TP Gestion des Logs

0 - Introduction

Au cours de ce TP, vous allez manipuler les différents outils de gestion des logs.
Nous commencerons par préparer l'infrastructure, puis nous continuerons par la visualisation et la configuration des logs sur les machines Linux et Windows. Nous terminerons par l'installation et la configuration des trois solutions de centralisation de logs vues en cours: stack Elastic, Graylog et Grafana Loki.

1 - Préparation de l’infrastructure

Dans cette section, vous devrez créer 5 machines virtuelles dans OpenStack avec les caractéristiques suivantes:

- 3 machines Ubuntu 20.04.3, 1 machine Ubuntu 20.04.3 - Docker Ready et 1 machine Windows 10 (BONUS)
- 2 vCPU
- 4GB RAM
- 10GB d’espace disque

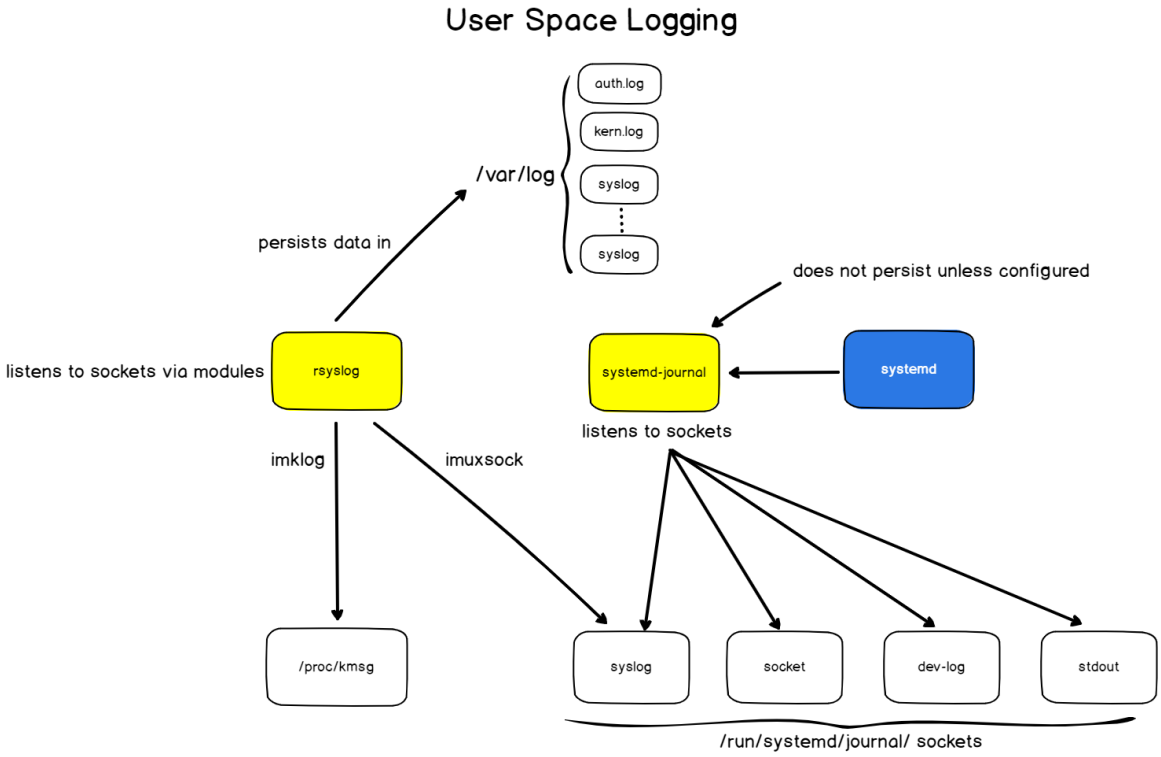
Ces machines doivent avoir des hostnames suivants:

- [num_etu]-nginx-server (Ubuntu)
- [num_etu]-graylog (Ubuntu)
- [num_etu]-elastic (Ubuntu)
- [num_etu]-loki (Ubuntu - Docker Ready)
- [num_etu]-windows-web-server (Windows 10) (BONUS)

Où “[num_etu]” est votre numéro d'étudiant.

2 - Logs Linux

Dans cette section, nous verrons comment fonctionnent les logs Linux. Nous allons visualiser, configurer et générer les logs sur la machine “nginx-server”.



Pour rappel l’architecture des logs de l’espace utilisateur

Pour commencer, vous devez installer un serveur *nginx* via le gestionnaire de packages *apt*. Vérifiez si le port 80 est bien ouvert dans *OpenStack* et testez si le serveur *nginx* répond bien aux requêtes des utilisateurs.

Rsyslog

Rsyslog est une implémentation du protocol *syslog* et est fourni par défaut sur la plupart des systèmes Linux modernes. Syslog est utilisé comme un standard pour produire, transmettre et collecter des logs.

Rsyslog récupère les logs de l’espace noyau et du journal de systemd et les persiste dans des fichiers.

Analysez les fichiers de configuration de *rsyslog* (“/etc/rsyslog.conf” et les fichiers dans “*/etc/rsyslog.d/*”).

- Quels modules sont activés par défaut et que font-ils?
- Quelles logs sont écrites dans le fichier “/var/log/syslog”?
- Dans quel fichier sont écrits les logs du noyau?

Configurez *rsyslog* pour qu’il envoie tous les logs contenant le mot “ssh” dans le fichier “/var/log/ssh.log”. (Créez un fichier de configuration dans “/etc/rsyslog.d/”, n’oubliez pas de redémarrer le rsyslog).

- Que mettez-vous dans le fichier de configuration?

Vérifiez si le fichier "/var/log/ssh.log" existe et contient des entrées avec le mot "ssh".

Systemd Journal

Le *systemd* est un gestionnaire de processus et de services qui implémente son propre service de journalisation appelé *systemd-journald (journald)*. Les services *systemd* envoient les logs directement au *journald*. Les fichiers logs du *journald* sont stockés dans “/var/log/journal”.
Essayez de visualiser les fichiers logs du journal avec la commande “cat”.

- Les logs sont-ils lisibles? Expliquez le résultat.
- Quelle commande permet de visualiser les logs du *journald*?
- Quelle commande permet de visualiser les logs enregistrés au cours de la dernière heure?
- Quel est l’avantage de stocker des logs de cette manière?

Generation des logs
Dans cette section, vous allez générer les logs avec la ligne de commande.

Logs noyau
Envoyez la phrase “Hello world” à l’espace des logs du noyau et visualisez-la via la commande “dmesg”. (Envoyez les logs en tant qu'utilisateur *root* dans “/dev/kmsg”).

- Quelle commande utiliserez-vous pour le faire ?

Vérifiez si le log a été bien écrit dans le fichier “/var/log/syslog” par *rsyslog* en tant que log du noyau.

- Quel module de *rsyslog* est responsable de l'écriture des logs de l’espace noyau dans ce fichier?

Logs espace utilisateur
Envoyez la phrase “Hello world” au journal de systemd. (pour cela vous pouvez utiliser la commande “systemd-cat”).

Visualisez ces logs avec la commande “journalctl” et vérifiez que *rsyslog* a bien écrit ce log dans le fichier “/var/log/syslog”.

- Quel module de *rsyslog* est responsable de l'écriture des logs de journal systemd dans ce fichier?

Suppression des logs
Dans les systèmes Linux, vous pouvez facilement supprimer les logs.
Pour supprimer les logs écrits dans des fichiers avec *rsyslog,* il vous suffit de purger, supprimer ou éditer le fichier.
Supprimez tous les logs du fichier “/var/log/syslog”.
(***Attention!*** Ne supprimez pas le fichier lui-même!)

- Quelle commande utiliserez-vous pour le faire ?

Pour supprimer les logs du journal *systemd*, il suffit d'utiliser les options "--rotate" et "--vacuum-time" de la commande "journalctl". Supprimez tous les logs du journal systemd.

- Quelle commande utiliserez-vous pour le faire ?

Logrotate

Logrotate est un outil système qui gère la rotation, la compression et la suppression automatique des fichiers log. Sans ces mécanismes, les logs pourraient éventuellement consommer tout l'espace disque disponible sur un système et rendre le système inutilisable.

Visualisez le fichier de configuration de *logrotate* et trouvez la configuration de la rotation des logs pour le fichier "/var/log/syslog".

- Comment fonctionne la rotation des logs pour le fichier "/var/log/syslog" (la fréquence de rotation, la durée de rétention, la compression)?

Configurez la rotation pour le fichier "/var/log/ssh.log". La rotation doit avoir lieu tous les jours, les 7 derniers fichiers doivent être conservés, la compression doit être activée.

- Que mettez-vous dans le fichier de configuration?

Vérifiez si votre configuration est correcte et est prise en compte avec la commande "sudo logrotate /etc/logrotate.conf --debug". Que fait cette commande?

Fail2ban

Dans cette section, vous allez installer et configurer l'outil *fail2ban*. Cet outil analyse les fichiers logs et interdit les adresses IP qui montrent les signes malveillants.

Installez l'outil *fail2ban* via le gestionnaire des packages *apt*.

Les filtres

fail2ban est fourni par défaut avec plusieurs filtres.
Les filtres sont généralement des expressions régulières qui sont utilisées pour détecter les tentatives d'effraction, les échecs de mot de passe, etc. Les filtres sont stockés dans "/etc/fail2ban/filter.d".

Les actions

Une action définit une ou plusieurs commandes qui sont exécutées à des moments différents: lors du démarrage / de l'arrêt d'un jail, de l'interdiction / de la suppression d'un hôte, etc. Les actions sont stockés dans "/etc/fail2ban/action.d"

Les jails

Un jail est une combinaison d'un filtre et d'une ou plusieurs actions. Les configurations des jails sont stockés dans "/etc/fail2ban/jail.d"

- Quel jail est activé par défaut?
- Confirmez que le jail est bien activé en utilisant le client *fail2ban* "fail2ban-client". Quelle commande utiliserez-vous pour le faire?

Demandez à un de vos collègues de faire plusieurs tentatives de connexion avec le mauvais mot de passe à la machine via SSH jusqu'à ce qu'il soit bloqué par *fail2ban*.

- Après combien de tentatives de connexion échouées, fail2ban a bloqué l'accès?

Visualisez les *iptables* avec la commande "iptables -L".

- Quelle règle a été créé par *fail2ban*?

Visualisez l'état du jail *sshd* avec le client *fail2ban* “fail2ban-client”.

- Quelle commande utiliserez-vous pour le faire ?

Supprimez l’adresse IP de votre collègue de jail avec le client *fail2ban*.

- Quelle commande utiliserez-vous pour le faire ?

2 - BONUS. Logs Windows

Dans cette section, vous allez visualiser et manipuler les logs Windows.
Windows Event Logs contient les logs du système d'exploitation et des applications. Les logs utilisent un format de données structuré, cela facilite la recherche et l’analyse des logs.

Utilisez *Windows Event Viewer* afin de visualiser les logs Windows.

- Quelles sont les catégories de logs disponibles dans *Event Viewer*?
- Quelles logs contiennent chaque catégorie ?
- Dans quel dossier sont stockées les Event Logs?
- Est-il possible de supprimer une seule entrée des logs Windows?

Créez un *Custom View* avec les logs de démarrage du noyau (provenant de la source “Kernel-Boot”).

- Que permettent de faire les *Custom Views* dans Windows Event viewer?

Sauvegardez et effacez les logs de sécurité, puis ouvrez le fichier sauvegardé avec Windows Event Viewer.

- Quel log de sécurité est créé lorsque vous effacez les logs?
- Dans quelle catégorie le fichier de logs ouvert apparaît-il?

3 - Centralisation des logs

Dans cette section, vous allez planifier et effectuer la centralisation des logs de toutes les machines créées auparavant. Pour cela vous allez déployer et utiliser les 3 solutions de centralisation des logs vues en cours.

- Pourquoi est-il important de centraliser les logs?

3.0 - Planification de collecte des logs

Comme dans le cadre de ce TP la volumétrie des logs n'est pas importante, nous allons choisir la stratégie de collecte de logs semi-maximaliste.
Pour les machines Linux, nous allons collecter et envoyer les logs *syslog*, les logs d'authentification et les logs du serveur *nginx*.
Pour la machine Windows - tous les Event Logs.

- Dans un environnement de production avec un volume de logs très important, cette stratégie est-elle viable?

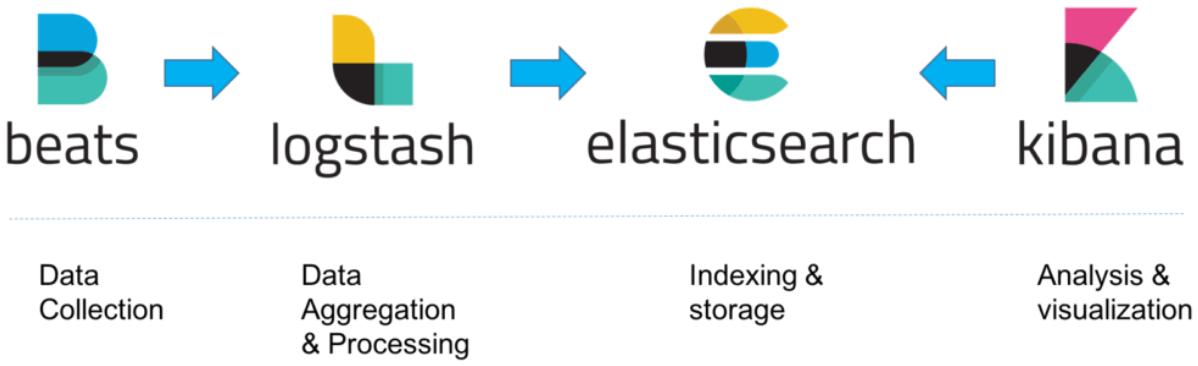
Pour collecter et envoyer les logs des machines, vous allez utiliser un agent.

- Dans quel cas l’utilisation d’un agent n’est-elle pas possible?

Pour faciliter le travail dans le cadre de ce TP, nous n'allons pas configurer d'authentification ou de canaux sécurisés pour transférer les logs. En revanche, c’est obligatoire en production, car les logs peuvent contenir des informations sensibles.

3.1 - Elastic Stack

Elastic Stack est une solution de monitoring et de gestion des logs très populaire. De manière plus générale, cette solution permet de récupérer des données de manière fiable et sécurisée à partir de n'importe quelle source, dans n'importe quel format, puis de les rechercher, les analyser et les visualiser en temps réel. Mais nous limiterons son utilisation à la centralisation et à l'agrégation des logs. Dans cette partie, vous allez donc déployer et configurer un Elastic Stack.



Architecture de stack Elastic

3.1.1 - Installation

Dans cette section, vous allez installer *Elasticsearch*, *Kibana et Logstash* sur la machine “elastic” et un agent collecteur *Beats* (*Filebeat ou Winlogbeat*) sur chaque machine créée auparavant.

Installation de l’Elasticsearch

Elasticsearch est un moteur de recherche, de stockage et d'analyse distribué basé sur Apache Lucene, qui supporte des volumes des données très importantes.

Installez l’Elasticsearch avec les commandes suivantes:

```
$ curl -fsSL https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo
apt-key add -
$ echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" |
sudo tee -a /etc/apt/sources.list.d/elastic-7.x.list
$ sudo apt update
$ sudo apt install elasticsearch
```

Ensuite, testez que Elasticsearch a démarré avec succès en envoyant une requête HTTP GET à l'API REST avec la commande “curl” sur le port 9200.

```
$ sudo systemctl start elasticsearch
$ sudo systemctl enable elasticsearch
$ curl -X GET "localhost:9200"
```

- Quel est le résultat de la commande “curl”?

Installation et configuration de Kibana

Kibana est une interface Web qui permet de visualiser et d’analyser les données stockées dans Elasticsearch.

Installez et démarrez *Kibana*.

```
$ sudo apt install kibana
$ sudo systemctl enable kibana
$ sudo systemctl start kibana
```

Étant donné que Kibana est configuré pour écouter uniquement sur “localhost”, nous devons configurer un reverse proxy pour autoriser l'accès externe. Nous allons utiliser *nginx* comme reverse proxy.

Installation nginx

```
$ sudo apt install nginx
```

Kibana ne propose pas de l’authentification dans sa version gratuite, mais nous pouvons l’ajouter avec un *reverse proxy*. Pour ce faire, vous allez créer un utilisateur et un mot de passe administrateur.

```
$ echo "kibanaadmin:`openssl passwd -apr1`" | sudo tee -a
/etc/nginx/htpasswd.users
```

Configuration du reverse proxy nginx

Créez le fichier de configuration nginx “*/etc/nginx/sites-available/kibana*”

```
server {
    listen 80;

    auth_basic "Restricted Access";
    auth_basic_user_file /etc/nginx/htpasswd.users;

    location / {
        proxy_pass http://localhost:5601;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
```

Supprimez la configuration nginx par défaut et activez la nouvelle configuration

```
$ sudo rm /etc/nginx/sites-enabled/default
$ sudo ln -s /etc/nginx/sites-available/kibana
/etc/nginx/sites-enabled/kibana
$ sudo systemctl reload nginx
```

Vérifiez que le port 80 de la machine “elastic” est bien ouvert dans l’Openstack.

Accédez à Kibana via un navigateur Web en utilisant l’adresse IP de la machine “elastic” et l’utilisateur administrateur créé précédemment.

Visualisez la page *http://IP_ADDR_MACHINE_ELASITC/status* pour vérifier que tout fonctionne correctement.

Installation et configuration de Logstash

Logstash est un agrégateur qui collecte des données à partir de diverses sources d'entrée, exécute différentes transformations, puis les envoie à Elasticsearch.

Installez le *Logstash.*

```
$ sudo apt install logstash
```

Configuration de Logstash

Lorsque vous configurez Logstash, il peut être utile de voir Logstash comme un pipeline qui prend des données à une extrémité, les traite d'une manière ou d'une autre et les envoie à sa destination. Un pipeline Logstash a deux éléments obligatoires, *une entrée* et *une sortie*, et un élément facultatif, *un filtre*.
Pour configurer ce pipeline, nous allons créer deux fichiers de configuration. Un fichier pour configurer l'entrée et l'autre fichier pour configurer la sortie.

Créez le fichier de configuration de l'entrée "/etc/logstash/conf.d/01-beats-input.conf"

```
input {
  beats {
    port => 5044
  }
}
```

Logstash écoutera le port 5044 et attendra les logs au format Beats. Ouvrez le port 5044 de la machine "elastic" dans l'Openstack.

Créez le fichier de configuration de la sortie "/etc/logstash/conf.d/02-elasticsearch-output.conf"

```
output {
  if [@metadata][pipeline] {
    elasticsearch {
    hosts => ["localhost:9200"]
    manage_template => false
    index => "%{[@metadata][beat]}-%{[@metadata][version]}-%{+YYYY.MM.dd}"
    pipeline => "%{[@metadata][pipeline]}"
    }
  } else {
    elasticsearch {
    hosts => ["localhost:9200"]
    manage_template => false
    index => "%{[@metadata][beat]}-%{[@metadata][version]}-%{+YYYY.MM.dd}"
    }
  }
}
```

Logstash enverra des logs dans Elasticsearch.

Démarrez le *logstash* et activez le démarrage automatique

```
$ sudo systemctl start logstash
$ sudo systemctl enable logstash
```

3.1.2 - Configuration des agents collecteurs sur Linux

Dans cette section, vous allez utiliser *Filebeat* comme agent collecteur des logs. Cet agent doit être installe et configuré sur toutes les machines créées au cours de ce TP. Vous allez commencer par la machine “elastic”, car vous devrez effectuer des actions supplémentaires sur cette machine.

Installation et configuration de Filebeat sur Linux

Sur toutes les machines sauf “elastic” vous devez ajouter les repos *Elastic* dans le gestionnaire des packages *apt*.

```
$ curl -fsSL https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo
apt-key add -
$ echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" |
sudo tee -a /etc/apt/sources.list.d/elastic-7.x.list
$ sudo apt update
```

Installez le Filebeat

```
$ sudo apt install filebeat
```

Configuration de Filebeat

Par défaut, *Filebeat* essaye d’envoyer les logs directement à *Elasticsearch*. Dans notre cas, nous voulons que *Filebeat* envoye les logs à *Logstash*. Pour ce faire, nous allons modifier la configuration de Filebeat. Dans le fichier “/etc/filebeat/filebeat.yml” commentez la section “output.elasticseach”.

```
#output.elasticsearch:
 # Array of hosts to connect to.
 #  hosts: ["localhost:9200"]
```

Ensuite decommentez et configurez l’envoie des logs à Logstash

```
output.logstash:
 # The Logstash hosts
 hosts: ["ADRESSE_IP_DE_LA_MACHINE_ELASTIC:5044"]
```

Vous avez configuré la sortie des logs, ils seront envoyés à Logstash.

Vous devez maintenant indiquer à Filebeat quels logs il doit collecter. Vous avez deux possibilités:

- Modifier manuellement le fichier de configuration Filebeat et spécifier les fichiers des logs
- Utiliser des modules Filebeat. Filebeat supporte plusieurs modules afin d’indiquer les sources des logs et d’étendre ses fonctionnalités

Dans cette section, nous utiliserons le module *system*, qui collecte les logs *syslog* (/var/log/syslog) et les logs d'authentification (/var/log/auth.log).

```
$ sudo filebeat modules enable system
$ sudo filebeat setup --pipelines --modules system
```

Activez le module *nginx* sur les machines avec un serveur *nginx*.

- Quelle commande utiliserez-vous?

Une fois que *Filebeat* connaît quels logs à collecter et où les envoyer, il faut créer un indice pour stocker les logs dans *Elasticsearch* et une description des pipelines de traitement des logs. Cette opération ne doit être effectuée que sur la machine “elastic”.

```
$ sudo filebeat setup --index-management -E output.logstash.enabled=false -E
'output.elasticsearch.hosts=["localhost:9200"]'
$ sudo filebeat setup --pipelines --modules system,nginx -E
output.logstash.enabled=false -E
'output.elasticsearch.hosts=["localhost:9200"]'
```

Démarrez le *Filebeat* et activez le démarrage automatique

```
$ sudo systemctl start filebeat
```

```
$ sudo systemctl enable filebeat
```

Vérifiez si Elasticsearch reçoit des données, interrogez l'index Filebeat avec la commande suivante. Cette opération ne doit être effectuée que sur la machine “elastic”.

```
$ curl -XGET 'http://localhost:9200/filebeat-*/_search?pretty'
```

- Elasticsearch reçoit-il des données?

3.1.3 - Visualisation et dashboards dans Kibana

Rappel: le lien pour accéder à l’instance Kibana est http://ADRESSE_IP_DE_LA_MACHINE_ELASTIC/

Dashboards

Filebeat est livré avec quelques tableaux de bord Kibana prédéfinis qui vous permettent d'afficher les données Filebeat dans Kibana.

Importez les dashboards dans Kibana avec la commande suivante. Cette opération ne doit être effectuée que sur la machine “elastic”.

```
$ sudo filebeat setup -E output.logstash.enabled=false -E
output.elasticsearch.hosts=['localhost:9200'] -E
setup.kibana.host=localhost:5601
```

Trouvez et visualisez le dashboard “*[Filebeat System] Syslog dashboard ECS*” dans Kibana.

- Quelles autres dashboards de type *“[Filebeat System]”* sont disponibles dans Kibana?

Visualisation

Visualisez les logs dans l’interface Kibana. (Kibana->Discover)

Trouvez tous les logs provenant de la machine “nginx-server”.

- Quelle requête utiliserez-vous?

3.1.4 - Bonus. Configuration des agents collecteurs sur Windows

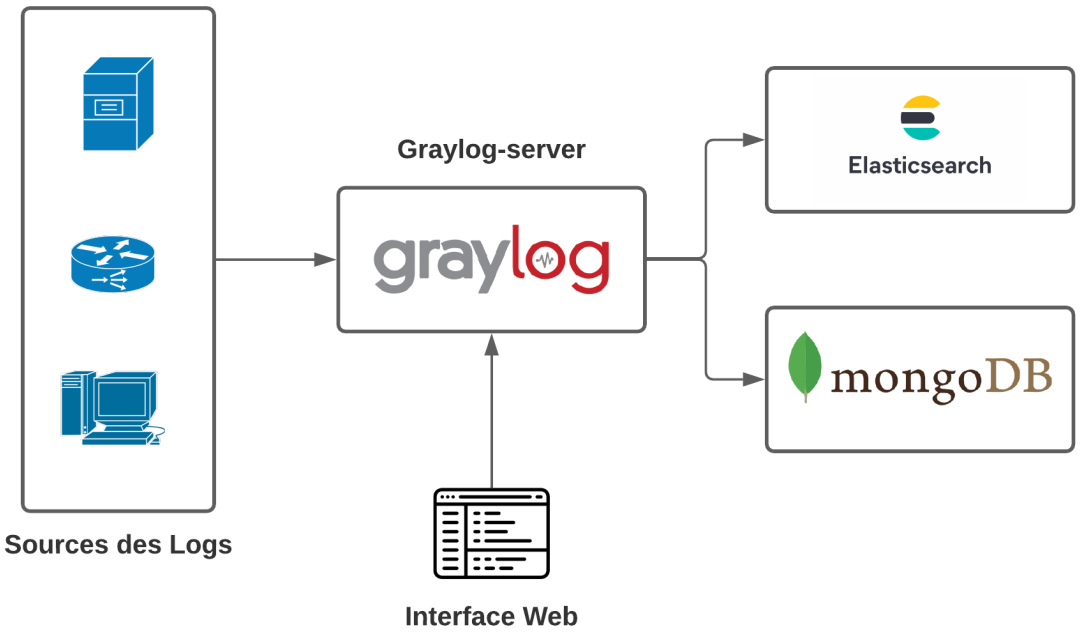
Installez et configurez le *Winlogbeat* sur la machine *Windows*. Le *Winlogbeat* doit collecter et envoyer tous les Event Logs.

3.1.5 - Conclusion

Dans cette section, vous avez installé et manipulé un Elastic Stack.
Si vous envisagez de déployer un Elastic Stack en production, il faudra penser à ajouter au moins de la sécurité sur les inputs (tunnelling et authentification), créer un cluster Elasticsearch, rajouter une solution de buffering devant Logstash (par exemple Redis), ajouter des équilibreurs de charge.
Même si vous avez utilisé Elastic Stack pour la centralisation des logs, il est capable de traiter tous types de messages. C'est en partie pourquoi il est relativement difficile à configurer (ajout des pipelines, création des indices et etc). De plus, de nombreuses fonctionnalités, comme l'alerting, l'authentification, l'apprentissage automatique, ne sont disponibles que dans la version payante.
Dans la section suivante, vous allez mettre en place une solution spécialement conçue pour la centralisation des logs. Cette solution est plus facile à configurer et à maintenir et possède de nombreuses fonctionnalités intéressantes fournies gratuitement.

3.2 - Graylog

Dans cette section, vous allez déployer et configurer la solution de centralisation des logs Graylog avec son système de gestion centralisée des sources de logs Graylog Sidecar.



Architecture de Graylog

3.2.1 - Installation

Vous allez installer *Elasticsearch*, *Mongodb* et *Graylog-server* sur la machine “graylog”. Ensuite, vous allez installer un agent collector *Filebeat ou Winlogbeat* et un agent de gestion de configuration *Graylog Collector* sur chaque machine.

Preparation de la machine “graylog”

Installez les composants nécessaires pour Graylog.

```
$ sudo apt update
$ sudo apt install apt-transport-https openjdk-8-jre-headless uuid-runtime
pwgen
```

Installation de MongoDB

Installez MongoDB

```
$ sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80
--keyserver-options http-proxy=http://proxy.univ-lyon1.fr:3128 --recv
9DA31620334BD75D9DCB49F368818C72E52529D4
$ echo "deb [ arch=amd64 ] https://repo.mongodb.org/apt/ubuntu
bionic/mongodb-org/4.0 multiverse" | sudo tee
/etc/apt/sources.list.d/mongodb-org-4.0.list
$ sudo apt update
$ sudo apt install -y mongodb-org
```

Activez le démarrage automatique de *MongoDB* lors du démarrage du système et vérifiez qu'il est en cours d'exécution.

```
$ sudo systemctl start mongod
$ sudo systemctl enable mongod
$ sudo systemctl --type=service --state=active | grep mongod
```

Installation et configuration de l'Elasticsearch

Graylog ne peut être utilisé qu'avec Elasticsearch 7.x. Dans cette section, nous allons installer la version open source d'Elasticsearch.

```
$ wget -q https://artifacts.elastic.co/GPG-KEY-elasticsearch -O myKey
$ sudo apt-key add myKey
$ echo "deb https://artifacts.elastic.co/packages/oss-7.x/apt stable main" |
sudo tee -a /etc/apt/sources.list.d/elastic-7.x.list
$ sudo apt-get update && sudo apt-get install elasticsearch-oss
```

Configurez Elasticsearch

Pour améliorer la sécurité d'Elasticsearch, nous allons en restreindre l'accès. Pour ce faire, dans "*/etc/elasticsearch/elasticsearch.yml*", recherchez la ligne qui spécifie "*network.host*", décommentez-la et remplacez sa valeur par "localhost" comme ceci:

```
network.host: localhost
```

Modifiez le "*cluster.name*" et ajoutez la ligne "*action.auto_create_index: false*" à la fin du fichier de configuration.

```
cluster.name: graylog
…
action.auto_create_index: false
```

Démarrez et testez Elasticsearch

Démarrez et configurez le service Elasticsearch pour qu'il démarre automatiquement à chaque démarrage du serveur. Ensuite, testez que Elasticsearch est fonctionnel en envoyant une requête HTTP GET à l'API REST avec la commande *curl* sur le port 9200.

```
$ sudo systemctl start elasticsearch
$ sudo systemctl enable elasticsearch
$ sudo systemctl --type=service --state=active | grep elasticsearch
$ curl -X GET "localhost:9200"
```

- Quel est le résultat de la commande “curl”?

Installation et configuration de Graylog

Graylog est un serveur de collecte et de visualisation des logs. C'est le composant clé de la solution Graylog. Graylog stocke les données de configuration dans MongoDB. Elasticsearch est utilisé pour stocker les logs.

Installez le Graylog Server

```
$ wget
https://packages.graylog2.org/repo/packages/graylog-4.2-repository_latest.de
b
$ sudo dpkg -i graylog-4.2-repository_latest.deb
$ sudo apt-get update && sudo apt-get install graylog-server
```

Configuration de Graylog

Le fichier de configuration du serveur Graylog est “/etc/graylog/server/server.conf”. Pour pouvoir démarrer le serveur, vous devez configurer les valeurs de “password_secret” et “root_password_sha2”.
Le “password_secret” est utilisé pour l’encryption de certaines données dans le MongoDB (mots des passe utilisateurs).

Générez un “password_secret” avec

```
$ pwgen -N 1 -s 96
```

Le “root_password_sha2” est le hash du mot de passe de l’utilisateur root (“admin” par défaut).

Créez un hash de mot de passe de l’utilisateur root

```
$ echo -n "Enter Password: " && head -1 </dev/stdin | tr -d '\n' | sha256sum
| cut -d" " -f1
```

Ajoutez ces deux valeurs dans le fichier de configuration “/etc/graylog/server/server.conf”

Configuration de l’interface Web

Pour accéder à l’interface Web Graylog, vous allez installer et utiliser un reverse proxy *nginx*.

Installation nginx

```
$ sudo apt install nginx
```

Créez le fichier de configuration nginx “*/etc/nginx/sites-available/graylog*”

```
server {
    listen 80;

    location / {
        proxy_pass http://localhost:9000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
```

Supprimez la configuration *nginx* par défaut et activez la nouvelle configuration

```
$ sudo rm /etc/nginx/sites-enabled/default
$ sudo ln -s /etc/nginx/sites-available/graylog
/etc/nginx/sites-enabled/graylog
$ sudo systemctl reload nginx
```

Vérifiez si le port 80 de la machine “graylog” est ouvert dans Openstack.

Dans “/etc/graylog/server/server.conf” décommentez l’option “http_external_uri” et mettez l’adresse IP de la machine “graylog”

```
# Default: $http_publish_uri
http_external_uri = http://ADRESSE_IP_DE_LA_MACHINE_GRAYLOG/
```

Démarrez et testez Graylog-server

```
$ sudo systemctl enable graylog-server
$ sudo systemctl start graylog-server
$ sudo systemctl --type=service --state=active | grep graylog
```

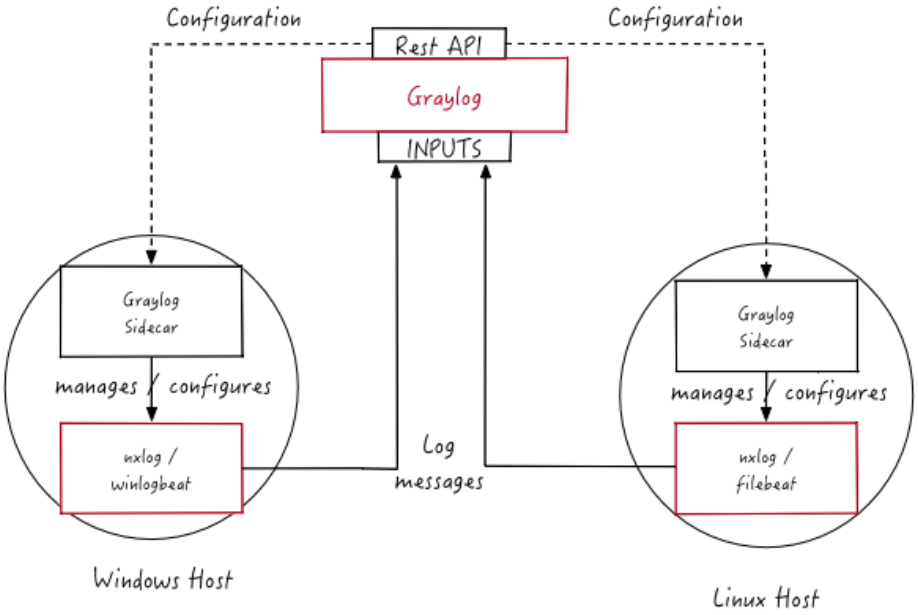
Attendez une minute et vérifiez si l’interface Web Graylog fonctionne correctement. L’interface web Graylog doit être disponible à l’adresse http://ADRESSE_IP_DE_LA_MACHINE_GRAYLOG/

- L'interface Web Graylog fonctionne-t-elle ?

Vous pouvez vous authentifier sur l'interface web Graylog avec l'utilisateur admin et le mot de passe créé précédemment.

3.2.2 - Configuration de Graylog Sidecar sur Linux

Dans cette section, nous n'allons pas configurer les collecteurs de logs manuellement comme nous l'avons fait pour Elastic Stack. Au lieu de cela, nous allons automatiser et centraliser la gestion de la configuration avec Graylog Sidecar. Pour ce faire, nous allons installer le collecteur de logs Filebeat, si cela n'a pas été fait précédemment, et le gestionnaire de configuration Graylog Sidecar sur chaque machine.



Fonctionnement de Graylog avec Sidecar

La configuration des sources des logs sera stockée dans la base de données *Graylog* et sera disponible via l'API REST.
Le *Graylog Sidecar* contacte périodiquement l'API REST du serveur *Graylog* afin de récupérer la configuration et lance le collector de logs avec cette configuration.
Dans notre cas, Graylog Sidecar récupérera la configuration du serveur Graylog et lancera une instance Filebeat avec cette configuration.

Installation de Filebeat

Si vous avez effectué la partie de TP sur Elastic Stack. Le Filebeat doit déjà être présent sur toutes les machines. Sinon,

vous trouverez les instructions d'installation de *Filebeat* dans la section sur Elastic Stack.

Installation de Graylog Sidecar

Installez le Graylog Sidecar

```
$ wget
https://packages.graylog2.org/repo/packages/graylog-sidecar-repository_1-2_a
ll.deb
$ sudo dpkg -i graylog-sidecar-repository_1-2_all.deb
$ sudo apt-get update && sudo apt-get install graylog-sidecar
```

Création du Token pour Graylog Sidecar

Pour que Graylog Sidecar puisse contacter l'API REST du serveur Graylog, vous devez générer un token accès pour le Graylog Sidecar. Vous pouvez le faire via l'interface web Graylog (System -> Sidecars -> Create or reuse a token for the *graylog-sidecar* user). Donnez un nom à votre token et cliquez sur "Create Token". Vous n'avez besoin que d'un seul token pour toutes les machines de ce TP.

Configuration de Graylog Sidecar

Ajoutez le token et l'adresse de l'API Graylog dans le fichier de configuration "/etc/graylog/sidecar/sidecar.yml"

```
# The URL to the Graylog server API.
server_url: "http://ADRESSE_DE_LA_MACHINE_GRAYLOG/api/"

# The API token to use to authenticate against the Graylog server API.
# This field is mandatory
server_api_token: "VOTRE_TOKEN"
```

Démarrage de Graylog Sidecar

Créez le service et démarrez le Graylog Sidecar en utilisant les commandes suivantes

```
$ sudo graylog-sidecar -service install
$ sudo systemctl enable graylog-sidecar
$ sudo systemctl start graylog-sidecar
```

Après quelques secondes, vos Sidecars devraient être visibles dans l'interface web Graylog (System -> Sidecars).

3.2.3 - Configuration des sources et des entrées des logs Linux

Dans cette section, vous allez configurer les sources et les entrées de logs de manière centralisée via l'interface web Graylog.

Configuration des entrées

Pour pouvoir envoyer des logs au serveur Graylog, vous devez créer des entrées (Inputs). Graylog prend en charge une multitude de types d'entrées: AWS, Beats, CEF, GELF, Syslog... Les entrées Graylog sont configurables via l'interface web Graylog (System -> Inputs).
Configurez une entrée de type "Beats", nommez l'entrée et laissez tous les autres paramètres par défaut.

- Quel protocole de sécurité est disponible pour l'entrée de type Beats?
- L'authentification est-elle configurable pour ce type d'entrée?

Par défaut, le port pour l'entrée de type "Beats" est 5044. Ouvrez ce port pour la machine "graylog" dans Openstack.

Configuration des sources

Pour envoyer les logs des machines vers le serveur Graylog, vous allez créer et attribuer la configuration des sources de logs via l'interface web Graylog. Cette configuration sera récupérée par Graylog Sidecar de chaque machine. Une instance de *Filebeat* avec la bonne configuration sera automatiquement lancée sur chaque machine. Ensuite, les journaux seront envoyés par Filebeat au serveur Graylog.

Pour commencer, vous devez créer une configuration *Filebeat* dans l'interface web *Graylog*. (System -> Sidecars -> Manage Sidecar -> Configuration -> Create Configuration)

Créez une configuration pour le collector "filebeat on Linux". Configurez les inputs et l'output *Filebeat*.

- Le *Filebeat* doit collecter et envoyer les logs suivantes

    - “/var/log/syslog”
    - “/var/log/auth.log”
    - “/var/log/nginx/access.log”
    - “/var/log/nginx/error.log”
- Le *Filebeat* doit tout envoyer au “ADRESSE_IP_DE_LA_MACHINE_GRAYLOG:5044” (Ne changez pas le type d’output)
- Quelle configuration allez-vous mettre dans le champ de configuration ?

Puis, il faut attribuer la configuration créée à un Graylog Sidecar. (System -> Sidecars -> Manage sidecar).
Il faut choisir “filebeat”, cliquer sur “Configure” et choisir la configuration créée précédemment. Après la confirmation, Graylog Sidecar récupérera cette configuration et lancera *Filebeat.*

Si tout à été correctement configuré, vous allez voir le status de Filebeat “Running” et vous allez pouvoir visualiser les logs dans la section “Search”.

Attribuez la configuration aux Sidecars de toutes les machines.

Confirmez que toutes les machines envoient des logs au serveur Graylog.

3.2.4 - BONUS. Installation de Graylog Sidecar et configuration des sources sur Windows

Si vous voulez aller plus loin, configurez *Graylog Sidecar* sur la machine Windows. La machine Windows doit envoyer tous les Event Logs au serveur Graylog.
Pour collecter et envoyer les Event Logs, il faudra utiliser *Winlogbeat* au lieu de *Filebeat*.

3.2.5 - Extracteurs et visualisation des logs

Dans les sections précédentes, vous avez configuré Graylog avec une gestion centralisée de la configuration des sources. Désormais, chaque machine envoie des logs à Graylog et nous pouvons les visualiser et les manipuler.

Création des extracteurs

Les extracteurs dans Graylog sont utilisés pour extraire et transformer les données de texte du log en champs pour un filtrage et une analyse plus faciles.

Prenons comme exemple un log du reverse proxy nginx.

```
172.29.27.162 - fa9506c3-f2d6-4d75-bcc4-8f877c798ac2 [08/Jan/2021:09:35:40
+0100] "POST /api/cluster/metrics/multiple HTTP/1.1" 200 295
"http://192.168.246.68/search?q=&rangetype=relative&relative=300"
"Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:84.0) Gecko/20100101
Firefox/84.0"
```

Cette entrée présente de nombreuses informations intéressantes. Par exemple, l'adresse IP du client qui a fait cette demande ou l'URL à laquelle le client a accédé. Vous pouvez extraire ces informations dans des champs séparés. Ces champs seront indexés. En conséquence, les recherches basées sur ces champs seront très rapides. Cela vous permettra ensuite de créer des statistiques et des tableaux de bord basés sur ces champs.

Dans cette section, vous allez extraire l’adresse IP du client avec une expression régulière.
Pour ce faire, trouvez un log *nginx* dans la section “Search” de l’interface web Graylog. Cliquez sur le log trouvé pour voir le message en détail. Sur le champ “message”, cliquez sur la petite flèche, puis “Create extractor”.
Créez un extracteur de type “Regular expression”.
Dans le champ “Regular expression” mettez l’expression régulière suivante

```
^((?<![0-9])(?:(?:[0-1]?[0-9]{1,2}|2[0-4][0-9]|25[0-5])[.](?:[0-1]?[0-9]{1,2
}|2[0-4][0-9]|25[0-5])[.](?:[0-1]?[0-9]{1,2}|2[0-4][0-9]|25[0-5])[.](?:[0-1]
?[0-9]{1,2}|2[0-4][0-9]|25[0-5]))(?![0-9])).*
```

Cliquez sur “Try” pour voir si l’adresse IP a été bien extraite des logs *nginx*.
Mettez une condition d’extraction “Only attempt extraction if field contains string” et mettez “HTTP” dans “Field contains string”. Nommez l'extracteur et le nouveau champ “source_ip”.

Si nous configurons l'extracteur de cette façon, Graylog créera un nouveau champ “source_ip” contenant l'adresse IP du client pour les entrées où le “message” contient la chaîne “HTTP”.

Revenez sur la page de recherche et confirmez que tous les logs *nginx* ont un nouveau champ “source_ip” avec l’adresse IP du client.
Cliquez sur la petite flèche sur “source_ip”, puis “Show top values” pour afficher les adresses IP qui utilisent le plus les serveurs *nginx*.

- Combien d’adresses IP sont affichées dans “Top values”?

Visualisation des logs
Jouez avec l'interface de recherche pour comprendre son fonctionnement (la période de recherche, la mise à jour automatique et etc).

- Quelle requête utilisez-vous pour trouver tous les messages contenant votre adresse IP?
- Quelle requête utilisez-vous pour trouver tous les messages contenant le champ “source_ip”?

Vous pouvez trouver plus d'informations sur la syntaxe des requêtes ici:

- https://docs.graylog.org/en/4.0/pages/searching/query_language.html

BONUS. Creation d’un dashboard
Dans cette section, vous allez créer un tableau de bord qui affichera quelques statistiques des serveurs nginx.

Ce dashboard doit contenir:

- Une liste des serveurs nginx
- Le nombre total des messages provenant des serveurs nginx
- Une liste des adresses IP qui utilisent le plus les serveurs nginx
- Une liste des URLs les plus consultées (pour cela, il faudra créer un extracteur pour extraire les URLs des logs nginx)
    - Quelle expression régulière utiliserez-vous pour créer cet extracteur?

3.2.7 - Rotation et la période de rétention des logs

La rotation des logs est un élément très important de la gestion des logs qui permet d'économiser de l'espace disque, de maintenir des temps d'ouverture et de recherche raisonnables et, dans certains cas, d'augmenter les performances d'écriture. Graylog écrit des messages dans des index sets. Un index set est une configuration de rétention, de partitionnement et de réplication des données stockées.

Dans cette section, vous allez configurer la rotation et la période de rétention de l'index set principal. (System -> Indices -> Default index set -> Edit).

Pour configurer la rotation et la période de rétention de l'index set, vous devrez modifier les paramètres dans "Index Rotation Configuration".
Configurez la stratégie de rotation "Index Time" pour la fréquence de rotation toutes les 12 heures.

- Que mettez-vous dans le champ "Rotation period"?

Configurez la période de rétention pour une rétention des logs pendant 12 mois.

- Combien d'indices faut-il conserver pour avoir la période de rétention des logs de 12 mois?

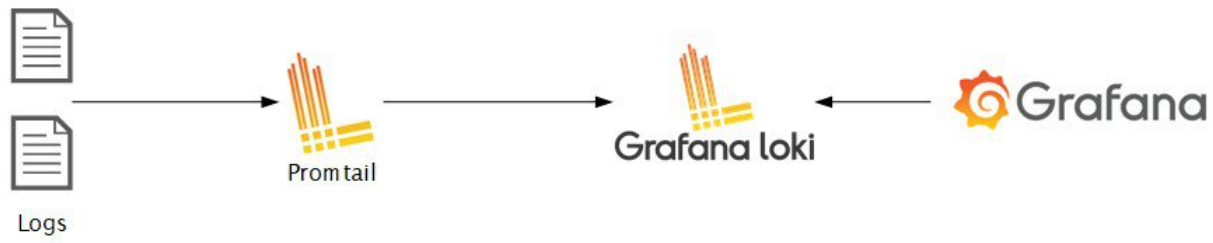
3.2.8 - Conclusion

Dans cette section, vous avez installé, configuré et manipulé le Graylog avec Sidecar. Ayant un temps très limité, même si nous avons fait beaucoup de choses, nous n'avons pas pu voir toutes les fonctionnalités de Graylog. Je vous conseille fortement de regarder la documentation officielle Graylog et d'essayer d'aller plus loin (Alerts, Streams, Users, Groups, Roles, Content Packs).
Pour un déploiement en production, il faudra au moins penser à sécuriser les entrées, passer l'API et l'interface Web à HTTPS et créer un cluster Elasticsearch. Vous pouvez également déployer plusieurs instances du serveur Graylog et de les placer derrière un load balancer.
L'un des principaux avantages de Graylog est qu'il est spécialement conçu pour la centralisation et la gestion des logs. Même dans sa version gratuite, il offre beaucoup de fonctionnalités très avancées (Gestion des utilisateurs et des rôles, Authentification via AD/LDAP, Alerting, fonctionnalités de Threat Intelligence, plugin AWS...).
De plus, une fois configuré, la plupart des actions peuvent être effectuées via son interface web. Et si vous automatisez le déploiement des agents collecteurs et de Graylog Sidecar avec Puppet ou Ansible, vous ne toucherez presque plus jamais les fichiers de configuration.

3.3 - Grafana Loki (PLG Stack)

Grafana Loki est une solution simple, légère et facile à utiliser. La principale différence avec les solutions vues précédemment est que Grafana Loki n'utilise pas Elasticsearch pour le stockage des logs. Cette solution n'indexe que les métadonnées et n'indexe pas le contenu des logs donc nécessite moins de ressources pour son fonctionnement. Cette solution est très utilisée pour la centralisation des logs dans un cluster Kubernetes et peut utiliser le stockage objet qui est un type de stockage relativement peu coûteux.

Dans cette section, vous allez déployer et configurer la solution de centralisation des logs Grafana Loki en mode monolithique avec Docker.



Architecture de Grafana Loki

3.3.1 - Installation et configuration

Dans cette section, vous allez lancer Grafana Loki et Grafana dans des containers Docker sur la machine “loki”. Ensuite, vous allez installer et lancer le binaire Promtail sur chaque machine créée précédemment.

Creation des volumes Docker

Avant de commencer, vous allez créer deux volumes Docker. Le premier volume contiendra la base de données Loki et le second la configuration de l’interface web Grafana. L’utilisation des volumes Docker est utile pour assurer la persistance en cas de recréation du conteneur.

```
docker volume create loki
docker volume create grafana
```

Grafana Loki

Grafana Loki est le composant principal du Stack PLG. Il est responsable de l'agrégation et du stockage des logs.

Créez le fichier de configuration “loki-config.yaml” avec le contenu suivant:

```
auth_enabled: false

server:
 http_listen_port: 3100
 grpc_listen_port: 9096

common:
 path_prefix: /loki
 storage:
   filesystem:
     chunks_directory: /loki/chunks
     rules_directory: /loki/rules
 replication_factor: 1
 ring:
   instance_addr: 127.0.0.1
   kvstore:
     store: inmemory

schema_config:
 configs:
   - from: 2020-10-24
     store: boltdb-shipper
     object_store: filesystem
     schema: v11
     index:
       prefix: index_
       period: 24h

ruler:
 alertmanager_url: http://localhost:9093
```

Lancez Grafana Loki avec Docker

```
docker run -d --name loki -v
$(pwd)/loki-config.yaml:/mnt/config/loki-config.yaml -v loki:/loki -p
3100:3100 grafana/loki:2.4.1 -config.file=/mnt/config/loki-config.yaml
```

Vérifiez si Grafana Loki a été bien lancé

```
docker ps
```

Vérifiez si le port 3100 est bien ouvert sur la machine “loki” dans Openstack, sinon ouvrez-le.

Grafana

Grafana est un outil de visualisation (Web UI) qui affiche les données stockées par Loki.

Pour commencer, définissez le nom d'utilisateur et le mot de passe administrateur avec les variables d'environnement

```
export GF_SECURITY_ADMIN_USER=admin
export GF_SECURITY_ADMIN_PASSWORD=MOT_DE_PASSE_ADMIN
```

Lancez Grafana avec Docker

```
docker run -d --name grafana -e
GF_SECURITY_ADMIN_USER=$GF_SECURITY_ADMIN_USER -e
GF_SECURITY_ADMIN_PASSWORD=$GF_SECURITY_ADMIN_PASSWORD -v
grafana:/var/lib/grafana -p 80:3000 grafana/grafana:latest
```

Vérifiez si Grafana a été bien lancé

```
docker ps
```

L’interface web Grafana sera disponible sur http://ADRESSE_IP_DE_LA_MACHINE_LOKI. Authentifiez-vous avec les identifiants administrateur choisis précédemment.

- Pouvez-vous accéder au Grafana?

Ajout de la source de données Loki dans Grafana

Ajoutez la source de données Loki dans Grafana (Configuration -> Datasources -> Add Datasource -> Loki). Lors de l'ajout de Loki en tant que source de données, mettez http://ADRESSE_IP_DE_LA_MACHINE_LOKI:3100 comme URL de Loki et laissez tous les autres paramètres par défaut. Cliquez sur “Save & Test”.
Si tout a été démarré et configuré correctement, vous verrez le message “Data source connected and labels found.”.

- Avez-vous vu ce message?

3.3.2 - Configuration des sources et des entrées des logs Linux

Dans cette section, vous allez déployer le binaire et configurer Promtail sur toutes les machines Linux créées dans ce TP.

Vérifiez si le port 3100 est bien ouvert sur la machine “loki” dans Openstack, sinon ouvrez-le.

Installation de Promtail

Sur chaque machine Linux, exécutez les commandes suivantes

```
sudo apt install -y unzip
curl -O -L
"https://github.com/grafana/loki/releases/download/v2.4.1/promtail-linux-amd64.zip"
unzip promtail-linux-amd64.zip
sudo mv promtail-linux-amd64 /usr/local/bin/promtail
```

Créez un service Promtail dans systemd. Pour cela, vous devez créer le fichier “/etc/systemd/system/promtail.service” avec le contenu suivant

```
[Unit]
Description=Promtail Service

[Service]
User=root
Group=root
ExecStart=/usr/local/bin/promtail -config.file=/opt/promtail-config.yaml

Restart=always

[Install]
WantedBy=multi-user.target
```

Configuration et démarrage de Promtail

Créez le fichier de configuration Promtail dans “/opt/promtail-config.yaml”

- Attention! Dans ce fichier de configuration, vous devez modifier “ADRESSE_IP_DE_LA_MACHINE_LOKI” et “HOSTNAME_DE_LA_MACHINE_COURANTE”.

```
server:
```

```
  http_listen_port: 9080
  grpc_listen_port: 0

positions:
  filename: /tmp/positions.yaml

clients:
  - url: http://ADRESSE_IP_DE_LA_MACHINE_LOKI:3100/loki/api/v1/push

scrape_configs:
- job_name: system
  static_configs:
  - targets:
      - localhost
      labels:
      host: HOSTNAME_DE_LA_MACHINE_COURANTE
      job: varlogs
      __path__: /var/log/*log
```

- A partir de cette configuration, pouvez-vous dire quels fichiers seront surveillés par Promtail pour collecter les logs?

Démarrez le service Promtail, activez son démarrage automatique et vérifiez s'il fonctionne correctement

```
sudo systemctl daemon-reload
sudo systemctl start promtail
sudo systemctl enable promtail
sudo systemctl status promtail
```

- Avez-vous réussi à démarrer le Promtail ?

N'oubliez pas de configurer et de lancer Promtail sur chaque machine Linux.

3.3.4 - Visualisation des logs

Si tout a été configuré correctement, vous devriez pouvoir consulter les logs via Grafana dans la section “Explore”.

Utilisez Log Browser pour explorer les logs et les étiquettes (labels) disponibles.

- Quels labels sont disponibles dans le Log Browser?

- Quelle requête Loki utiliseriez vous pour afficher toutes les logs provenant des fichiers “/var/log/syslog” de toutes les machines?

3.3.5 - Conclusion

Dans cette section, vous avez installé et configuré la solution de centralisation des logs Grafana Loki. Grafana Loki est une solution très légère et adaptée à l'environnement avec des microservices. Cette solution montre toute sa puissance et son potentiel d'évolutivité lorsqu'elle est utilisée dans un cluster Kubernetes.
Grafana Loki est très intéressant pour centraliser les logs d'une petite infrastructure, car il est facile à installer et a peu de composants.
L'utilisation de Grafana Loki peut également être intéressante si votre stack de monitoring est basé sur Prometheus. Dans ce cas, vous n'aurez qu'une seule interface web Grafana pour consulter les logs et les données de la supervision.

Bravo! Vous avez fini le TP!