



## Gestion des logs

Vladimir Ostapenco  
pro@vladost.com

# Planning

1. Introduction
2. Logs Linux
3. Logs Windows
4. Gestion des logs
5. Solutions de gestion des logs
6. Demo Time

- **CM:** 3h
- **TP:** 6h

# Qu'est-ce qu'un log ?

- Un fichier qui contient des événements, des messages et d'autres données provenant d'applications, de systèmes d'exploitation ou d'appareils
- Produit automatiquement chaque fois que certains événements se produisent dans un système informatique
- Permet de documenter et de suivre les activités des
  - Utilisateurs, serveurs, réseaux, applications
- Entrées de ce fichier sont
  - Classées par ordre chronologique
  - Horodatées

/var/log/auth.log

```
Dec  7 20:22:03 node 1 sshd[657131]: Failed password for root from  
x.x.x.x port 54512 ssh2
```

/var/log/dpkg.log

```
2020-12-07 13:22:17 trigproc libc-bin:amd64 2.32-0ubuntu3 <none>  
2020-12-07 13:22:17 status half-configured libc-bin:amd64 2.32-0ubuntu3  
2020-12-07 13:22:18 status installed libc-bin:amd64 2.32-0ubuntu3
```

/var/log/nginx/access.log





```
10.0.1.31 - - [07/Dec/2020:19:42:20 +0000] "POST /loki/api/v1/push HTTP/1.1" 204 0 "-" "promtail/2.0.0"  
10.0.1.51 - - [07/Dec/2020:19:42:21 +0000] "POST /loki/api/v1/push HTTP/1.1" 204 0 "-" "promtail/2.0.0"  
10.0.1.32 - - [07/Dec/2020:19:42:21 +0000] "POST /loki/api/v1/push HTTP/1.1" 204 0 "-" "promtail/2.0.0"
```

# Types et sources des logs

- Types des logs

- Logs d'audit
- Logs des transactions
- Logs d'événements
- Logs d'erreurs
- Logs de sécurité
- Logs de messages

59	0000:000004cd	login: ASPNET	INSERT	2012/01/15 20:30:09:730
59	0000:000004cc	login: ASPNET	CREATE TABLE	2012/01/15 20:30:09:713

	Error	28/04/2016 20:25:51	Service...	7026	None
	Error	28/04/2016 20:25:49	Service...	7000	None
	9/5/2016 4:05:4...	7179	102	Information	Application ESENT
	9/5/2016 4:05:4...	7180	300	Information	Application ESENT

- Sources des logs

- Infrastructure réseau, stockage, serveurs
- Postes de travail
- Dispositifs de sécurité
- Systèmes d'exploitation
- Hyperviseurs et systèmes de gestion des conteneurs
- Applications
  - Serveurs Web
  - Serveurs d'authentification
  - Proxies / passerelles Web
  - ...

# Pourquoi les logs c'est important ?

- Les logs permettent de
  - Comprendre ce qui se passe
  - Détecter et comprendre une erreur
  - Détecter et comprendre un événement ou une panne
  - Détecter un incident de sécurité
  - Suivre les actions des utilisateurs
  - Établir des statistiques

# Logs Linux

- **Logs espace noyau**

- Erreurs, warnings ou messages du noyau
- Stockés sur le **Kernel Ring Buffer**
  - Structure de données qui stocke les logs lorsque le système démarre
  - Matérialisé par un fichier de périphérique `/dev/kmsg` et `/proc/kmsg`
  - Visualisable avec la commande **dmesg**
  - Écrit dans un fichier par **rsyslogd** (ou **klogd** sur les anciens systèmes)

- **Logs espace utilisateur**

- Liés aux processus ou aux services qui s'exécutent sur la machine hôte
- Basés sur le protocole Syslog

# Logs Linux - Syslog

- **Standard** pour produire, transmettre et collecter les logs
- **Protocole (RFC 5424)** qui spécifie
  - Format des messages
  - Comment transmettre les logs sur un réseau
- **Service**, qui reçoit et traite les messages Syslog
  - Écriture de messages dans un fichier local
  - Transfert de messages vers un serveur distant
  - Implémentations les plus connus pour Linux : **syslog-ng** et **rsyslogd**

# Logs Linux - Syslog Format

<165>1 2019-08-01T15:30:54.001Z ubuntu-box apache 200 20031 - " The Apache Server encountered an error"



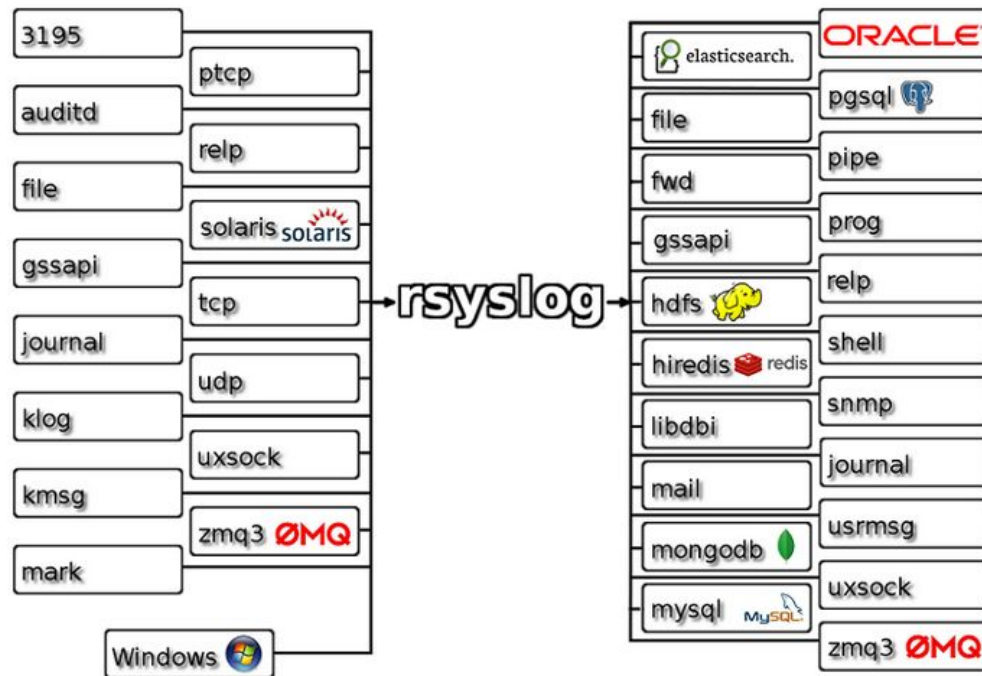
The diagram illustrates the Syslog format components. The string "<165>1 2019-08-01T15:30:54.001Z ubuntu-box apache 200 20031 - \" The Apache Server encountered an error\"" is shown. Below it, three blue brackets identify the parts: "PRI" under "<165>1", "HEADER" under "2019-08-01T15:30:54.001Z ubuntu-box", and "MSG" under "apache 200 20031 - \" The Apache Server encountered an error\"".

- Message Syslog est constitué d'une valeur de priorité (PRI), d'un en-tête standardisé (HEADER) et d'un message contenant le log (MSG)
- Définit trois termes importants
  - **Facility level** – utilisé pour déterminer le programme ou la partie du système qui a produit le log
    - **Exemples** : kern (0) - messages du noyau; daemon (3) - démons système; auth (4) - messages de sécurité
    - Plus de 23 niveaux de facility différentes
  - **Severity level** – utilisé pour connaître la gravité d'un événement
    - **Exemples** : debug (7), warning (4), error (3), critical (2), emergency (0)
  - **Priority value (PRI)** = Facility level \* 8 + Severity level



# Logs Linux - Rsyslog

- Système de traitement des logs
- Implémente Syslog
- Architecture modulaire
- Capable d'accepter les entrées d'une grande variété de sources, de les transformer et d'envoyer les résultats vers diverses destinations
- Peut livrer plus d'un million de messages par seconde vers des destinations locales
- Installé par défaut sur la plupart des systèmes Linux modernes



Source: <https://www.rsyslog.com/>

# Logs Linux - Rsyslog - Configuration

Configuration par défaut */etc/rsyslog.conf*

```
# /etc/rsyslog.conf configuration file for rsyslog
#
# For more information install rsyslog-doc and see
# /usr/share/doc/rsyslog-doc/html/configuration/index.html
#
# Default logging rules can be found in /etc/rsyslog.d/50-default.conf

#####
#### MODULES ####
#####

module(load="imuxsock") # provides support for local system logging
#module(load="imark") # provides --MARK-- message capability

# provides UDP syslog reception
#module(load="imudp")
#input(type="imudp" port="514")

# provides TCP syslog reception
#module(load="imtcp")
#input(type="imtcp" port="514")

# provides kernel logging support and enable non-kernel klog messages
module(load="imklog" permitnonkernelfacility="on")
```

Règles par défaut */etc/rsyslog.d/50-default.conf*

```
# Default rules for rsyslog.
#
# For more information see rsyslog.conf

#
# First some standard log files. Log by facility.
#
auth,authpriv.* /var/log/auth.log
*.*;auth,authpriv.none -/var/log/syslog
#cron.* /var/log/cron.log
#daemon.* -/var/log/daemon.log
kern.* -/var/log/kern.log
#lpr.* -/var/log/lpr.log
mail.* -/var/log/mail.log
#user.* -/var/log/user.log

#
# Logging for the mail system. Split it up so that
# it is easy to write scripts to parse these files.
#
#mail.info -/var/log/mail.info
#mail.warn -/var/log/mail.warn
mail.err /var/log/mail.err
```

# Logs Linux - Emplacement des fichiers des logs

- Stockés dans `/var/log`
- Fichiers des logs les plus importants
  - `/var/log/syslog` et `/var/log/messages` - tous les logs d'activité globale du système, y compris les messages de démarrage
  - `/var/log/auth.log` et `/var/log/secure` - tous les événements liés à la sécurité tels que les connexions, les actions de l'utilisateur root et les messages des modules PAM
  - `/var/log/kern.log` - événements du noyau, y compris les erreurs et les warnings
  - `/var/log/cron` - informations sur les tâches planifiées CRON

# Logs Linux - Systemd

- Gestionnaire de systèmes et de services
- Implémente son propre service de journalisation appelé **systemd-journald (journald)**
  - Solution de gestion centralisée des logs de tous les processus du noyau et de l'espace utilisateur
  - Gère tous les messages produits par le noyau, les services et les applications
- Logs **journald**
  - Stockés dans un format binaire
  - Indexés et structurés
  - Bénéficie des mécanismes supplémentaires : rotation automatique et contrôle d'accès



# Logs Linux - Journald

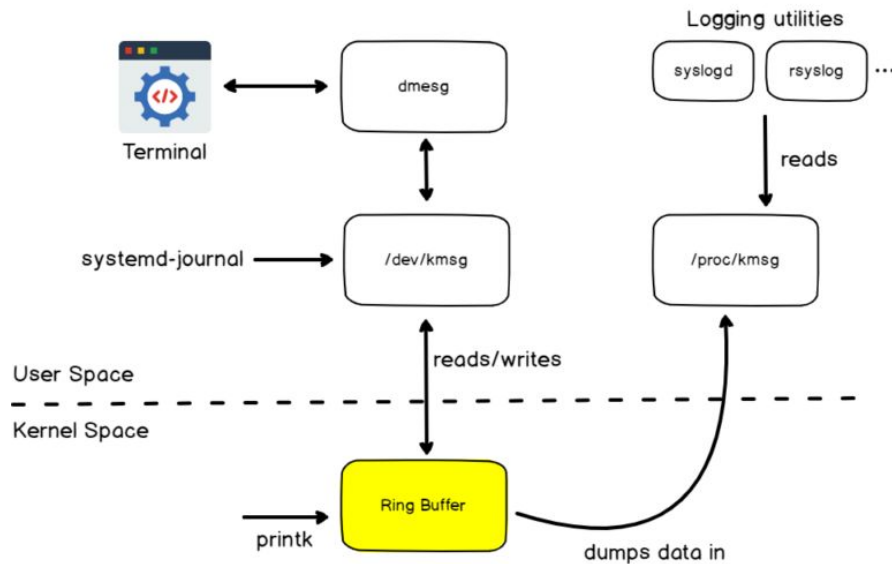
- Prend en charge deux modes de stockage
  - In-memory (Stockage dans la RAM) : les logs sont écrits dans `/run/log/journal`
  - Persistent (Stockage sur disque) : les logs sont écrits dans `/var/log/journal`
- Consultation des logs avec “**journalctl**”
  - **journalctl --since yesterday** – voir tous les logs depuis hier
  - **journalctl -u nginx.service** – voir tous les logs du service Nginx
  - **journalctl -k** – voir tous les messages du noyau (équivalent au dmesg)
  - **journalctl -f** – suivre activement les logs au fur et à mesure de leur écriture
- Maintenance des logs **journald**
  - **journalctl --disk-usage** – voir l'espace qu'occupent les logs sur le disque
  - **journalctl --vacuum-size=1G** / **journalctl --vacuum-time=1day** – supprimer les logs

# Logs Linux - Journald et Rsyslog

- Systèmes sans **systemd**
  - Logs sont collectés par **syslog (rsyslogd)**
- Systèmes avec **systemd**
  - Logs collectés par **systemd-journal**
  - Écrits dans des fichiers par **rsyslogd**
- Ces deux systèmes coexistent principalement pour les raisons de rétrocompatibilité
  - Des applications peuvent utiliser des bibliothèques **syslog** ou **journald** afin d'envoyer des logs

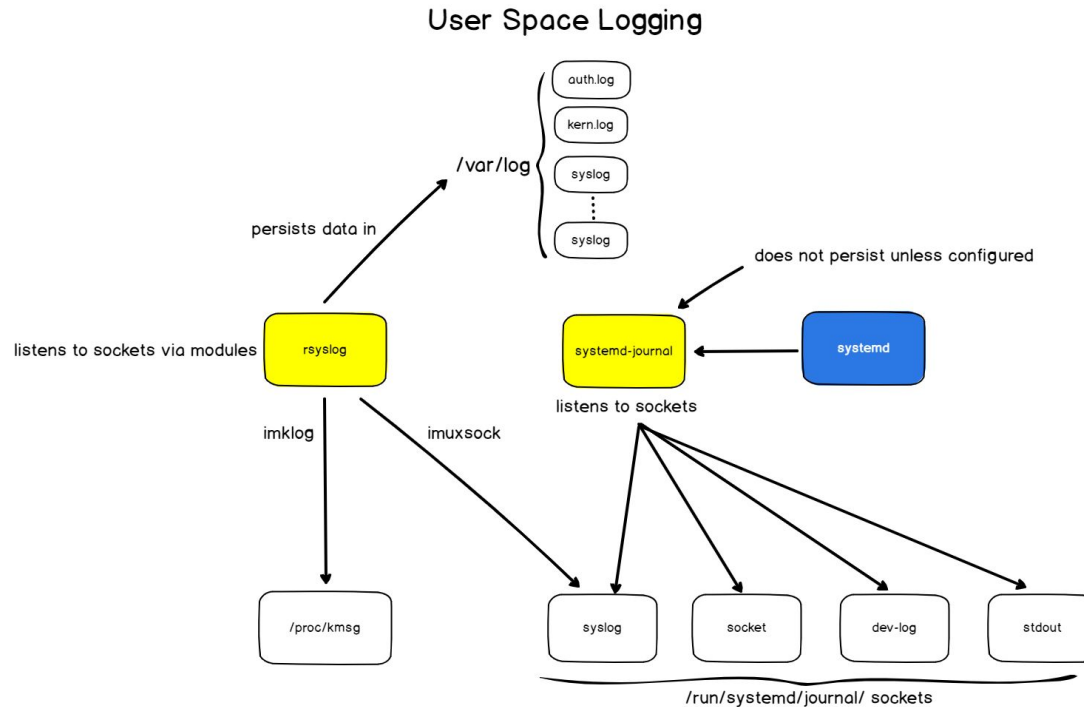
# Logs Linux - Espace noyau

## Kernel Logging Complete



Source : <https://devconnected.com/linux-logging-complete-guide/>

# Logs Linux - Espace utilisateur



Source : <https://devconnected.com/linux-logging-complete-guide/>



# Logs Linux - Fail2ban - Première analyse des logs

- Logiciel de prévention des intrusions conçu pour empêcher les attaques brute-force
- Analyse les logs et interdit les adresses IP qui montrent des signes de comportement malveillant
  - Trop d'échecs d'authentification
  - Recherche d'exploits
- Met à jour les règles de pare-feu (iptables) afin de rejeter les adresses IP pendant une période de temps
- Peut faire une autre action arbitraire (Envoi d'un mail)
- Livré avec des filtres préconfigurés pour divers services (> 90 filtres)
  - **SSH, Apache, MySQL**



**fail2ban**

# Logs Linux - Fail2ban

- Installation et configuration facile
  - Disponible dans la plupart des gestionnaires de paquets Linux
  - Par défaut est configuré pour protéger votre serveur SSH
- À utiliser sur tous vos serveurs exposés à Internet !

- **Statut Fail2ban sur un de mes serveurs**

```
Status for the jail: sshd
|- Filter
|  |- Currently failed: 10
|  |- Total failed:    93795
|  `-- File list:      /var/log/auth.log
`- Actions
   |- Currently banned: 7
   |- Total banned:    15128
   `-- Banned IP list:  [redacted]
```

# Logs Windows - Windows Event Logs

- **Windows Event Logs** contient les logs du système d'exploitation et des applications
  - *SQL Server* ou *Internet Information Services (IIS)*
  - Logs sont stockés dans `C:\Windows\System32\winevt\Logs`
- Logs utilisent un format de données structuré
  - Facilite la recherche et l'analyse
- Certaines applications écrivent les logs directement dans des fichiers
  - Logs accès *Internet Information Services (IIS)*
- **Windows Event Viewer** peut être utilisé pour visualiser *Windows Event Logs*
  - Permet d'afficher, parcourir, rechercher, filtrer, exporter, configurer et effacer les logs

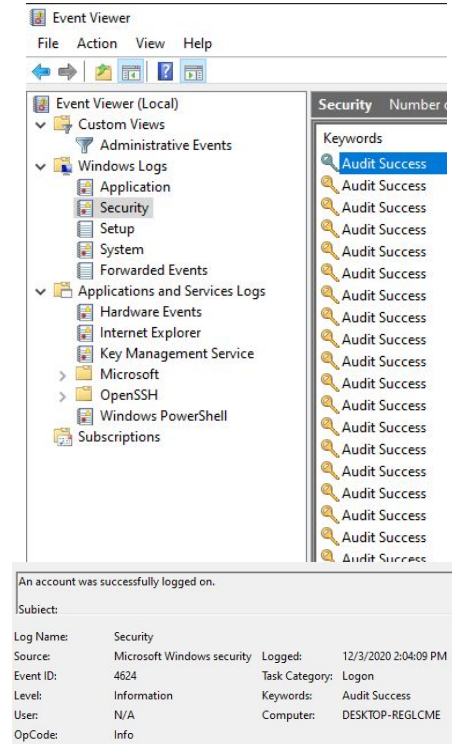
# Logs Windows - Windows Event Viewer

- **Catégories des logs**

- **Application** – logs des applications hébergées sur la machine locale
- **Security** – logs relatifs aux tentatives de connexion, élévation des privilèges et autres événements de sécurité audités
- **Setup** – logs générés lors de l'installation et de la mise à niveau de l'OS
- **System** – logs générés par l'OS
- **Forwarded Events** – logs transmis par d'autres ordinateurs

- **Application and Services Logs**

- Logs par application ou par service



# Gestion des Logs

- Terme générique qui décrit toutes les activités et processus utilisés pour *générer, collecter, centraliser, analyser, transmettre, stocker* et *archiver* des logs générés par des systèmes informatiques
- **Pourquoi la gestion des logs est-elle importante ?**
  - Stockage unifié et centralisé
    - Analyse plus simple
    - Logs standardisés (normalisés) → gain de temps lors de la recherche d'informations
  - Surveillance des systèmes et alertes
    - Alertes personnalisables en temps réel → plus une réactivité en cas de problème
  - Sécurité améliorée
    - En cas de piratage d'une machine → logs ne sont pas perdus
  - Dépannage plus rapide
    - Analyse centralisée des logs → meilleure compréhension des processus et identification de la source du problème plus rapide
  - Parsing des logs et analyse des données

# Gestion des Logs - Étapes

- Collecte des logs
- Agrégation centralisée des logs
- Stockage des logs
  - Durée de rétention des logs
  - Rotation des fichiers de logs
- Analyse des logs

# Gestion des Logs - Collecte des logs

- Processus de capture des logs à partir de fichiers, de flux de sortie standard des applications (stdout), de prises réseau et d'autres sources
- Déterminer comment collecter et envoyer des logs
- Étapes
  - Identifier les sources des logs
  - Choisir une stratégie de collecte
  - Identifier une méthode de collecte
  - Identifier une méthode de transfert des logs

# Gestion des Logs - Collecte des logs - Sources des logs

- **Infrastructure réseau**
  - Commutateurs, routeurs, contrôleurs sans fil et points d'accès
- **Dispositifs de sécurité**
  - Pare-feus
  - IDP/IPS
  - Endpoint Security (EDR, AV, etc.)
  - Outils de gestion des informations et des événements de sécurité (SIEM)
- **Serveurs**
  - Logs système Linux et Windows
- **Serveurs Web**
  - Apache, Nginx, Tomcat, IIS
- **Serveurs d'authentification**
  - Active Directory, LDAP
- **Proxies / passerelles Web**
- **Hyperviseurs**
- **Systèmes de gestion des conteneurs**
  - Kubernetes, Swarm, Mesos
- **Infrastructure SAN**
- **Applications**
- **Postes de travail**



# Gestion des Logs - Collecte des logs - Stratégie de collecte

- **Minimaliste**

- Collecter et envoyer que le nécessaire
- ✓ Moins de bruit dans les données
- ✓ Coûts opérationnels réduits
- ✗ Peut être difficile à identifier et à paramétrer

- **Maximaliste**

- Collecter et envoyer tout
- Toutes les données sont importantes
- ✗ Coûts opérationnels importants
- ✗ Performances réduites
- ✓ Plus facile à paramétrer

# Gestion des Logs - Collecte des logs - Méthode de collecte

- Identifier la méthode de collecte pour chaque source des logs
- Sans agent
  - Source envoie des logs via un protocole et dans un format connu
    - *Syslog*
- Avec un agent
  - Source envoie des logs dans un fichier ou en utilisant un format propriétaire
  - **Agents les plus utilisés** : *Rsyslog, NXLog, Filebeat, Winlogbeat, Promtail, Fluentd*

# Gestion des Logs - Collecte des logs - Méthode de transfert des logs

- Identifier la méthode et le moyen sûr et fiable pour transférer les logs
- **Logs peuvent contenir des données sensibles !**
- Il est préférable de
  - Utiliser un protocole de transport fiable (TCP/IP)
  - Transférer des logs uniquement par des canaux sécurisés (TLS)
  - Utiliser une méthode d'authentification sécurisée (certificats)

# Gestion des Logs - Agrégation centralisée des logs

- Processus d'agrégation de tous les logs en un seul endroit
- **Défis**
  - Volumétrie des données
    - Centaines de gigaoctets de logs par jour pour une grande organisation
  - Vélocité des données
    - Plusieurs dizaines des milliers des messages par seconde
  - Véracité des données
    - Événements peuvent ne pas être précis (horloges désynchronisées, adresses IP incorrectes)
  - Normalisation des données
    - Logs sont produits dans différents formats
- Agrégateurs des logs (*Logstash, Graylog, Loki*)

# Gestion des Logs - Stockage des logs

- **Où et comment stocker les logs ?**
  - On-premise ou dans un cloud externe
  - Dans des fichiers ou dans une base de données telle que
    - *Elasticsearch, Cassandra, DynamoDB, Bigtable*
  - Quel type de stockage utiliser ?
    - HDD ou SDD
    - Objet

# Gestion des Logs - Stockage des logs - Durée de rétention des logs

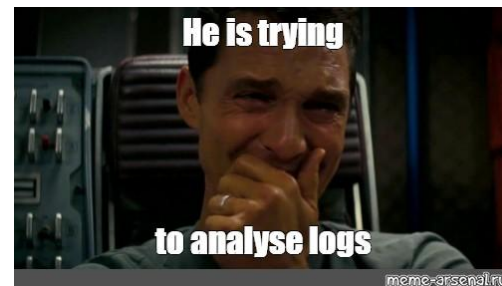
- Combien de temps faut-il stocker les logs ?
  - Stocker les logs pendant une période illimitée = impossible
- Suivre les meilleures pratiques et réglementations de l'industrie
  - Stocker les logs durant au moins 1 an au cas d'une enquête
- Plusieurs durées de rétention possibles

# Gestion des Logs - Stockage des logs - Rotation des logs

- Processus automatique de compression, renommage, déplacement ou suppression des fichiers de logs trop volumineux ou trop anciens
- Permet de
  - Économiser de l'espace
  - Garder le temps d'ouverture des fichiers raisonnable
  - Augmenter les performances d'écriture
- Solutions permettant la rotation des logs
  - *Logrotate*
  - *Graylog* (Rotation de l'index)

# Gestion des Logs - Analyse des logs

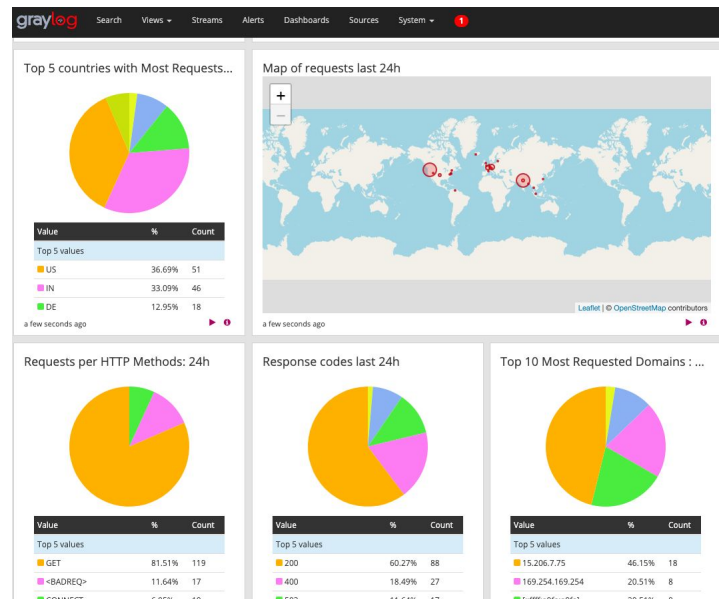
- Utiliser les logs collectés et stockés pour
  - Recherches avancées
  - Détection des problèmes et dépannage des systèmes
  - Réponse aux incidents de sécurité
  - Étude de conformité (politiques de sécurité, audit ou réglementation)
  - Analyse de comportement des utilisateurs
  - Analyse de performance et de sécurité
  - Recherche des corrélations entre les événements et les données
- Peut être automatisée avec des outils
  - Analyse en temps réel (définitions des conditions, des seuils et alerting)
  - Analyse après stockage (traitement avec des algorithmes)





# Gestion des Logs - Analyse des logs - Rapports et tableaux de bord

- Gestion centralisée des logs permet
  - Calcul et visualisation des statistiques
  - Génération des rapports et des tableaux de bords
- Outils
  - *Graylog*
  - *Kibana*
  - *Grafana*



# Récap. Gestion des Logs - Étapes

- Collecte des logs
  - Sources des logs
  - Stratégie de collecte
  - Méthode de collecte
  - Méthode de transfert des logs
- Agrégation centralisée des logs
- Stockage des logs
  - Durée de rétention des logs
  - Rotation des fichiers de logs
- Analyse des logs



# Solutions de gestion des logs

- **ELK Stack** (Elasticsearch, Logstash, Kibana)
- **EFK Stack** (Elasticsearch, Fluentd, Kibana)
- **Graylog**
- **PLG Stack** (Promtail, Loki and Grafana)



# ELK Stack



- Solution de monitoring et de gestion des logs très populaire
- Actuellement Elastic Stack
- Composants
  - **Elasticsearch** – moteur de recherche, de stockage et d'analyse distribué basé sur Apache Lucene
  - **Logstash** – agrégateur qui collecte des données à partir de diverses sources d'entrée, exécute différentes transformations, puis les envoie à Elasticsearch
  - **Kibana** – interface Web qui permet de visualiser et d'analyser les données stockées dans Elasticsearch
  - **Beats** – agents de collecte et de transfert de données
  - **Elastic Agent** – agent qui unifie la collecte des logs, des métriques et des données de sécurité
    - Positionné comme un remplaçant de **Beats**
    - Peut être géré de manière centralisée avec **Fleet** (module dans Kibana)

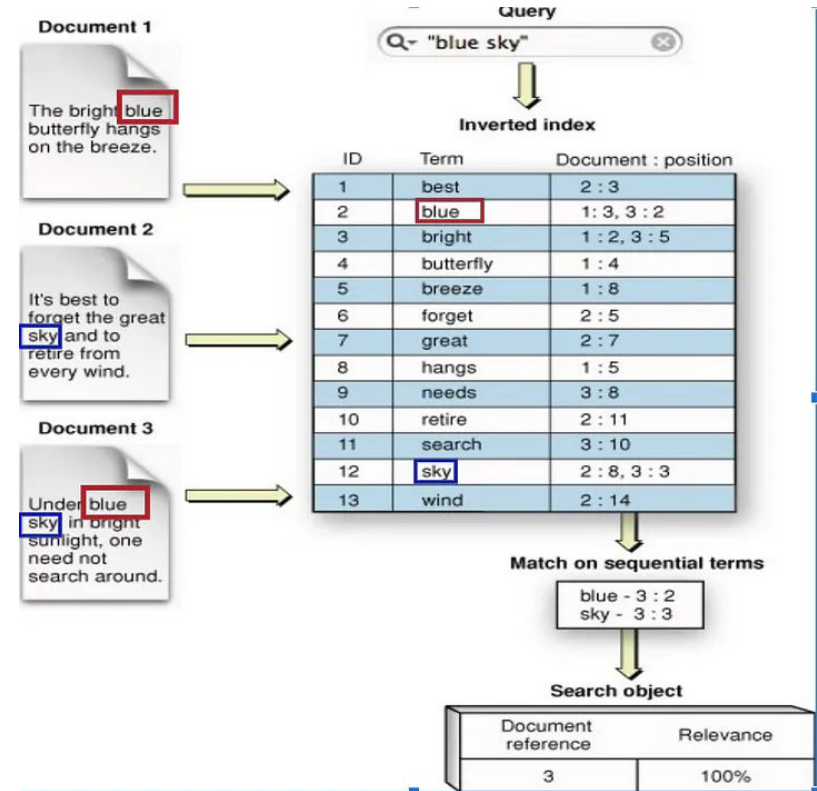
# ELK Stack - Elasticsearch

- Moteur de recherche et d'analyse distribué et open source (?)
- Base de données No SQL
- Stocke les données de manière non structurée en tant que des objets JSON
- Prend en charge des volumes de données très importantes
- Données sont groupées dans des **indexes**
  - Collection de documents qui ont des caractéristiques similaires
  - Peut être considéré comme une "base de données" mais avec beaucoup plus de flexibilité
    - MySQL → Bases de données → Tables → Colonnes/Lignes
    - Elasticsearch → Indexes → Types → Documents avec propriétés



# ELK Stack - Elasticsearch

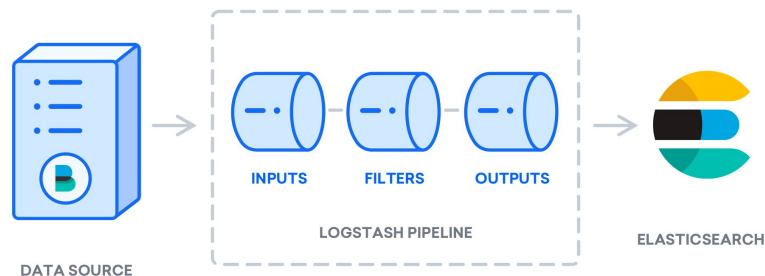
- Indexe toutes les données (*Inverted index*)
- Toutes les données sont consultables en quasi-temps-réel (en ~1 seconde)
- Nécessite plus d'espace de stockage que le format texte



Source : <https://kaushik-jeeyaraman.medium.com/elasticsearch-search-features-capabilities-20ed99e497ec>

# ELK Stack - Logstash

- Agrégateur des données
- Peut être vu comme un pipeline qui
  - Prend des données à une extrémité
  - Les traite d'une manière ou d'une autre
  - Les envoie vers une destination
- Pipeline Logstash est composée de
  - Deux éléments obligatoires
    - **Entrée** (input)
    - **Sortie** (output)
  - Un élément facultatif
    - **Filtre** (filter)



# ELK Stack - Beats

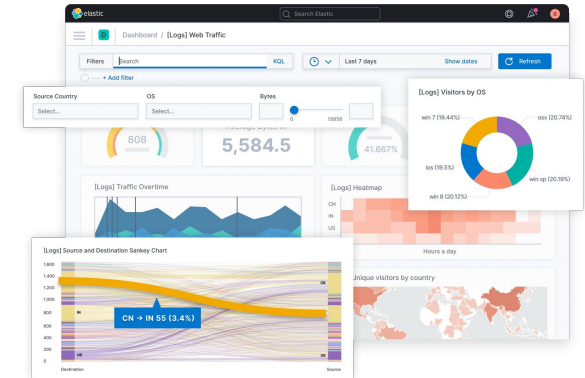
- Agents collecteurs installés sur des hôtes
- Collectent différents types de données
  - **Filebeat** – fichiers de logs
    - Prend en charge des modules qui simplifient la collecte de logs
  - **Winlogbeat** – Windows event logs
  - **Metricbeat** – métriques du système (CPU, RAM, I/O) et des services (Apache, Nginx...)
  - **Packetbeat** – analyseur de paquets réseau
  - **Auditbeat** et **Heartbeat**
- Les transfèrent vers
  - **Logstash**
  - Elasticsearch



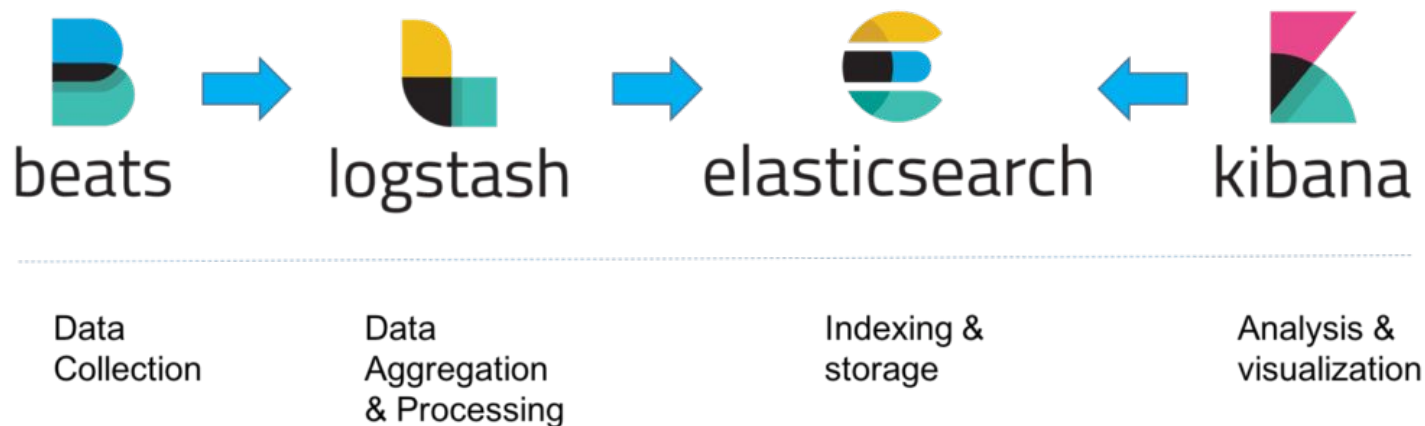


# ELK Stack - Kibana

- Outil de visualisation et d'exploration de données utilisé pour
  - Analyse des logs et des séries chronologiques
  - Surveillance des applications
  - Intelligence opérationnelle
- Permet de visualiser les données stockées dans Elasticsearch
- Offre des fonctionnalités puissantes et faciles à utiliser telles que
  - Recherche rapide dans de grandes quantités de données
  - Création des histogrammes, graphiques linéaires, diagrammes circulaires...
  - Création des dashboards
  - Configuration et gestion des alertes
  - Machine learning pour analyser les données
  - Gestion de la stack Elastic



# ELK Stack - Architecture



**Source :** <https://logz.io/learn/complete-guide-elk-stack/>

# ELK Stack - Études de cas



- **Netflix**
  - Utilise ELK stack pour surveiller et analyser les logs de sécurité des opérations du service client. Il leur permet d'indexer, de stocker et de rechercher les logs de plus d'une quinzaine de clusters comprenant près de 800 nœuds.
- **LinkedIn**
  - Utilise ELK stack pour surveiller les performances et la sécurité. Leur infrastructure ELK comprend plus de 100 clusters dans six datacentres différents.
- **Medium**
  - Utilise ELK stack pour déboguer les problèmes de production. Grâce à ELK, la société peut prendre en charge 25 millions de lecteurs uniques ainsi que des milliers de publications par semaine.

**Source :** <https://www.guru99.com/elk-stack-tutorial.html>

# EFK Stack



- Adaptée pour les microservices hébergés sur Docker / Kubernetes
- Composants
  - **Elasticsearch**
  - **Fluentd**
    - Collecteur de données qui unifie la collecte et la consommation des données
    - Structure les données en JSON autant que possible
    - Possède un système de plugins flexible avec plus de 500 plugins fournis par la communauté
    - Utilise peu de ressources (utilise 30 à 40 Mo de mémoire et peut traiter 13 000 événements/seconde/cœur)
  - **Kibana**

# EFK Stack - Architecture



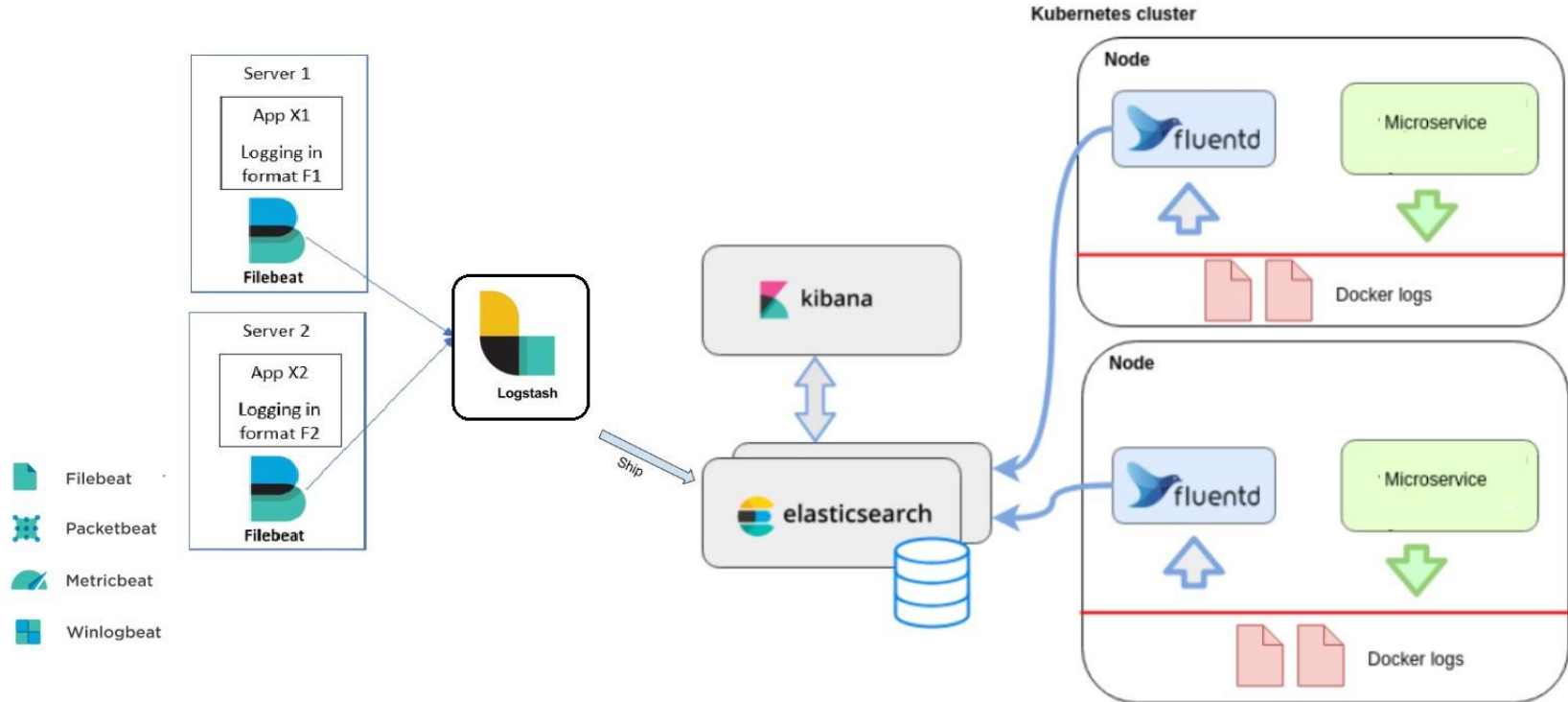
**Source :** <https://www.cncf.io/blog/2020/07/27/logging-in-kubernetes-efk-vs-plg-stack/>

# Fluentd vs Logstash



- **Fluentd** est bien adapté pour les microservices hébergés sur Docker / Kubernetes
- **Fluentd** possède plus de plugins que **Logstash**
- **Fluentd** ne nécessite pas de runtime Java
- **Fluentd** intègre par défaut les parsers **JSON**, **CSV** et **RegEx**
- **Logstash** doit être déployé avec **Redis** pour garantir la fiabilité entre les redémarrages
- **Logstash** nécessite un outil supplémentaire pour obtenir des données
- **Logstash** consomme plus de ressources que **Fluentd**
- **Logstash** prend en charge les métriques système / conteneur

# Architecture hybride ELK-EFK



Source : <https://medium.com/techmanyu/logstash-vs-fluentd-which-one-is-better-adaaba45021b>

# Fluent Bit et Fluentd

- Construit sur la base de l'architecture et des principes de conception de **Fluentd**
- Très léger et a de très bonnes performances
- Développé en **C**
- Moins d'empreinte mémoire que **Fluentd** (~1MB)
- Moins des plugins disponibles qu'avec **Fluentd** (~100)
- Tend à remplacer **Fluentd** et est considéré comme une solution next gen





# Graylog

- Outil **très puissant** et spécialement conçu pour la gestion des logs
- Composants
  - **Stockage des logs (Data node)**
    - **Elasticsearch** ou *OpenSearch*
  - **MongoDB** – base No SQL utilisée pour stocker la configuration et des paramètres
  - **Graylog-server** – serveur de collecte et de visualisation des logs



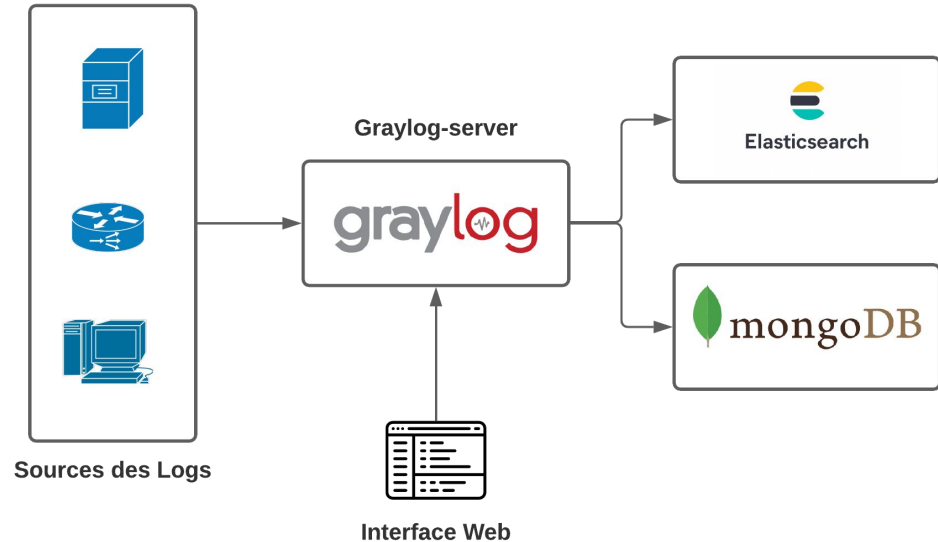
# Graylog - Planification du déploiement

- **Nœuds Graylog-server**
  - CPU le plus puissant possible
- **Nœuds Elasticsearch/OpenSearch doivent disposer**
  - Autant de RAM que possible
  - Disques les plus rapides possibles
  - Vitesse d'E/S est très importante
- **MongoDB**
  - Pas besoin de beaucoup de ressources



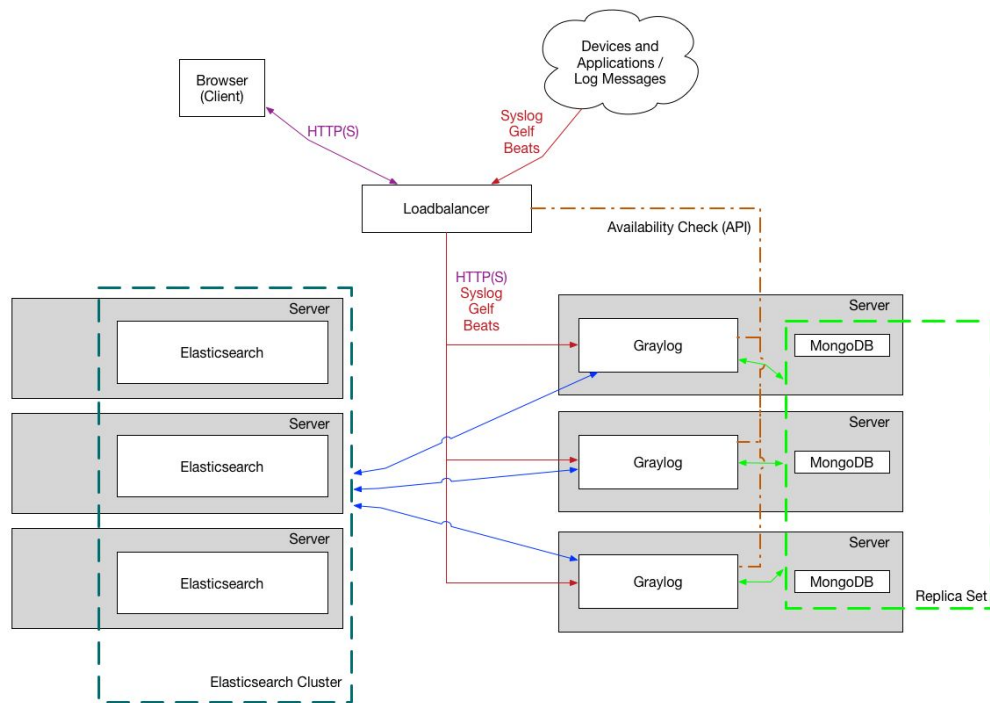
# Graylog - Architecture - Déploiement simple

- Petite infrastructure non critique
- Déploiement de test
- Aucun des composants n'est redondant
- Installation et configuration très simples



# Graylog - Architecture - Déploiement en production

- Déploiement pour des environnements de production plus importants
- Plusieurs nœuds Graylog derrière un load balancer
- Cluster Elasticsearch
- Replica Set de MongoDB



Source : <https://docs.graylog.org/docs/architecture>

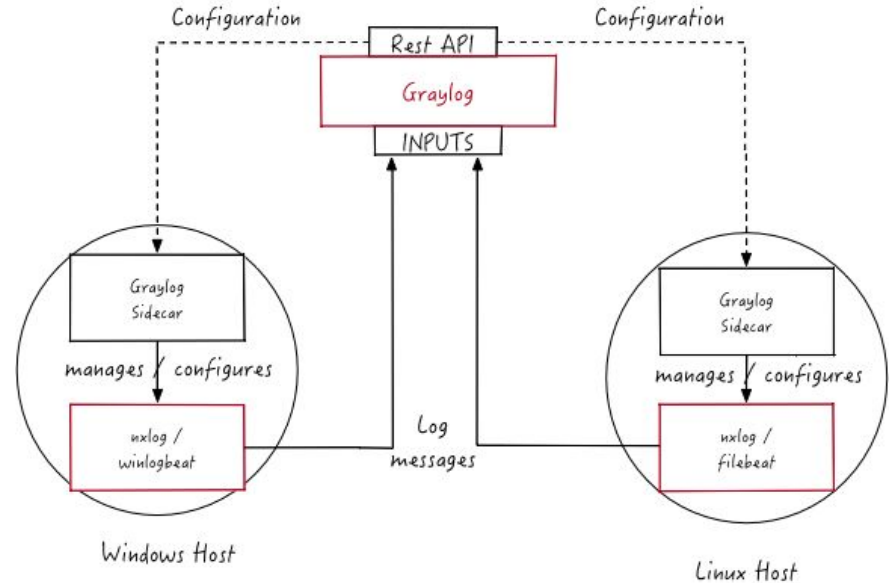
# Graylog - Inputs

- Graylog reçoit des logs via les Inputs
- **Types des inputs**
  - Listener inputs (Ouvre un port TCP ou UDP)
    - *Beats, Syslog, GELF, CEF, Netflow, RAW*
  - Pull inputs (Interroge un endpoint)
    - *AWS, GCP, Gmail, Office 365, JSON*
- **Sécurisation des échanges**
  - *TLS et authentification via des certificats*
- **Collecteurs de logs**
  - *NXLog*
  - *Elastic Beats (Filebeat, Winlogbeat)*
  - *Autres (Sysmon, auditd...)*



# Graylog - Sidecar

- Système de gestion de configuration des collecteurs de logs (NxLog, Filebeat, Winlogbeat...)
- Facile à installer et à configurer
- Gère la configuration des collecteurs de manière centralisée via l'interface Web Graylog

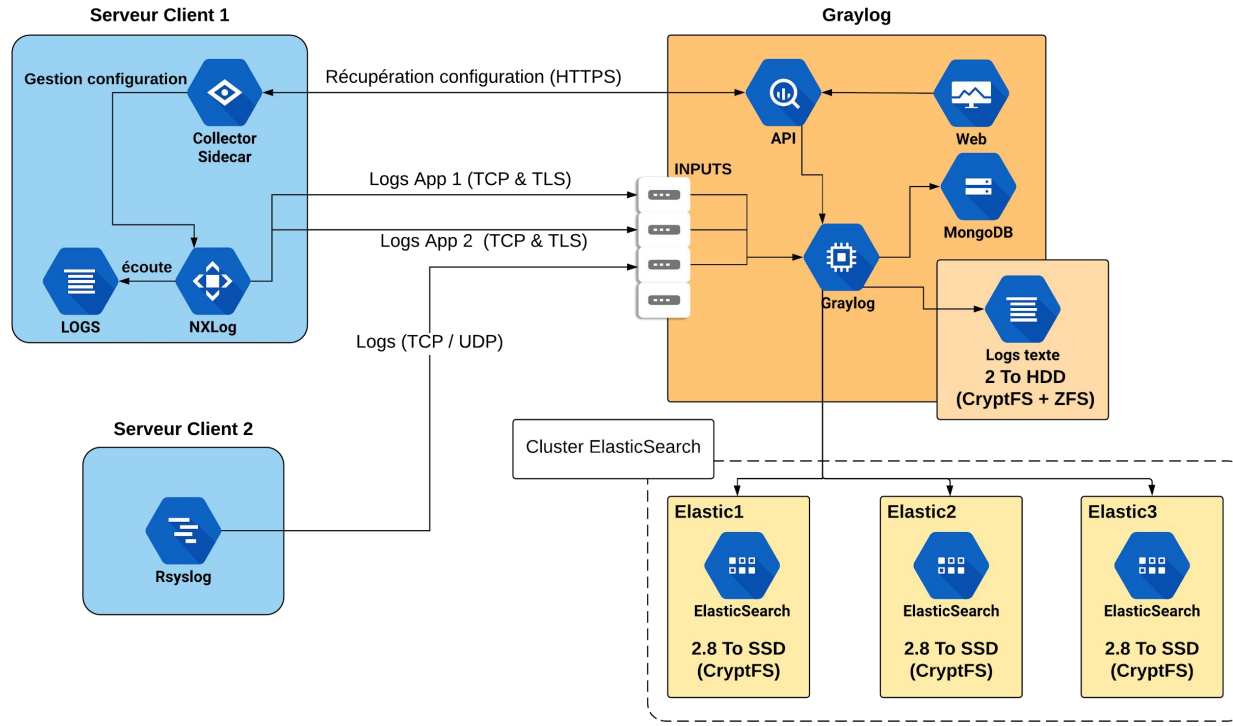


# Graylog - Avantages



- Outil **gratuit** et **open-source**
- Installation rapide et facile
- Peut recevoir des logs directement depuis une application ou un appareil
- Propose deux types d'Inputs (listener et pull)
- Gestion des utilisateurs, intégration avec LDAP et autres mécanismes d'authentification
- Gestion des alertes intégrée et gratuite
- Multitude des plugins et content pack disponible dans **Graylog Marketplace**
- Archivage et rotation des logs
- Gestion des sources des logs centralisée avec **Graylog Sidecar**

# Graylog - Exemple de déploiement





# PLG Stack

- Connu sous le nom de **Grafana Loki**
- Solution d'agrégation de logs simple, légère et facile à utiliser
- Indexe uniquement les métadonnées et n'indexe pas le contenu des logs
  - Nécessite moins de ressources que les autres solutions
  - Requêtes sur le contenu des logs sont moins performantes
- Utilise un langage de requête appelé **LogQL** pour interroger les logs («grep» distribué)
- Dispose d'une intégration native avec *Kubernetes*
- Peut utiliser **le stockage objet** (*Amazon S3* ou *GCS*)



# PLG Stack - Composants

- **Promtail**

- Agent installé sur tous les nœuds sources de logs
- Collecte les logs et attache des étiquettes aux logs
- Envoie les logs du système local vers le cluster Loki

- **Loki**

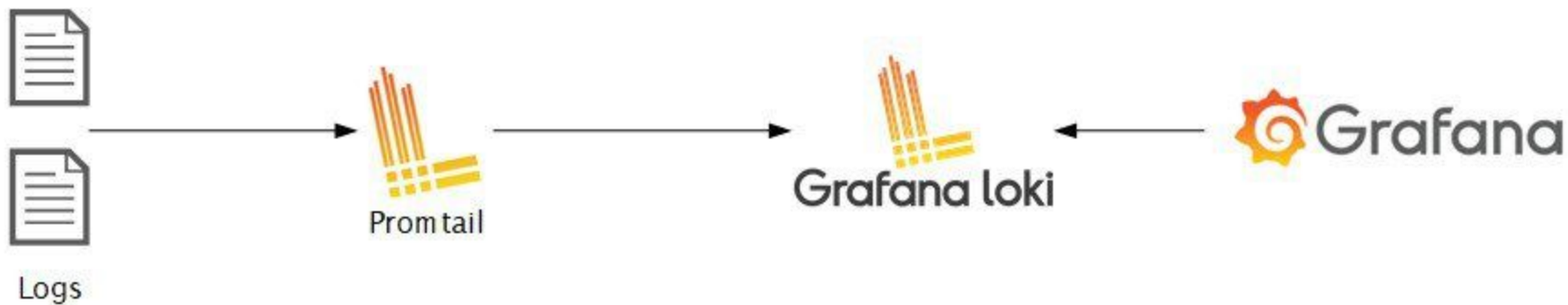
- Cœur de la stack PLG
- Agrège et stocke les logs
- Évolutif horizontalement, hautement disponible et inspiré de Prometheus

- **Grafana**

- Outil de visualisation
- Affiche des données stockées par Loki



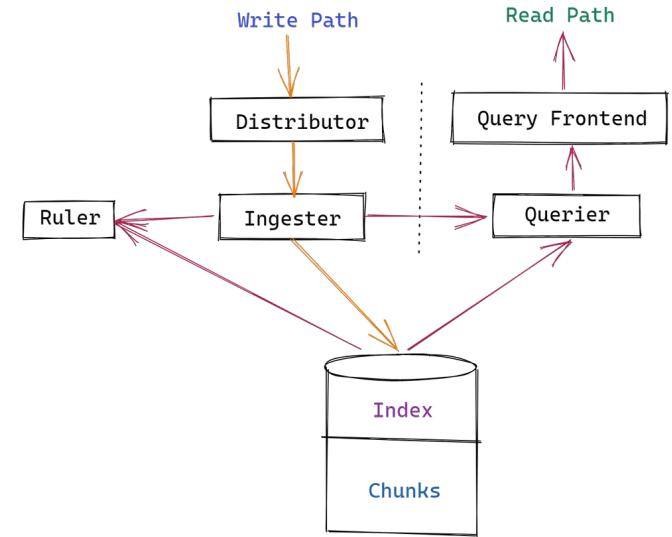
# PLG Stack - Architecture



**Source :** <https://www.cncf.io/blog/2020/07/27/logging-in-kubernetes-efk-vs-plg-stack/>

# PLG Stack - Loki - Fonctionnement

- Loki découple les requêtes en chemins de lecture et d'écriture séparés
  - Afin que vous puissiez les mettre à l'échelle indépendamment
  - **Chemin de lecture** – lecture des données (traitement des demandes)
  - **Chemin d'écriture** – écrire des données sur le stockage



Source : <https://hackernoon.com/grafana-loki-architecture-summary-and-running-in-kubernetes>

# PLG Stack - Loki - Composants - Chemin de lecture

- **Query frontend (Optionnel)**

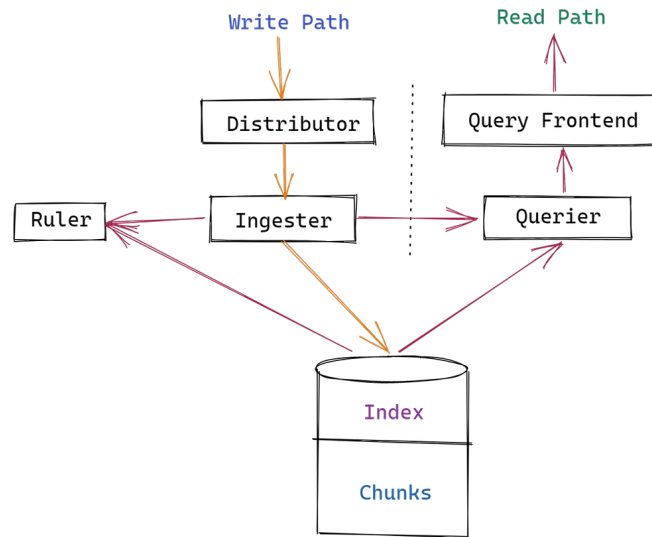
- Utilisé pour accélérer la lecture
- Garantit que les requêtes volumineuses seront exécutées
- Distribue la charge entre les instances de **Querier**
- A une file d'attente interne pour les requêtes

- **Querier**

- Traite les requêtes de lecture **LogQL**
- Récupère les logs à la fois des **Ingesters** et du stockage

- **Ruler**

- Évalue en permanence un ensemble de requêtes
- Effectue une action en fonction du résultat (Alerting)



Source : <https://hackernoon.com/grafana-loki-architecture-summary-and-running-in-kubernetes>

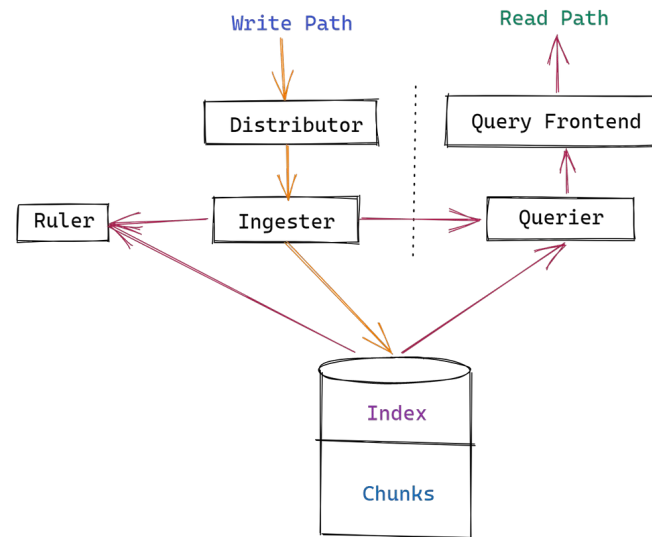
# PLG Stack - Loki - Composants - Chemin d'écriture

- **Distributeur**

- Traite les flux entrants
- Valide les données d'entrée (validité des labels, timestamps, longueur des logs)
- Normalise les labels
- Envoie des données à **Ingester**

- **Ingester**

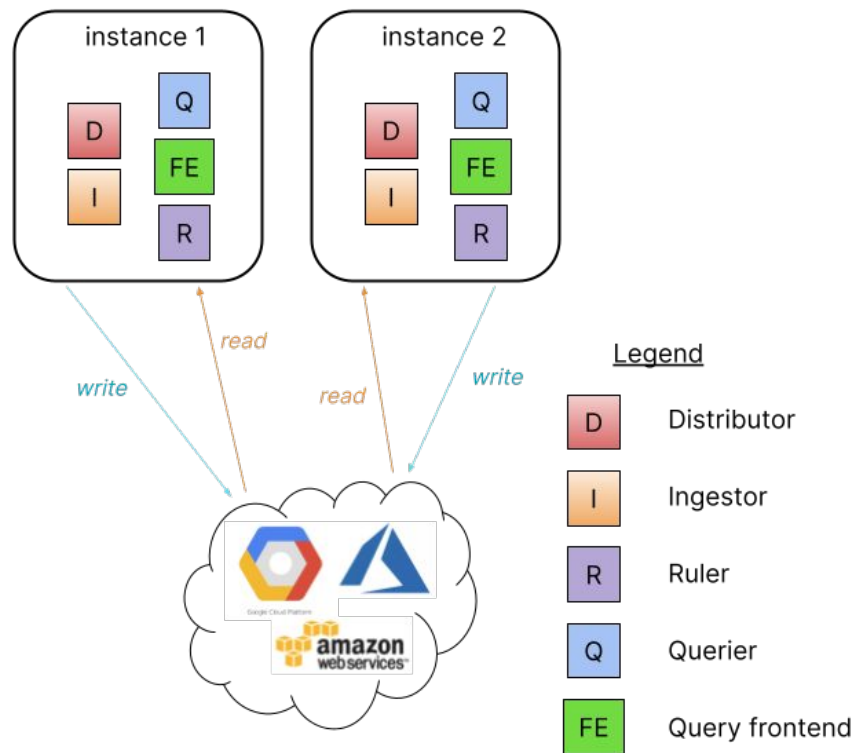
- Écrit les logs sur les backends de stockage
- Renvoie les logs s'ils sont dans la mémoire lors des demandes de lecture



Source : <https://hackernoon.com/grafana-loki-architecture-summary-and-running-in-kubernetes>

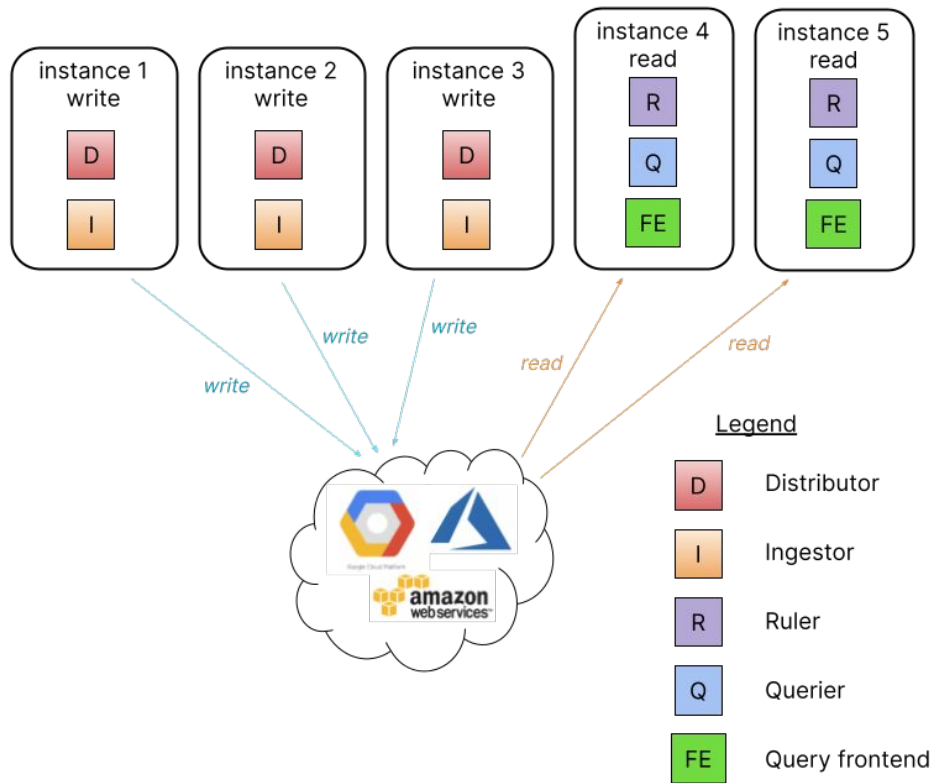
# PLG Stack - Loki - Déploiement - Mode Monolithique

- Tous les composants sont lancés à l'intérieur d'un seul processus
- Expérimenter avec Loki
- Petits volumes de lecture/écriture
  - Environ 20 Go par jour



# PLG Stack - Loki - Déploiement - Mode de déploiement simple et évolutif

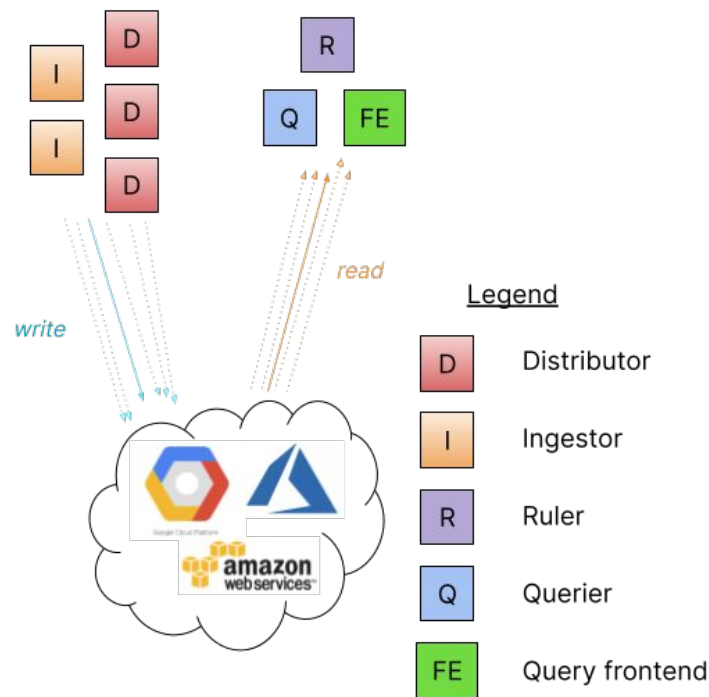
- Moyen préféré de déployer Loki pour la plupart des installations
- Séparer la lecture et l'écriture des logs
  - Plus grande disponibilité pour le chemin d'écriture
  - Chemin de lecture séparé et évolutif
- Composants peuvent être mises à l'échelle indépendamment
- Peut évoluer jusqu'à plusieurs To de logs par jour





# PLG Stack - Loki - Déploiement - Mode microservices

- Recommandé pour les très gros clusters Loki
- Chaque composant est lancé dans un microservice individuel
- Plus de contrôle sur la mise à l'échelle
- Complexe à mettre en place et à maintenir
- Conçu pour les déploiements Kubernetes



# PLG vs ELK et Graylog



- **PLG** est moins coûteux à opérer, car n'indexe pas le contenu des logs
- **PLG** est facile à configurer
- **PLG** est très évolutif
- **PLG** est plus adapté aux environnements cloud-native
- **ELK et Graylog** ont un moteur de recherche beaucoup plus puissant et mature
- **ELK et Graylog** permettent des recherches très rapides sur le contenu des logs
- **ELK et Graylog** offrent plus de fonctionnalités pour analyser les logs
- **ELK et Graylog** proposent une gestion centralisée des collecteurs de logs via une interface web

# Demo Time

- Elastic Stack
- Graylog
- Grafana Loki

Merci pour votre attention!

