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

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

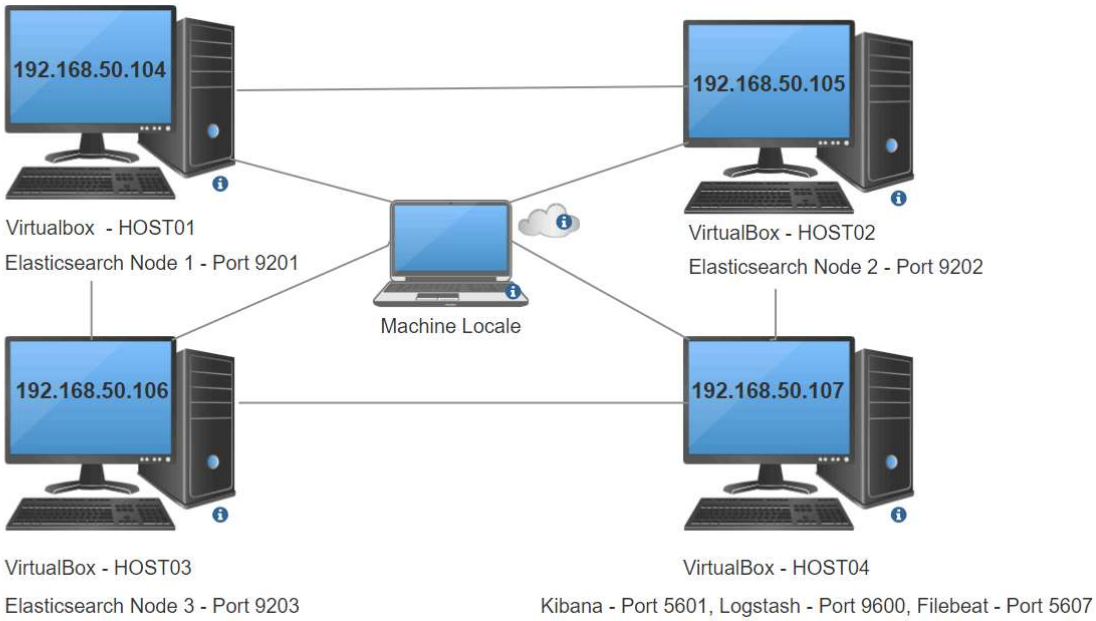
eraste.akande@gmail.com

Objectifs du TP

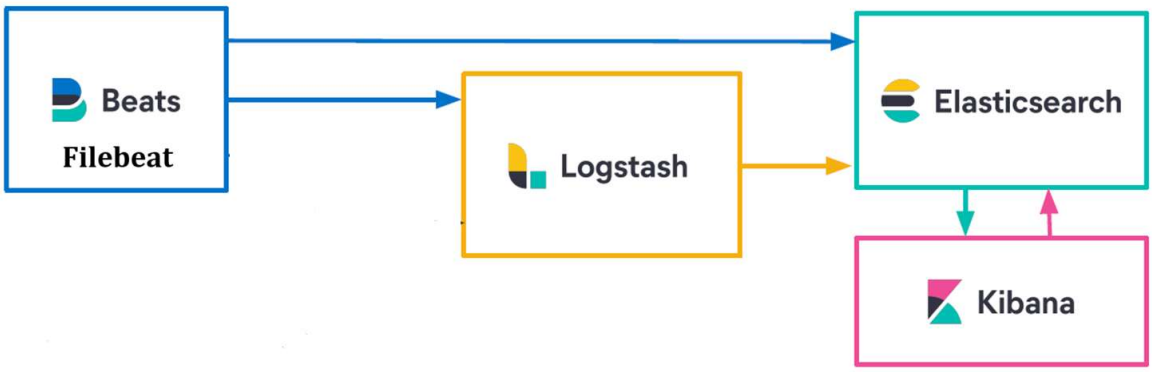
Dans ce TP, nous allons déployer 4 machines virtuelles en utilisant Vagrant. Un cluster Elasticsearch de 3 nœuds sera formé avec chaque nœud installé sur une machine virtuelle. Le cluster sera sécurisé et les nœuds auront des certificats. Sur la quatrième machine, on installera Kibana, Logstash et Filebeat.
Le but du TP consiste à collecter des fichiers de logs d'une application Spring Boot avec Filebeat. Filebeat enverra les logs a Logstash qui se chargera de les formater avant de les envoyer à Elasticsearch. Ensuite on fera une visualisation sur Kibana qui indiquera en temps réel les différents types de logs reçus et les détails des logs d'erreur.

Etape 1: Architecture du TP

Les machines virtuelles



Les applications



Etape 2: 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** dans le répertoire GitHub et placez-le dans un dossier de votre convenance. Exécutez ce qui suit depuis cet emplacement

  ```
  cd $vagrant_file_repository
  vagrant up
  ```

  $vagrant_file_repository représente le répertoire dans lequel vous avez déposé le fichier Vagrantfile.

- 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 3: Installation et configuration des applications

1. Les noeuds Elasticsearch

HOST 01

- Connexion SSH au 192.168.50.104

  vagrant ssh HOST01

- 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 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 par le contenu du fichier **node01.yml**.

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

- 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 copiant au préalable le contenu du fichier **instances.yml** dans le répertoire qui suit**.**

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

```
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
```

Vous venez de générer dans le dossier **/opt/training/elasticsearch-8.13.0/config/certificates/** le certificat d'autorité et les certificats des trois nœuds elasticsearch et ainsi que celui de Kibana**.**

```
sudo ls config/certificates
```

- 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
```

C'est mieux de le démarrer comme un processus pour qu'il ne bloque pas l'utilisation de la console**.**

- Démarrer le nœud Elasticsearch comme un processus Linux et vérifier les logs

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

Pour arrêter Elasticsearch, exécutez pkill -F pid. Mais il faut noter que le processus va s'arrêter si le host s'éteint et ne se relancera pas automatiquement au redémarrage. Je vous montrerai prochainement l'installation comme un service Linux.

HOST 02

- Connexion SSH au 192.168.50.105

  vagrant ssh HOST02

- 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 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 par le contenu du fichier **node02.yml**.

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

- 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 générés précédemment sur le HOST 01

  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

- Démarrer le nœud Elasticsearch comme un processus Linux et vérifier les logs

  ./bin/elasticsearch -d -p pid

HOST 03

- Connexion SSH au 192.168.50.106

  vagrant ssh HOST03

- 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 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 en utilisant le contenu du fichier **node03.yml**.

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

- 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 HOST 01

  ```
  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
```

- Démarrer le nœud Elasticsearch comme un processus Linux et vérifier les logs

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

Connexion au cluster

- Connexion SSH au 192.168.50.104 avec le user training créé plus haut
- Générer et sauvegarder quelque part 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 de l’utilisateur **elastic** sauvegardé.

- Vérifier que les trois nœuds ont rejoint 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'
  ```

Félicitations votre cluster Elasticsearch est bien installé et a démarré.

2. Le serveur Kibana

Pour cet exercice, Kibana est sur un host différent des nœuds du cluster Elasticsearch.
Connexion SSH au **192.168.50.107.**

vagrant ssh HOST04

- 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 par le contenu du fichier **kibana.yml**.

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

  Ouvrez **kibana.yml** et remplacez **$kibana_system_password** par le mot de passe de l'utilisateur **kibana_system** sauvegardé précédemment.

- Copier les certificats depuis le HOST 01

  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, allez sur l'interface de Kibana https://192.168.50.107:5601 et connectez-vous avec le user **elastic** et son mot de passe généré précédemment.

Félicitations votre instance Kibana est bien installée et a démarré.

3. L'ETL Logstash

La configuration de Logstash se fera sur le HOST04

- Se connecter au HOST04 depuis votre machine en utilisant Vagrant

  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.

  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
```

Félicitations votre ETL Logstash est bien installé et a démarré.

4. L'agent Filebeat

La configuration de Filebeat se fera sur le HOST04

- Se connecter au HOST04 depuis votre machine en utilisant Vagrant

```
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 filebeat.service.

- Afficher les logs du service Filebeat

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

Félicitations votre agent Filebeat est bien installée et a démarré.

Etape 4: Collecte des logs Spring Boot : De Filebeat vers Logstash et transfert vers Elasticsearch

Ce LAB se déroulera sur le HOST04.

- Se connecter au HOST04 depuis votre machine en utilisant Vagrant

  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.

  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.

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

  Remplacer **$elastic_password** dans **filebeat-pipeline.conf** par le mot de passe de l'utilisateur elastic enregistré précédemment.

- Redémarrer le service Logstash

  sudo systemctl restart logstash.service

- Mettre à jour le index template de filebeat sur **Kibana > Dev Tools** en utilisant le fichier **filebeat-index-template** situé dans le répertoire GitHub.
- Configurer filebeat.yml en utilisant le fichier **filebeat.yml** situé dans le répertoire GitHub.

  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
  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.
- Sur Kibana, allez sur **Lens** et amusez-vous à faire des Dashboards en utilisant les métriques qui vous semblent pertinentes.

Etape 5: Visualisation des Logs et création de Dashboards avec Kibana

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.

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

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