

ELK / EFK

Pipeline ELK – Étapes de mise en place

Étape 1 : Préparation de l'environnement ELK

1. Installer Elasticsearch

- Moteur de recherche + stockage des logs
- Configurer la mémoire, les limites, les ports (9200/9300)

2. Installer Kibana



- Interface de visualisation
- Connectée à Elasticsearch (port 5601)

3. Installer Logstash

- Collecte + traitement des logs
 - Ports à exposer : 5044 (Filebeat) ou lecture locale de fichiers
-

Étape 2 : Log Shipping (🔴 c'est ce que tu viens de faire)

1. Récupérer les logs depuis les applications distantes

-  Docker logs depuis `pipeline-ci` : Joget, MySQL, SonarQube
-  Transfert régulier des fichiers `.log`, `.gz` vers la machine `monitoring`

2. Centraliser tous les fichiers dans un répertoire commun

- Exemple : `/home/vagrant/monitoring-data/fichiers-logs/`
-

Étape 3 : Ingestion et parsing des logs avec Logstash

1. Créer une configuration `logstash.conf`

- `input` : lire les fichiers de logs

- `filter` : grok / dissect / date / mutate (selon la structure)
- `output` : envoyer dans Elasticsearch (index personnalisé par app)

2. Test local de la configuration

- `docker logs logstash` pour vérifier qu'il n'y a pas d'erreurs
- Utiliser un fichier à la fois si besoin

Étape 4 : Indexation et stockage dans Elasticsearch

1. Créer ou personnaliser des index

- Nommer par application (`joget-*` , `sonarqube-*` , etc.)
- Gérer les mappings si besoin

2. Vérifier l'ingestion avec la requête :

```
curl -X GET http://localhost:9200/_cat/indices?v
```

Étape 5 : Visualisation dans Kibana

1. Créer un index pattern dans Kibana

- Exemple : `joget-*`
- Sélectionner le champ `@timestamp` pour le tri

2. Créer des dashboards personnalisés

- Nombre d'erreurs, warning, heatmap, top logs par source, etc.

1. Trouver les fichiers de logs dans chaque conteneur

On va d'abord « entrer » dans chaque conteneur pour chercher les fichiers de logs à l'intérieur.

A. SonarQube

```
bash
CopierModifier
```

```
docker exec -it sonarqube bash
```

Puis, à l'intérieur du conteneur :

```
bash
CopierModifier
ls /opt/sonarqube/logs
```

Tu y verras des fichiers comme `web.log` , `ce.log` , `es.log` , etc.

Tu peux ensuite faire par exemple :

```
bash
CopierModifier
cat /opt/sonarqube/logs/web.log | tail -n 20
```

Pour voir les 20 dernières lignes.

B. Joget

```
bash
CopierModifier
docker exec -it joget bash
```

Puis cherche ici :

```
bash
CopierModifier
ls /opt/joget/apache-tomcat/logs
```

Les fichiers seront des logs Tomcat comme `catalina.out` , `localhost.log` , etc.

C. MySQL

```
bash
CopierModifier
docker exec -it jogetdb bash
```

Puis tente :

```
bash
CopierModifier
ls /var/log/mysql
```

OU parfois c'est plutôt dans :

```
bash
CopierModifier
ls /var/lib/mysql
```

Tu peux faire :

```
bash
CopierModifier
cat /var/log/mysql/error.log | tail -n 20
```

Transférer les logs de VM **pipeline-ci** → VM **monitoring** chaque 5 min


On va créer un **script bash** ou **script Python** + une **tâche cron** sur la machine **pipeline-ci** qui :

- Se connecte à **monitoring** (**192.168.56.150**)

- Transfère tous les logs des conteneurs vers :

`/home/vagrant/monitoring-data/fichiers-logs/`

Option 1 : Script Bash (simple, efficace)

 **Crée un script** `/home/vagrant/send-logs.sh` **sur** `pipeline-ci` :

```
bash
CopierModifier
#!/bin/bash

# Dossier de logs dans chaque conteneur
LOG_DIRS=(
    "/opt/joget/apache-tomcat/logs"
    "/var/log"
    "/opt/sonarqube/logs"
)

# Nom des conteneurs
CONTAINERS=("joget" "jogetdb" "sonarqube")

# Dossier temporaire
TMP_DIR="/tmp/logs_to_send"
mkdir -p "$TMP_DIR"

# Récupération des logs de chaque conteneur
for i in "${!CONTAINERS[@]}; do
    container="${CONTAINERS[$i]}"
    log_path="${LOG_DIRS[$i]}"
    docker cp "$container:$log_path" "$TMP_DIR/$container"
done

# Envoi vers monitoring
scp -i /home/vagrant/.ssh/monitoring_key -o StrictHostKeyChecking=no -r
"$TMP_DIR"/* vagrant@192.168.56.150:/home/vagrant/monitoring-data/fichier
s-logs/
```

```
# Nettoyage temporaire  
rm -rf "$TMP_DIR"
```

Donne les permissions d'exécution :

```
bash  
CopierModifier  
chmod +x /home/vagrant/send-logs.sh
```



Configurer le cron (chaque 5 minutes)

Lance :

```
bash  
CopierModifier  
crontab -e
```

Ajoute la ligne suivante :

```
bash  
CopierModifier  
*/5 * * * * /home/vagrant/send-logs.sh >> /home/vagrant/send-logs.log 2>&1
```



Tu peux tester à la main d'abord :

```
bash  
CopierModifier  
/home/vagrant/send-logs.sh
```

Ensuite vérifie sur la VM `monitoring` :

```
bash
```

```
CopierModifier
```

```
ls /home/vagrant/monitoring-data/fichiers-logs
```