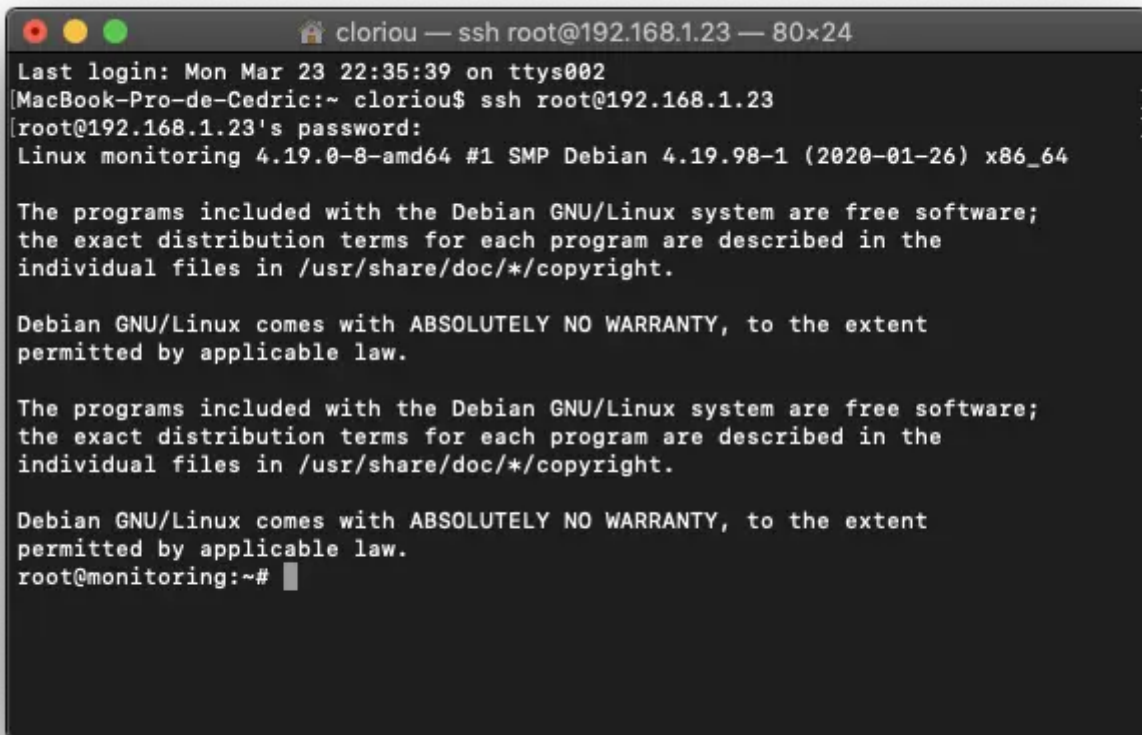


Ajouter un MOTD dynamique

C'est quoi un MOTD ?

Comme indiqué plus haut, lorsque l'on se connecte sur une machine Linux (*Débian dans mon cas*) en ligne de commande (*ici en SSH donc*), il y a tout un message qui s'affiche. **C'est celui-ci que l'on va modifier afin d'y afficher les valeurs de notre choix !**



```
cloriou — ssh root@192.168.1.23 — 80x24
Last login: Mon Mar 23 22:35:39 on ttys002
[MacBook-Pro-de-Cedric:~ cloriou$ ssh root@192.168.1.23
]
[root@192.168.1.23's password:
]
Linux monitoring 4.19.0-8-amd64 #1 SMP Debian 4.19.98-1 (2020-01-26) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@monitoring:~#
```

MOTD de base

Dans mon cas, je souhaite afficher **le nom de la machine**, ainsi que quelques informations sur la machine comme **le CPU, la mémoire libre, l'adresse IP**, etc...

Voici un exemple du résultat que j'ai obtenu.

```
cloriou — ssh root@192.168.1.22 — 80x24
[MacBook-Pro-de-Cedric:~ cloriou$ ssh root@192.168.1.22
[root@192.168.1.22's password:
]
]

jeedom-master

Debian GNU/Linux 10 (buster) (kernel 4.19.0-6-amd64)

Processeur : Common KVM processor
Charge CPU : 0.01 (1min) / 0.03 (5min) / 0.04 (15min)
Adresse IP : 192.168.1.22
RAM : 162MB / 987MB
Uptime : up 1 week, 1 day, 9 hours, 17 minutes

root@jeedom-master:~# █
```

MOTD Personnalisé

Comment changer ce fameux MOTD ?

En fait, lorsque l'on creuse, on se rend compte que c'est intégré nativement à Linux (*je le rappelle, dans mon cas ce sont des distributions Debian*). Pour faire simple, il suffit de créer des scripts, et de les mettre dans un dossier bien spécifique. Lorsque c'est fait, ils seront automatiquement exécutés lorsque vous allez vous connecter.

Ici, je souhaite avoir le **nom de ma machine** avec une police originale et en couleur. Je vais donc commencer par installer l'utilitaire qui permet d'écrire de cette façon, à savoir « **figlet** » .

Pour cela, il faut saisir `apt-get update && apt-get install figlet`


```
cloriou — ssh root@192.168.1.23 — 125x30
-rw-r----- 1 root shadow 1033 mars 23 22:59 shadow
-rw-r----- 1 root shadow 1033 mars 23 22:59 shadow-
-rw-r--r-- 1 root root 116 mars 23 21:54 shells
drwxr-xr-x 2 root root 4096 mars 23 21:54 skel
drwxr-xr-x 2 root root 4096 mars 23 22:59 snmp
drwxr-xr-x 2 root root 4096 mars 23 22:34 ssh
drwxr-xr-x 4 root root 4096 mars 23 22:21 ssl
-rw-r--r-- 1 root root 44 mars 23 22:58 subgid
-rw-r--r-- 1 root root 21 mars 23 22:24 subgid-
-rw-r--r-- 1 root root 44 mars 23 22:58 subuid
-rw-r--r-- 1 root root 21 mars 23 22:24 subuid-
drwxr-xr-x 2 root root 4096 mars 23 22:53 subversion
-rw-r--r-- 1 root root 2351 mai 31 2018 sysctl.conf
drwxr-xr-x 2 root root 4096 mars 23 22:06 sysctl.d
drwxr-xr-x 5 root root 4096 mars 23 22:06 systemd
drwxr-xr-x 2 root root 4096 mars 23 22:06 terminfo
-rw-r--r-- 1 root root 13 mars 23 22:07 timezone
drwxr-xr-x 2 root root 4096 août 20 2019 tmpfiles.d
-rw-r--r-- 1 root root 1260 déc. 14 2018 ucf.conf
drwxr-xr-x 4 root root 4096 mars 23 22:08 udev
drwxr-xr-x 3 root root 4096 mars 23 22:21 ufw
drwxr-xr-x 2 root root 4096 mars 23 22:06 update-motd.d
drwxr-xr-x 2 root root 4096 mars 23 21:56 vim
-rw-r--r-- 1 root root 4942 avril 5 2019 wgetrc
drwxr-xr-x 6 root root 4096 mars 23 22:52 X11
-rw-r--r-- 1 root root 642 mars 1 2019 xattr.conf
drwxr-xr-x 4 root root 4096 mars 23 22:52 xdg
-rw-r--r-- 1 root root 289 févr. 5 2018 xinetd.conf
drwxr-xr-x 2 root root 4096 mars 23 22:59 xinetd.d
root@monitoring:/etc#
```

Le dossier est bien présent

Si celui-ci n'existe pas, il faut alors le créer et lui donner les droits en exécution via `mkdir /update-motd.d && chmod 644 /update-motd.d`.

Si ce dossier existe déjà, vous pouvez vous apercevoir qu'il y a déjà un script `10-uname`. Personnellement je ne vais pas l'utiliser je vais donc le supprimer via `rm 10-uname`.

```
cloriou — ssh root@192.168.1.23 — 125x30
drwxr-xr-x 2 root root 4096 mars 23 22:59 snmp
drwxr-xr-x 2 root root 4096 mars 23 22:34 ssh
drwxr-xr-x 4 root root 4096 mars 23 22:21 ssl
-rw-r--r-- 1 root root 44 mars 23 22:58 subgid
-rw-r--r-- 1 root root 21 mars 23 22:24 subgid-
-rw-r--r-- 1 root root 44 mars 23 22:58 subuid
-rw-r--r-- 1 root root 21 mars 23 22:24 subuid-
drwxr-xr-x 2 root root 4096 mars 23 22:53 subversion
-rw-r--r-- 1 root root 2351 mai 31 2018 sysctl.conf
drwxr-xr-x 2 root root 4096 mars 23 22:06 sysctl.d
drwxr-xr-x 5 root root 4096 mars 23 22:06 systemd
drwxr-xr-x 2 root root 4096 mars 23 22:06 terminfo
-rw-r--r-- 1 root root 13 mars 23 22:07 timezone
drwxr-xr-x 2 root root 4096 août 20 2019 tmpfiles.d
-rw-r--r-- 1 root root 1260 déc. 14 2018 ucf.conf
drwxr-xr-x 4 root root 4096 mars 23 22:08 udev
drwxr-xr-x 3 root root 4096 mars 23 22:21 ufw
drwxr-xr-x 2 root root 4096 mars 23 22:06 update-motd.d
drwxr-xr-x 2 root root 4096 mars 23 21:56 vim
-rw-r--r-- 1 root root 4942 avril 5 2019 wgetrc
drwxr-xr-x 6 root root 4096 mars 23 22:52 X11
-rw-r--r-- 1 root root 642 mars 1 2019 xattr.conf
drwxr-xr-x 4 root root 4096 mars 23 22:52 xdg
-rw-r--r-- 1 root root 289 févr. 5 2018 xinetd.conf
drwxr-xr-x 2 root root 4096 mars 23 22:59 xinetd.d
root@monitoring:/etc# cd update-motd.d/
root@monitoring:/etc/update-motd.d# ls -l
total 4
-rwxr-xr-x 1 root root 23 avril 4 2017 10-uname
root@monitoring:/etc/update-motd.d#
```

Le principe est simple, **tous les scripts se trouvant dans ce dossier sont exécutés lors de la connexion**. Il suffit de les nommer **00-xxx**, **10-xxx**, **20-xxx**, etc. pour qu'ils soient exécutés dans l'ordre de votre choix.

Pour créer mon MOTD, je me suis basé sur plusieurs sources (*je vous les donnerais toutes à la fin de l'article*). Dans l'une de ces sources, il y avait un fichier permettant d'utiliser de la couleur, ce qui est plus sympa !

Je vais donc créer ce fameux fichier, dans mon dossier `/etc/update-motd.d` via la commande `nano colors`.

On va ensuite y coller le code suivant :

```
NONE="\033[m"
WHITE="\033[1;37m"
GREEN="\033[1;32m"
RED="\033[0;32;31m"
YELLOW="\033[1;33m"
BLUE="\033[34m"
CYAN="\033[36m"
LIGHT_GREEN="\033[1;32m"
LIGHT_RED="\033[1;31m"
```

On va maintenant passer au script pour **afficher le nom du serveur**. Toujours dans le dossier `/update-motd.d` nous allons créer un fichier `00-hostname` via `nano 00-hostname` et dont le code est le suivant :

```
#!/bin/sh

. /etc/update-motd.d/colors

printf "\n"$LIGHT_RED
figlet "  "$(hostname -s)
printf $NONE
printf "\n"
```

Ensuite, on passe au second fichier, qui se nomme `10-banner`, et qui va afficher **la version Debian** qui est utilisée. On le crée donc avec la commande `nano 10-banner`, et on y ajoute le code suivant :

```
#!/bin/bash
#
```

```

# Copyright (C) 2009-2010 Canonical Ltd.
#
# Authors: Dustin Kirkland <kirkland@canonical.com>
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License along
# with this program; if not, write to the Free Software Foundation, Inc.,
# 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

. /etc/update-motd.d/colors

[ -r /etc/update-motd.d/lsb-release ] && . /etc/update-motd.d/lsb-release

if [ -z "$DISTRIB_DESCRIPTION" ] && [ -x /usr/bin/lsb_release ]; then
    # Fall back to using the very slow lsb_release utility
    DISTRIB_DESCRIPTION=$(lsb_release -s -d)
fi

re='(.*\(\)(.*)\(\).*)'
if [[ $DISTRIB_DESCRIPTION =~ $re ]]; then
    DISTRIB_DESCRIPTION=$(printf "%s%s%s%s%s" "${BASH_REMATCH[1]}" "${YELLOW}"
"${BASH_REMATCH[2]}" "${NONE}" "${BASH_REMATCH[3]}")
fi

echo -e " $DISTRIB_DESCRIPTION "(kernel "$(uname -r)")\n"

# Update the information for next time
printf "DISTRIB_DESCRIPTION=\"%s\" " "$(lsb_release -s -d)" > /etc/update-motd.d/lsb-release &

```

Maintenant, je souhaite dans mon cas, ajouter différentes informations comme **le CPU, la charge du système, la RAM libre**, etc. J'ai donc pour cela, créé un script que j'ai nommé `20-sysinfo`

dont le code est le suivant :

```
#!/bin/bash
proc=`cat /proc/cpuinfo | grep -i "^model name" | awk -F": " '{print $2}'`
memfree=`cat /proc/meminfo | grep MemFree | awk {'print $2'}`
memtotal=`cat /proc/meminfo | grep MemTotal | awk {'print $2'}`
uptime=`uptime -p`
addrip=`hostname -I | cut -d " " -f1`
# Récupérer le loadavg
read one five fifteen rest < /proc/loadavg

# Affichage des variables
printf " Processeur : $proc"
printf "\n"
printf " Charge CPU : $one (1min) / $five (5min) / $fifteen (15min)"
printf "\n"
printf " Adresse IP : $addrip"
printf "\n"
printf " RAM : $((($memfree/1024))MB libres / $((($memtotal/1024))MB"
printf "\n"
printf " Uptime : $uptime"
printf "\n"
printf "\n"
```

*Comment afficher de façon plus sympa les caractéristiques CPU. En effet, lorsque que l'ont a plusieurs coeurs,on fini par avoir plusieurs fois la ligne qui s'affichait. Ici, cela permet d'indiquer le nombre de coeurs/processeurs avant le type de CPU. Dans le script **20-sysinfo**, il faut remplacer la ligne `proc` par :*

```
proc=`(echo $(more /proc/cpuinfo | grep processor | wc -l ) "x" $(more /proc/cpuinfo | grep 'model name' | uniq |awk -F":" '{print $2}') )`
```

Maintenant, nous devons **rendre ces fichiers exécutable**, afin qu'ils soient interprétés à la connexion. Pour cela, on utilise la commande `chmod 755 00-hostname`, même chose pour les deux autres fichiers en changeant bien sûr le nom du fichier.

```
cloriou — ssh root@192.168.1.23 — 125x30
root@monitoring:/etc/update-motd.d# chmod 755 00-hostname
root@monitoring:/etc/update-motd.d# chmod 755 10-banner
root@monitoring:/etc/update-motd.d# chmod 755 20-sysinfo
root@monitoring:/etc/update-motd.d# ls -l
total 16
-rwxr-xr-x 1 root root 114 mars 30 23:25 00-hostname
-rwxr-xr-x 1 root root 1533 mars 30 23:25 10-banner
-rwxr-xr-x 1 root root 666 mars 30 23:26 20-sysinfo
-rw-r--r-- 1 root root 173 mars 30 23:24 colors
root@monitoring:/etc/update-motd.d#
```

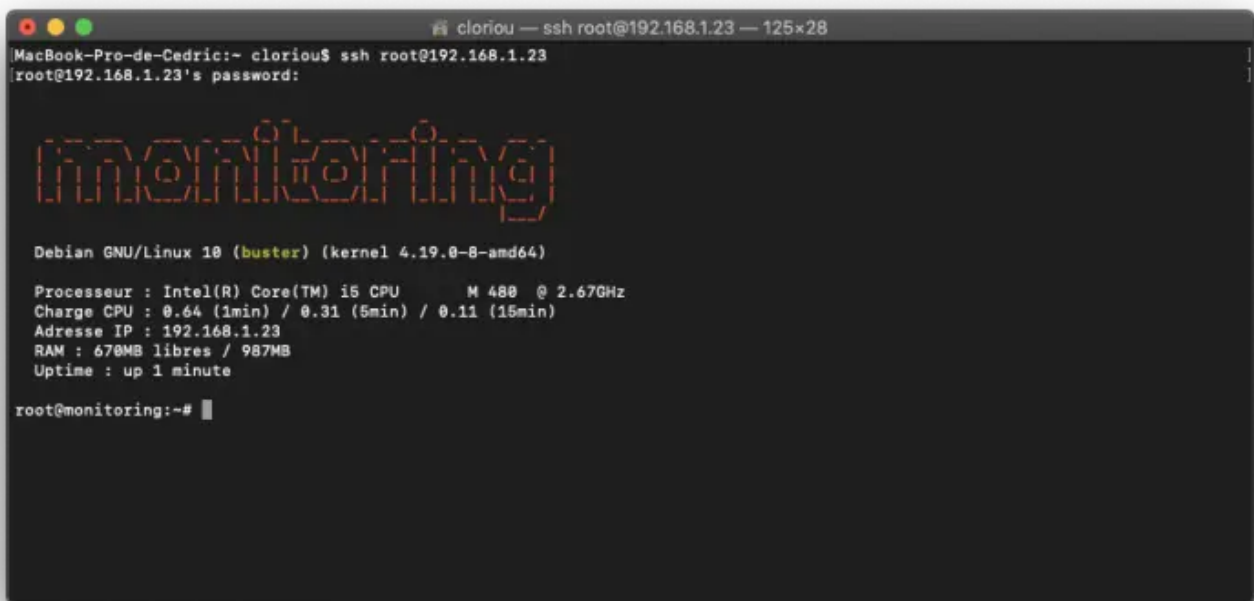
Nos 3 scripts sont maintenant exécutables.

Il nous reste à supprimer l'ancien MOTD, via la commande `rm /etc/motd`. Il faut maintenant le récréer via la commande `ln -s /var/run/motd /etc/motd`.

```
cloriou — ssh root@192.168.1.23 — 125x30
root@monitoring:/etc# rm /etc/motd
root@monitoring:/etc# ln -s /var/run/motd /etc/motd
root@monitoring:/etc#
```

Suppression de l'ancien MOTD

Et voilà, à partir de maintenant, lors de la prochaine connexion en SSH, vous devriez avoir votre MOTD, qui correspond aux différents scripts que nous avons vus juste avant.



```
MacBook-Pro-de-Cedric:~ cloriou$ ssh root@192.168.1.23
root@192.168.1.23's password:

  monitoring

Debian GNU/Linux 10 (buster) (kernel 4.19.0-8-amd64)

Processeur : Intel(R) Core(TM) i5 CPU      M 480 @ 2.67GHz
Charge CPU : 0.64 (1min) / 0.31 (5min) / 0.11 (15min)
Adresse IP : 192.168.1.23
RAM : 670MB libres / 987MB
Uptime : up 1 minute

root@monitoring:~#
```

Voilà notre beau MOTD ☐☐

Created 2025-02-04 14:36:02 UTC by Nicolas
Updated 2025-02-13 22:15:27 UTC by Nicolas