|  |  |
| --- | --- |
| Best 20 NuGet syslog Packages - NuGet Must Haves Package | **Fichiers journaux syslog**  **Centraliser les journaux** |

# Présentation générale

En tant que protocole, ***syslog* se compose d'une partie cliente et d'une partie serveur**.  
La partie cliente émet les informations sur le réseau, à l’aide du **port UDP 514**.   
Les serveurs collectent l'information et se chargent de créer les journaux.

L'intérêt de syslog est donc de **centraliser les journaux d'événements**, permettant de **repérer plus rapidement et efficacement les défaillances d'hôtes présents sur un réseau**.

syslog est donc un démon (*daemon*) dédié à l'enregistrement des journaux (*log*).   
Les journaux log sont stockés dans le **répertoire** **/var/log/**.

|  |
| --- |
| Jul 6 18:59:56 core kernel: [ 3.630480] EXT4-fs (dm-0): re-mounted. Opts: errors=remount-ro  Jul 6 18:59:56 core kernel: [ 3.918179] Adding 1044476k swap on /dev/mapper/turnkey-swap\_1. Priority:-1 extents:1 a$  Jul 6 18:59:56 core kernel: [ 4.494498] ADDRCONF(NETDEV\_UP): eth0: link is not ready  Jul 6 18:59:56 core kernel: [ 4.496426] e1000: eth0 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX  Jul 6 18:59:56 core kernel: [ 4.496757] ADDRCONF(NETDEV\_CHANGE): eth0: link becomes ready  Jul 6 18:59:56 core acpid: starting up with netlink and the input layer  Jul 6 18:59:56 core acpid: 1 rule loaded  Jul 6 18:59:56 core acpid: waiting for events: event logging is off  Jul 6 18:59:56 core cron[2016]: (CRON) INFO (pidfile fd = 3)  Jul 6 18:59:56 core cron[2017]: (CRON) STARTUP (fork ok)  Jul 6 18:59:56 core cron[2017]: (CRON) INFO (Running @reboot jobs)  Jul 6 18:59:57 core ntpd[2048]: ntpd 4.2.6p5@1.2349-o Fri Apr 10 18:48:35 UTC 2015 (1)  Jul 6 18:59:57 core ntpd[2049]: proto: precision = 0.885 usec  Jul 6 18:59:57 core ntpd[2049]: unable to bind to wildcard address 0.0.0.0 - another process may be running - EXITING  Jul 6 18:59:58 core kernel: [ 10.097508] postgres (2083): /proc/2083/oom\_adj is deprecated, please use /proc/2083/oom$  Jul 6 18:59:59 core ntpdate[1664]: 142.137.247.109 rate limit response from server.  Jul 6 19:00:03 core kernel: [ 15.296056] eth0: no IPv6 routers present  Jul 6 19:00:09 core ntpdate[1708]: step time server 142.137.247.109 offset 1.707286 sec  Jul 6 19:00:11 core ntpdate[1664]: step time server 206.108.0.131 offset 1.708011 sec |

*Exemple d’un journal*

Un journal (*log*) est un fichier texte dont les évènements sont enregistrés, un par ligne.

Dans **chaque ligne d'évènement on distingue** …

* La **date** à laquelle l'évènement a été déclenché ;
* Le **processus déclencheur** de l'évènement ;
* Le **processus ayant demandé l'ajout du message** correspondant au journal :
* Le **niveau de gravité** du message (*priority*)

Pour afficher les dernières procédures d’ouverture de session, l'heure des tentatives, si elles ont échouées ou réussies …  
**>>** **sudo** **tail --follow /var/log/auth.log**

Le fichier **/var/log/auth.log est le journal des authentifications**.

# Format syslog

Un journal au format syslog comporte dans l'ordre les informations suivantes …

* la **date** à laquelle a été émis le log,
* le **nom de l'équipement** (*hostname*) ayant généré l’événement,
* une **information sur le processus** qui a déclenché cette émission,
* le **niveau de priorité** de l’événement,
* un **identifiant du processus** ayant généré l’événement   
  et
* enfin un **corps de message**.

Certaines de ces informations sont optionnelles.

Voici un exemple d’entrée dans un journal …  
**Sep 14 14:09:09 ordinono dhcp service [warning] 110 message**

**Remarque importante** …  
L'**heure du système doit être à la bonne heure et à la bonne date**, sinon la datation des messages sera erronée, ce qui complique, si besoin est, la recherche d'anomalies de fonctionnement du système dans les messages enregistrés dans les fichiers de journalisation.   
Il faudra s’assurer que tous les **composants du réseau sont synchronisés sur un serveur NTP**.

# Centralisation des journaux

La **centralisation des journaux de plusieurs** serveurs sur un seul peut présenter un grand intérêt au niveau de la sécurité au sein d’un système d’information. En effet, il est **plus facile pour des outils d’analyse de journaux de comparer**, **lire** et **balayer des fichiers** se situant **sur un seul et unique serveur** plutôt que de le faire à distance ou via des agents distants. De plus en **cas de panne d’un serveur**, l’administrateur sera en mesure de **récupérer les erreurs** et **actions menées** sur le serveur avant que celui-ci ne tombe en panne, facilitant ainsi la remise en activité de celui-ci et sa sécurisation future.

Par défaut sur la plupart des distributions Linux modernes, l’outil de gestion des journaux est rsyslog

Le **fichier de configuration à modifier sera /etc/rsyslog.conf**.   
Il est néanmoins possible de créer des fichiers spécifiques et de les sauvegarder dans le **répertoire /etc/rsyslog.d/**et d’utiliser le **paramètre include** **dans le fichier de configuration principal**.

La prochaine étape est de configurer le serveur afin qu’il s'exécute en mode serveur.

## Configuration du serveur

Pour le serveur, il faut simplement **paramétrer ce dernier pour qu’il accepte les événements venant de l’extérieur** en ouvrant le port adéquat.

### Définition du protocole et du port du serveur rsyslog

Pour commencer, on doit définir le protocole et le port sur lesquels on souhaite recevoir les journaux.  
il est possible de choisir l’utilisation du protocole UDP ou du protocole TCP et le port de son choix.

**Remarque** …  
La réception du syslog TCP est plus fiable que le syslog UDP et est suffisamment efficace et rapide.   
UDP peut subir une perte de message. Cela se produit lorsque le serveur syslog doit recevoir de grandes rafales de messages. Si le tampon système pour UDP est plein, tous les autres messages seront supprimés.   
Avec TCP, cela n'arrive pas.   
Toutefois, il peut s’avérer utile de configurer également un serveur UDP.   
Certains périphériques (comme les aiguilleurs (*routers*)) sont incapables d'envoyer des messages syslog à l,aide du protocole TCP.   
Dans ce cas, il faudra deux types de serveurs syslog pour la récolte des messages.

La configuration suivante de syslog permet l’utilisation des protocoles UDP et TCP pour la réception des journaux sur les ports 514 et 50514 respectivement.

Par défaut, le syslog UDP est reçu sur le port 514.

### Activation de la réception syslog à l’aide d’UDP

Il faut accéder au fichier de configuration ryslog afin de le modifier …  
**>>** **sudo** **nano /etc/rsyslog.conf**

On décommente les lignes pour la réception UDP dans la section MODULES …  
**...  
#################  
#### MODULES ####  
#################  
...  
# provides UDP syslog reception  
module(load="imudp")  
input(type="imudp" port="514")**

### Activation de la réception syslog à l’aide de TCP

La réception des messages à l’aide du protocole TCP peut avoir besoin d'utiliser un port différent.  
Il peut arriver que le service RPC utilise également ce port.

Pour configurer rsyslog pour qu'il s'exécute sur un autre port TCP (TCP 50514), on doit décommente les lignes de réception TCP et modifier le port de réception …  
**# provides TCP syslog reception  
module(load="imtcp")  
input(type="imtcp" port="50514")**

On enregistre et on quitte le fichier.

Pour que le service tienne en compte les modification du fichier de configuration, on doit remémarrer le service …  
**>>** **sudo** **systemctl restart rsyslog.service**

On peut vérifier que rsyslog écoute maintenant sur deux ports …  
**>>** **sudo** **ss -4altunp | grep 514  
udp UNCONN 0 0 0.0.0.0:514 0.0.0.0:\*  
tcp LISTEN 0 25 0.0.0.0:50514 0.0.0.0:\***ou  
**>> sudo netstat -4altunp | grep 514  
tcp 0 0 0.0.0.0:50514 0.0.0.0:\* LISTEN 3961/rsyslogd   
udp 0 0 0.0.0.0:514 0.0.0.0:\* 3961/rsyslogd**

On remarque peut-être que le port UDP n'a pas en état LISTEN car il est sans connexion et n'a aucun concept d’écoute, établi ou fermé.

## Définition des hôtes autorisés

Il est possible de définir de manière explicite les clients distants autorisés à envoyer des messages syslog à rsyslogd.   
Afin d’y arriver, on doit définir une directive globale à l'aide $AllowedSender.

Les listes d'expéditeurs autorisés peuvent être définies séparément pour les expéditeurs UDP et TCP.

La syntaxe pour les spécifier est …  
**$AllowedSender [UDP/TCP], IP[/CIDR], IP[/CIDR]**

IP[/CIDR] est une adresse IP d’un réseau ou d’un hôte comme 192.168.1.0/24 ou 192.168.1.100.   
Si la partie /CIDR est omise, un seul hôte est supposé.

Des noms d'hôtes, avec et sans caractères génériques, peuvent également être précisés.   
Si tel est le cas, le résultat de la résolution DNS inversée est utilisé pour le filtrage.   
Plusieurs expéditeurs autorisés peuvent être spécifiés dans une liste délimitée par des virgules.

**Remarque** …  
C’est une bonne pratique de spécifier les hôtes avec un volume de trafic élevé avant ceux avec un volume plus faible.

Pour autoriser des hôtes spécifiques pour la journalisation UDP ou TCP …  
**>> sudo /etc/rsyslog.conf**...  
**###########################  
##### GLOBAL DIRECTIVES ####  
###########################  
# $AllowedSender – Spécification des hôtes distants qui sont autorisés à envoyer des messages syslog à rsyslogd $AllowedSender UDP, 192.168.1.0/24, [::1]/128, \*.tux.info, srv01.tux.info  
$AllowedSender TCP, 192.168.2.0/24, [::1]/128, \*.tux.info, srv02.tux.info**

Les noms d'hôtes doivent pouvoir être résolus car avant la mise à jour d'ACL, ils seront résolus par leurs adresses IP individuelles.

**Remarques** …  
Les directives précédentes autorisent uniquement la réception UDP à partir du réseau 192.168.1.0/24 et la réception TCP à partir de 192.168.2.0/24.  
En plus d'autoriser des hôtes spécifiques à l’aide des directives précédentes, une autre technique estd'imposer des limitations d'expéditeurs autorisés à l’aide du pare-feu.

### Vérification de la syntaxe du fichier de configuration

Après l’ajout de modifications au fichier /etc/rsyslog.conf, on doit redémarrer le service rsyslog.   
Toutefois, avant de redémarrer le service, il est préférable d’effectuer une vérification du fichier de configuration …  
**>> sudo rsyslogd -f /etc/rsyslog.conf -N1  
rsyslogd: version 8.2102.0, config validation run (level 1), master config /etc/rsyslog.conf  
rsyslogd: End of config validation run. Bye.**

**Remarque** …  
L’option -f spécifie le fichier de configuration à vérifier. Cette option est facultative.  
L’option -N1 ne vérifie que l'exactitude du fichier de configuration.

Après la vérification du fichier de configuration, on redémarre le service …  
**>> sudo systemctl restart rsyslog**

La configuration de rsyslogd est maintenant complétée et ce dernier est prêt à recevoir les journaux des hôtes distants.

## Configuration du (des) client(s)

On doit s’assurer, dans un premier temps que paquet rsyslog est installé.

Afin de configurer le client pour envoyer ses événements au serveur central de journalisation, il faut aussi modifier le fichier **/etc/rsyslog.conf**.

Comme configuration de base, il suffit de rediriger tous les événements vers le serveur en ajoutant cette ligne à la fin du fichier …  
**#--- Envoi des messages syslog au serveur –>  
\*.\* @˂Adresse IP du serveur˃:514**

**Remarque** …  
Le @ signifie que les messages seront redirigés vers un serveur.

Il faut ensuite redémarrer le service rsyslog …  
**>> systemctl restart rsyslog.service**

Il suffit, comme test, de générer des événements …

* redémarrer un service quelconque ;
* ouvrir et fermer une session (avec bon et mauvais mot de passe) ;
* …

Il faudra alors par la suite scruter les fichiers journaux dans le répertoire /var/log/ du serveur.  
Pour ce faire, l’utilisation de la commande **journalctl** utile.

Pour envoyer les journaux d'authentification sur le port UDP 514, on ajoute la ligne suivante à la fin du fichier de configuration /etc/rsyslog.conf ...   
**# Envoyer les journaux au serveur syslog distant à l’aide du protocole UDP  
auth,authpriv.\* @192.168.1.100:514**

Pour envoyer tous les journaux sur le port TCP 50514, on ajoute la ligne suivante à la fin du fichier de configuration /etc/rsyslog.conf …  
**# Envoyer les journaux au serveur syslog distant à l’aide du protocole TCP 50514  
 \*.\*@@192.168.1.100:50514**

**Remarques** …  
Le serveur ayant l’adresse IP 192.168.1.100 va recevoir les messages de l’hôte local.  
Cette configuration doit être spécifiée à partir d'hôtes des réseaux 192.168.1.0/24 et 192.168.2.0/24, conformément à la directive AllowedSender du serveur qui centralise les messages syslog.

Il faut ensuite redémarrer le service rsyslog …  
**>> systemctl restart rsyslog.service**

Configuration d’un modèle rsyslog

Les modèles sont une fonctionnalité clé de rsyslog.   
Toute sortie générée par rsyslog peut être modifiée et mise en forme selon ses besoins à l'aide de modèles.

Pour créer un modèle, on utilise la syntaxe suivante dans le fichier de configuration /etc/rsyslog.conf …  
**$template <Nom du modèle>,"<Texte %PROPERTY% <Texte supplémentaire", [OPTION]**

Voici un exemple pratique …  
**# provides TCP syslog reception  
module(load="imtcp")  
input(type="imtcp" port="50514")**

**#Modèle pour générer le nom du fichier journal   
$template RemInputLogs, "/var/log/remotelogs/%FROMHOST-IP%/%PROGRAMNAME%.log"  
\*.\* ?RemInputLogs**

Cette configuration classera les journaux reçus de l'hôte distant dans des fichiers journaux pour des programmes spécifiques responsables de la génération de ce journal.

On redémarre le service …

**>> systemctl restart rsyslog.service**

### Mise en cache des messages

Afin de se protéger au cas où le serveur rsyslog distant tombe en panne et que les journaux sont si importants que l’on ne veut pas risque de les perdre, on peut définir une file d'attente du disque rsyslog pour la mise en mémoire tampon dans le fichier de configuration /etc/rsyslog.conf des hôtes clients …  
**# Définition du tampon de la file d'attente au cas où le serveur tomberait en panne  
  
# Définition d’un nom de fichier pour l'assistance disque  
$ActionQueueFileName queue**

**# Définition de la taille maximale que tous les fichiers de file d'attente utiliseront ensemble sur le disque  
$ActionQueueMaxDiskSpace 1g**

**# Spécification que les données doivent être enregistrées à l'arrêt  
$ActionQueueSaveOnShutdown on**

**# Spécification que les messages en file d'attente demeure en mémoire, ce qui rend le processus très rapide  
$ActionQueueType LinkedList**

**# Spécification que rsyslog ne supprime des messages lors de la nouvelle tentative de connexion si le serveur ne répond pas  
$ActionResumeRetryCount -1**

Il faut ensuite redémarrer le service rsyslog …  
**>> systemctl restart rsyslog.service**

## Vérifcation de la réception des messages sur le serveur

On se connecte au serveur rsyslog et on vérifier la bonne réception des messages …  
**>> ls /var/log/remotelogs/**

**>> ls /var/log/remotelogs/192.168.60.12/**

### Redirection ou copie selon la «*facility*» ou la priorité (*priority*)

Sous Linux, lorsque l’on parle de gestion de journaux, le**s «facility» représentent des catégories dans lesquelles les événements vont se « ranger » afin de mieux les archiver et les trier**.

**Liste des services reconnus par *syslog***

|  |  |
| --- | --- |
| Service | Utilisation |
| *auth* | Messages relatifs à la sécurité du système et à l'authentification des utilisateurs. |
| *cron* | Messages générés par les daemons *cron* et at. |
| *daemon* | Messages générés par les autres daemons systèmes. |
| *kern* | Messages générés par le noyau (en provenance de *klogd*). |
| *lpr* | Messages relatifs au service d'impression. |
| *mail* | Messages provenant du système de mail. |
| *news* | Messages générés par la gestion des news *USENET*. |
| *syslog* | Messages générés en interne par *syslog*. |
| *user* | (C'est le service par défaut). Messages en provenance des utilisateurs. |
| *uucp* | Messages générés par le système UUCP (*Unix-to-Unix Copy*). |

À ces services, il faut rajouter ces derniers …

* **local0 à local7 qui sont des services définis localement** par l'administrateur système
* **\* qui désigne toutes les « facilities »**  
  et
* **none qui désigne aucune « facility »**.

En plus de ces « *facilities* », on retrouve, pour chacune d’entre elles, un **niveau de gravité (aussi appelé priorité) qui va du plus grave à la simple information** …

**Liste des priorités (par ordre décroissant) reconnues par syslog**

|  |  |
| --- | --- |
| Priorité | Signification |
| emerg (emergency) | Système inutilisable |
| alert (alerte) | Une intervention immédiate est nécessaire |
| crit (critique) | Erreur critique pour le système |
| err (erreur) | Erreur de fonctionnement |
| warning (avertissement) | Avertissement (une erreur peut intervenir si aucune action n'est prise). |
| notice (note) | Événement normal méritant d'être signalé |
| info (information) | Message à titre informatif |
| debug (debogage) | Message de mise au point |

On retrouve ainsi plusieurs façons de spécifier les événements qui nous intéressent et, par le fait même, ceux que l’on va rediriger.

**Voici quelques exemples** …

* Redirection vers le serveur de journalisation 192.168.19.145 uniquement les messages critiques et supérieurs concernant les courriels sur le port UDP 514 …  
  **mail.err @192.168.19.145:514**
* Redirection de tous les événements relatifs aux courriels …  
  **mail.\* @192.168.19.145:514**

Il est également possible de saisir en une ligne plusieurs types de « *facilities* » et de priorité.   
**\*.=debug;\  
 auth,authpriv.none;\  
 news.none;mail.none -/var/log/debug  
\*.=info;\*.=notice;\*.=warn;\  
 auth,authpriv.none;\  
 cron,daemon.none;\  
 mail,news.none -/var/log/messages**

Dans l’exemple précédent…

* toutes les priorités debug sont redirigées vers le fichier /var/log/debug.
* toutes les priorités info,notice et warn se retrouveront dans /var/log/messages.

Pour que ces filtres soient redirigés vers le serveur central de journalisation, il suffit de spécifier l’IP du serveur ainsi que son port.

## Principaux journaux

Voici une liste des principaux journaux …

|  |  |
| --- | --- |
| Journal | Fonction |
| /var/log/messages | Messages globaux du système, incluant les messages enregistrés lors du démarrage et de l’arrêt du système |
| /var/log/auth.log  journal des authentifications | Fichier contenant les informations d‘autorisation du système, y compris les connexions des utilisateurs et des authentifications sur des hôtes spécifiés |
| var/log/daemon.log | Fichier regroupant les informations consignées par les divers démons qui sont en fonction sur le système |
| /var/log/kern.log | Fichier qui contient les informations consignées pour le noyau et qui sont utiles pour l’utilisateur qui veut dépanner un noyau personnalisé |
| /var/log/cron.log | Chaque fois que le daemon *cron* ou *anacron* commence une tache CRON, il enregistre les informations sur cette tâche dans ce fichier |
| /var/log/lpr.log | Fichier qui contient les informations relatives au démon responsable de l’impression |
| /var/log/mail.log | Fichier contenant les journaux supplémentaires à partir du serveur de messagerie |
| /var/log/user.log | Fichier qui contient des informations sur tous les journaux de niveau utilisateur |
| /var/log/httpd ou  /var/log/apache2 | Fichier regroupant les demandes de services (*access\_log*) du serveur web  Apache et les événements de type *erreur\_log* |
| /var/log/sssd/ | Fichier qui contient les messages de sécurité démon qui gère l’accès aux répertoires distants et ses mécanismes d’authentification |

**Remarque** …  
La commande **dmesg est également très utile pour voir les événements survenus lors du démarrage**.

## Suivi des logs systèmes en temps réel

Un atout dans les systèmes GNU/Linux, c’est principalement la verbosité du système.  
Cela signifie, que lorsqu’un système Linux est installé, soit sur un hôte local, soit sur un serveur distant, il est possible de consulter en temps réel les informations renvoyées par le système (dans une console / fenêtre terminal en local, ou connecté en SSH à un serveur distant).

Les **principales remontées d’informations sont les journaux systèmes**, le plus généralement inscrits dans le fichier **/var/log/messages**.

Pour suivre l’activité du fichier, la commande **tail** est fort utile …  
**>> tail -f /var/log/messages**Avec cette commande, les informations vont défiler automatiquement sur la sortie standard (la console).

Pour interrompre la consultation …  
**CTRL+C**

Il est également possible de conserver le processus actif, et reprendre la main sur la console, en plaçant le processus en arrière-plan …  
**>> tail -f /var/log/messages**

Pour interrompre la consultation …  
**CTRL+Z**La tâche reste active en arrière-plan (*background*).

**Remarque** …  
Après avoir pressé **CTRL+Z**, la tâche est numérotée (ici 1, la première en arrière-plan), et se retrouve à l’état *stopped*. **bg** permet de déplacer la tâche devient active.

Il est alors possible de saisir d’autres commandes, tout en ayant la visibilité sur les événements du système.   
Les événements s’affichent quoi que l’on fasse, ce qui peut parasiter la saisie des commandes.

L’intérêt de la chose, est, par exemple, de pouvoir monitorer en temps réel plusieurs fichiers de journalisation.

**Première étape** …  
Analyse des logs système  
**>> tail -f /var/log/messages**  
**CTRL+Z** - Activation de la tâche en arrière-plan  
**>> bg**

**Seconde étape** …  
Analyse simultanée des logs apache  
**>> tail -f /var/log/httpd/error\_log**  
**CTRL+Z** - Activation de cette autre tâche en arrière-plan  
**>> bg**(le 2 indique qu’il s’agit de la deuxième tâche tournant en arrière-plan).

**Troisième étape** …  
Lancement des services pour visualiser le résultat …  
**>> systemctl restart apache2.service**(apache se doit d’être installé sur l’hôte pour cet exemple)  
Le défilement des informations relatifs au redémarrage du service sont affichées.

**Quatrième étape** …  
Remise des tâches en premier plan afin de les interrompre  
**>> fg 1  
CTRL+C** pour interrompre la tâche.   
On répète la commande *fg* jusqu’à l’apparition du message  
**bash: fg: current : tâche inexistante**Aucun processus demeure en arrière-plan.

Ceci est évidemment utilisable avec tous les événements renvoyés par le système, événements généralement disponibles dans le répertoire **/var/log**.