machine s'éteint et ne se relancera pas automatiquement au redémarrage.

Connexion au Cluster

- Connexion SSH au 192.168.50.104 avec l'utilisateur `training` créé plus haut
- Générer et sauvegarder les mots de passe des utilisateurs `elastic` et `kibana_system`

```
cd /opt/training/elasticsearch-8.13.0
bin/elasticsearch-reset-password -u elastic
bin/elasticsearch-reset-password -u kibana_system
```

- Vérifier que le cluster est en bon état (statut "green")

```
curl --cacert /opt/training/elasticsearch-
8.13.0/config/certificates/ca/ca.crt -u elastic:$elastic_password -XGET
'https://192.168.50.104:9201/_cluster/health?pretty'
```

Remplacer `$elastic_password` par le mot de passe du user `elastic` sauvegardé.

- Vérifier que le nœud est bien identifié sur le cluster

```
curl --cacert /opt/training/elasticsearch-
8.13.0/config/certificates/ca/ca.crt -u elastic:$elastic_password -XGET
'https://192.168.50.104:9201/_cat/nodes?pretty'
```

Remplacer `$elastic_password` par le mot de passe du user `elastic` sauvegardé.

2. Le serveur Kibana

Pour cet exercice, Kibana est sur un host différent des nœuds du cluster Elasticsearch. Il peut se faire qu'il soit sur le même host qu'un des nœuds.

- Connexion SSH au 192.168.50.107
- Créer un utilisateur et l'ajouter au groupe `sudo`. Supposons que votre utilisateur s'appelle `training`.

```
sudo adduser training
sudo passwd training
sudo usermod -aG sudo training
su - training
```

- Création du dossier d'installation

```
sudo mkdir -p /opt/training
```

- Téléchargement de Kibana 8.13.0

```
cd /opt/training
sudo wget https://artifacts.elastic.co/downloads/kibana/kibana-8.13.0-
amd64.deb
```

- Décompression de Kibana 8.13.0

```
sudo dpkg -i kibana-8.13.0-amd64.deb
sudo rm kibana-8.13.0-amd64.deb
```

- Modifier le fichier de configuration `kibana.yml`.

```
sudo rm /etc/kibana/kibana.yml
sudo nano /etc/kibana/kibana.yml
```

Remplacer `$kibana_system_password` dans le fichier `kibana.yml` par le mot de passe du user `kibana_system` sauvegardé.

```
#kibana.yml
server.port: 5601
server.host: 192.168.50.107
server.name: "kibana"
elasticsearch.hosts: ["https://192.168.50.104:9201",
"https://192.168.50.105:9202", "https://192.168.50.106:9203"]
elasticsearch.username: "kibana_system"
elasticsearch.password: "$kibana_system_password"
#HTTP SSL
server.ssl.enabled: true
server.ssl.certificate: /opt/training/certificates/kibana/kibana.crt
server.ssl.key: /opt/training/certificates/kibana/kibana.key
server.ssl.certificateAuthorities: [ "/opt/training/certificates/ca/ca.crt" ]
#ES SSL
elasticsearch.ssl.certificateAuthorities: [
"/opt/training/certificates/ca/ca.crt" ]
elasticsearch.ssl.verificationMode: certificate
# bin/kibana-encryption-keys generate
xpack.encryptedSavedObjects.encryptionKey: 6f47131be984e3df38fc6e0c25b6865c
xpack.reporting.encryptionKey: 8046f6baea96389a900879c3081a9837
xpack.security.encryptionKey: 6c0074cd1a970076a1e9959881a709a9
```

- Copier les certificats depuis le HOST01

```
sudo mkdir -p /opt/training/certificates/ca
sudo mkdir -p /opt/training/certificates/kibana
sudo nano /opt/training/certificates/ca/ca.crt
sudo nano /opt/training/certificates/kibana/kibana.crt
sudo nano /opt/training/certificates/kibana/kibana.key
```

- Autoriser les ports sur le pare-feu

```
sudo ufw enable
sudo ufw allow 5601/tcp
sudo ufw status
```

- Démarrer le nœud Kibana et vérifier les logs

```
sudo /bin/systemctl daemon-reload
sudo systemctl enable kibana.service
sudo systemctl start kibana.service
sudo systemctl status kibana.service
```

Pour arrêter le service Kibana, exécutez `sudo systemctl stop kibana.service`.

- Afficher les logs du service Kibana

```
sudo journalctl --unit=kibana.service -n 100 --no-pager
```

- Sur votre machine aller sur l'interface de Kibana https://192.168.50.107:5601 et connecter vous avec l'utilisateur `elastic` et son mot de passe généré précédemment.

3. L'ETL Logstash

La configuration de Logstash se fera sur le HOST04

- Se connecter au HOST04 depuis votre machine en utilisant Vagrant

```
cd $vagrant_file_repository
vagrant ssh HOST04
```

- Téléchargement de Logstash 8.13.0

```
cd /opt/training
sudo wget https://artifacts.elastic.co/downloads/logstash/logstash-8.13.0-
amd64.deb
```

- Décompression de Logstash 8.13.0

```
sudo dpkg -i logstash-8.13.0-amd64.deb
sudo rm logstash-8.13.0-amd64.deb
```

- Modifier le fichier de configuration `logstash.yml` avec le contenu du fichier `logstash.yml` situé dans le répertoire GitHub https://github.com/iyanou/elastic-training-lab.

```
sudo rm /etc/logstash/logstash.yml
sudo nano /etc/logstash/logstash.yml
```

- Autoriser les ports sur le pare-feu

```
sudo ufw enable
sudo ufw allow 9600/tcp
sudo ufw allow 5044/tcp
sudo ufw status
```

- Donner les permissions nécessaires à Logstash

```
sudo chmod 755 /usr/share/logstash/data
sudo chown -R logstash:logstash /usr/share/logstash/data
```

- Démarrer Logstash

```
sudo /bin/systemctl daemon-reload
sudo systemctl enable logstash.service
sudo systemctl start logstash.service
sudo systemctl status logstash.service
```

Pour arrêter le service Logstash, exécuter `sudo systemctl stop logstash.service`.

- Afficher les logs du service Logstash

```
sudo journalctl --unit=logstash.service -n 100 --no-pager
```

- Aller sur l'interface de Logstash sur votre navigateur avec le lien http://192.168.50.107:9600/?pretty pour se rassurer que Logstash est démarré.

4. L'agent Filebeat

La configuration de Filebeat se fera sur le HOST04

- Se connecter au HOST04 depuis votre machine en utilisant Vagrant

```
cd $vagrant_file_repository
vagrant ssh HOST04
```

- Téléchargement de Filebeat 8.13.0

```
cd /opt/training
```

```
sudo wget https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-
8.13.0-amd64.deb
```

- Décompression de Filebeat 8.13.0

```
sudo dpkg -i filebeat-8.13.0-amd64.deb
sudo rm filebeat-8.13.0-amd64.deb
```

- Autoriser les ports sur le pare-feu

```
sudo ufw enable
sudo ufw allow 5067/tcp
sudo ufw status
```

- Démarrer Filebeat

```
sudo /bin/systemctl daemon-reload
sudo systemctl enable filebeat.service
sudo systemctl start filebeat.service
sudo systemctl status filebeat.service
```

Pour arrêter le service Filebeat, exécuter `sudo systemctl stop logstash.service`.

- Afficher les logs du service Filebeat

```
sudo journalctl --unit=filebeat.service -n 100 --no-pager
```

Etape 5 : Introduction à Filebeat

Explication du fonctionnement de Filebeat et présentation des différentes options et paramètres de configuration → (PPT Starting with Filebeat).

Etape 6 : Introduction à Logstash

Explication du fonctionnement de Logstash et présentation des différentes options et paramètres de configuration des pipelines Logstash → (PPT Starting with Logstash).

Etape 7 : Introduction à Elasticsearch (Mapping et Analysers)

Explication de quelques concepts importants liés à Elasticsearch et à son fonctionnement → (PPT Starting with Elasticsearch).

Etape 8 : Collecte des logs Spring Boot : De Filebeat vers Logstash et transfert vers Elasticsearch

Ce LAB se déroulera sur le HOST04.

Exercice

A- Vous avez un fichier de journal au format JSON nommé **spring-boot-app-logs.json**. Utilisez Filebeat pour traiter le fichier et l'envoyer à votre instance Logstash écoutant sur le port 5044.

B- Vous recevez des journaux d'application Spring Boot via Filebeat sur le port 5044. Configurez Logstash pour traiter ces journaux et envoyer les résultats vers Elasticsearch.

1. Utilisez le **filtre JSON** pour analyser le message et ajouter un champ nommé **dataset** avec la valeur "**Spring App**".
2. Renommez le champ **log_level** en **level**.
3. Supprimez le champ **event** et formatez le champ **event_date** avec le **filtre date**.
4. Appliquez un **filtre gsub** pour remplacer "**com.example.**" par un espace blanc **""** dans le champ **logger**.
5. Définissez pour le data stream "**filebeat-8.13.0**" un analyseur nommé **filebeat_analyzer**. Cet analyseur utilisera le **tokenizer standard** et les filtres de caractères **lowercase** et **synonym** pour que "**failed**" et "**error**" soient des synonymes.
6. Définissez un **index template** pour le data stream "**filebeat-8.13.0**".
7. Mappez ces champs comme suit (logger: keyword, thread_id: keyword, user_id: keyword, request_id: keyword, @timestamp: date, event_date: date, message: text, level: keyword).
8. Appliquez le **filebeat_analyzer** au champ **message**.
9. Envoyez les journaux vers votre sortie **Elasticsearch**.
10. Aller sur Kibana et vérifiez que le système fonctionne comme voulu.

Solution

- Se connecter au HOST04 depuis votre machine en utilisant Vagrant

```
cd $vagrant_file_repository
vagrant ssh HOST04
```

- Créer le fichier `spring-boot-app-logs.json` en utilisant le fichier `spring-boot-app-logs.json` situé dans le répertoire GitHub https://github.com/iyanou/elastic-training-lab.

```
sudo mkdir -p /opt/training/app/logs
sudo nano /opt/training/app/logs/spring-boot-app-logs.json
```

- Créer le pipeline Logstash pour le traitement des données venant de Filebeat. Utiliser le fichier `filebeat-pipeline.conf` situé dans le répertoire GitHub https://github.com/iyanou/elastic-training-lab.

```
 sudo nano /etc/logstash/conf.d/filebeat-pipeline.conf
```

- Redémarrer le service Logstash

```
sudo systemctl restart logstash.service
```

- Mettre à jour le index template de filebeat en utilisant le fichier filebeat-index-template.txt situé dans le répertoire GitHub https://github.com/iyanou/elastic-training-lab.
- Configuration de `filebeat.yml` en utilisant le fichier `filebeat.yml` situé dans le répertoire GitHub https://github.com/iyanou/elastic-training-lab.

```
sudo rm /etc/filebeat/filebeat.yml
sudo nano /etc/filebeat/filebeat.yml
```

- Supprimer le registre de Filebeat

```
sudo rm -r /var/lib/filebeat/registry
sudo rm /var/lib/filebeat/meta.json
udo rm /var/lib/filebeat/filebeat.lock
```

- Redémarrer le service Filebeat

```
sudo systemctl restart filebeat.service
```

- Afficher les logs du service Filebeat

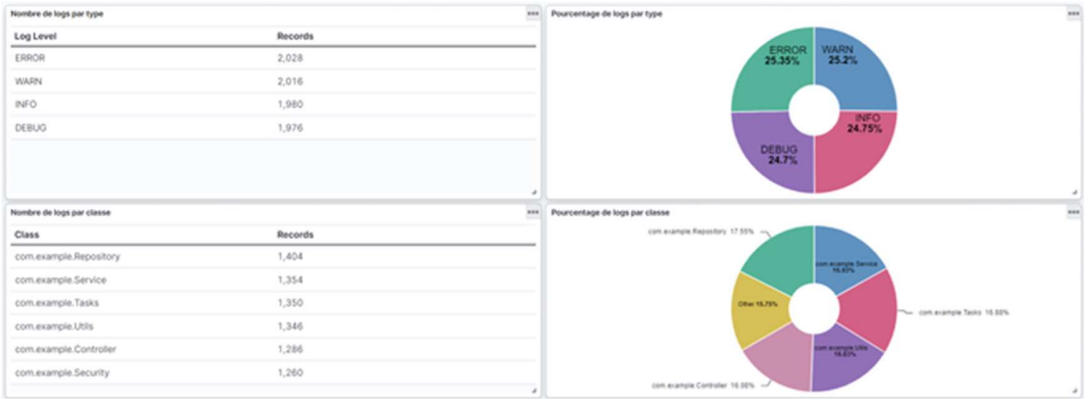
```
sudo journalctl --unit=filebeat.service -n 100 --no-pager
```

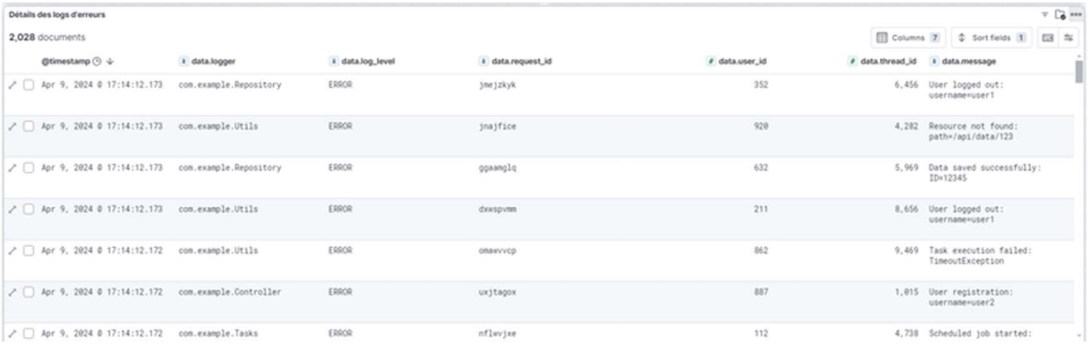
- Afficher les logs du service Logstash

```
sudo journalctl --unit=logstash.service -n 100 --no-pager
```

- Aller sur Kibana, choisir le data view nommé "**filebeat**" et aller dans la rubrique "**Discover**" pour vérifier que les logs ont été envoyés sur le cluster Elasticsearch.
- Tester sur Discover que l'analyser `filebeat_analyser` fonctionne comme voulu

Etape 9 : Visualisation des Logs et création de Dashboard avec Kibana





Bonus : Tester que le système est automatique

Sur le HOST04,

- Créer un nouveau fichier `spring-boot-app-logs-test.json` en utilisant le fichier `spring-boot-app-logs-test.json` situé dans le répertoire GitHub https://github.com/iyanou/elastic-training-lab.

```
sudo nano /opt/training/app/logs/spring-boot-app-logs-test.json
```

- Visualiser que les logs ont mis le Dashboard à jour.