Table des matières

[1. Installation de Loki sur la machine Rocky 1](#_Toc93590231)

[2. Installation de l’agent promtail sur la machine Debian 3](#_Toc93590232)

[3. Installation de l’agent promtail sur les machines Windows 6](#_Toc93590233)

[4. Mise en place de tableaux de bord dans Grafana 8](#_Toc93590234)

**Pour ce TP nous avons formé le binôme LEGER Lucas + LAFORGE Samuel. Nous avons fait un compte rendu pour deux et nous nous sommes réparti les tâches**

**La partie installation de Grafana n’est pas expliquée car nous l’avons fait dans un TP précédent**

**Avant de commencer le TP nous devons mettre la bonne date et heure sur nos machines afin d’éviter des problèmes lors de l’envoi des données vers le serveur. Pour la machine Debian nous avons fait cela directement dans les paramètres et pour la machine rocky nous avons suivi le site suivant après avoir passé la commande "timedatectl set-timezone Europe/Paris" :** [**https://wiki.crowncloud.net/?How\_to\_Sync\_Time\_in\_Rocky\_Linux\_8\_using\_Chrony**](https://wiki.crowncloud.net/?How_to_Sync_Time_in_Rocky_Linux_8_using_Chrony)

# 1. Installation de Loki sur la machine Rocky

Avant de pouvoir monitorer nos logs dans Grafana nous devons installer un serveur qui va permettre de stocker les données récupérées par les agents sur les machines et qui sera lu et afficher par Grafana. Nous allons donc installer un serveur nommé "Loki".

Pour procéder à l’installation du serveur nous devons commencer par créer un répertoire pour Loki et se déplacer dedans :

mkdir /opt/loki/ && cd /opt/loki/

Maintenant que nous avons un répertoire pour Loki nous pouvons télécharger le fichier zip du serveur puis le décompresser et donner les droits :

**/ ! \ Attention : Vérifier avant que la version téléchargée soit bien la dernière sortie sur le site suivant :** [**https://github.com/grafana/loki/releases/**](https://github.com/grafana/loki/releases/)

wget https://github.com/grafana/loki/releases/download/v2.4.2/loki-linux-amd64.zip

unzip loki-linux-amd64.zip

chmod a+x "loki-linux-amd64"

Nous devons aussi télécharger le fichier de configuration du serveur Loki :

wget https://raw.githubusercontent.com/grafana/loki/v2.4.2/cmd/loki/loki-local-config.yaml

Nous n’avons rien à modifier dans ce fichier car le serveur se trouve bien en localhost et il écoute sur les ports par défauts

Maintenant nous devons supprimer l’utilisateur actuel Loki et en recréer un nouveau et lui donner les droits sur le répertoire créé précédemment :

userdel loki

useradd -s /bin/bash -d /opt/loki/ loki

chown -R loki: /opt/loki/

Nous devons maintenant créer un fichier de service pour le serveur afin qu’il puisse se démarrer automatiquement à chaque démarrage :

nano /etc/systemd/system/loki.service

Dans ce fichier nous devons mettre les lignes suivantes :

[Unit]

Description=Loki Grafana

Wants=network-online.target

After=network-online.target

[Service]

Type=simple

User=loki

Group=loki

ExecStart=/opt/loki/loki-linux-amd64 -config.file=/opt/loki/loki-local-config.yaml

SyslogIdentifier=loki

Restart=always

[Install]

WantedBy=multi-user.target

Une fois tout ceci fait nous pouvons activer et démarrer le service puis vérifier son bon fonctionnement :

systemctl enable loki.service

systemctl start loki.service

systemctl status loki.service

journalctl -f -u loki.service

Nous pouvons vérifier que le serveur écoute bien sur le bon port (3100 et 9096) et sur toutes les IP :

[root@localhost loki]# netstat -tpln

Connexions Internet actives (seulement serveurs)

Proto Recv-Q Send-Q Adresse locale Adresse distante Etat PID/Program name

tcp6 0 0 :::3100 :::\* LISTEN 11558/loki-linux-am

tcp6 0 0 :::9096 :::\* LISTEN 11558/loki-linux-am

Nous devons aussi ouvrir le port 3100 en TCP pour accepter de recevoir les données des agents :

sudo firewall-cmd --zone=public --add-port=3100/tcp

Notre serveur Loki est maintenant prêt à l’utilisation

# 2. Installation de l’agent promtail sur la machine Debian

Pour la configuration du fichier yaml nous nous sommes aidés du site suivant : <https://grafana.com/docs/loki/latest/clients/promtail/configuration/>

Nous avons maintenant un serveur Loki fonctionnel sur notre machine Rocky Linux donc nous pouvons passer à l’installation de l’agent sur la machine Debian. Toute la procédure suivante se fera donc sur notre VM Linux Debian.

Nous devons donc commencer par créer un répertoire pour l’agent et se déplacer dedans :

mkdir /opt/promtail && cd /opt/promtail

Ensuite nous devons télécharger le zip de promtail et le décompresser puis lui donner les droits :

**/ ! \ Attention : Vérifier avant que la version téléchargée soit bien la dernière sortie sur le site suivant :** [**https://github.com/grafana/loki/releases/**](https://github.com/grafana/loki/releases/)

wget https://github.com/grafana/loki/releases/download/v2.4.2/promtail-linux-amd64.zip

unzip promtail-linux-amd64.zip

chmod a+x promtail-linux-amd64

Il faut ensuite télécharger le fichier de configuration de l’agent et le modifier pour qu’il soit lié à notre serveur :

wget https://raw.githubusercontent.com/grafana/loki/v2.2.1/cmd/promtail/promtail-local-config.yaml

Modification du fichier :

nano /opt/promtail/promtail-local-config.yaml

Contenu du fichier :

server:

http\_listen\_port: 9080

grpc\_listen\_port: 0

positions:

filename: /tmp/positions.yaml

clients:

- url: http://10.44.19.200:3100/loki/api/v1/push

scrape\_configs:

- job\_name: apache

static\_configs:

- targets:

- localhost

labels:

host: debian

job: apache2

\_\_path\_\_: /var/log/apache2/\*log

- job\_name: system

static\_configs:

- targets:

- localhost

labels:

host: debian

job: varlogs

\_\_path\_\_: /var/log/\*log

Le fichier est configuré de façon à récupérer tous les logs du répertoire **/var/log/** et aussi les logs d’apache dans **/var/log/apache2/** et à les envoyer vers le serveur Loki

Il faut maintenant créer un utilisateur promtail puis lui donner les droits sur le dossier et l’ajouter au groupe adm :

useradd -s /bin/bash -d /opt/promtail/ promtail

chown -R promtail: /opt/promtail/

usermod -a -G adm promtail

Nous devons ensuite créer un fichier pour ajouter promtail en tant que service :

nano /etc/systemd/system/promtail.service

Dans ce fichier nous mettons les lignes suivantes :

[Unit]

Description=Promtail Loki

Wants=network-online.target

After=network-online.target

[Service]

Type=simple

User=promtail

Group=promtail

ExecStart=/opt/promtail/promtail-linux-amd64 -config.file /opt/promtail/promtail-local-config.yaml

SyslogIdentifier=promtail

Restart=always

[Install]

WantedBy=multi-user.target

Nous pouvons maintenant activer et démarrer le service promtail puis vérifier son bon fonctionnement :

systemctl enable promtail.service

systemctl start promtail.service

systemctl status promtail.service

journalctl -f -u promtail.service

Si l’agent est bien configuré nous devrions récupérer les logs du fichier **/var/log/** :

root@debian10:/opt/promtail# cat /tmp/positions.yaml

positions:

/var/log/alternatives.log: "48798"

/var/log/auth.log: "1059428"

/var/log/daemon.log: "6347084"

/var/log/dpkg.log: "978481"

/var/log/faillog: "0"

/var/log/fontconfig.log: "3873"

/var/log/kern.log: "1930865"

/var/log/lastlog: "0"

/var/log/syslog: "10220118"

/var/log/user.log: "1887357"

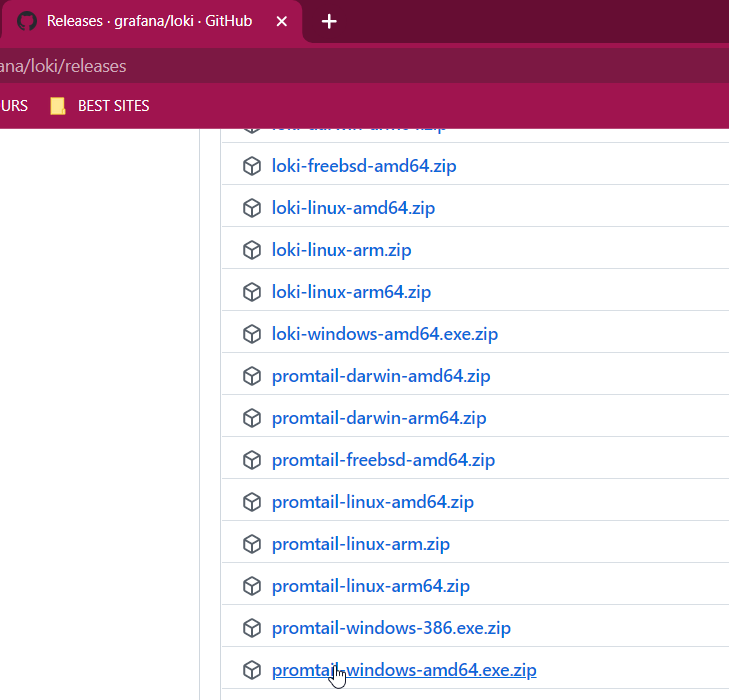
/var/log/vboxadd-install.log: "627"

/var/log/vboxadd-setup.log: "156"

Nous récupérons bien un certain nombre de lignes en fonction du type de log que nous voulons récupérer

# 3. Installation de l’agent promtail sur les machines Windows

Tout d’abord, il faut télécharger le fichier zip sur le Github de Grafana :



Après cela, on dézippe le fichier puis on créée le fichier de configuration que l’on nommera promtail-local-config.yaml dans le même dossier. Voici à quoi il ressemble pour notre utilisation :

Une image contenant texte

Description générée automatiquement

Une fois cela fait, on peut faire un test manuel en lançant cette commande :

C:\Users\lucas\Downloads>promtail-windows-amd64.exe --config.file= promtail-local-config.yaml

On peut désormais voir nos fichiers de log Windows remonter sur notre interface Grafana :

Une image contenant texte

Description générée automatiquement

Nous allons désormais créer un service lié afin de ne pas devoir lancer manuellement le fonctionnement de la remontée de logs.

Nous allons tout d’abord installer **nssm** en suivant ce lien :

<https://nssm.cc/download>

Ensuite, on décompresse le fichier téléchargé et on place l’exécutable dans le même dossier que notre promtail.

Une image contenant texte

Description générée automatiquement

On commence tout d’abord par créer le service :

C:\Users\lucas\Downloads>nssm.exe install promtail\_agent "C:\Users\lucas\Downloads\promtail-windows-amd64.exe"

Après ça on le renomme proprement :

C:\Users\lucas\Downloads>nssm.exe set promtail\_agent DisplayName "Promtail Agent" "C:\Users\lucas\Downloads\promtail-windows-amd64.exe"

Puis, on définit le fichier de log :

C:\Users\lucas\Downloads>nssm.exe set promtail\_agent set promtail\_agent AppStdout "C:\Users\lucas\Downloads\promtail.log"

C:\Users\lucas\Downloads>nssm.exe set promtail\_agent set promtail\_agent AppStderr "C:\Users\lucas\Downloads\promtail.log"

Une fois cela fait, on va mettre le paramètre nécessaire au fonctionnement dans le menu des services :

Une image contenant texte

Description générée automatiquement

Ensuite, il ne nous reste plus qu’à démarrer le service.

# 4. Mise en place de tableaux de bord dans Grafana

Maintenant que tous les agents sont configurés nous pouvons nous rendre dans Grafana.

On va créer une ressource Loki afin de pouvoir faire remonter graphiquement tous les logs ce qu’il récupère.

Une image contenant texte, moniteur, capture d’écran, écran

Description générée automatiquement

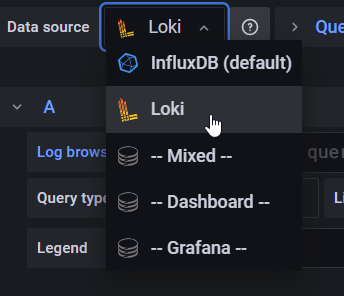
**Pour la machine Windows :**

On crée un nouveau Dashboard dans lequel on ajoute un panel :

Une image contenant texte, moniteur, capture d’écran, écran

Description générée automatiquement

On choisit notre data source Loki :



Puis, en cliquant sur le bouton « Log browser », on trouve tous les fichiers de log que Loki fait remonter :

Une image contenant texte

Description générée automatiquement

Attention ! Il existe une particularité pour Windows. Une fois que l’on a choisi le fichier que nous souhaitons interpréter, il faut remplacer tous les « \ » par « \\ » comme sur l’image suivante :



Une fois cela fait, nous obtenons nos log :

Une image contenant texte

Description générée automatiquement

Les logs restent néanmoins illisibles du fait que le format des fichiers evtx de Windows n’est pas lisible dans un autre logiciel que l’observateur d’évènements.

Pour avoir les logs du service IIS, il faut compléter le fichier de configuration YAML.

Une image contenant texte

Description générée automatiquement

Il ressemble désormais à cela.

Ensuite, le fichier de log du service remonte sur le Grafana et on peut interpréter les logs.

Ceux-ci sont lisibles contrairement aux précédents car c’est un .log interprétable avec du texte.

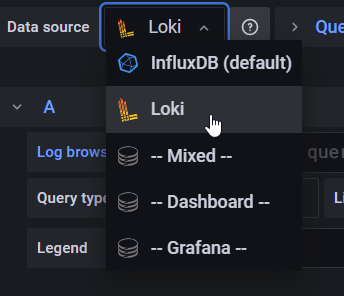
Une image contenant texte, capture d’écran, moniteur, intérieur

Description générée automatiquement

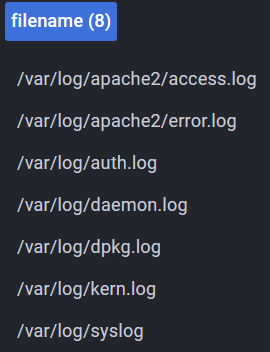
**Pour la machine Linux :**

Pour la machine Linux la création du Dashboard reste la même.

Nous ajoutons un Pannel que nous lions à notre datasource Loki :



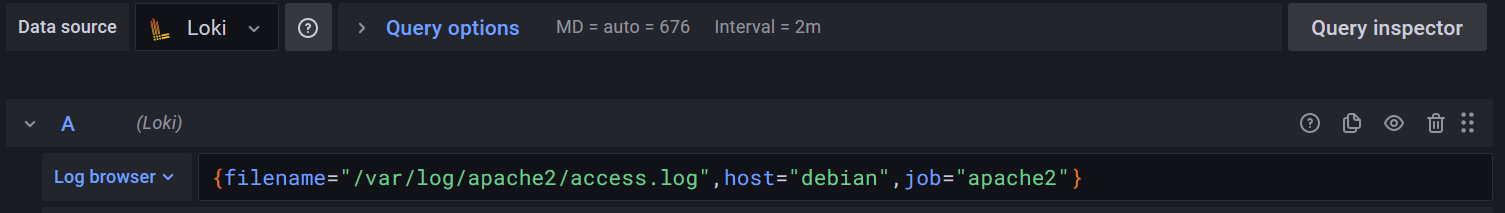
De même en cliquant sur le bouton log browser nous retrouvons a liste de tous les fichiers de logs qui sont remontés par le serveur Loki :



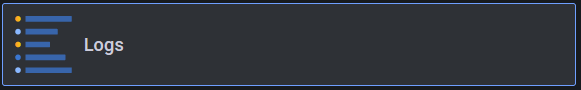
Nous créons un pannel par log donc pour avoir tous les logs nous devons créer 8 pannels.

Voici l’exemple de configuration pour 1 pannel :

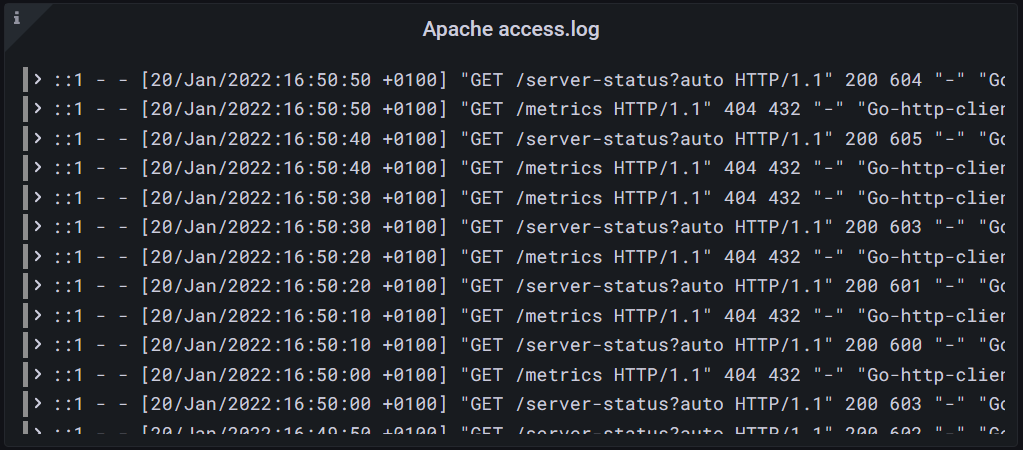
Donner la commande de récupération des logs :



Définir une visualisation sur logs :

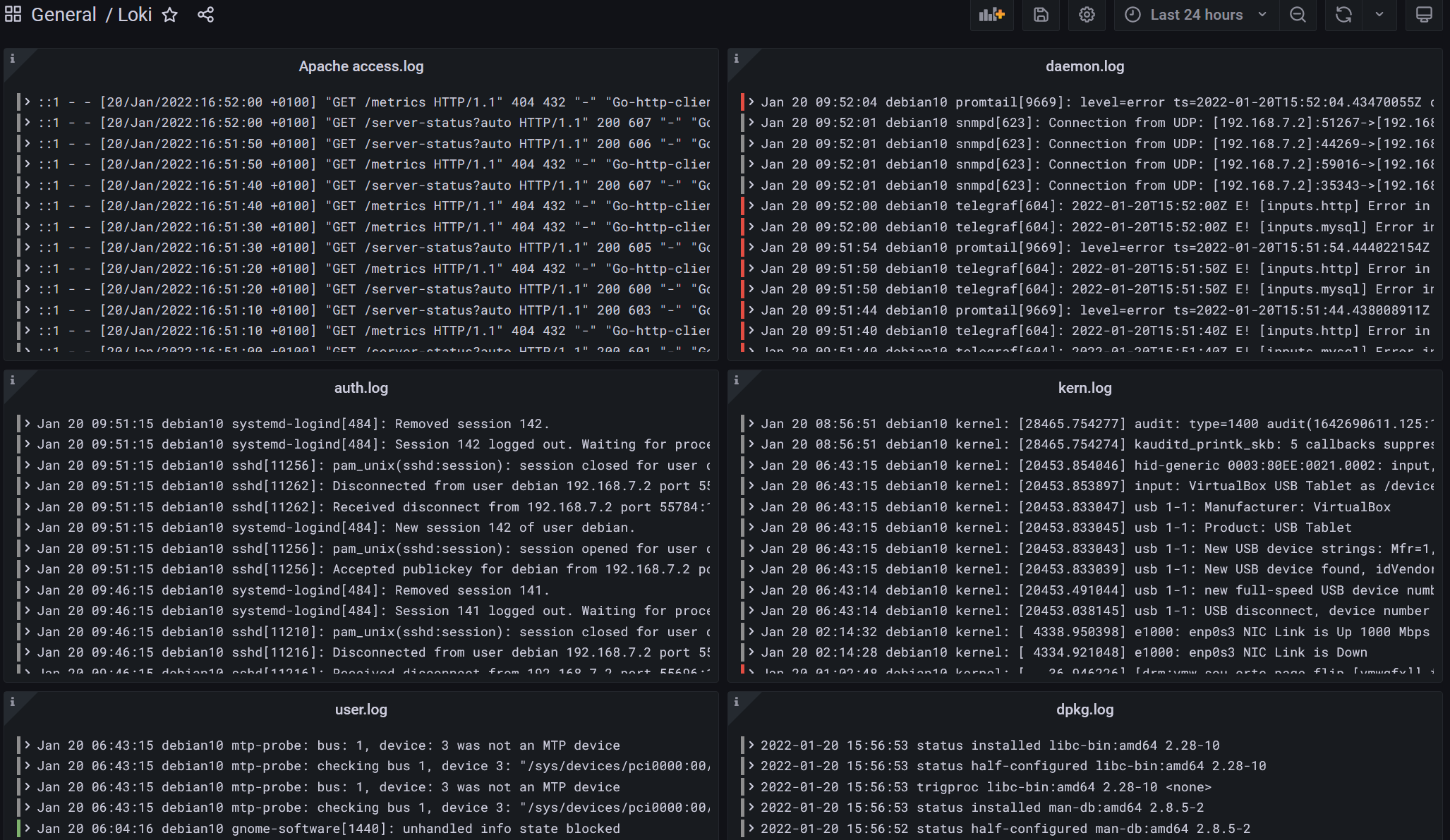


Appliquer la configuration et voici le pannel final :



Nous voyons bien pour le service web Apache que nous récupérons les logs à la date du 20 janvier

Voici donc le Dashboard final :



Nous récupérons bien tous les principaux logs de notre système Linux