

TIW3 - Administration des systèmes et des bases de données

Outils du sysadmin

Fabien RICO (fabien.rico@univ-lyon1.fr)

Univ. Claude Bernard Lyon 1

séance 2



- 1 Outils de gestion
 - Les services
 - Gestion du noyau
 - Le firewall

- 2 Outils de débogage
 - Les logs
 - Les outils de test
 - Sysdig



Introduction

Les outils de configurations sont une interface du système :

- Ils évoluent :
 - ▶ Passage à `systemd`
 - ▶ Généralisation des répertoires `*.d`
- Il faut *suivre la mode* :
 - ▶ Les changements peuvent avoir une bonne raison.
 - ▶ S'ils sont acceptés, c'est souvent une preuve de leur utilité.
 - ▶ S'ils sont acceptés, les outils dépréciés ne vont plus fonctionner.
- Attention par contre à ne pas trop anticiper.



Utilité de la documentation

Définition (Google is not your friend)

« Many people try to find documentation by Googling keywords, but this method is not generally productive. Most of the links are to other people asking the same question, or to out-dated third party documentation. It is easy to get lost in a mess of contradictory and confusing documentation, leading to frustration and a badly configured server.

Most third-party documentation and How-To's are wrong and outdated. We strongly suggest that you do not follow any documentation which is more than 4 years old. If you do follow such ancient documentation, the result will likely be a server that does not work. »

Documentation de freeradius

- La documentation doit suivre la version du logiciel.
- Le cycle des versions n'est pas forcément celui des tutoriels ni de la popularité sur google.
- Les grandes entreprises du web et les technologies associées ont réussi car leur évolution est très rapides.

Il faut **lire le manuel**



- 1 Outils de gestion
 - Les services
 - Gestion du noyau
 - Le firewall

- 2 Outils de débogage
 - Les logs
 - Les outils de test
 - Sysdig



Les services

Définition (Services)

Les *services* sont les programmes lancés au démarrage du système. Ce sont eux qui permettent effectivement son utilisation et lui donne un rôle particulier. Un grand nombre de logiciels utilisent des services

- les serveurs réseaux (web, mail,...) ;
- les serveurs internes (serveur graphique, d'impression, audit...)
- les systèmes de virtualisation (docker, virtualbox ...)

La grande évolution du coté des service est le quasi abandon de l'ancien système `systemV` pour `systemd`.



System V

Système basé sur des niveaux d'exécution et une série de scripts lancés dans un ordre défini.

- `/etc/inittab` définit le niveau d'exécution
- `/etc/init.d/` contient les scripts écrits la plupart du temps en `bash`
- `/etc/rc[0-6].d/` contient des liens vers les scripts dont le nom donne l'ordre d'exécution.
- `/etc/rc.local` est un script exécuté en dernier pour les modifications mineures.

Le système souffrait de plusieurs défauts mais avait l'immense avantage d'être simple et facile à prendre en main (Keep It Simple and Stupid).



Sytemd

C'est un système plus complexe donnant une grande liberté de configuration :

- possibilité de lancer les services en parallèle ;
- gestion des dépendances entre services ;
- utilisation des cgroups pour la gestion des processus liés au services ;
- possibilité de snapshot pour retrouver un état de fonctionnement ;
- ...

D'après Wikipedia, « La documentation de systemd comporte à elle seule actuellement 579 entrées, référencant 216 pages de manuel, soit 72 % de toutes les pages de manuel d'Unix v7 pour un seul logiciel. ».



Fonctionnement général

`systemd` gère des *unités* de différents types (liste non exhaustive) :

- `service` les services eux même ;
- `socket` les canaux de communications associés aux services (lancé a part et avant pour améliorer le parallélisme) ;
- `timer` pour la gestion des tâches régulières (`cron`)
- `target` groupes d'unités qui remplacent les niveaux d'exécution
- ...

Ces unités sont lancées par `systemd` en fonction des cibles configurées. Il lance les cible système (si le pid du processus est 1) ou, lors de chaque login, celles des utilisateurs.



Outil(s) de configuration : systemctl

system V n'avait pas d'outils intégrés mais des utilitaires proposés par les différentes distributions.

systemd propose des logiciels standard pour la configuration notamment `systemctl`

- Il est plutôt bien géré par la completion avancée.
- Il permet de lister des services : `systemctl list-units`.
- Il permet d'allumer/eteindre/tester les services
`systemctl (start|stop|status) NOMSERVICE`
- Il permet d'activer/désactiver le service au démarrage
`systemctl (enable|disable) NOMSERVICE`
- Il permet de changer la *cible de demarage* (ex runlevel)
`systemctl set-default (multi-user.target|graphical.target)`



Fichiers de configuration

Les unités sont décrites dans des fichiers :

- Dans la base de configurations par défaut de systemd (installée par les logiciels eux mêmes) :
 - ▶ `/usr/lib/systemd/system/` pour les unités du système ;
 - ▶ `/usr/lib/systemd/user/` pour les unités des utilisateurs ;
 - ▶ par exemple `/usr/lib/systemd/system/httpd.*`
- Dans la base de configuration du système pour les configurations spécifiques.
 - ▶ `/etc/systemd/system` et `/etc/systemd/user`
 - ▶ par exemple
`/etc/systemd/system/docker.service.d/proxy.conf` (en TP de cloud).



A voir pour Systemd

- <https://doc.fedora-fr.org/wiki/Systemd>
- <http://lea-linux.org/documentations/Systemd>
- <http://images.linuxide.com/systemd-vs-sysVinit-cheatsheet.pdf>



Le noyau

Le noyau est le cœur du système d'exploitation. Il est parfois nécessaire d'accéder à ses informations ou de modifier son comportement.

- Lire des données sur les processus, les services.
- Lire des données sur le système lui même.
- Modifier le comportement du noyau.
- Modifier les fonctionnalités du noyau (modules).

Attention, si vous utilisez *docker*, ce n'est pas de la virtualisation. Le noyau (donc les modifications) sont communes avec l'hôte. De plus, le docker n'aura sans doute pas la *capacité* (le droit) de faire ces modifications.



Système de fichier virtuels

Sous linux, vous disposez de 2 répertoires correspondant à des systèmes virtuels.

- `/proc/` contient des informations sur les processus, les variables du système d'exploitation que l'on peut modifier.
 - ▶ `/proc/sys/net/ipv4/ip_forward` le système accepte-t-il de transmettre des paquets ipv4 ?
 - ▶ `/proc/4256/cmdline` la ligne de commande utilisée par le processus 4256.
- `/sys/` contient des informations sur chaque driver du noyau
 - ▶ `/sys/class/thermal/thermal_zone4/temp` température d'un processeur ;
 - ▶ `/sys/devices/system/cpu/cpu0/cpufreq/scaling_cur_freq` fréquence courante du processeur 0.



Groupes de processus

Les cgroup sont un cas particulier de driver du noyau. On trouve leurs informations dans le système `/sys/`

- `/sys/fs/cgroup/` est l'accès aux différents *contrôleurs*
 - ▶ `/sys/fs/cgroup/memory` pour la mémoire
 - ▶ `/sys/fs/cgroup/cpu` pour les processeurs.
 - ▶ ...
- `systemd` utilise les cgroups et forme un groupe pour chaque service.
 - ▶ `/sys/fs/cgroup/memory/system.slice/
sshd.service/memory.usage_in_bytes`
est la consommation actuelle du service ssh et des processus qu'il a lancé.



Les variables

On peut lire/modifier les variables grâce au système de fichier

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Mais il existe un utilitaire pour cela `sysctl`

- `sysctl -a` affiche toutes les variables
- `sysctl net.ipv4.ip_forward=1` affecte une valeur à la variable
- `/etc/sysctl.conf` et `/etc/sysctl.d/*` sont les fichiers chargés au démarrage.



Gestion des modules

Les modules permettent de gérer les fonctionnalités du noyau. Ce sont notamment ceux qui contiennent les drivers du matériel.

- `lsmod` pour lister les modules.
- `modprobe insmod` pour ajouter dynamiquement un module.
- `modinfo` pour obtenir des informations sur un module.

Les modules sont maintenant gérés à part, via des paquets proposés par les constructeurs.

- Paquets `kmod-...` sous redhat.
- **Attention** au décalage entre la mise à jour du noyau et celle du paquet qui en dépend.
- On peut aussi utiliser des utilitaires de recompilation du module lorsque le noyau change : `akmod` sous redhat/fedora et `dkms` sous debian/ubuntu.



Pare-feu sous linux

Sous linux, le pare-feu est implémenté directement dans le noyau via des chaînes de règles appliquées aux paquets qui arrivent ou transitent par le système.

Il est organisé en 3 tables définissant les actions possibles sur le paquet :

- `filter` pour filtrer les paquets (actions ACCEPT, DROP, REJECT, LOG)
- `mangle` pour modifier les paquets (ajouter des tags ...)
- `nat` pour changer les adresses ou rediriger les paquets (actions REDIRECT, MASQUERADE, ...)



Exemple d'utilisation

Exemple (Portail captif)

- Les paquets normaux (non tagués) sont redirigé sur un site web demandant l'authentification (table `nat`).
- Les paquets avec un tag sont routés normalement et non filtrés.
- Si l'authentification réussit, le pare-feu est modifié pour ajouté un tag au paquets associé à cette carte wifi (table `mangle`).



Fonctionnement d'une table

Les tables ont déjà des chaînes pré-configurées dans lesquelles les paquets sont placés automatiquement. Par exemple pour la table **filter**

- INPUT pour les paquets qui sont à destination d'un processus du système.
- OUTPUT pour les paquets qui sont issus d'un processus du système.
- FORWARD pour les paquets qui sont en transit dans le système.

Dans ce cas, les 3 chaînes ne concernent pas les mêmes paquets. D'autres tables ont des chaînes différentes par exemple POSTROUTING les paquets en partance du système (issus d'un processus local ou qui transitent).



Configuration

L'utilitaire de configuration est iptables

- `iptables -t TABLE -A ...` pour ajouter une règle
- `iptables -t TABLE -D ...` pour supprimer une règle
- `iptables -t TABLE -F` pour supprimer toutes les règles
- `iptables -t TABLE -L` pour lister les règles (manque d'information)
- `iptables-save -f NOMFICHIER` pour obtenir les règles précises afin de les sauvegarder (ou les lire)
- `iptables-restore NOMFICHIER` pour rétablir les règles sauvegardées.



Remarques sur le pare-feu

- C'est un firewall à état, c'est à dire qu'on peut tracer les connexions (niveau 4) et filtrer en fonction de l'état d'une connexion TCP ou UDP.
- On peut créer ses propres chaînes et sous certaines conditions transférer des paquets vers ces chaînes.
 - ▶ Créer des traitement spécifiques pour certains services (par ex. fail2ban).
 - ▶ Filtrer en fonction des interfaces d'entrées (par ex. docker).
- On voit de plus en plus apparaître une politique de sécurité dépendant de *zone* (ZBF).
 - ▶ Une connexion wifi ou une carte fait partie d'une zone (public, privée, ...)
 - ▶ La politique est définie sur la zone.
 - ▶ Le gestionnaire de réseau permet de changer la zone.
- Voir <http://olivieraj.free.fr/fr/linux/information/firewall/firewall.html>



- 1 Outils de gestion
 - Les services
 - Gestion du noyau
 - Le firewall

- 2 Outils de débogage
 - Les logs
 - Les outils de test
 - Sysdig



Logs du système

La première action à faire pour résoudre un problème est de consulter les logs du système.

- Toutes les applications envoient leur message de log au gestionnaire du système.
- Ce dernier stocke les informations dans des fichiers textes.
- Cela pose plusieurs problèmes :
 - ▶ Problèmes de classement : l'organisation se fait via une multitude de fichiers.
 - ▶ Problèmes de gestion de la taille
 - ★ les fichiers de logs peuvent devenir tellement gros ou nombreux qu'ils pénalisent le système ;
 - ★ les outils de rotation de logs peuvent perturber les logiciels qui utilisent ces logs.
 - ▶ Problème de format : les fichiers textes utilisent une organisation trop simple et l'ajout d'informations, comme les erreurs java, les rendent illisibles.



Gestion centralisée

Il est possible de transférer les logs en réseau

- gestion centralisée ;
- utilisation des données pour la sécurité ;
- nécessité de mise en forme ;
- problème de stockage ;
- existence de suite de logiciels comme ELK : Logstash (filtre/mise en forme), Elasticsearch (stockage, recherche) Kibana pour la visualisation.

Pour sécuriser un système d'information, une des premières chose à faire est la centralisation des logs. Ces derniers sont aussi nécessaire pour des raisons légales (par exemple pour retrouver un utilisateur auteur d'une infraction).



Les fichiers

La configuration dépend du logiciel qui implémente le service syslog : rsyslog ou syslog-ng. Sous fédora :

- `/etc/rsyslog.conf` pour les configurations par défaut ;
- `/etc/rsyslog.d` pour les modifications.

Les logs sont finalement stockés dans `/var/log/....`

- `/var/log/message` ou `/var/log/syslog` pour les messages généraux ;
- `/var/log/secure` pour les messages en rapport l'authentification ;
- `/var/log/(httpd|apache)/*` pour les messages en rapport avec apache
- ...

On utilise les utilitaires habituels du système pour explorer les logs :

- `tail -f /var/log/...` pour consulter un log en temps réel ;
- `grep` pour rechercher un motif



Cas particulier de systemd

systemd utilise son propre système de logs : journald.

- C'est un service spécial `systemd-journald.service`.
- Il est configurable dans le fichier `/etc/systemd/journald.conf`
 - ▶ transfert de logs à syslog (par défaut non);
 - ▶ stockage persistant des logs...
- Les fichiers de stockage sont binaires, il faut les consulter via la commande `journalctl`
 - ▶ `journalctl -u NOMUNIT -f` consultation temps réel
 - ▶ `journalctl -xe` ouverture des derniers logs (option `e`) avec message explicatif (option `x`).

Attention cela n'affiche que les logs en rapport avec systemd (donc les services) ou les logiciels qui utilisent journald. C'est donc surtout utile pour comprendre pourquoi un service ne démarre pas.



Cas particulier des dockers

Dans un docker ce qui est écrit par le processus principal sur ses sorties est conservé pour les logs.

```
docker logs NOM_DU_DOCKER
```

permet de consulter ces logs.

Mais si le docker contient un logiciel qui écrit ses logs dans le syslog interne du docker, il n'est pas facile de le lire.

- Il est possible d'utiliser `docker exec` pour les consulter
`docker exec -it ockerapache tail -f /var/log/apache/error.log`
- Certaines images modifient les fichiers de logs des serveurs pour qu'ils correspondent au sortie du processus principales :
 - ▶ L'image de `nginx` crée un lien symbolique entre `/var/log/nginx/access.log` et `/dev/stdout` (idem pour `error.log` et `/dev/stderr`).
 - ▶ L'image de `apache` modifie la configuration du serveur pour écrire dans `/proc/self/fd/1` et `2`.
- Enfin il est toujours possible de partager comme volume le répertoire `/var/log/`.



Ça ne marche pas !

Pour mettre en place ou réparer une application réseaux vous disposez :

- Outils d'écoute sur le réseau pour voir ce qui est envoyé.
- Clients simples pour tester un service.
- Outils de connexion à distance pour effectuer une manipulation à distance.
- Historique d'évènements (Log) et mode de debuggage.



Écoute sur le réseau

Pourquoi ?

- Ce qu'on envoie n'est pas toujours ce qu'on croit.
- Les paquets n'arrivent pas toujours.
- Le service peut être défaillant

Outils :

- `wireshark` ou `tshark`
- `tcpdump`



Exemple :

Que vous apprend le logiciel wireshark sur le service ldap :

No..	Time	Source	Destination	Protocol	Info
1	0.00000	10.57.10.2	10.99.5.9	DNS	Standard query A ldap.tpr.univ-lyon1.fr
2	0.00096	10.57.10.2	10.99.5.9	DNS	Standard query AAAA ldap.tpr.univ-lyon1.fr
3	0.05884	10.99.5.9	10.57.10.2	DNS	Standard query response CNAME bertrand.tpr.univ-lyon1.fr A 10.99.5.9
4	0.06135	10.99.5.9	10.57.10.2	DNS	Standard query response CNAME bertrand.tpr.univ-lyon1.fr
5	0.06161	10.57.10.2	10.99.5.9	TCP	59078 > ldap [SYN] Seq=0 Win=5840 Len=0 MSS=1460 TSV=224757689 TSER=0 WS=7
6	0.11767	10.99.5.9	10.57.10.2	TCP	ldap > 59078 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 TSV=1315127039 T
7	0.11769	10.57.10.2	10.99.5.9	TCP	59078 > ldap [ACK] Seq=1 Ack=1 Win=5888 Len=0 TSV=224757745 TSER=1315127039
8	0.11794	10.57.10.2	10.99.5.9	LDAP	bindRequest(1) "uid=moiA2,ou=People,dc=tpr,dc=univ-lyon1.fr" simple
9	0.17792	10.99.5.9	10.57.10.2	TCP	ldap > 59078 [ACK] Seq=1 Ack=61 Win=5888 Len=0 TSV=1315127096 TSER=224757745
10	0.17793	10.99.5.9	10.57.10.2	LDAP	bindResponse(1) invalidCredentials
11	0.17794	10.57.10.2	10.99.5.9	TCP	59078 > ldap [ACK] Seq=61 Ack=15 Win=5888 Len=0 TSV=224757805 TSER=131512709
12	0.18206	10.57.10.2	10.99.5.9	TCP	59078 > ldap [FIN, ACK] Seq=61 Ack=15 Win=5888 Len=0 TSV=224757809 TSER=1315
13	0.23926	10.99.5.9	10.57.10.2	TCP	ldap > 59078 [FIN, ACK] Seq=15 Ack=62 Win=5888 Len=0 TSV=1315127160 TSER=224
14	0.23928	10.57.10.2	10.99.5.9	TCP	59078 > ldap [ACK] Seq=62 Ack=16 Win=5888 Len=0 TSV=224757866 TSER=131512716

Exemple :

Que vous apprend le logiciel **wireshark** par rapport au service telnet :

File Edit View Go Capture Analyze Statistics Telephony Tools Help

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
1	0.0000	AsustekC_bd:29:49	Broadcast	ARP	Who has 192.168.1.11? Tell 192.168.1.10
2	0.00005	HonHaiPr_16:73:8b	AsustekC_bd:29:49	ARP	192.168.1.11 is at 00:26:5e:16:73:8b
3	0.00109	192.168.1.10	192.168.1.11	TCP	46104 > telnet [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK PERM=
4	0.00113	192.168.1.11	192.168.1.10	TCP	telnet > 46104 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)

- Ethernet II, Src: AsustekC_bd:29:49 (bc:ae:c5:bd:29:49), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
- Address Resolution Protocol (request)

```

0000  ff ff ff ff ff ff bc ae  c5 bd 29 49 08 06 00 01  ..... ..)I....
0010  08 00 06 04 00 01 bc ae  c5 bd 29 49 c0 a8 01 0a  ..... ..)I....
0020  00 00 00 00 00 00 c0 a8  01 0b 00 00 00 00 00  .....
0030  00 00 00 00 00 00 00 00  00 00 00 00  .....
  
```

File: "/test" 318 Bytes... Packets: 4 Displayed: 0 Marked: 0 Load time: 0:00.113 Profile: Default



Client simple

Pourquoi ?

- Il faut tester chaque service.
- Les applications qu'on utilise donnent des informations de haut niveau peu utiles pour le débogage.
- Cela permet d'avoir des informations fiables.

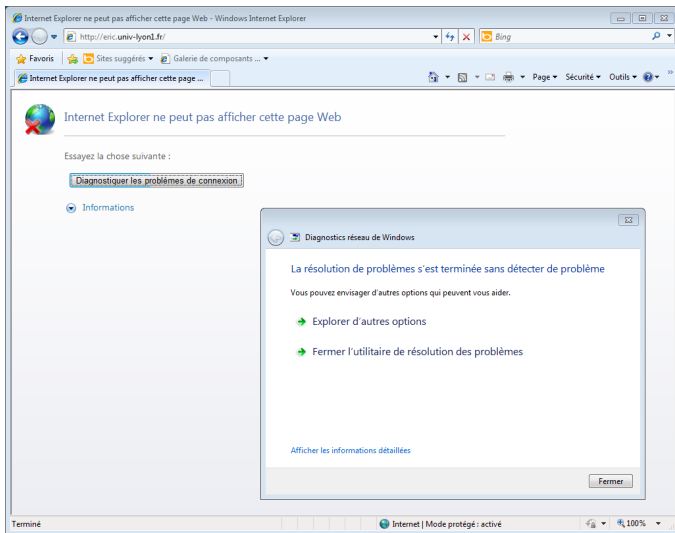
Outils :

- ping pour tester la connectivité ;
- telnet pour ouvrir une connexion tcp ;
- nc (netcat) pour ouvrir des serveurs ;
- selon le serveur qu'on doit contacter ldapsearch, mysql, ftp, ...



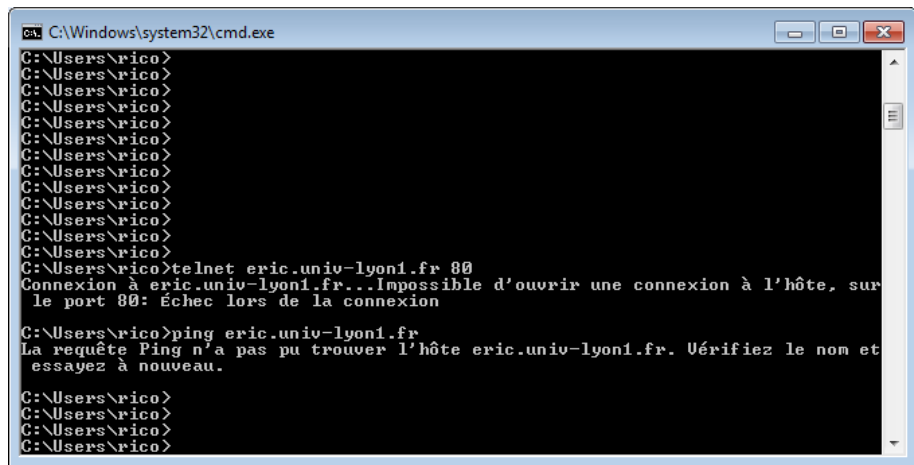
Exemple :

Quel est l'erreur ?



Exemple :

Quel est l'erreur ?



```
C:\Windows\system32\cmd.exe
C:\Users\rico>
C:\Users\rico>
C:\Users\rico>
C:\Users\rico>
C:\Users\rico>
C:\Users\rico>
C:\Users\rico>
C:\Users\rico>
C:\Users\rico>
C:\Users\rico>
C:\Users\rico>
C:\Users\rico>
C:\Users\rico>
C:\Users\rico>
C:\Users\rico>
C:\Users\rico>telnet eric.univ-lyon1.fr 80
Connexion à eric.univ-lyon1.fr...Impossible d'ouvrir une connexion à l'hôte, sur
le port 80: Échec lors de la connexion

C:\Users\rico>ping eric.univ-lyon1.fr
La requête Ping n'a pas pu trouver l'hôte eric.univ-lyon1.fr. Vérifiez le nom et
essayez à nouveau.

C:\Users\rico>
C:\Users\rico>
C:\Users\rico>
C:\Users\rico>
```



Outils de connexion à distance

Pourquoi ?

- Vérifier les configurations.
- Deux machines différentes n'ont pas le même environnement.

Outils :

- `telnet`
- `ssh`
- RDP (Bureau à distance)



Historique d'évènement

Pourquoi ?

- Détails sur ce qui arrive.
- Analogie avec le debuggage d'un programme.
- Voir les effets de ce que vous faites.

Outils :

- Logs `/var/log/...`
- *outils d'administrations* → *observateur d'évènement*



Exercice



Sysdig

Définition (Sysdig)

C'est un système de monitoring bas niveau. Il est capable de récolter, filtrer et mettre en forme les événements du système.

- monitoring en temps réel ;
- récupération de trace ;
- interface ncurses ;
- utilise des script *lua*.

Il permet de rendre les mêmes services que plusieurs utilitaires différents. `sysdig` est aussi disponible (avec des fonctionnalités limitées) sous macos et windows.



Utilisation de base

Par défaut sysdig permet de lister les événements avec ou sans filtre.

- `sysdig -l` liste des filtres
- `sysdig [filter]` affiche les événements reconnus par le filtre.

affichage des événements associé à sshd

```
sysdig proc.name=sshd
```

- `sysdig -w NOMFICHER.scap` sauvegarde les événements dans un fichier de trace.

Le fichier de trace peut être utilisé en entrée pour toutes les commandes suivantes.



Script

L'utilitaire dispose de nombreux scripts qui formatent les résultats appelés *chisels*.

- `sysdig -cl` listes les *chisels* disponibles
- `sysdig -c NOMCHISEL [-r NOMFICHIER.scap]` utilise le chisel

équivalent de la commande ps

```
sysdig -c ps
```

afficher les processus classés par utilisation du réseau

```
sysdig -c topprocs_net
```

- Voir <https://github.com/draios/sysdig/wiki/sysdig-examples>



Gestion des conteneurs

sysdig est capable de monitorer les conteneurs docker et de les étudier séparément. L'option `-p` permet de choisir un format d'affichage adapté.

- `-pc` pour les conteneurs docker
- `-pk` pour *kubernetes*
- `-pm` pour *mesos*

- `csysdig -vcontainers` lister les conteneurs.
- `sysdig -pc -c topprocs_cpu container.name=wordpress1`
appliquer un chisel en filtrant selon un conteneur.



csysdig

csysdig est une interface ncurses à sysdig. Elle permet de naviguer dans les différents affichages, les conteneurs ...

- `csysdig -r NOMFICHIER.scap` pour la lancer sur un fichier trace.
- `csysdig -pc` pour utiliser la possibilité de visualiser les conteneurs.
- Voir <https://www.digitalocean.com/community/tutorials/how-to-monitor-your-ubuntu-16-04-system-with-sysdig>

