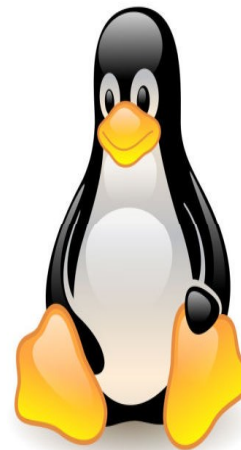


+ Module Linux

Equipe Pédagogique

S. BEN YAALA

H. SLIMANI



Administration réseau

+ Plan

- NetworkManager
- Nmcli
- Nmtui
- Configuration à l'aide des fichiers
- OpenSSH

Mise en réseau

+Introduction à NetworkManager

- Depuis Red Hat Enterprise Linux 7, le service réseau est géré par NetworkManager
- NetworkManager est un démon qui permet la configuration dynamique des périphériques réseau.
- NetworkManager utilise toujours les fichiers de configurations ifcfg
- NetworkManager supporte l'Ethernet, Bridge, Bond, Teams, Wifi, Mobile

Outil	Description
nmtui	Un outil TUI (console) pour effectuer les configuration de NetworkManager
nmcli	Un outil en ligne de commande pour configurer les interfaces
nm-connection-editor	Un outil en mode GUI (interface graphique) pour configurer les cartes

+ Les fichiers de configurations réseau

- Red Hat EL 7 utilise les fichiers de configuration pour conserver les informations TCP/IP des interfaces
 - /etc/sysconfig/network-scripts/ifcfg-eno13536

```
#cat /etc/sysconfig/network-scripts/ifcfg-eno123654
TYPE=Ethernet
NAME=eno123654
UUID=567fa98b00-cf642324-ad456456
DEVICE=eno123654
ONBOOT=YES
BOOTPROTO=dhcp
```

+ La commande nmcli

- Gérer les interfaces avec nmcli
 - Visualiser l'état de NetworkManager

```
# nmcli general status
ÉTAT    CONNECTIVITÉ WIFI-HW  WIFI  WWAN-HW  WWAN
connecté plein    activé activé activé activé
```

- Afficher les interfaces

```
# nmcli connection show

NOM                UUID                                TYPE      PÉRIPHÉRIQUE
Ethernet automatique 45dafcbd-b1b2-41ab-9e20-61a2a5aa0f58 802-3-ethernet --
eno16777728         92ae0d5d-987a-4c6f-b8c3-708e2b03368d 802-3-ethernet
eno16777728
```

+ La commande nmcli

7

- Gérer les interfaces avec nmcli

- État des périphériques

```
#nmcli device status
PÉRIPHÉRIQUE TYPE   ÉTAT   CONNEXION
eno16777728 ethernet connecté eno16777728
```

- Connecter une interface ethernet

```
#nmcli device connect eno33554968
Device 'eno33554968' successfully activated with '3de303f7-2fd6-4ed5-a95a-e9311e594a28'.
```

- Déconnecter une interface ethernet

```
#nmcli device disconnect eno33554968
Device 'eno33554968' successfully disconnected.
```

+ La commande nmcli

- Gérer les interfaces avec nmcli
 - Configurer la pile IP pour une carte

```
# nmcli connection add type ethernet con-name eno33554968 ifname eno33554968 \
ip4 192.168.1.100 gw4 192.168.1.1
Connexion « eno33554968 » (0065aa95-aa86-48af-a7d3-c431b2b78c0a) ajoutée avec
succès.
```

- Ajouter des serveurs DNS

```
# nmcli connection modify eno33554968 ipv4.dns "8.8.8.8 8.8.4.4"
```

- Afficher les détails de l'interface ethernet

```
#nmcli -p connection show eno33554968
```


+ Commande nmcli : Résumé

COMMAND	PURPOSE
nmcli dev status	Show the NetworkManager status of all network interfaces.
nmcli con show	List all connections.
nmcli con show <i>name</i>	List the current settings for the connection <i>name</i> .
nmcli con add con-name <i>name</i>	Add a new connection named <i>name</i> .
nmcli con mod <i>name</i>	Modify the connection <i>name</i> .
nmcli con reload	Reload the configuration files (useful after they have been edited by hand).
nmcli con up <i>name</i>	Activate the connection <i>name</i> .
nmcli dev dis <i>dev</i>	Deactivate and disconnect the current connection on the network interface <i>dev</i> .
nmcli con del <i>name</i>	Delete the connection <i>name</i> and its configuration file.

+ La commande ip

- Gérer les interfaces avec la commande ip
 - Afficher les interfaces

```
# ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eno16777728: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast master
    kvmbr0 state UP mode DEFAULT qlen 1000
    link/ether 00:0c:29:6e:04:3b brd ff:ff:ff:ff:ff:ff
```

- Connecter, déconnecter une carte réseau

```
#ip link set eno33554968 down

#ip link set eno33554968 up
```

+ La commande ip

- Gérer les interfaces avec la commande ip
 - Afficher la configuration ip des interfaces

```
# ip addr show
2: eno16777736: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
qlen 1000
    link/ether 00:0c:29:a2:78:c6 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.43/24 brd 192.168.1.255 scope global dynamic eno16777736
```

- Supprimer, ajouter une adresse ip

```
#ip addr del 192.168.1.43/24 dev eno16777736

#ip addr add 192.168.1.100/24 dev eno16777736
```

+ La commande ip

- Gérer les interfaces avec la commande ip
 - Afficher les routes, passerelles

```
# ip route  
default via 192.168.1.1 dev eno16777736 proto static metric 100  
192.168.1.0/24 dev eno16777736 proto kernel scope link src 192.168.1.43
```

- Supprimer, ajouter une route

```
#ip route add 192.168.2.1 dev eno16777736  
  
# ip route del 192.168.2.1 dev eno16777736  
  
# ip route add default via 192.168.2.1 dev eno16777736
```

OpenSSH

+ Introduction à OpenSSH

- OpenSSH est une suite d'outils qui permet :
 - L'ouverture des sessions distantes sécurisées
 - L'échange des fichiers entre plusieurs machines
 - Le "tunneling" de toutes les applications TCP/IP
- OpenSSH chiffre toutes les communications entre les clients et les serveurs
- OpenSSH est un projet libre maintenu par l'équipe OpenBSD et diffusé sous licence BSD

+ Introduction à OpenSSH

- OpenSSH utilise des méthodes de cryptographie à clés symétriques et asymétriques
- OpenSSH fournit une alternative à la gestion des mots de passe "classique" Linux
- OpenSSH fonctionne en mode client/serveur. Il faut donc installer les deux paquets

```
#yum -y install openssh-server openssh-clients
```

+ Introduction à OpenSSH

- OpenSSH fournit une suite de logiciels :

```
#rpm -ql openssh-clients  
/usr/bin/scp  
/usr/bin/sftp  
/usr/bin/slogin  
/usr/bin/ssh  
/usr/bin/ssh-add  
/usr/bin/ssh-agent  
/usr/bin/ssh-copy-id  
/usr/bin/ssh-keyscan
```

- Slogin (lien) ou ssh pour initier des Shells distants
- SCP pour copier des arborescences de manière sécurisée
- SFTP n'est pas du ftp sur SSH, c'est un outil qui offre le transfert de fichiers, plus complet que scp (limite de bande passante, compression)

+ Accès distant avec ssh

- Ouvrir une session distante SSH

```
root@rhel7#ssh ludo@mon_serveur_ssh  
root@rhel7#ssh ludo@192.168.0.10
```

- Ouvrir une session distante avec redirection du protocole X

```
root@rhel7#ssh -X ludo@mon_serveur_ssh  
root@rhel7#ssh -X ludo@192.168.0.10  
ludo@192.168.0.10$firefox
```

+ Exécuter des commandes à distance

- Décompresser et désarchiver un tar à distance

```
root@rhel7#ssh root@192.168.1.20 tar xvfz file.tar.gz
root@rhel7#ssh ludo@192.168.0.20 ls -l
-rw-r--r-- 1 root root 37M Sep 22 11:19 file.tar.gz
drwxrwxrwt 13 root root 4,0K Sep 18 14:13 tmp
```

+ Copier des arborescences avec scp

- Copier un répertoire sur une machine distante

```
root@rhel7#scp -r /home/ludo ludo@mon_serveur_ssh:/home/  
root@rhel7#scp -r /home/ludo ludo@192.168.0.10:/home/
```

- Copier un répertoire d'une machine distante en local

```
root@rhel7#scp -r root@192.168.0.10:/home/ /home/
```

+ Gérer vos fichiers avec sftp

- Se connecter en sftp sur une machine distante

```
root@rhel7#sftp root@192.168.0.10  
sftp>help  
Available commands:
```

- Manipuler des fichiers distants

```
sftp> mkdir from_sftp  
sftp> ls  
stp> ll  
sftp> lcd /srv/
```

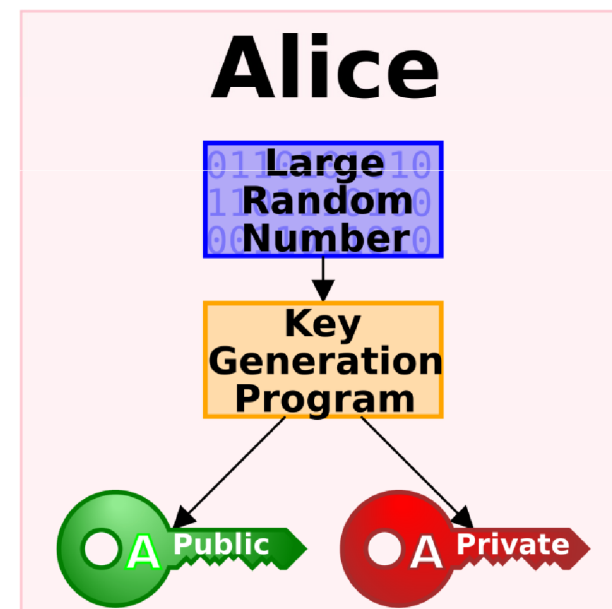
OpenSSH

La gestion des clés

+ Introduction sur la notion de clés

- Dans un système à clé asymétrique ou à clé publique
- Un utilisateur génère un porte clé contenant deux clés :

- Une clé privée : conserver jalousement
- Une clé publique : diffusé publiquement



+ Bref introduction sur la notion de clés

- Alice veut se connecter sur le serveur SSH en tant qu'utilisateur Alice
- Le serveur demande à Alice de prouver son identité
- Le serveur refuse l'authentification par mot de passe, il attend un challenge de clés
- Le serveur souhaite une authentification forte
- Alice génère alors deux clés, publique et privée
- Alice copie la clé publique sur le serveur
- Lors de la connexion d'Alice sur le serveur, ce dernier, vérifie la correspondance des clés entre clé publique et clé privée

+ Générer ses clés avec ssh

- L'utilitaire ssh-keygen permet de générer sa paire de clés
 - Générer une paire de clés de type DSA

```
ludo@rhel7$ssh-keygen -t dsa -C «clé de ludo »  
Generating public/private dsa key pair.  
Enter file in which to save the key (/home/ludo/.ssh/id_dsa):  
Created directory '/home/ludo/.ssh'.  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Passphrases do not match. Try again.  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/ludo/.ssh/id_dsa.  
Your public key has been saved in /home/ludo/.ssh/id_dsa.pub.
```

- Où sont ses clés

```
ludo@rhel7$ls -l .ssh/
```


+ Copie des clés sur les machines

- L'utilitaire ssh-copy-id permet copier la clé publique sur le serveur
 - Copie de la clé publique de ludo sur le serveur

```
ludo@rhel7$ssh-copy-id 192.168.1.20
/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are
prompted now it is to install the new keys
ludo@192.168.1.20's password:
```

Number of key(s) added: 1

Now try logging into the machine, with: "ssh '192.168.1.20'"
and check to make sure that only the key(s) you wanted were added.

- Où est la clé?

```
ludo@192.168.1.20$cat .ssh/authorized_keys
```