SAÉ B - Déploiement d'une application

Semaine 1 - Présentation générale et mise en place

1 Scénario

Une entreprise de développement logiciel souhaite améliorer les modes de communication à la fois des collaborateurs entre eux, mais également entre les collaborateurs et les clients de l'entreprise (notamment à des fins de support).

Pour cela, elle souhaite mettre en place un outil répondant aux contraintes suivantes:

- le mode de communication proposé doit supporter une conversation synchrone et asynchrone ;
- une conversation peut avoir lieu entre deux personnes (communication 1 vers 1), en petit groupe (discussion entre développeurs) ou en audience plus large (*forum* de discussion) ;
- les communications doivent être chiffrées de bout en bout afin d'assurer la confidentialité des échanges ;
- le service de discussion doit pouvoir être accessible en utilisant une application web, une application lourde ou une application mobile.

Après une étude des solutions existantes, le groupe de travail chargé de choisir la solution à déployer vous demande de mettre en place une solution basée sur le standard Matrix qui répond à tous les critères demandés.

1.1 Matrix

Matrix est un projet open source qui publie un **standard** pour une communication sécurisée, distribuée et temps réelle.

Un standard est l'ensemble des documentations de référence permettant à plusieurs implémentations respectant celui-ci de se comprendre. En d'autres termes, il peut y avoir plusieurs logiciels qui implémentent le même standard. Ces logiciels vont pouvoir inter-opérer car il se comprennent, parle la même langue, grâce au standard.

Le standard matrix défini la liste et le format de message permettant:

- à un serveur de discussion (*back-end*) de dialoguer avec un client (*front-end*). Ainsi, plusieurs client différents (web, mobile, application lourde, client simple en mode texte…) peuvent communiquer avec le serveur, laissant le choix de l'interface à l'utilisateur ;
- à plusieurs serveurs de discussion de *fédérer* leurs utilisateurs et leur canaux. Ainsi, les utilisateurs de différents fournisseurs pourront communiquer entre eux.

1.2 Implémentation de référence

En plus du standard, la fondation matrix propose une implémentation de référence du standard: Synapse, un serveur écrit en python. C'est cette version du serveur que vous allez devoir mettre en place.

Pour la partie cliente, c'est la solution Element Web que vous devrez proposer.

Pour réaliser cette SAÉ, vous allez travailler sur des machines virtuelles. Contrairement à ce que vous avez utilisé au département jusqu'à présent, il n'est pas question d'utiliser des machines avec interface graphique, mais de travailler à distance

sur les serveurs que vous allez administrer.

2 Consigne importante

Lors de cette SAÉ, vous devrez exécuter un certains nombre d'actions d'administration système, dont beaucoup devront être répétées et refaites plusieurs fois. Le plus long, en administration système, n'est pas de **faire** les choses mais de trouver **comment** les faire. Il est donc très important de noter vos actions et le résultat de vos recherches.

> **Documentez vos actions.** Pour chaque étape, pensez à rédiger des procédures indiquant les actions à effectuer pour réaliser l'étape. Vous pourrez plus facilement refaire ces actions par la suite.

3 Un peu de vocabulaire et de convention

Afin de bien se comprendre tout au long de cette SAÉ, voici le vocabulaire qui sera employé dans les sujets :

- **machine physique** : la machine qui est devant vous. Celle sur laquelle vous êtes connecté.e.s dans la salle TP ;
- **machine de virtualisation** : la machine qui fera fonctionner vos machines virtuelles. Vous vous y connecterez à distance depuis la *machine physique*. Il s'agit d'un serveur dédié, situé dans la salle serveur du département ;
- **machine virtuelle** : une machine "émulée" par un logiciel de virtualisation. Les *machines virtuelles* seront accessibles *via* la *machine de virtualisation* ;
- **réseau physique** : il s'agit du réseau des salles de TP de l'IUT. Les *machines physiques* et *de virtualisation* y sont connectées directement via une de leur interface réseau (`eth0` pour les salles de TP, `ens10f0` pour le serveur) ;
- **réseau virtuel principal** : il s'agit d'un réseau virtuel, reliant la *machine de virtualisation* aux *machines virtuelles* s'y exécutant. Sur la *machine de virtualisation*, ce réseau correspond à l'interface `lxcbr0`.

Pendant cette SAÉ, les sujets contiendront souvent des commandes à exécuter, avec parfois le résultat attendu de la commande, comme par exemple:

```
$ ls
README.html       README.md         network-sae.svg   sujet1.html       sujet1.md
```

Le prompt de votre invite de commande est indiqué par le caractère `$`, il ne faudra donc pas le saisir. Le prompt pourra également être indiqué par le caractère `#` à la place de `$`. Dans ce cas, il faudra exécuter la commande en tant qu'utilisateur `root`. Par exemple:

```
# adduser toto
```

Vous ne pourrez donc exécuter ces commandes que sur vos *machines virtuelles*.

Si nécessaire, le prompt indiquera également précisemment l'utilisateur et la machine sur laquelle la commande doit s'exécuter sous la forme `utilisateur@machine`. Concernant la machine, il s'agira soit:

- de `phys` pour représenter la machine physique ;
- de `virtu` pour représenter la machine de virtualisation ;
- de `vm` pour représenter la machine virtuelle dans le cas où il n'y a pas d'ambiguïté ;
- d'un nom de machine virtuelle s'il est nécessaire de le préciser.

Dans le cas du nom d'utilisateur, le seul cas particulier sera `login`. Ce cas se produira pour les commandes à saisir sur la machine physique ou la machine de virtualisation et il voudra simplement dire que vous devez exécuter la commande en utilisant votre compte étudiant.

Par exemple, si la commande `ls` est à exécuter sur la machine physique:

```
login@phys$ ls
```

Si la commande `id` est à exécuter en tant que l'utilisateur `root` de la machine virtuelle nommée `matrix`:

```
root@matrix# id
```

La convention `utilisateur@machine` pourra également être utilisée pour désigner les paramètres de connexion de la commande `ssh`. Les mêmes noms particuliers s'appliqueront.

4 Connexion à distance

Afin de se connecter facilement sur des machines à distance, nous allons commencer par prendre en main la commande `ssh`.

4.1 Première connexion à la machine de virtualisation

La machine de virtualisation est la même pour tout le monde. Il s'agit de la machine

```
dattier.iutinfo.fr
```

Retenez bien ce nom de machine, vous devrez l'utiliser pour toute la SAÉ.

Afin de vous connecter sur la machine de virtualisation utilisez la commande suivante :

```
login@phys$ ssh dattier.iutinfo.fr
```

Une fois la commande lancée, si vous ne vous étiez jamais connecté en SSH sur la machine indiquée, vous aurez un affichage similaire à celui-ci, ne validez rien tant que vous n'aurez pas effectué les vérifications indiquées après:

```
The authenticity of host 'dattier.iutinfo.fr (172.18.48.20)' can't be established.
ED25519 key fingerprint is SHA256:QynRpdPucTVcwhMrD3824pqUviVFCgPwxwhkDyGyVSg.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

Ce message vous indique que votre client SSH ne peut vérifier l'identité du serveur SSH. Il vous demande si vous souhaitez faire **confiance** à ce serveur.

Pour vérifier si le serveur SSH est bien celui auquel vous pensez vous connecter, il faut vérifier **l'empreinte** de sa clé.

Les empreintes des clés du serveur de virtualisation sont les suivantes (une clé différente par algorithme supporté par le serveur):

	Fingerprint	Machine	Algo
3072	SHA256:mPCb5nJD8F6YOg/aDCRjqF/ZW3Ei9iLpzXw5UDCIH8g	dattier.iut-infobio.priv.univ-lille1.fr	(RSA)
256	SHA256:+XNYpzmoYKDnwaB1xqCA2Yu7mBZEK5zvtfXYw1zDO1Y	dattier.iut-infobio.priv.univ-lille1.fr	(ECDSA)
256	SHA256:QynRpdPucTVcwhMrD3824pqUviVFCgPwxwhkDyGyVSg	dattier.iut-infobio.priv.univ-lille1.fr	(ED25519)

Utilisez cette liste pour vérifier l'empreinte de la clé et **uniquement** si celle-ci correspond, répondez yes à la question de votre client SSH. Votre client SSH ajoutera alors le serveur dans le fichier `$HOME/.ssh/known_hosts` pour indiquer que vous lui faites confiance.

En condition réelle, c'est l'administrateur du serveur ou votre hébergeur qui vous donnera cette empreinte.

Après avoir répondu yes, saisissez votre mot de passe quand il vous est demandé. Vous obtenez alors un shell qui vous permet d'exécuter des commandes sur la machine de virtualisation.

4.2 Faciliter la connexion

Devoir saisir son mot de passe à chaque connexion peut vite s'avérer pénible, surtout si on doit le faire souvent. SSH permet de s'authentifier autrement qu'avec un simple mot de passe.

4.2.1 Un peu de théorie (mais vraiment un peu)

Cet autre mode d'authentification passe par l'utilisation d'un mécanisme cryptographique plus sécurisé qu'un simple mot de passe[1]. Ce mécanisme se base sur de la cryptographie asymétrique (ou à clé publique). Sans rentrer dans les détails, vous allez créer une **paire de clés** constituée d'une clé **privée** et d'une clé **publique**. Comme leurs noms l'indiquent, la clé *publique* peut être connue par tout le monde, la clé *privée* doit être connue uniquement par vous.

Le principe d'utilisation pour l'authentification SSH est le suivant: vous donnez au serveur SSH votre clé *publique* et, au moment de la connexion, un *challenge cryptographique* permettra au serveur de vérifier que vous possédez bien la clé *privée* associée à la clé *publique* sans que votre clé privée ne soit jamais diffusée en dehors de votre machine.

4.2.2 Mise en pratique

Pour mettre en place ce que nous venons de décrire, nous devons réaliser ces étapes:

1. fabriquer une paire de clés ;
2. transmettre la clé publique au serveur.

4.2.2.1 Fabriquer une paire de clés

Si vous avez déjà une paire de clés, vous pouvez passer cette étape.

Pour fabriquer une paire de clé, vous allez utiliser la commande `ssh-keygen` avec les paramètres par défaut[2]. Lors de l'utilisation, la commande vous demandera deux choses:

1. un nom de fichier : vous pouvez laisser le nom de fichier par défaut. Notez le. Il est bon de le connaître ;
2. une *passphrase*: c'est un mot de passe qui permet de chiffrer le fichier contenant votre clé privé. Il est très important d'utiliser un mot de passe pertinent. Ainsi, si on vous vole le fichier, le voleur ne pourra pas se servir de votre clé.

Voici un exemple d'exécution de la commande:

```
login@phys$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/infoetu/login/.ssh/id_rsa):
Created directory '/home/infoetu/login/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/infoetu/login/.ssh/id_rsa
Your public key has been saved in /home/infoetu/login/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:fMB2KbRIhY0++6x3VcQ8Zmzz9taiJdE79RREr7oPlLw login@phys
The key's randomart image is:
```

```
+---[RSA 3072]----+
|      .=o    +oo |
|     .o+.. .  @..|
|     .. * o  * +o|
|      oo + ...o.=|
|       oS . +o.=+|
|      .  . .oo= =|
|       o   .E+ + |
|        + . .o   |
|       .o .  ...  |
+----[SHA256]-----+
```

Une fois votre clé en main, il reste à la diffuser sur le serveur.

4.2.2.2 Transmettre la clé publique au serveur

Quand un utilisateur tente de se connecter à un serveur SSH, celui-ci consulte le fichier `$HOME/.ssh/authorized_keys`[3] à la recherche de clés publiques autorisées à se connecter pour l'utilisateur.

Ce fichier, dont un exemple est donné ci-dessous, est constitué d'une clé publique par ligne.

```
ssh-rsa AAAAB3NzaC1 [...] +1ts5x6ZXE= login@phys
ssh-rsa AAAAB3NzaC1 [...] o/6cLqTmM8= commentaire
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIGHaJOp3Vx34PVVEt6ZzTii60Sd3Hl5CZTiqgx37f8hp user@machine
```

Les deux premières lignes on été tronquée pour des raisons de lisibilité. Le format d'une ligne est

```
[options] type-clé clé commentaire
```

Si vous regardez le contenu du fichier `$HOME/.ssh/id_rsa.pub` (cette fois côté client), vous verrez qu'il respecte le même format, à la différence qu'il ne contient qu'une seule clé (et donc qu'une seule ligne).

Pour transmettre votre clé au serveur, il suffit donc d'ajouter le contenu du fichier `id_rsa.pub` (de votre machine physique) au fichier `authorized_keys` (de votre machine de virtualisation).

Pour se faire, deux solutions:

1. le faire manuellement (on vous laisse réfléchir à comment s'y prendre) ;
2. utiliser la commande `ssh-copy-id` qui est faite pour ça.

Lisez le manuel de `ssh-copy-id` et transmettez votre clé publique sur la machine de virtualisation.

> Retenez bien toutes ces manipulations, vous devrez le refaire plusieurs fois avec vos différentes machines virtuelles.

Retentez maintenant la connexion depuis votre machine physique à votre machine de virtualisation. Voici à gros grain les étapes de la connexion:

1. le client propose au serveur des identifiants de clé avec lesquels il peut s'authentifier (par défaut, la clé présente dans les fichiers `id_dsa`, `id_ecdsa`, `id_ed25519` et `id_rsa`) ;
2. si la clé publique est présente dans le fichier `authorized_keys`, le serveur génère un challenge aléatoire et le transmet au client ;
3. le client renvoie une signature du challenge qu'il effectue avec sa clé privée ;
4. le serveur vérifie la signature à l'aide de la clé publique ;
5. le client est identifié, la connexion est autorisée.

Lors de l'étape 3, le client SSH a besoin d'accéder au fichier `id_rsa` qui contient votre clé privée. Or, rappelez vous, ce fichier est **chiffré** avec la passphrase que vous avez utilisée à la création de la clé.

Le client doit donc vous demander cette passphrase. En général, il ne le fait pas directement, mais demande à utiliser votre clé à travers un **agent ssh**. Cet agent est un processus qui tourne en arrière plan pendant toute la durée de votre session et va retenir, en mémoire vive, une version déchiffrée de votre clé privée pour pouvoir l'utiliser. Cet agent est généralement lancé automatiquement par votre environnement de bureau (mate, gnome, kde…).

A la première utilisation de votre clé, l'agent vous demande la passphrase pour déchiffrer le fichier `id_rsa`. Ensuite, lors d'utilisations successives pendant votre session, il n'aura plus à le faire. Vous pourrez donc vous connecter plusieurs fois sans avoir à saisir un mot de passe ou une passphrase.

5 Créer et gérer des machines virtuelles

Attention vous devez exécuter toutes les commandes de cette section sur votre **machine de virtualisation**.

Nous vous fournissons un script nommé `vmiut` qui vous permet de gérer vos machines virtuelles. Ce script n'est pas situé dans un chemin standard du système, vous ne pouvez donc pas l'exécuter directement.

Lancer la commande `vmiut` sans paramètre affichera un message d'aide.

5.1 Création d'une machine virtuelle

Pour créer votre machine virtuelle, utilisez la commande suivante:

```
$ vmiut creer matrix
```

La création de la machine va prendre un peu de temps. Si celle-ci se passe bien, vous aurez une sortie similaire à :

```
Virtual machine 'matrix' is created and registered.
UUID: 903447fe-d6ca-4f2f-9272-3f465a026540
Settings file: '/usr/local/virtual_machine/infoetu/login/matrix/matrix.vbox'
0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%
Clone medium created in format 'VMDK'. UUID: 99d2712d-e40d-4661-995f-b1147131414b
# Paramètres vmiut
MACHINE=matrix
VBOXES=/usr/local/virtual_machine/infoetu/login
RESEAU=vboxtap0
MEMOIRE=1024
VRDEPORT=5000-5050
MODELE=/home/public/vm/disque-5Go-bullseye.vdi

# Paramètres VirtualBox
name=matrix
UUID=903447fe-d6ca-4f2f-9272-3f465a026540
path=/usr/local/virtual_machine/infoetu/login/matrix
memory=1024
etat=poweroff
vrdeport=-1
mac=08:00:27:ba:9a:8d
```

Votre machine est créée, mais elle n'est pas encore démarrée. Lancez la commande:

```
$ vmiut lister
```

Vous devez constater qu'une machine nommée `matrix` apparaît dans la liste. L'identifiant codé par une succession de chiffres hexadécimaux est un identifiant unique, généré aléatoirement. Il ne sera pas le même pour vous.

```
"matrix" {903447fe-d6ca-4f2f-9272-3f465a026540}
```

Si, au cours de la SAÉ, le résultat de la commande `vmiut lister` ressemble à

```
"<inaccessible>" {903447fe-d6ca-4f2f-9272-3f465a026540}
```

alors soit:

1. vous ne vous trouvez pas sur la machine de virtualisation ;
2. les fichiers de votre machine virtuelle ont été effacés.

5.2 Démarrage de la machine virtuelle

Pour démarrer votre VM, lancez la commande:

```
$ vmiut demarrer matrix
Waiting for VM "matrix" to power on...
VM "matrix" has been successfully started.
```

5.3 Arrêt et suppression de la machine virtuelle

Pour arrêter et supprimer la machine virtuelle:

```
$ vmiut arreter matrix
```

puis

```
$ vmiut supprimer matrix
```

Attention: à la fin d'une séance, pensez à **arrêter** vos machines virtuelles pour ne pas qu'elles utilisent des ressources inutilement. Elles ne seront pas probablement pas supprimées et vous pourrez les redémarrer à la séance suivante.

Vous pouvez maintenant re-créer une machine virtuelle nommée `matrix` et la démarrer.

5.4 Obtenir des informations sur la machine virtuelle

Une fois la VM démarrée, vous pouvez obtenir des informations sur elle en utilisant:

```
$ vmiut info
```

Dans les informations affichées, vous devriez observer la ligne suivante :

```
ip-possible=
```

ou

```
ip-possible=10.42.xx.yy
```

Dans le premier cas, la VM est démarrée mais n'a pas encore obtenu d'adresse IP, il faudra attendre un peu avant de relancer la commande `vmiut info` et obtenir une ligne correspondante au deuxième cas (xx sera remplacé par un nombre).

5.5 Quelques informations sur le réseau et la VM

Comme vous pouvez le constater, votre machine virtuelle a obtenu une adresse IP dans le réseau 10.42.0.0/16. Il s'agit du *réseau virtuel principal*. Dans ce réseau, on a:

Machine	Adresse
Machine de virtualisation	10.42.0.1
Routeur, DNS	10.42.0.1
Adresses dynamiques (attribuées automatiquement)	10.42.1.0-10.42.99.255

La machine virtuelle a été créée à partir d'un modèle. Voici les caractéristiques du modèle:

- Distribution: Debian GNU/Linux 12 (bookworm)
- Utilisateur standard: `user`, mot de passe: `user`
- Administrateur: `root`, mot de passe: `root`
- empreinte des clés SSH serveur:

```
SHA256:C+oy3vfY9fGCAmwzHCUADu75cFUiOGpp7Y5/zOLJIB4 (RSA)
SHA256:jq4fycPE9bXnOsphH/mkP0ue3KLQP4WEFmXDuYCpLf0 (ECDSA)
SHA256:5CmKzEIqY6qbp0w+sXfHe7/jUDjsPtySwcioO5+BeVo (ED25519)
```

5.6 Utilisation de la machine virtuelle

Pour utiliser vos machines virtuelles, deux solutions:

1. utiliser une console virtuelle: simule un clavier et un écran qui serait connectés physiquement à la machine virtuelle
2. se connecter en SSH.

5.6.1 Console virtuelle

Pour utiliser la console virtuelle, il faut lancer la commande

```
$ vmiut console matrix
```

Vous aurez le message d'erreur suivant:

```
ERROR: Failed to open display:
```

Ceci est du au fait que la console virtuelle est une application graphique mais que vous êtes connectés à distance sur la machine de virtualisation. L'application graphique ne peut donc pas afficher sa fenêtre.

Pour palier à ce problème, nous allons utiliser une fonctionnalité de SSH qui permet de *rediriger* une application graphique par la connexion SSH. Pour cela, déconnectez vous de la machine de virtualisation et reconnectez vous avec la commande suivante (en remplaçant `virtu` par le nom de la machine de virtualisation) :

```
login@phys$ ssh -X virtu
```

l'option `-X` de SSH permet d'effectuer la redirection graphique. Vous pouvez maintenant retenter la commande `vmiut console matrix`.

Connectez vous en tant que `root` et utilisez les commandes `ip addr show` et `ip route show` pour constater que les paramètres de réseau correspondent bien à ce qui est attendu.

5.6.2 Connexion SSH

Pour vous connecter en ssh, il suffit d'utiliser la commande (en remplaçant xx par le nombre correspondant, obtenu à l'aide de `vmiut info` ou de la sortie de `ip addr show` exécuté dans la console virtuelle):

```
$ ssh user@10.42.xx.yy
```

5.6.3 Changement de la configuration réseau

Votre machine virtuelle sera un serveur, hébergeant un service. Il est préférable qu'elle ait donc toujours la même adresse IP. Nous vous avons attribué une plage d'adresse IP dans le réseau 10.42.0.0/16.

> Consulter le fichier disponible sur moodle pour connaître la plage d'adresse qui vous est attribuée

Dans ce fichier, la plage sera indiquée sous la forme `10.42.xx.1-254`. Retenez bien la valeur xx. Si les consignes demandent d'utiliser l'adresse ip `.1`, alors il faudra utiliser l'adresse complète `10.42.xx.1`.

Pour cette première machine, nous utiliserons l'adresse `.1` (et donc `10.42.xx.1`)

Depuis la console virtuelle[4], coupez l'interface réseau à l'aide de la commande (remplacez `enp0s3` par le nom de l'interface ayant

l'adresse en `10.42.xx.yy` si nécessaire)

```
root@vm# ifdown enp0s3
```

Modifiez les fichiers `/etc/network/interfaces` et `/etc/resolv.conf` de façon à ce que la VM ait l'adresse statique `10.42.xx.1` et qu'elle utilise le routeur `10.42.0.1` et serveur DNS `10.42.0.1`.

Aidez vous des pages de manuel `interfaces(5)` et `resolv.conf(5)` pour la syntaxe.

Vous pouvez redémarrer l'interface réseau à l'aide de la commande:

```
root@vm# ifup enp0s3
```

Utilisez les commandes `ip addr show`, `ip route show` et `host www.univ-lille.fr` pour vérifier, respectivement, l'adresse de l'interface, l'adresse du routeur et si la configuration DNS fonctionne correctement.

Utilisez la commande suivante pour redémarrer la machine virtuelle et vérifier que la configuration réseau est bien persistante

```
root@vm# reboot
```

6 Configurer et mettre à jour la machine virtuelle

6.1 Connexion `root` et SSH

Essayez de vous connecter à la machine virtuelle en SSH sur le compte `root`.

1. Quelle commande avez vous utilisée ?
2. Que se passe-t'il ?
3. Pourquoi ?

Reconnectez vous, cette fois avec le compte `user` et lisez la page de manuel `su(1)`

1. Quelle est la signification de l'option `--login` ?
2. Pourquoi est-il intéressant de l'utiliser ?

Utilisez la commande `su` pour passer `root`

6.2 Accès extérieur pour les VM

6.2.1 Un peu de réseau

Comme nous l'avons vu, votre machine virtuelle est connectée au *réseau virtuel principal*. Ce réseau est privé, spécifique à la machine de virtualisation et n'est pas routé. Autrement dit, aucune machine, autre que la machine de virtualisation et vos machines virtuelles, n'a accès à ce réseau.

Pour que vos VM aient accès à l'extérieur de ce réseau, elles passent par le routeur 10.42.0.1. Ce routeur est équivalent à la box internet que vous pouvez avoir chez vous. Il ne fait pas qu'un simple routage mais de la *translation d'adresse* (ou *NAT: Network Address Translation* en anglais).

Le principe du NAT est que le routeur remplace l'adresse des machines qui veulent sortir du réseau par son adresse *publique* (son adresse qui lui permet, lui, de sortir de son réseau). Quand votre machine virtuelle se connecte sur une machine extérieure au réseau virtuel principal (à une autre machine de salle de TP par exemple), celle ci pensera que c'est le routeur (et donc la machine de virtualisation) qui la contacte directement.

Une fois cette étape passée, votre VM est *vue* comme la machine de virtualisation.

6.3 Mise à jour

Le modèle que nous vous avons préparé date de quelques temps. Il y a probablement eu des mises à jour publiées par Debian. Pour mettre à jour votre machine virtuelle, vous allez utiliser le système de gestion de paquet Debian: APT.

Ce système de gestion de paquets est disponible *via* plusieurs interfaces utilisateurs. Nous allons utiliser `apt`, l'interface en ligne de commande recommandée pour une utilisation interactive[5]. Nous reviendrons par la suite sur APT. Pour la mise à jour du système, utilisez la commande suivante:

```
# apt update && apt full-upgrade
```

et laissez se terminer la mise à jour. Si cette dernière vous pose une question au sujet de GRUB, cochez la case `[ ] /dev/sda` à l'aide de la barre d'espacement.

Une fois la mise à jour terminée, comme il est probable qu'une nouvelle version du noyau ait été installée, redémarrez la machine virtuelle:

```
# reboot
```

6.4 Installation d'outils

Lisez la page de manuel `apt(1)` ainsi que le chapitre du guide de l'administrateur debian qui lui est consacré et installez les outils

suivants dans votre machine virtuelle:

- vim
- less
- tree
- rsync

7 Quelques trucs en plus

Le client SSH peut être configuré pour régler certains paramètres de façon spécifique pour chaque hôte auquel vous vous connectez.

Cette configuration se fait dans le fichier `$HOME/.ssh/config`. Lisez la page de manuel `ssh_config(5)` et configurer le client SSH de votre machine physique de façon à :

1. créer un alias `virt` pour votre machine de virtualisation. Ainsi, utiliser la commande `ssh virt` effectura en réalité l'équivalent de la commande `ssh dattier.iutinfo.fr` ;
2. effectuer un *transfert* de votre agent SSH quand vous vous connectez sur la machine de virtualisation. Vous pourez ainsi diffuser votre clé publique sur votre VM et utiliser votre clé privé (via l'agent de votre machine physique) depuis la machine de virtualisation ;
3. créer un alias `vm` pour pouvoir se connecter à votre VM directement depuis votre machine physique (directive `ProxyJump`). En d'autre terme, la commande `ssh vmjump` se connectera d'abord à la machine de virtualisation puis, automatiquement à la VM, sans autre manipulation de votre part.

Votre environnement de travail est maintenant prêt. Nous pourrons attaquer le vif du sujet dans le prochain sujet.

---

1. même si nous allons utiliser ce mécanisme pour des raisons pratiques, il existe avant tout pour des raisons de sécurité, notamment pour éviter le vol de mot passe par un serveur compromis↩

2. on pourrait utiliser d'autres paramètres pour générer des clés plus sécurisées en suivant les recommandations de l'ANSSI par exemple.↩

3. Ici `$HOME` représente le répertoire home de l'utilisateur **sur le serveur** sur lequel celui-ci tente de se connecter.↩

4. et pas la connexion SSH, à vous de deviner pourquoi.↩

5. Pour une utilisation dans des scripts, l'interface `apt-get` est conseillée car délibéremment stable. Plus d'information à ce sujet disponible dans le Guide de l'administrateur debian.↩