

Documentation de mon système de supervision

Procédure rédigée par : CHERIF Ahmed

Table des matières

I - Interface nagios présentant tous les services de tous les serveurs	2
II - Description des sondes choisis avec des solutions proposées	3
II.1 - check_http	3
II.2 - check_ping.....	4
II.3 - check_ssh.....	5
Nagios surveillance avec NRPE	6
II.4 – check_procs.....	7
II.5 – check_diskstat.sh	8
II.6 – check_load	9
II.7 – check_ram.sh	10
II.8 – check_users	11
II.9 - check_disk	12
II.10 – check_swap.....	13
II.11 - check_apd	14
II.12 - check_spam.py	15
III – Centraliser les logs avec Rsyslog	15
III.1 – Les logs systèmes	15
III.2 – Les logs du serveur apache	17

I- \ Interface nagios présentant tous les services de tous les serveurs

Current Network Status

Last Updated: Wed Sep 29 16:27:01 CEST 2021
Updated every 90 seconds
Nagios® Core™ 4.4.6 - www.nagios.org
Logged in as nagiosadmin

[View History For all hosts](#)
[View Notifications For All Hosts](#)
[View Host Status Detail For All Hosts](#)

Host Status Totals

Up	Down	Unreachable	Pending
2	0	0	0
All Problems		All Types	
0		2	

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
22	0	0	1	0
All Problems		All Types		
1		23		

Service Status Details For All Hosts

Limit Results:

Host	Service	Status	Last Check	Duration	Attempt	Status Information
Nagios Serveur	HTTP sur Nagios Server	OK	09-29-2021 16:25:49	0d 7h 7m 13s	1/3	HTTP OK: HTTP/1.1 200 OK - 8220 octets en 0,001 secondes de temps de réponse
	affiche le nombre de processus	OK	09-29-2021 16:26:16	3d 15h 10m 47s	1/3	PROCS OK: 113 processus
	check IO	OK	09-29-2021 16:26:44	4d 15h 20m 0s	1/3	summary: 0 io/s, read 0 sectors (0kB/s), write 7544 sectors (0kB/s) in 59204 seconds
	check apt	CRITICAL	09-29-2021 16:26:25	0d 0h 15m 37s	3/3	APT CRITICAL: 2 packages available for upgrade (2 critical updates).
	ping	OK	09-29-2021 16:22:13	5d 1h 24m 30s	1/3	PING OK - Paquets perdus = 0%, RTA = 0.07 ms
	ssh sur Nagios Serveur	OK	09-29-2021 16:22:42	6d 21h 28m 16s	1/3	SSH OK - OpenSSH_7.9p1 Debian-10+deb10u2 (protocol 2.0)
	verifier la charge du processeur	OK	09-29-2021 16:23:10	5d 7h 21m 32s	1/3	OK - Charge moyenne: 0.00, 0.01, 0.00
	verifier la memoire sur Nagios Server	OK	09-29-2021 16:23:40	5d 16h 56m 26s	1/3	OK: 64% free memory
	verifier le nombre utilisateurs	OK	09-29-2021 16:24:07	5d 8h 39m 37s	1/3	UTILISATEURS OK - 1 utilisateurs actuellement connectés sur
	verifier le stockage	OK	09-29-2021 16:24:36	5d 15h 31m 51s	1/3	DISK OK - free space: / 16047 MB (88% inode=95%):
	verifier le swap	OK	09-29-2021 16:25:04	5d 9h 47m 19s	1/3	SWAP OK - 100% libre (1021 MB sur un total de 1021 MB)
serveurproduction	HTTP sur Production Server	OK	09-29-2021 16:25:37	0d 0h 12m 24s	1/3	HTTP OK: HTTP/1.1 200 OK - 20723 octets en 0,088 secondes de temps de réponse
	SSH sur le serveur de production	OK	09-29-2021 16:22:06	0d 7h 9m 57s	1/3	SSH OK - OpenSSH_7.9p1 Debian-10+deb10u2 (protocol 2.0)
	affiche le nombre de processus	OK	09-29-2021 16:26:31	0d 7h 10m 31s	1/3	PROCS OK: 106 processes
	check IO	OK	09-29-2021 16:26:01	0d 0h 12m 0s	1/3	summary: 0 io/s, read 0 sectors (0kB/s), write 616 sectors (1kB/s) in 300 seconds
	check apt	OK	09-29-2021 16:22:49	0d 0h 15m 12s	1/3	APT OK: 0 packages available for upgrade (0 critical updates).
	check spam	OK	09-29-2021 16:22:27	0d 7h 9m 34s	1/3	SPAM OK - Tout est normal dans les 4 dernières heures .
	ping	OK	09-29-2021 16:22:56	0d 7h 9m 5s	1/3	PING OK - Paquets perdus = 0%, RTA = 0.86 ms
	verifier la charge du processeur	OK	09-29-2021 16:22:25	0d 1h 10m 36s	1/3	OK - load average: 0.19, 0.23, 0.12
	verifier la memoire sur serveur de production	OK	09-29-2021 16:23:53	2d 12h 3m 36s	1/3	OK: 54% free memory
	verifier le nombre d utilisateur	OK	09-29-2021 16:24:21	5d 11h 27m 46s	1/3	USERS OK - 0 users currently logged in
	verifier le stockage sur serveur de production	OK	09-29-2021 16:24:51	1d 22h 21m 46s	1/3	DISK OK - free space: /var/tmp 15 GB (88% inode=95%):
	verifier le swap	OK	09-29-2021 16:25:20	1d 22h 17m 27s	1/3	SWAP OK - 100% free (1021 MB out of 1021 MB)

Results 1 - 23 of 23 Matching Services

II-\ Description des sondes choisis avec des solutions proposées

1-\ check_http

Ce plugin teste le service HTTP sur l'hôte spécifié. Il peut tester serveurs normaux (http) et sécurisés (https), suivez les redirections, recherchez des chaînes et des expressions régulières, vérifier les temps de connexion et générer des rapports sur délais d'expiration des certificats.

Exemple d'utilisation :

check_http -H \$HOSTADDRESS\$ -w 0,1 -c 0,2 \$ARG1\$ \$ARG2\$

```
./check_http -H localhost -w 0,1 -c 0,2 -u /nagios/main.php -a nagiosadmin:pass
```

-H : nom de l'hôte

-w : Temps de réponse pour aboutir à un état d'avertissement

-c : Temps de réponse pour aboutir à un état critique

-u : URI to GET or POST

-a : Nom d'utilisateur : mot de passe sur les sites avec authentification de base

```
nagios@serveurNagios:/usr/lib/nagios/plugins$ ./check_http -H localhost -w 0,1 -c 0,2 -u /nagios/main.php -a nagiosadmin:pass
HTTP OK: HTTP/1.1 200 OK - 8220 octets en 0,002 secondes de temps de réponse |time=0,002429s;0,100000;0,200000;0,000000;10,000000 size=8220B;;;0
```

Solution :

- Redémarrer le serveur web
- Vérifier s'il y a un problème de réseau (par exemple check_ping)
- Vérifier la configuration du serveur web
- Vérifier s'il y a un problème de surcharge

2-\ check_ping

Ce plugin permet de vérifier les statistiques de connexion d'un hôte distant.

Exemple d'utilisation :

```
check_ping -H $HOSTADDRESS$ -w 40,40% -c 60,60%
```

-H, --hostname=HTE

hôte à ping

-w, --warning=SEUIL

Paire de seuils d'avertissement

-c, --critical=SEUIL

Paire de seuils critiques

```
nagios@serveurNagios:/usr/lib/nagios/plugins$ ./check_ping -H 192.168.1.51 -w 40,40% -c 60,60%
PING OK - Paquets perdus = 0%, RTA = 0.68 ms | rta=0.679000ms;40.000000;60.000000;0.000000 pl=0%;40;60;0
```

Solution :

- Le ping sert de tester pour voir si un périphérique en réseau est accessible.
- Vérifier l'adressage
- A l'aide traceroute , on peut voir le chemin traversé pour arriver au hôte destination, et à partir de ce résultat on pourra intervenir pour la correction et la modification

3-\ check_ssh

Ce plugin permet de vérifier la connexion en SSH sur le serveur distant.

Exemple d'utilisation :

```
check_ssh -H $HOSTADDRESS$
```

```
nagios@serveurNagios:/usr/lib/nagios/plugins$ ./check_ssh 192.168.1.51  
SSH OK - OpenSSH_7.9p1 Debian-10+deb10u2 (protocol 2.0) | time=0,009604s;;;0,000000;10,000000
```

Renvoie CRITICAL si l'Agent ne parvient pas à ouvrir de session SSH.

Si ce n'est pas le cas, renvoie OK.

Solution :

- Vérifier le retour de ping pour voir si la machine est joignable
- Vérifier que OpenSSH est bien installé et configurer

Nagios – Surveillance avec NRPE

NRPE pour Nagios Remote Plugin Executor est un plugin de Nagios qui permet d'exécuter des commandes de supervision sur des machines distantes.

Pour superviser nos serveur Linux il faut installer plugin NRPE nécessaire

```
apt-get install nagios-nrpe-server
```

Editer le fichier nrpe.cfg présent dans le dossier /etc/nagios/

```
nano /etc/nagios/nrpe.cfg
```

Commencer par trouver la ligne:

```
allowed_hosts=192.168.1.50
```

Et mettre à la place de 127.0.0.1 l'adresse IP de votre serveur nagios

Descendre plus bas dans ce fichier de configuration jusqu'à trouver les lignes de check :

```
command[check_users]=/usr/lib/nagios/plugins/check_users -w 5 -c 10
command[check_load]=/usr/lib/nagios/plugins/check_load -w 0.7,0.6,0.5 -c 0.9,0.8,0.7
command[check_sda1]=/usr/lib/nagios/plugins/check_disk -u GB -w 20% -c 10% -p /dev/sda1
command[check_zombie_procs]=/usr/lib/nagios/plugins/check_procs -w 5 -c 10 -s Z
command[check_total_procs]=/usr/lib/nagios/plugins/check_procs -w 150 -c 200
command[check_swap]=/usr/lib/nagios/plugins/check_swap -w 20 -c 10
command[check_io]=/usr/lib/nagios/plugins/check_diskstat.sh -d sda -w 200,100000,100000 -c 300,200000,200000
command[check_spam]=/usr/lib/nagios/plugins/check_spam.py -H 192.168.1.51 -P 3306 -u wordpress_user -p password -d wordpress -w 4 -c 10 -i 4
```

Remarque : tous les plugins et les scripts sont mis dans /usr/lib/nagios/plugins/

4-\ check_procs

Ce plugin permet de vérifier tous les processus et génère des états AVERTISSEMENT ou CRITIQUE si la métrique spécifiée est en dehors des plages de seuil requises. La métrique est par défaut le nombre de processus. Des filtres de recherche peuvent être appliqués pour limiter les processus à vérifier.

Exemple d'utilisation :

```
./check_procs -w 150 -c 200
```

-w Générer un état d'avertissement si la métrique est en dehors de cette plage

-c Générer un état critique si la métrique est en dehors de cette plage

```
nagios@serveurNagios:/usr/lib/nagios/plugins$ ./check_procs -w 150 -c 200  
PROCS OK: 134 processus | procs=134;150;200;0;
```

Solution :

Lorsque le nombre de processus est assez grand, il peut engendrer le ralentissement de système. Donc il faut vérifier la charge de processeur, et on peut décider soit d'arrêter les processus qu'on n'a pas besoin, soit de limiter l'utilisation de processeur en utilisant quelques outils par exemple CPULimit

CPULimit contrôle les processus individuels sur un système Linux par ID de processus. Donc, pour limiter un programme qui s'exécute avec une utilisation trop importante du processeur, nous devons trouver l'ID de processus exact.

5-\ check_diskstat.sh

Ce plug-in vérifie les statistiques d'E/S d'un (ou de tous) disque. Il peut être utilisé pour envoyer des alertes lorsque le nombre maximal d'E/S de disque dur ou de secteurs en lecture/écriture/s est atteint.

Exemple d'utilisation :

```
check_diskstat.sh -d sda -w 200,100000,100000 -c  
300,200000,200000
```

-d DEVICE doit être sans /dev (ex : -d sda)

-w/c TPS,READ,WRITE

- TPS signifie transfert par seconde (aka IO/s)
- LECTURE et ÉCRITURE sont en secteurs par seconde

```
nagios@serveurNagios:/usr/lib/nagios/plugins$ ./check_diskstat.sh -d sda -w 200,100000,100000 -c 300,200000,200000  
summary: 0 io/s, read 0 sectors (0kB/s), write 688 sectors (0kB/s) in 31817 seconds | tps=0io/s;;; read=0b/s;;; write=11b/s;;;  
nagios@serveurNagios:/usr/lib/nagios/plugins$
```

Solution :

- Performances disque insuffisantes
- Sous-systèmes de disque mal configurés ou défectueux
- Applications qui génèrent des activités d'E/S redondantes
- Requêtes sql mal conçues ou non optimisées

6-\ check_load

Ce plugin teste la moyenne de charge actuelle du système.

Exemple d'utilisation :

```
check_load -r -w 0.7,0.6,0.5 -c 0.9,0.8,0.7
```

-w, --warning=WLOAD1,WLOAD5,WLOAD15

Sortie avec l'état d'AVERTISSEMENT si la moyenne de charge dépasse WLOADn

-c, --critical=CLOAD1,CLOAD5,CLOAD15

Sortie avec l'état CRITICAL si la charge moyenne dépasse CLOADn

le format de moyenne de charge est le même que celui utilisé par "uptime" et "w"

-r, --percpu

Divisez les moyennes de charge par le nombre de processeurs (si possible)

```
nagios@serveurNagios:/usr/lib/nagios/plugins$ ./check_load -r -w 0.7,0.6,0.5 -c 0.9,0.8,0.7
OK - Charge moyenne: 0.00, 0.00, 0.00|load1=0.000;0.700;0.900;0; load5=0.000;0.600;0.800;0; load15=0.000;0.500;0.700;0;
nagios@serveurNagios:/usr/lib/nagios/plugins$
```

Solution :

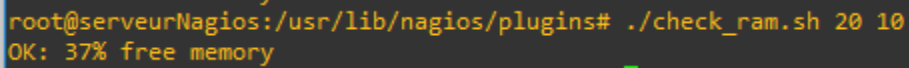
Si on arrive au seuil d'avertissement ou critique, il faut arrêter les processus en cours d'exécution et que je n'ai pas besoin

7-\ check_ram.sh

Ce script génère le pourcentage de mémoire RAM libre et des alertes (avertissement/critique) en fonction des seuils spécifiés.

Exemple d'utilisation :

```
check_ram.sh [warn] [crit]
```



```
root@serveurNagios:/usr/lib/nagios/plugins# ./check_ram.sh 20 10
OK: 37% free memory
```

S'il reste 20 % de la ram libre une alerte d'avertissement s'affiche.

S'il reste 10% de la ram libre une alerte de critique s'affiche

Solution :

Avec la commande Top je peux voir les processus en cours d'exécution avec l'espace ram utilisé, et par la suite je peux décider si je vais arrêter ces processus.

Si besoin on ajoute de la ram pour la VM concernée

8-\ check_users

Ce plugin vérifie le nombre d'utilisateurs actuellement connectés sur le local système et génère une erreur si le nombre dépasse les seuils spécifiés.

Exemple d'utilisation :

```
check_users -w <utilisateurs> -c <utilisateurs>
```

-w, --warning=ENTIER

Définir le statut d'AVERTISSEMENT si plus de INTEGER utilisateurs sont connectés

-c, --critical=ENTIER

Définir l'état CRITIQUE si plus de INTEGER utilisateurs sont connectés

```
check_users -w 5 -c 10
```

```
nagios@serveurNagios:/usr/lib/nagios/plugins$ ./check_users -w 5 -c 10
UTILISATEURS OK - 1 utilisateurs actuellement connectés sur |users=1;5;10;0
nagios@serveurNagios:/usr/lib/nagios/plugins$
```

Solution :

Contrôler les accès à ton serveur.

Déconnecter des utilisateurs en cas de besoin

```
skill -KILL -u utilisateur
```

9-\ check_disk

Ce plugin vérifie la quantité d'espace disque utilisé sur un système de fichiers monté et génère une alerte si l'espace libre est inférieur à l'une des valeurs de seuil

Exemple d'utilisation :

```
check_disk -w 20% -c 10% -p /dev/sda1
```

-w, --warning=PERCENT%

Affiche l'état d'AVERTISSEMENT si moins de POURCENT d'espace disque est libre

-c, --critical=PERCENT%

Affiche l'état CRITIQUE si moins de POURCENT d'espace disque est libre

```
nagios@serveurNagios:/usr/lib/nagios/plugins$ ./check_disk -w 20% -c 10% -p /dev/sda1
DISK OK - free space: / 16049 MB (89% inode=95%);| /=1983MB;15216;17118;0;19021
nagios@serveurNagios:/usr/lib/nagios/plugins$
```

Solution :

Débarrassez-vous des packages qui ne sont plus nécessaires

Effacer les journaux du journal systemd

Nettoyer le cache de thumbnail

10-\ check_swap

Ce plugin vérifie le swap de la machine

Exemple d'utilisation :

```
check_swap -w 30% -c 20%
```

-w, --warning=PERCENT%%

Affiche le statut d'AVERTISSEMENT si moins de POURCENT du swap est libre

-c, --critical=PERCENT%%

Affiche le statut CRITIQUE si moins de POURCENT du swap est libre

Solution :

Effacer le swap de votre système, il vous suffit de désactiver l'échange. Cela déplace toutes les données du swap vers la RAM. Cela signifie également que vous devez vous assurer que vous disposez de la RAM pour prendre en charge cette opération. Un moyen simple de le faire est d'exécuter 'free -m' pour voir ce qui est utilisé dans le swap et dans la RAM. Une fois que vous l'avez éteint, vous pouvez attendre un temps arbitraire (30 secondes environ) pour laisser le temps à l'opération de se terminer, puis rallumer le swap. Cela efface le cache du swap et le réactive.

```
Check space: # free -m  
Disable swap: # swapoff -a  
Wait approx 30 sec  
Enable swap: # swapon -a
```

Augmenter la taille du swap

Si vous constatez un pourcentage élevé d'utilisation du swap, il est temps d'ajouter plus de RAM physique au système Linux.

11-\ Check_apt

Cette une commande qui permet de savoir si des mises à jour sont disponibles pour toutes versions de Linux basée sur Debian et utilisant donc apt.

Exemple d'utilisation :

```
./check_apt
```

```
APT OK: 0 packages available for upgrade (0 critical updates).
```

Renvoi CRITIQUE, si des mises à jour de sécurité sont disponibles

Solution:

Si ce plugin affiche un état alerte ou critique , il faut faire la mise à jour en tapant la commande :

```
apt-get update
```

12-\ Check_spam.py

Ce script permet de vérifier le nombre de commentaires reçu dans les 4 derniers heures.

Exemple d'utilisation :

```
check_spam.py -H 192.168.1.51 -P 3306 -u wordpress_user -p password -d  
wordpress -w 4 -c 10 -i 4
```

-H Nom d'hôte ou adresse IP du serveur de base de données, localhost par défaut

-P Spécifiez un port, 3306 par défaut (TCP uniquement)

-u Spécifiez le nom d'utilisateur autorisé à se connecter à la base de données

-p Spécifiez le mot de passe utilisateur pour se connecter à la base de données

-d Spécifiez le nom de la base de données à laquelle vous souhaitez vous connecter

-w Spécifiez le seuil d'avertissement, avec un nombre de commentaires, 4 par défaut

-c Spécifiez le seuil critique, avec un nombre de commentaires, 10 par défaut

-i Intervalle en heures pour le comptage des commentaires, 4 par défaut

Solution :

Bloquer des adresses emails et mentionné des adresses ip en tant que spam

III-\ Centraliser les logs avec Rsyslog

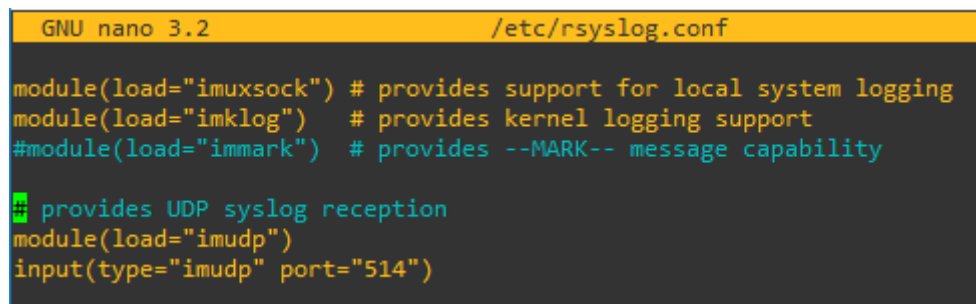
1-\ les logs systèmes :

Coté serveur nagios :

nano /etc/rsyslog.conf

Décommenter ces 2 instructions :

```
module(load="imudp")
input(type="imudp" port="514")
```



```
GNU nano 3.2 /etc/rsyslog.conf
module(load="imuxsock") # provides support for local system logging
module(load="imklog")   # provides kernel logging support
#module(load="immark")  # provides --MARK-- message capability

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

Après l'enregistrement redémarrer le service

```
Systemctl restart rsyslog
```

Coté serveur de production :

nano /etc/rsyslog.conf

Ajouter cette ligne qui permet d'envoyer tous les logs vers le serveur nagios

```
*.* @192.168.1.50:514
```


192.168.1.51 - PuTTY

```
GNU nano 3.2 /etc/rsyslog.conf
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
*. *                      @192.168.1.50:514
```

Après l'enregistrement redémarrer le service

```
Systemctl restart rsyslog
```

Tester le fonctionnement de cette configuration :

Aller au serveur de centralisation nagios et afficher les fichiers logs

192.168.1.50 - PuTTY

```
root@serveurNagios:~#
root@serveurNagios:~#
root@serveurNagios:~# nano /etc/rsyslog.conf
root@serveurNagios:~# tail /var/log/auth.log
Sep 29 10:03:53 serveurproduction sshd[2123]: Received disconnect from 192.168.1
.50 port 59972:11: disconnected by user
Sep 29 10:03:53 serveurproduction sshd[2123]: Disconnected from user nagios 192.
168.1.50 port 59972
Sep 29 10:03:53 serveurproduction sshd[2106]: pam_unix(sshd:session): session cl
osed for user nagios
Sep 29 10:03:53 serveurproduction systemd-logind[330]: Session 25 logged out. Wa
iting for processes to exit.
Sep 29 10:03:53 serveurproduction systemd-logind[330]: Removed session 25.
Sep 29 10:04:03 serveurproduction systemd: pam_unix(systemd-user:session): sessi
on closed for user nagios
Sep 29 10:05:01 serveurNagios CRON[1807]: pam_unix(cron:session): session opened
```

2-\ les logs du serveur apache

Ajouter dans le fichier `wordpress.conf` et `000-default.conf` qui se trouvent dans `/etc/apache2/sites-available/` les lignes suivantes :

```
ErrorLog "|/usr/bin/logger -t apache -p local6.info"
CustomLog "|/usr/bin/logger -t apache -p local6.info" combined
LogLevel info
```

```
192.168.1.51 - PuTTY
GNU nano 3.2 /etc/apache2/sites-available/wordpress.conf

<VirtualHost *:80>
    ServerAdmin admin@your_domain.com
    DocumentRoot /var/www/html/wordpress
    ServerName 192.168.1.51

    <Directory /var/www/html/wordpress>
        Options FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/your-domain.com_error.log
    CustomLog ${APACHE_LOG_DIR}/your-domain.com_access.log combined
    ErrorLog "|/usr/bin/logger -t apache -p local6.info"
    CustomLog "|/usr/bin/logger -t apache -p local6.info" combined
    LogLevel info

</VirtualHost>
```

Après l'enregistrement redémarrer le service

```
Systemctl restart apache2
```

Tester le fonctionnement de cette configuration :

Aller au serveur de centralisation nagios et afficher les fichiers logs

```
192.168.1.50 - PuTTY
root@serveurNagios:~# tail /var/log/messages
Sep 29 10:27:59 serveurproduction apache: 192.168.1.55 - - [29/Sep/2021:10:27:58 +0200] "GET /wp-content/themes/twentyfifteen/js/functions.js?ver=20171218 HTTP/1.1" 200 1937 "http://192.168.1.51/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.61 Safari/537.36"
Sep 29 10:27:59 serveurproduction apache: 192.168.1.55 - - [29/Sep/2021:10:27:58 +0200] "GET /wp-content/themes/twentyfifteen/js/skip-link-focus-fix.js?ver=20141028 HTTP/1.1" 200 795 "http://192.168.1.51/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.61 Safari/537.36"
Sep 29 10:27:59 serveurproduction apache: 192.168.1.55 - - [29/Sep/2021:10:27:58 +0200] "GET /wp-content/themes/twentyfifteen/css/blocks.css?ver=20190102 HTTP/1.1" 200 2387 "http://192.168.1.51/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.61 Safari/537.36"
Sep 29 10:27:59 serveurproduction apache: 192.168.1.55 - - [29/Sep/2021:10:27:58 +0200] "GET /wp-content/themes/twentyfifteen/style.css?ver=20201208 HTTP/1.1" 200 14643 "http://192.168.1.51/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.61 Safari/537.36"
Sep 29 10:27:59 serveurproduction apache: 192.168.1.55 - - [29/Sep/2021:10:27:58 +0200] "GET /wp-content/themes/twentyfifteen/genericons/genericons.css?ver=20201208 HTTP/1.1" 200 16920 "http://192.168.1.51/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.61 Safari/537.36"
Sep 29 10:27:59 serveurproduction apache: 192.168.1.55 - - [29/Sep/2021:10:27:59 +0200] "GET /wp-content/uploads/2021/09/accueil.png HTTP/1.1" 304 182 "http://192.168.1.51/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.61 Safari/537.36"
Sep 29 10:28:06 serveurproduction apache: ::1 - - [29/Sep/2021:10:28:06 +0200] "OPTIONS * HTTP/1.0" 200 126 "-" "Apache/2.4.38 (Debian) (internal dummy connection)"
Sep 29 10:28:07 serveurproduction apache: ::1 - - [29/Sep/2021:10:28:07 +0200] "OPTIONS * HTTP/1.0" 200 126 "-" "Apache/2.4.38 (Debian) (internal dummy connection)"
Sep 29 10:28:08 serveurproduction apache: ::1 - - [29/Sep/2021:10:28:08 +0200] "OPTIONS * HTTP/1.0" 200 126 "-" "Apache/2.4.38 (Debian) (internal dummy connection)"
```

Webography

<https://itsfoss.com/free-up-space-ubuntu-linux/>

<https://www.redhat.com/sysadmin/clear-swap-linux>

<https://www.it-connect.fr/centralisez-vos-logs-avec-rsyslog/>