

# Documentation

## L'émulateur NetKit

### Présentation de Netkit

Netkit (<http://wiki.netkit.org/>) est un émulateur de réseau. Il permet d'effectuer facilement des expérimentations réseau en déployant un ensemble de machines virtuelles munies de cartes réseau type Ethernet, reliées entre elles via des domaines de diffusion virtuels. Netkit s'exécute au sein d'une machine physique que l'on nommera par la suite machine hôte. L'environnement Netkit se compose d'une console sur la machine hôte dans laquelle vous entrez les commandes et d'un terminal pour chaque machine virtuelle créée. L'environnement de travail doit être par conséquent sous X11 pour permettre l'affichage des terminaux des machines virtuelles. Netkit est un outil simple et léger qui n'utilise aucune commande ayant besoin de droits administrateur. Les commandes de Netkit commencent par la lettre `v` ou `l`. Elles servent à manipuler les machines virtuelles afin de constituer un réseau. Ces commandes sont entrées sur le shell. Les commandes de Netkit sont décrites dans le manuel d'Unix en ligne (commande `man`). Les commandes courantes de machines virtuelles sont :

- vstart** Pour démarrer une machine virtuelle équipée d'un certain nombre de cartes réseau reliées à des domaines de diffusion.
- vlist** Pour prendre connaissance des machines virtuelles actuellement actives.
- vhalt** Pour arrêter proprement une machine virtuelle (sauvegarde du filesystem dans le fichier `.disk`).
- vcrash** Pour arrêter brutalement une machine virtuelle.
- vclean** commande "sous panique", pour supprimer tous les processus Netkit ainsi que les fichiers de la machine virtuelle.
- vconfig** Pour ajouter à la volée une carte réseau à une machine virtuelle.

### Création d'un réseau

La création d'un réseau va consister à créer des machines et de préciser leur connectivité physique. Une machine virtuelle peut être aussi bien un hôte ou un routeur. Une machine avec 2 interfaces réseaux devient un routeur. En fait, un routeur est une machine avec une table de routage correctement configurée et exécutant éventuellement un logiciel ou daemon de routage dynamique.

La création d'une machine s'effectue par la commande `vstart <nom_machine>`. Ceci produit la création d'un nouveau terminal qui est la console de la machine créée. L'utilisateur de ce terminal a les droits de root. Le répertoire `$HOME` de la machine hôte est accessible via `/hosthome`. Ce répertoire sert à faire les échanges entre la machine virtuelle et la machine hôte.

La création d'une machine avec une carte réseau, s'effectue en indiquant la carte réseau et le nom du domaine de collision :

```
vstart -eth0=<domaine> <nom_machine>
```

Par exemple la création d'une machine PC1 sur le réseau Ethernet A s'effectue par la commande `vstart -eth0=A PC1`.

La configuration de l'interface s'effectue de manière classique à tous systèmes Unix à l'aide de la commande `ifconfig`. Des routes peuvent s'ajouter avec la commande `route`. De manière générale, les outils réseaux standards sont disponibles comme `tcpdump`, `ping` ou `netstat`.

Une machine virtuelle s'arrête soit depuis la console (commande `vhalt`), soit dans le terminal de la machine virtuelle par la commande `halt`.

### Sauvegarde d'un réseau : notion de Lab

Un Lab est un ensemble de scripts pour le lancement automatique de toutes les machines virtuelles constituant un réseau, et pour leur configuration automatique. Les deux principaux avantages par rapport à l'utilisation de commandes en `v` sont :

- Un démarrage et une configuration plus rapide de l'ensemble des machines virtuelles.
  - La possibilité de sauvegarder une topologie de réseau et sa configuration.
- La création d'un Lab s'effectue dans un répertoire distinct. Ce répertoire contiendra :
- un fichier `lab.conf` qui contiendra les créations des machines. Des balises spécifiques en début de fichier permettent d'ajouter des informations sur le contenu du Lab.
  - un répertoire `<nom_machine>` par machine qui contiendra les commandes et les fichiers propres à la machine. Les éléments dans ce répertoire sont montés à partir de la racine de la machine virtuelle (uniquement si il n'y a pas de fichier `.disk`)
  - un fichier de configuration par machine appelé `<nom_machine>.startup`. Ce fichier est un script qui est exécuté par la machine virtuelle à son démarrage.

Il faut veiller à garder le même nom pour une machine dans les fichiers et les répertoires.

Les commandes de Lab commencent par la lettre `l` :

**lstart** démarrage du Lab du répertoire courant.

**lhalt** arrêt ordonné des machines du Lab en cours d'exécution.

**lcrash** arrêt brutal de toutes les machines du Lab en cours d'exécution.

**linfo** présente le Lab du répertoire courant sans démarrer le Lab.

**lclean** supprime tous les fichiers temporaires créés par le Lab.

Le contrôle de l'exécution du Lab s'effectue par la syntaxe `lcommand -d <lab_directory> [machine ...]` avec `lab_directory` le répertoire contenant le Lab et optionnellement la liste des machines auxquelles s'appliquent la commande. `lcommand` peut être `lstart`, `lhalt` ou `lcrash`. Comme pour les commandes en `v`, les commandes de Lab sont décrites dans le manuel en ligne d'Unix. Le répertoire du Lab est accessible depuis les machines virtuelles via [/hostlab](#).

## Préparer un Lab

Le fichier `lab.conf` décrit les virtuelles machines et la topologie du réseau qui interconnecte les machines virtuelles. Les liens sont des domaines de collision. Le fichier `lab.conf` commence par ces balises de description qui s'affiche au démarrage du Lab :

```
LAB_DESCRIPTION="une description du lab."
LAB_VERSION=1.2
LAB_AUTHOR="auteur 1"
LAB_EMAIL=name@domaine.org
LAB_WEB=http://www.unsiteweb.org/
```

Le fichier `lab.conf`, se termine par la description de la connexion des interfaces réseau de la manière suivante :

```
machine[x]=networkname
```

avec `x` indique le numéro d'interface Ethernet et `networkname` le nom. Par exemple, si l'interface `eth0` des machines `PC1` et `PC2` sont connectées sur le même lien, ceci se déclare de cette manière :

```
PC1[0]=net1
PC2[0]=net1
```

NetKit démarre une machine virtuelle pour chaque sous répertoire contenu dans le répertoire du Lab. Le contenu du sous répertoire est monté sous la racine (`/`) du système de fichiers de la machine virtuelle. Ceci se produit au produit tant qu'il n'y a pas l'image disque (`.disk`) de la machine virtuelle.

Enfin, pour que le Lab soit complet, il convient d'indiquer la configuration de chaque machine virtuelle. Ceci se décrit par les scripts shells dont le nom est suffixé par `.startup`. Une machine virtuelle au démarrage exécute les scripts : `shared.startup`, `VmName.startup`. Le fichier `shared.startup` s'applique à toutes les machines virtuelles. Le script shell propre à la machine virtuelle et le script commun sont exécutés à l'intérieur de la machine virtuelle.

## Démarrage d'un environnement avec Netkit

1) Si votre poste de travail n'a pas l'installation Netkit. Le plus simple (rapide) est de démarrer une machine Linux dans l'environnement de virtualisation `virtualbox`.

Dans `virtualbox` :

- Créer une machine Linux de type Linux 2.6. Un disque dur n'est pas nécessaire. Prévoir 512 Mo de mémoire vive.
  - Faire fonctionner la carte réseau **en pont** (et non en NAT)
  - Faire démarrer cette machine Linux à partir du live CD ISO. Demander à l'enseignant l'image iso ou récupérer la dernière version [http://wiki.netkit.org/index.php/Download\\_Official](http://wiki.netkit.org/index.php/Download_Official) et en miroir à la Réunion <http://lim.univ-reunion.fr/staff/panelli/ressources/Netkit-2.8-K2.8-F5.2.iso>. Attention : la taille de l'image est conséquente (2.6 Go).
  - Au boot de la machine Linux, entrer : `knoppix lang=fr` afin d'avoir une disposition du clavier et des paramètres en français, ainsi qu'un démarrage en mode console moins consommateur en ressources.
- 2) Une fois, la machine Linux démarrée, nous allons l'utiliser comme un serveur de l'application Netkit, c'est à dire de votre poste de travail vous ouvrirez un terminal distant sur ce serveur. Ce terminal sera nommé par la suite console. Pour cela, vous allez permettre la connexion à distance sur le serveur :
- Ouvrir un terminal dans l'environnement du serveur.
  - Associer un mot de passe à l'utilisateur par défaut knoppix : `sudo passwd knoppix`.
  - Démarrer le serveur de connexion à distance : `sudo /etc/init.d/ssh start`
  - Récupérer l'adresse IP du serveur à l'aide de la commande `ifconfig -a`
- 3) Ensuite, vous pouvez quitter l'environnement du serveur et retourner dans l'environnement de votre poste de travail afin de mettre en place la console :
- Démarrer X11.
  - Ouvrir un terminal (application `xterm`).
  - Démarrer une session distante à l'aide de la commande `ssh -X knoppix@addripduserveur`
  - Démarrer le shell bash par la commande `bash`.
- Désormais, vous avez la possibilité de faire des copies avec la souris dans l'environnement de votre machine. Votre environnement de travail est terminé, vous êtes prêt à commencer le TP.

## Mise en place d'un Lab

- Ouvrir une session sur le serveur Netkit qui vous a été alloué. L'ouverture d'une session doit être se faire avec la commande `ssh -X` de manière à autoriser l'affichage de nouveaux terminaux sur votre écran. L'environnement graphique de votre station de travail doit être X11.
  - Récupérer de la page du module le fichier archive (tar.gz) contenant le Lab à utiliser et installer le sur votre serveur Netkit (appelé par la suite machine hébergeuse). Ceci peut se faire soit par la commande `wget` depuis la machine hébergeuse ou depuis votre poste de travail et transfert du fichier par la commande de copie sur site distant (`scp`).
  - Extraire sur la machine hébergeuse le contenu de l'archive par la commande `tar`.
  - Démarrer le Lab qui créera un réseau virtuel correspond à la topologie présentée dans le sujet de TP.
- Typiquement les commandes de mise en place d'un Lab sont :

```
ssh -X user1-netkit@10.230.135.xxx
```

Puis téléchargement depuis un navigateur sur votre poste de travail en recopie sur la machine hébergeuse :

```
scp lab.tar.gz user1-netkit@10.230.135.xxx:.
```

enfin sur le terminal de la machine hébergeuse 10.230.135.xxx :

```
tar xvzf lab.tar.gz
```

Vous êtes maintenant prêt à commencer le sujet du TP.

## screen : Création de terminaux pour une machine virtuelle

Afin d'avoir plusieurs terminaux dans une console d'une machine virtuelle, il faut utiliser la commande `screen`. Celle-ci crée des terminaux virtuels dans la console. Ensuite, avec des raccourcis clavier, il devient très aisé de gérer les terminaux virtuels. Les principaux raccourcis sont :

- CTRL A c pour créer un terminal virtuel
- CTRL A n pour passer au terminal virtuel suivant
- CTRL A " pour sélectionner un terminal virtuel
- CTRL A k pour supprimer un terminal virtuel

Pour en savoir plus sur les raccourcis de la commande `screen`, il faut se référer au manuel : `man screen`.

## tcpdump : Capture de trafic

La commande `tcpdump` est un outil de capture de paquets très puissant. Pour diminuer le "bruit", `tcpdump` utilise des expressions pour filtrer (conserver) les paquets :

- `icmp6` pour filtrer le trafic ICMPv6
- `ip6` pour filtrer tout le trafic IPv6
- `tcp` pour filtrer tout le trafic TCP

Enfin, la commande `tcpdump` comporte des options pouvant s'avérer utiles pour afficher plus d'informations, citons :

- `-XX` pour l'affichage du contenu de la trame
- `-i` pour indiquer l'interface où doit se faire l'écoute
- `-n` affichage des adresses (à la place du nom)
- `-s 512` avec ici 512 nombre maximum d'octets capturés par paquet
- `-A` pour l'affichage du contenu des paquets (en ASCII)
- `-e` pour l'affichage des adresses MAC de la trame
- `-t` sans l'affichage de l'horodatage des traces
- `-vv` pour une interprétation détaillée des champs des en-têtes.

Par exemple, la capture des paquets IPv6 se spécifie de la manière suivante :

```
tcpdump -i eth0 -n -t -s 512 -X -vv ip6
```

Pour en savoir plus, consulter le manuel en ligne de `tcpdump` et éventuellement `pcap_filter`.

## vim : Edition de texte

Le passage du mode édition au mode commande s'effectue par la touche `[ESC]`.

### L'ouverture de fichier

Il est toujours mieux de donner un nom au fichier dès l'appel de `vi`.

- `vi nom_fic` : ouvre le fichier `nom_fic` en positionnant le curseur au début du fichier, crée le fichier sinon.
- `vi + nom_fic` : ouvre le fichier `nom_fic` en positionnant le curseur à la fin du fichier, crée le fichier sinon.

### Les commandes d'insertion

(pour passer en mode édition)

- `i` insérer le texte avant le caractère courant
- `a` insérer le texte après le caractère courant
- `I` insérer le texte au début de la ligne courante
- `A` insérer le texte à la fin de la ligne courante

### Les commandes de suppression

- `dd` coupe la ligne courante
- `n dd` coupe `n` lignes à partir de la ligne courante
- `D` supprime jusqu'à la fin de la ligne
- `x` supprime le caractère courant
- `X` détruit le caractère précédent

### Les commandes de déplacement

- `w` déplace le curseur au début du prochain mot
- `$` déplace le curseur à la fin de la ligne courante
- `-` déplace le curseur au début de la ligne précédente
- `+` déplace le curseur au début de la ligne suivante
- `Ctrl-F` avance d'une page
- `Ctrl-B` recule d'une page

### Copier / coller

- `yy` copie la ligne courante
- `p` colle le contenu du tampon
- `P` colle le contenu du tampon avant le curseur

### Les commandes de sauvegarde et de sortie

- `:w` sauvegarde sans sortir
- `:q!` sort sans sauvegarde
- `:wq` sort avec sauvegarde