TP - Monitoring des logs d'une application Spring Boot avec ELK

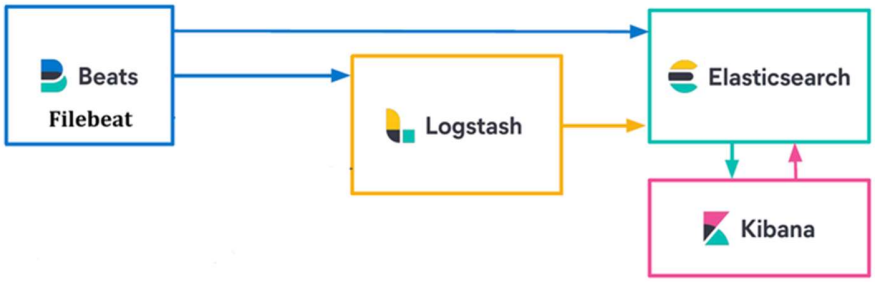
Réalisé par **Mr. Iyanou Eraste AKANDE**, Ingénieur des données Telecom à Synaptique Maghreb, Ingénieur Certifié Elasticsearch.

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



Etape 3 : Installation et configuration des applications

1- Les nœuds Elasticsearch

Les 3 nœuds Elasticsearch et l'instance Kibana seront installés sur la même machine virtuelle. Nous travaillerons dans ce TP avec la version 8.13.0 de la pile ELK. Vous devez connaitre au préalable l'adresse IP de la machine virtuelle et l'adresse publique du serveur.

Dans cet exemple, l'adresse IP de la machine virtuelle est le `172.20.0.5`. L'adresse publique du serveur est le `52.149.157.216`.

Elasticsearch Node 1

- Connexion SSH a l'adresse publique du serveur avec les identifiants. Vous pouvez utiliser `putty`(https://www.putty.org). Dans cet exercice c'est l'adresse `52.149.157.216` avec le nom d'utilisateur `azureuser` et le mot de passe `Azure@P@ssword_MONITOR`.
- Vérifier l'adresse IP de la machine virtuelle

- Créer un utilisateur pour le TP et l'ajouter au groupe `sudo`. Supposons que l'utilisateur s'appelle `training`.

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

Il ne faut pas exécuter Elasticsearch avec l'utilisateur 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
```

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

```
sudo mv /opt/training/elasticsearch-8.13.0 /opt/training/node01
cd /opt/training/node01
sudo rm config/elasticsearch.yml
sudo nano config/elasticsearch.yml

#elasticsearch.yml
cluster.name: es-training
node.name: node01
network.host: 172.20.0.5
http.port: 9201
transport.port: 9301
discovery.seed_hosts: ["172.20.0.5:9301", "172.20.0.5:9302",
"172.20.0.5:9303"]
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 la mémoire allouée au `heap` selon les ressources de la machine. Pour cet exercice, la valeur du `heap` sera de 2G.

```
sudo rm config/jvm.options
sudo nano config/jvm.options

#jvm.options
################################################################
##
## JVM configuration
##
################################################################
##
## WARNING: DO NOT EDIT THIS FILE. If you want to override the
## JVM options in this file, or set any additional options, you
## should create one or more files in the jvm.options.d
## directory containing your adjustments.
##
## See https://www.elastic.co/guide/en/elasticsearch/reference/8.13/jvm-
options.html
## for more information.
##
################################################################
################################################################
## IMPORTANT: JVM heap size
################################################################
##
## The heap size is automatically configured by Elasticsearch
## based on the available memory in your system and the roles
## each node is configured to fulfill. If specifying heap is
## required, it should be done through a file in jvm.options.d,
## which should be named with .options suffix, and the min and
## max should be set to the same value. For example, to set the
## heap to 4 GB, create a new file in the jvm.options.d
## directory containing these lines:
##
-Xms2g
-Xmx2g
##
## See https://www.elastic.co/guide/en/elasticsearch/reference/8.13/heap-
size.html
## for more information
##
################################################################
################################################################
## Expert settings
################################################################
##
## All settings below here are considered expert settings. Do
## not adjust them unless you understand what you are doing. Do
## not edit them in this file; instead, create a new file in the
## jvm.options.d directory containing your adjustments.
##
################################################################
```

```
-XX:+UseG1GC
## JVM temporary directory
-Djava.io.tmpdir=${ES_TMPDIR}
# Leverages accelerated vector hardware instructions; removing this may
# result in less optimal vector performance
20-:--add-modules=jdk.incubator.vector
## heap dumps
# generate a heap dump when an allocation from the Java heap fails; heap
dumps
# are created in the working directory of the JVM unless an alternative path
is
# specified
-XX:+HeapDumpOnOutOfMemoryError
# exit right after heap dump on out of memory error
-XX:+ExitOnOutOfMemoryError
# specify an alternative path for heap dumps; ensure the directory exists and
# has sufficient space
-XX:HeapDumpPath=data
# specify an alternative path for JVM fatal error logs
-XX:ErrorFile=logs/hs_err_pid%p.log
## GC logging
-
Xlog:gc*,gc+age=trace,safepoint:file=logs/gc.log:utctime,level,pid,tags:filec
ount=32,filesize=64m
```

- 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
```

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

```
cd /opt/training/node01
sudo mkdir config/certificates
sudo nano config/certificates/instances.yml

#instances.yml
instances:
  - name: node01
    ip:
      - 172.20.0.5
  - name: node02
    ip:
      - 172.20.0.5
  - name: node03
    ip:
      - 172.20.0.5
  - name: kibana
    ip:
      - 172.20.0.5

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/node01
```

- Démarrer le nœud Elasticsearch et vérifier les logs

```
cd /opt/training/node01
./bin/elasticsearch
```

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

- Arrêter le nœud Elasticsearch pour le redémarrer 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.

Elasticsearch Node 2

- Décompression de Elasticsearch 8.13.0

```
cd /opt/training
sudo tar -xzf elasticsearch-8.13.0-linux-x86_64.tar.gz
```

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

```
sudo mv /opt/training/elasticsearch-8.13.0 /opt/training/node02
cd /opt/training/node02
sudo rm config/elasticsearch.yml
sudo nano config/elasticsearch.yml

#elasticsearch.yml
cluster.name: es-training
```

```
node.name: node02
network.host: 172.20.0.5
http.port: 9202
transport.port: 9302
discovery.seed_hosts: ["172.20.0.5:9301", "172.20.0.5:9302",
"172.20.0.5:9303"]
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 la mémoire allouée au `heap` selon les ressources de la machine. Pour cet exercice, la valeur du `heap` sera de 2G.

```
sudo rm config/jvm.options
sudo nano config/jvm.options

#jvm.options
################################################################
##
## JVM configuration
##
################################################################
##
## WARNING: DO NOT EDIT THIS FILE. If you want to override the
## JVM options in this file, or set any additional options, you
## should create one or more files in the jvm.options.d
## directory containing your adjustments.
##
## See https://www.elastic.co/guide/en/elasticsearch/reference/8.13/jvm-
options.html
## for more information.
##
################################################################
################################################################
## IMPORTANT: JVM heap size
################################################################
##
## The heap size is automatically configured by Elasticsearch
## based on the available memory in your system and the roles
## each node is configured to fulfill. If specifying heap is
## required, it should be done through a file in jvm.options.d,
## which should be named with .options suffix, and the min and
## max should be set to the same value. For example, to set the
## heap to 4 GB, create a new file in the jvm.options.d
```

```
## directory containing these lines:
##
-Xms2g
-Xmx2g
##
## See https://www.elastic.co/guide/en/elasticsearch/reference/8.13/heap-
size.html
## for more information
##
################################################################
################################################################
## Expert settings
################################################################
##
## All settings below here are considered expert settings. Do
## not adjust them unless you understand what you are doing. Do
## not edit them in this file; instead, create a new file in the
## jvm.options.d directory containing your adjustments.
##
################################################################
-XX:+UseG1GC
## JVM temporary directory
-Djava.io.tmpdir=${ES_TMPDIR}
# Leverages accelerated vector hardware instructions; removing this may
# result in less optimal vector performance
20-:--add-modules=jdk.incubator.vector
## heap dumps
# generate a heap dump when an allocation from the Java heap fails; heap
dumps
# are created in the working directory of the JVM unless an alternative path
is
# specified
-XX:+HeapDumpOnOutOfMemoryError
# exit right after heap dump on out of memory error
-XX:+ExitOnOutOfMemoryError
# specify an alternative path for heap dumps; ensure the directory exists and
# has sufficient space
-XX:HeapDumpPath=data
# specify an alternative path for JVM fatal error logs
-XX:ErrorFile=logs/hs_err_pid%p.log
## GC logging
-
Xlog:gc*,gc+age=trace,safepoint:file=logs/gc.log:utctime,level,pid,tags:filec
ount=32,filesize=64m
```

- Copier les certificats depuis le premier nœud

```
cd /opt/training/node02
sudo cp -r /opt/training/node01/config/certificates config/
```

- Autoriser les ports sur le pare-feu

```
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/node02
```

- Démarrer le nœud Elasticsearch et vérifier les logs

```
cd /opt/training/node02
./bin/elasticsearch
```

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

- Arrêter le nœud Elasticsearch pour le redémarrer 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.

Elasticsearch Node 3

- Décompression de Elasticsearch 8.13.0

```
cd /opt/training
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`.

```
sudo mv /opt/training/elasticsearch-8.13.0 /opt/training/node03
cd /opt/training/node03
sudo rm config/elasticsearch.yml
sudo nano config/elasticsearch.yml

#elasticsearch.yml
cluster.name: es-training
node.name: node03
network.host: 172.20.0.5
http.port: 9203
transport.port: 9303
discovery.seed_hosts: ["172.20.0.5:9301", "172.20.0.5:9302",
"172.20.0.5:9303"]
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 a jour la mémoire allouée au `heap` selon les ressources de la machine. Pour cet exercice, la valeur du `heap` sera de 2G.

```
sudo rm config/jvm.options
sudo nano config/jvm.options

#jvm.options
################################################################
##
## JVM configuration
##
################################################################
##
## WARNING: DO NOT EDIT THIS FILE. If you want to override the
## JVM options in this file, or set any additional options, you
## should create one or more files in the jvm.options.d
## directory containing your adjustments.
##
## See https://www.elastic.co/guide/en/elasticsearch/reference/8.13/jvm-
options.html
## for more information.
##
################################################################
################################################################
## IMPORTANT: JVM heap size
################################################################
##
## The heap size is automatically configured by Elasticsearch
## based on the available memory in your system and the roles
## each node is configured to fulfill. If specifying heap is
## required, it should be done through a file in jvm.options.d,
## which should be named with .options suffix, and the min and
## max should be set to the same value. For example, to set the
## heap to 4 GB, create a new file in the jvm.options.d
## directory containing these lines:
##
-Xms2g
-Xmx2g
##
## See https://www.elastic.co/guide/en/elasticsearch/reference/8.13/heap-
size.html
## for more information
##
################################################################
################################################################
## Expert settings
################################################################
##
## All settings below here are considered expert settings. Do
```

```
## not adjust them unless you understand what you are doing. Do
## not edit them in this file; instead, create a new file in the
## jvm.options.d directory containing your adjustments.
##
################################################################
-XX:+UseG1GC
## JVM temporary directory
-Djava.io.tmpdir=${ES_TMPDIR}
# Leverages accelerated vector hardware instructions; removing this may
# result in less optimal vector performance
20-:--add-modules=jdk.incubator.vector
## heap dumps
# generate a heap dump when an allocation from the Java heap fails; heap
dumps
# are created in the working directory of the JVM unless an alternative path
is
# specified
-XX:+HeapDumpOnOutOfMemoryError
# exit right after heap dump on out of memory error
-XX:+ExitOnOutOfMemoryError
# specify an alternative path for heap dumps; ensure the directory exists and
# has sufficient space
-XX:HeapDumpPath=data
# specify an alternative path for JVM fatal error logs
-XX:ErrorFile=logs/hs_err_pid%p.log
## GC logging
-
Xlog:gc*,gc+age=trace,safepoint:file=logs/gc.log:utctime,level,pid,tags:filec
ount=32,filesize=64m
```

- Copier les certificats depuis le premier nœud

```
cd /opt/training/node03
sudo cp -r /opt/training/node01/config/certificates config/
```

- Autoriser les ports sur le pare-feu

```
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/node03
```

- Démarrer le nœud Elasticsearch et vérifier les logs

```
cd /opt/training/node03
./bin/elasticsearch
```

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

- Arrêter le nœud Elasticsearch pour le redémarrer 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.

Connexion au cluster

- Générer et sauvegarder les mots de passe des utilisateurs `elastic` et `kibana_system`

```
cd /opt/training/node01
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/node01/config/certificates/ca/ca.crt -u
elastic:$elastic_password -XGET
'https://172.20.0.5:9201/_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/node01/config/certificates/ca/ca.crt -u
elastic:$elastic_password -XGET 'https://172.20.0.5:9201/_cat/nodes?pretty'
```

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

- Vous pouvez aussi faire ces vérifications sur votre navigateur sur l'interface https://52.149.157.216:9201/_cluster/health?pretty ou https://52.149.157.216:9201/_cat/nodes?pretty en vous connectant avec le user `elastic` et son mot de passe généré précédemment.

2- Le serveur Kibana

- 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: 172.20.0.5
server.name: "kibana"
elasticsearch.hosts: ["https://172.20.0.5:9201", "https://172.20.0.5:9202",
"https://172.20.0.5:9203"]
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 cp -r /opt/training/node01/config/certificates /opt/training/
```

- Autoriser les ports sur le pare-feu

```
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://52.149.157.216:5601 et connecter vous avec le user `elastic` et son mot de passe généré précédemment.

3- L'ETL Logstash

- 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`.

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

#logstash.yml
node.name: "logstash"
http.host: 0.0.0.0
http.port: 9600
path.logs: /usr/share/logstash/logs
log.format: json
log.level: info
config.reload.automatic: true
config.reload.interval: 3s
config.debug: false
api.enabled: true
api.environment: logstash
```

- Autoriser les ports sur le pare-feu

```
sudo ufw allow 9600/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://52.149.157.216:9600/?pretty pour se rassurer que Logstash est démarré.

4- L'agent Filebeat

- 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 allow 5044/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 4 : Introduction à Filebeat

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

Etape 5 : 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 6 : Introduction à Elasticsearch (Mapping et Analysers)

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

Etape 7 : Collecte des logs Spring Boot : De Filebeat vers Logstash et transfert vers Elasticsearch

Exercice

A- Vous avez un fichier de journal au format JSON nommé **spring-boot-app-logs.json** situé dans le répertoire GitHub https://github.com/iyanou/elastic-training-lab. 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

- 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.

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

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

```
#filebeat-pipeline.conf
input {
  beats {
    port => 5044
  }
}

filter {

  json { source => "message" }

  mutate {
        add_field => { "dataset" => "Spring App" }
        rename => { "log_level" => "level" }
    gsub => [
        "logger", "com.example.", ""
    ]
        remove_field => ["event"]
  }

  date {
    match => ["event_date", "YYYY-MM-dd HH:mm:ss"]
    target => "event_date"
  }
}

output {
  elasticsearch {
    hosts => ["https://172.20.0.5:9201", "https://172.20.0.5:9202",
"https://172.20.0.5:9203"]
    user => "elastic"
    password => "$elastic_password"
    cacert   => "/opt/training/certificates/ca/ca.crt"
    ssl      => "true"
    ssl_certificate_verification   => "true"
    manage_template => false
    index => "%{[@metadata][beat]}-%{[@metadata][version]}"
    action => "create"
  }
}
```

- Redémarrer le service Logstash

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

- Mettre a jour le index template de filebeat sur Dev Tools dans Kibana en utilisant le fichier `filebeat-index-template.txt`.

```
#filebeat-index-template.txt
PUT _index_template/filebeat-8.13.0
{
  "index_patterns": ["filebeat-8.13.0"],
  "data_stream": {},
  "template": {
    "settings": {
      "analysis": {
        "analyzer": {
          "filebeat_analyzer": {
            "type": "custom",
            "tokenizer": "standard",
            "filter": ["lowercase", "synonym"]
          }
        },
        "filter": {
          "synonym": {
            "type": "synonym",
            "synonyms": ["failed, error"]
          }
        }
      }
    },
    "mappings": {
      "properties": {
        "logger": {
          "type": "keyword"
        },
        "thread_id": {
          "type": "keyword"
        },
        "user_id": {
          "type": "keyword"
        },
        "request_id": {
          "type": "keyword"
        },
        "@timestamp": {
          "type": "date"
        },
        "event_date": {
          "type": "date"
        },
        "message": {
          "type": "text",
          "analyzer": "filebeat_analyzer"
        },
        "level": {
          "type": "keyword"
        }
      }
    }
  }
```

```
}
```

- Configuration de `filebeat.yml` en utilisant le fichier `filebeat.yml`.

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

#filebeat.yml
filebeat.inputs:
- type: filestream

  # Unique ID among all inputs, an ID is required.
  id: my-filestream-id

  # Change to true to enable this input configuration.
  enabled: true

  # Paths that should be crawled and fetched. Glob based paths.
  paths:
    - /opt/training/app/logs/*.json

filebeat.config.modules:
  # Glob pattern for configuration loading
  path: ${path.config}/modules.d/*.yml

  # Set to true to enable config reloading
  reload.enabled: false

  # Period on which files under path should be checked for changes
  #reload.period: 10s


# ------------------------------ Logstash Output -------------------------------
---
output.logstash:
  # The Logstash hosts
  enabled: true
  hosts: ["172.20.0.5:5044"]

# ================================== Processors
==================================
processors:
  - add_host_metadata:
      when.not.contains.tags: forwarded
  - add_cloud_metadata: ~
  - add_docker_metadata: ~
  - add_kubernetes_metadata: ~



# ================================== HTTP Connection
==================================
#http.enabled: true
#http.port: 5067
#http.host: 0.0.0.0
#disable internal monitoring
```

```
monitoring.enabled: false
```

- Supprimer le registre de Filebeat

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

- Redémarrer le service Filebeat

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

- Afficher les logs du service Filebeat

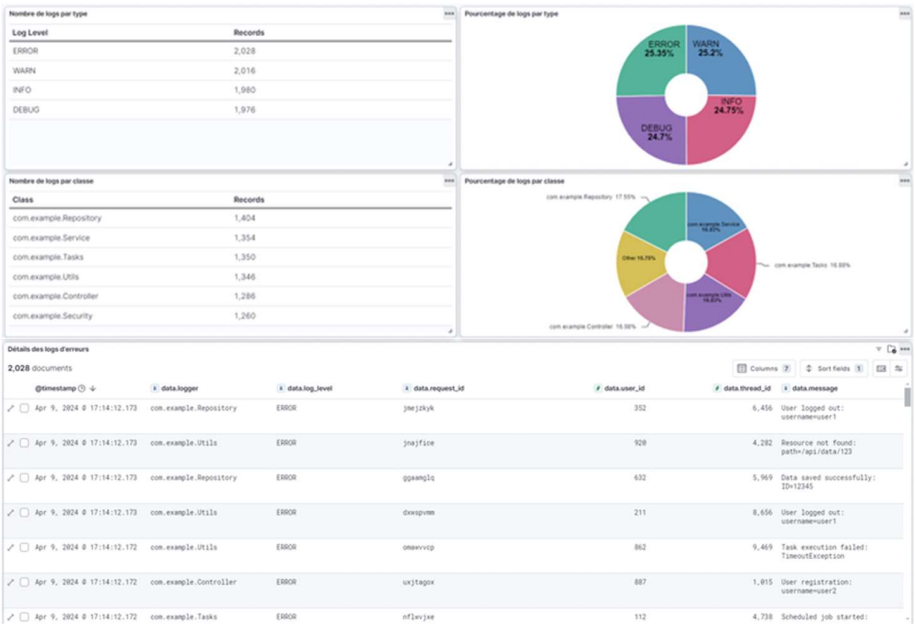
```
sudo journalctl --unit=filebeat.service -n 100 --no-pager
```

- 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 8 : Visualisation des Logs et création de Dashboards avec Kibana



Bonus : Tester que le système est automatique

- 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.