La première des choses à faire

1. copie du template
2. changement d'adresse ip : 10.31.96.30
3. nom du template:Munin

Installation de Munin :

sur le serveur

```
apt-get install munin munin-node munin-plugins-extra
```

sur le client

```
apt-get install munin-node munin-plugins-extra
```

Cependant, pour la configuration du serveur Munin, cela se fait dans le fichier

```
/etc/munin/munin.conf
```

. Dans ce fichier, il faut décommenter les lignes et les adapter en fonction de notre réseau :

Section globale

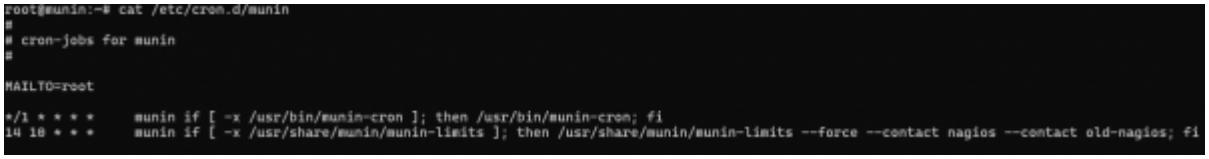
```
dbdir   /var/lib/munin
htmldir /var/cache/munin/www
logdir /var/log/munin
rundir  /var/run/munin
```

Les nœuds

```
[munin.m2l.org]
    address 127.0.0.1
    use_node_name yes

[srv6.m2l.org]
    address 10.31.96.1
    use_node_name yes

[ns.m2l.org]
    address 10.31.96.53
    use_node_name yes

[ns3.m2l.org]
    address 10.31.96.63
    use_node_name yes

[web.m2l.org]
```

```
    address 10.31.96.80
    use_node_name yes

[backup.m2l.org]
    address 10.31.96.99
    use_node_name yes

[ftp.m2l.org]
    address 10.31.96.21
    use_node_name yes
```

Lancement de munin



Configuration du client

dans le Fichier de configuration : /etc/munin/munin-node.conf

- La directive “host_name” est utilisable si le nom d'hôte du client diffère de celui déclaré sur le
- serveur.
- La directive “allow” est une expression régulière indiquant quelles machines sont autorisées à
- interroger le nœud.
- La directive “host” indique sur quelles interfaces Munin doit fonctionner

sur le serveur

```
#host_name localhost.localdomain

# A list of addresses that are allowed to connect.  This must be a
# regular expression, since Net::Server does not understand CIDR-style
# network notation unless the perl module Net::CIDR is installed.  You
# may repeat the allow line as many times as you'd like

allow ^127\.0\.0\.1$
allow ^::1$

# If you have installed the Net::CIDR perl module, you can use one or more
# cidr_allow and cidr_deny address/mask patterns.  A connecting client must
# match any cidr_allow, and not match any cidr_deny.  Note that a netmask
# *must* be provided, even if it's /32
#
# Example:
#
# cidr_allow 127.0.0.1/32
# cidr_allow 192.0.2.0/24
# cidr_deny  192.0.2.42/32

# Which address to bind to;
```

```
host *
# host 127.0.0.1

# And which port
port 4949
```

sur le client

```
nano 7.2
/etc/munin/munin-node.conf
#
# Example config-file for munin-node
#

log_level 4
log_file /var/log/munin/munin-node.log
pid_file /var/run/munin/munin-node.pid

background 1
setsid 1

user root
group root

# This is the timeout for the whole transaction.
# Units are in sec. Default is 15 min
#
# global_timeout 900

# This is the timeout for each plugin.
# Units are in sec. Default is 1 min
#
# timeout 60

# Regexps for files to ignore
ignore_file [\#~]$
ignore_file DEADJOE$
ignore_file \.bak$
ignore_file %$
ignore_file \.dpkg-(tmp|new|old|dist)$
ignore_file \.rpm(save|new)$
ignore_file \.pod$

# Set this if the client doesn't report the correct hostname when
# telnetting to localhost, port 4949
#
host_name munin.m2l.org

# A list of addresses that are allowed to connect.  This must be a
# regular expression, since Net::Server does not understand CIDR-style
# network notation unless the perl module Net::CIDR is installed.  You
```

```
# may repeat the allow line as many times as you'd like

allow ^10\.31\.96\.20$
allow ^::1$

# If you have installed the Net::CIDR perl module, you can use one or more
# cidr_allow and cidr_deny address/mask patterns.  A connecting client must
# match any cidr_allow, and not match any cidr_deny.  Note that a netmask
# *must* be provided, even if it's /32
#
# Example:
#
# cidr_allow 127.0.0.1/32
# cidr_allow 192.0.2.0/24
# cidr_deny  192.0.2.42/32

# Which address to bind to;
host *
# host 127.0.0.1

# And which port
port 4949
```

Lancer munin-node sur le serveur et sur le client et utiliser la commande suivante pour le reeboot:

```
systemctl restart munin-node
```

Vérification qu'il écoute le port 4949:

```
root@munin:~# netstat -nat
Connexions Internet actives (serveurs et établies)
Proto Recv-Q Send-Q Adresse locale          Adresse distante        Etat
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN
tcp        0      0 10.31.96.30:41918       10.31.96.53:4949        TIME_WAIT
tcp        0      0 10.31.96.30:34764       10.31.96.80:4949        TIME_WAIT
tcp        0      0 127.0.0.1:33740         127.0.0.1:4949          TIME_WAIT
tcp        0      0 10.31.96.30:55866       10.31.96.99:4949        TIME_WAIT
tcp        0      0 10.31.96.30:37994       10.31.96.21:4949        TIME_WAIT
tcp        0      0 10.31.96.30:47706       10.31.96.63:4949        TIME_WAIT
tcp6       0      0 ::1:25                  :::*                    LISTEN
tcp6       0      0 :::80                   :::*                    LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
tcp6       0      0 :::443                  :::*                    LISTEN
tcp6       0      0 :::4949                 :::*                    LISTEN
```

Plugins

La commande munin-node-configure permet de visualiser les différents plugins installés.

```
acpi                          | yes |
cpu                           | yes |
cpuspeed                      | yes |
df                            | yes |
df_inode                      | yes |
diskstats                     | yes |
entropy                       | yes |
forks                         | yes |
fw_conntrack                  | yes |
fw_forwarded_local            | yes |
fw_packets                    | yes |
if_                           | yes | eth0
if_err_                       | yes | eth0
interrupts                    | yes |
irqstats                      | yes |
load                          | yes |
memory                        | yes |
munin_stats                   | yes |
netstat                       | yes |
open_files                    | yes |
open_inodes                   | yes |
proc_pri                      | yes |
processes                     | yes |
swap                          | yes |
threads                       | yes |
uptime                        | yes |
users                         | yes |
vmstat                        | yes |
```

Ces plugins sont des scripts écrits en perl, bash, ... et se trouvent dans le répertoire /usr/share/munin/plugins

Les plugins activés se trouvent dans /etc/munin/plugins.

Pour activer un plugin qui ne l'est pas, il faut créer un lien symbolique dans le répertoire des plugins:

- cd /etc/munin/plugins
- ln -s /usr/share/munin/plugins/cpu .
- ln -s /usr/share/munin/plugins/if_ if_enp0s3