TP Gestion des Logs

0 - Introduction

Au cours de ce TP, vous allez manipuler les différents outils de gestion des logs. Nous allons commencer par la préparation de l'infrastructure, puis nous allons continuer par la visualisation et configuration des logs sur des machines Linux et Windows. Nous allons finir par l'installation et la configuration des deux solutions de centralisation des logs vus en cours: stack Elastic et Graylog.

1 - Préparation de l'infrastructure

Dans cette section il faudra créer 4 machines virtuelles dans l'OpenStack avec les caractéristiques suivantes:

- 3 machines Ubuntu 20.04.1 et 1 machine Windows 10 (BONUS)
- 2 vCPU
- 4GB RAM
- 10GB Espace disque

Ces machines auront des hostnames suivants:

- nginx-server (Ubuntu)
- graylog (Ubuntu)
- elastic (Ubuntu)
- windows-web-server (Windows 10) (BONUS)

2 - Logs Linux

Dans cette section nous allons voir comment fonctionnent les logs Linux en pratique.

Architecture des logs de l’espace utilisateur

Nous allons visualiser, configurer et générer les logs sur la machine “nginx-server”.

Pour commencer vous allez installer un serveur *nginx* avec le gestionnaire des packages *apt* et ouvrir le port 80 dans *l’Openstack* (Vous pouvez ajouter le security group HTTP à votre machine).

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

Analysez le fichier de configuration de rsyslog (/etc/rsyslog.conf).

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

Configurez rsyslog pour qu'il envoie tous les logs qui contiennent le mot "ssh" dans le fichier "/var/log/ssh.log". (Créez un fichier de configuration dans "/etc/rsyslog.d/", n'oubliez pas à redemarrer le rsyslog)

Trouvez un moyen de vérifier si le fichier "/var/log/ssh.log" à et contient les entrées qui contiennent 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".

- Est-ce que les logs sont 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 nous allons voir comment nous pouvons générer les logs avec la ligne de commande.

Logs noyau

Envoyez la phrase "Hello world" dans l'espace logs noyau et visualisez la via la commande "dmesg". (Envoyez les logs en tant qu'utilisateur *root* sur "/dev/kmsg").

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

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

Logs espace utilisateur

Envoyez la phrase “Hello world” dans le 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 pour l'écriture des logs du journal de 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 suffit de purger, supprimer ou éditer le fichier.
Supprimez tous les logs du fichier “/var/log/syslog”. (Ne supprimez pas le fichier lui même)

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

Logrotate

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

Visualisez le fichier de configuration de logrotate.

- 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 se produire chaque jour, 7 derniers fichiers doivent être conservés, la compression doit être activée.

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 nous allons 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'une prison, de l'interdiction / de la suppression d'un hôte, etc.
Les actions sont stockés dans “/etc/fail2ban/action.d”

Les prisons (jails)

Une prison (jail) est une combinaison d'un filtre et d'une ou plusieurs actions. Vous pouvez configurer les jails 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”

Essayez d’effectuer plusieurs tentatives de connexion avec un mot de passe erroné au serveur SSH jusqu'à être bloqué par *fail2ban*.

- Combien de tentatives de connexion ont échoué avant que le fail2ban vous bloque?

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”.

Supprimez votre adresse IP du jail avec le client *fail2ban*.

2 - BONUS. Logs Windows

Dans cette section nous allons 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 des logs disponibles dans *Event Viewer*?
- Quelles logs sont disponibles dans chaque catégorie?
- Dans quel dossier sont stockées les Event Logs?
- Pouvez-vous 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-les dans l'Event Viewer.

3 - Centralisation des logs

- 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'autorisation et les logs du serveur *nginx*.
Pour la machine Windows - tous les Event Logs.

- Est-ce que dans un environnement de production avec une volumétrie des logs très importante, cette stratégie est-elle viable?

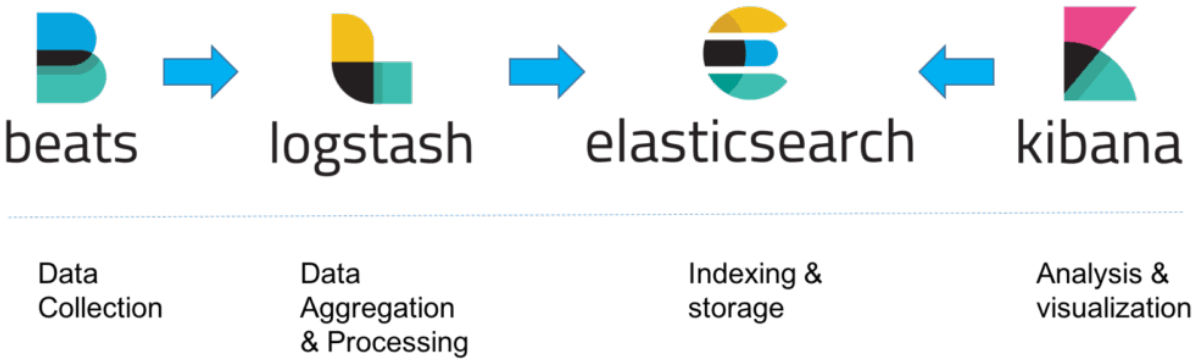
Pour collecter et envoyer les logs nous allons utiliser un agent.

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

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

3.1 - Elastic Stack

Dans cette section vous allez 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.

Installation et configuration de l’Elasticsearch

```
$ 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
```

Configurez Elasticsearch

Pour restreindre l'accès et améliorer la sécurité 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
```

Démarrez et testez Elasticsearch

Démarrez et configurez le démarrage automatique du service *elasticsearch* à chaque démarrage de serveur. Puis testez que

Elasticsearch a été bien démarré 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

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 cela vous allez créer un utilisateur et un mot de passe administrateur.

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

Configuration 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
```

Ouvrez le port 80 de la machine “elastic” dans l’Openstack (Vous pouvez ajouter le security group HTTP à votre machine).

Accédez au 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

Installez le *Logstash*

```
$ sudo apt install logstash
```

Configuration de Logstash

Lorsque vous configurez le Logstash, il est utile de considérer 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.

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.

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
```

Ouvrez le port 5044 de la machine “elastic” dans l’Openstack (Vous pouvez ajouter le security group LOGSTASH à votre machine).

3.1.2 - Configuration des agents collecteurs sur Linux

Dans cette section nous allons utiliser *Filebeat* comme agent collecteur des logs. Cet agent doit être installe et configuré sur toutes les machines créées pendant ce TP. Vous allez commencer par la machine “elastic”, car pour cette machine il faudra faire des actions supplémentaires.

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 dans *Elasticsearch*. Dans notre cas, nous voulons que *Filebeat* envoye les logs dans *Logstash*.

Pour faire cela 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 la section

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

Filebeat supporte des modules afin d'étendre les fonctionnalités. Dans le cadre de ce TP nous utiliserons le module *system*, qui collecte les logs Syslog (/var/log/syslog) et les logs d'autorisation (/var/log/auth.log).

```
$ sudo filebeat modules enable 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 est à faire uniquement 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
```

Testez si Elasticsearch reçoit effectivement des données, interrogez l'index Filebeat avec la commande suivante. **Cette opération est à faire uniquement sur la machine "elastic".**

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

- Est-ce que Elasticsearch reçoit 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 des exemples des tableaux de bord *Kibana* qui vous permettent de visualiser les données *Filebeat* dans *Kibana*.

Importez les dashboards avec *Filebeat* depuis la machine "elastic"

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

- Quel est le résultat d'exécution de cette commande?

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

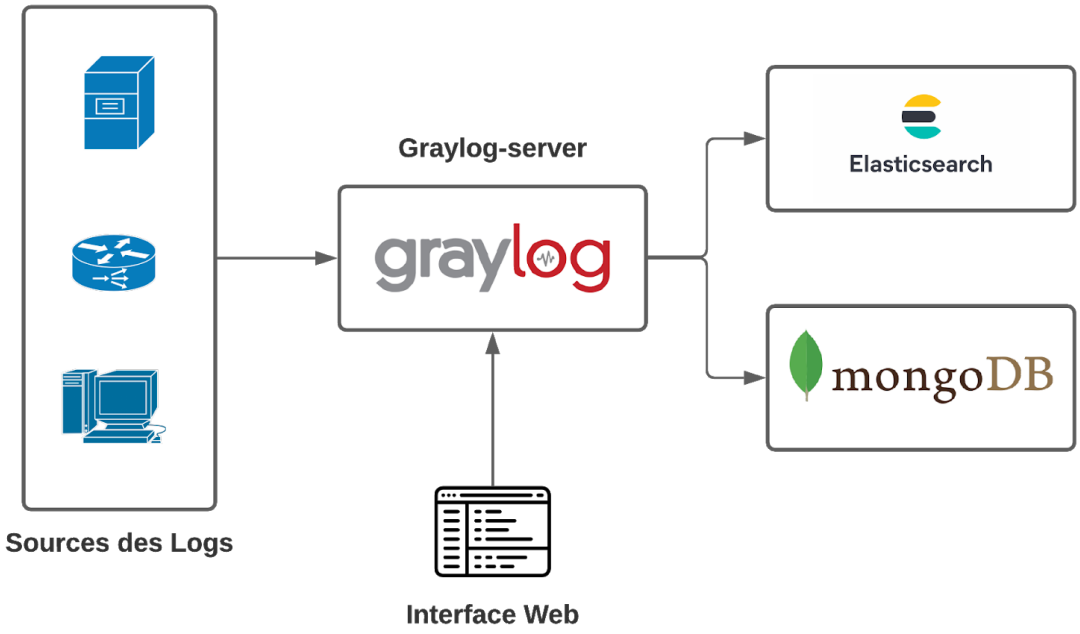
Installez et configurez le *Winlogbeat* sur la machine *Windows*. Le *Winlogbeat* doit collecter et envoyer tous les Event Logs.

3.1.5 - Conclusion

Dans cette section vous avez installé et manipulé un stack Elastic.
Si vous pensez à déployer un stack Elastic 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 on a utilisé le stack Elastic pour la centralisation des logs, il est capable de traiter tous les types de messages. C'est en partie pourquoi il est relativement difficile à configurer et à maintenir.
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 nous allons déployer une solution qui est spécialement conçue pour la centralisation des logs. Cette solution est plus simple à configurer et a beaucoup plus de fonctionnalités fournies gratuitement.

3.2 - Graylog

Dans cette section vous allez déployer et configurer Graylog avec Graylog Sidecar.



Architecture de Graylog

3.2.1 - Installation

Vous allez installer *Elasticsearch*, *Mongodb* et *Graylog-server* sur la machine “graylog”. Puis 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 packages nécessaires pour le fonctionnement de *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 supporte Elasticsearch 7.x, par contre 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 restreindre l'accès et améliorer la sécurité 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
```

Changez le *cluster.name* et ajouter la ligne "*action.auto_create_index: false*"

```
cluster.name: graylog
…
```

```
action.auto_create_index: false
```

Démarrez et testez Elasticsearch

Démarrez et configurez le démarrage automatique du service *elasticsearch* à chaque démarrage de serveur. Puis testez que Elasticsearch est bien 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

Installez le Graylog Server

```
$ wget
https://packages.graylog2.org/repo/packages/graylog-4.0-repository_latest.de
b
$ sudo dpkg -i graylog-4.0-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 rajouter les valeurs dans “password_secret” et “root_password_sha2”.

Le “password_secret” est utilisé pour l’encryption des 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 defaut).
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 nous allons 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
```

Ouvrez le port 80 de la machine “graylog” dans l’Openstack (Vous pouvez ajouter le security group HTTP à votre machine).

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

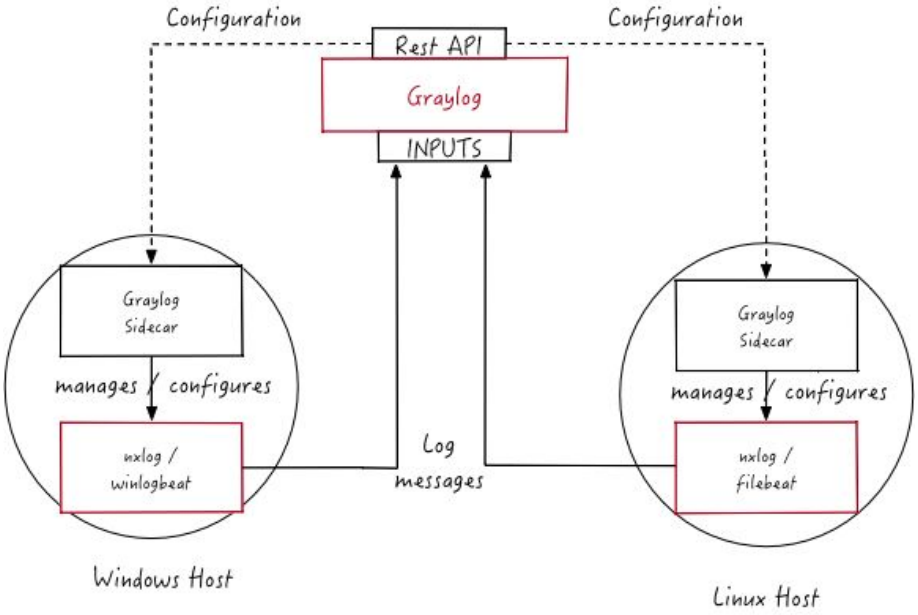
```
$ sudo systemctl enable graylog-server
$ sudo systemctl start graylog-server
$ sudo systemctl --type=service --state=active | grep graylog
```

Attendez une minute. L’interface web Graylog doit être disponible sur http://ADRESSE_IP_DE_LA_MACHINE_GRAYLOG/

- Est-ce que l’interface web Graylog fonctionne?

3.2.2 - Configuration de Graylog Sidecar sur Linux

Dans cette section nous allons installer et configurer le collecteur des logs *Filebeat* 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 le REST API.

Le *Graylog Sidecar* contacte périodiquement le REST API du serveur *Graylog* afin de récupérer la configuration et lancer le collector des logs avec cette configuration.
Dans notre déploiement *Graylog Sidecar* récupérera la configuration depuis le serveur Graylog et lancera une instance *Filebeat* avec cette configuration.

Installation de Filebeat

Si vous avez effectué la partie de TP sur le stack Elastic. Le Filebeat doit déjà être présent sur toutes les machines. Sinon vous trouverez les instructions de l'installation de *Filebeat* dans la section sur le stack Elastic.

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

Vous avez besoin de 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 avez besoin 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

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 doivent être visibles dans l'interface Web (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 des logs via l'interface web Graylog.

Configuration des entrées

Les nœuds Graylog acceptent les données via des entrées. Graylog supporte une multitude de types d'entrées de logs: 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 de l'entrée "Beats" est 5044. Ouvrez ce port pour la machine "graylog" dans Openstack. (Vous pouvez ajouter le security group BEATS à votre machine).

Configuration des sources

Dans cette section nous allons créer et attribuer la configuration de sources des logs via l'interface web *Graylog*. Cette configuration sera récupérée par Graylog Sidecar et sera utilisée pour la configuration de *Filebeat*.

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

Créez une configuration "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)

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 *Filbeat.*

Si tout à été configuré correctement, vous pouvez visualiser les logs dans la section "Search".

Attribuez la configuration aux Sidecars de toutes les machines.

Confirmez que toutes les machines envoient des logs à 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 les Event Logs à Graylog.
Pour collecter et envoyer les Event Logs, il faudra utiliser *Winlogbeat* au lieu de *Filebeat*.

3.2.5 - Extracteurs et visualisation des logs

Création des extracteurs

Les extracteurs sur Graylog sont utilisés pour extraire et transformer des données texte en champs pour un filtrage et une analyse plus faciles.
Dans cette section nous allons créer un extracteur simple.

Prenons par 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"
```

Ce message présente beaucoup d'informations intéressantes. Par exemple l'adresse IP client qui a effectué cette requête ou l'URL accède par le client. Nous pouvons extraire cette information dans des champs séparés. Ensuite nous pourrons effectuer les recherches et l'analyse sur ces champs.

Dans cette section nous allons extraire l'adresse IP client avec une expression régulière.
Pour faire cela, 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 sollicitent 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

3.2.6 - Creation des dashboards

Dans cette section vous allez créer un dashboard 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 sollicitent le plus les serveurs *nginx*

- Une liste des URLs les plus consultées (pour cela il faudra créer un extracteur afin d'extraire les URLs des logs *nginx*)
    - Quelle expression régulière utiliserez-vous pour créer cet extracteur?

BONUS: Si il vous reste du temps, créez un dashboard qui affiche les statistiques des logs d'accès.

3.2.7 - Rotation et la période de rétention des logs

La rotation des logs est un élément très important dans la gestion des logs qui permet d'économiser de l'espace disque, de garder les temps d'ouverture et de recherche raisonnables et, dans certains cas, d'augmenter les performances d'écriture.

Un flux Graylog écrit des messages dans des index sets, qui 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 de l'index set, il faudra changer les paramètres dans "Index Rotation Configuration".
Configurez la stratégie de rotation "Index Time" avec la fréquence de rotation toutes les 12 heures.

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

Pour configurer la période de rétention de l'index set, il faudra changer les paramètres dans "Index Rotation Configuration".
Configurez la période de rétention pour une rétention des logs pendant 12 mois.

- Combien d'indices doivent être conservés 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 minimum penser à sécuriser les inputs, passer l'API et l'interface Web en HTTPS et créer un cluster Elasticsearch. Vous pouvez aussi déployer plusieurs instances du serveur Graylog et de les mettre 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 propose 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 du Graylog Sidecar avec Puppet ou Ansible, vous ne toucherez presque plus jamais les fichiers de configuration.

Bravo! Vous avez fini le TP!