

# Procédure déploiement d'application. SAE 3.03

## Groupe J

Renan Declercq  
Clément Delerue

## Repo Gitlab

- <https://gitlab.univ-lille.fr/utilisateur/sae-deploiement.git>

## 1. Table des matières

- [Procédure déploiement d'application. SAE 3.03](#)
  - [1. Table des matières](#)
- [2. TP01](#)
  - [2.1 Un peu de vocabulaire et de convention](#)
  - [2.2 Connexion à distance](#)
    - [2.2.1 Première connexion à la machine de virtualisation](#)
    - [2.2.2 Faciliter la connexion](#)
  - [2.3 Créer et gérer des machines virtuelles](#)
    - [2.3.1 Création d'une machine virtuelle](#)
    - [2.3.2 Démarrer la machine virtuelle](#)
    - [2.3.3 Arrêt et suppression de la machine virtuelle](#)
    - [2.3.4 Obtenir des informations sur la machine virtuelle](#)
    - [2.3.5 Quelques informations sur le réseau et la VM](#)
    - [2.3.6 Utilisation de la machine virtuelle](#)
      - [2.3.6.1 Solution 1 : Console virtuelle](#)
      - [2.3.6.2 Solution 2 : Connexion SSH](#)
    - [2.3.7 Changement de la configuration réseau](#)
  - [2.4 Configurer et mettre à jour la machine virtuelle](#)
    - [2.4.1 Connexion root et SSH](#)

- 2.4.2 Accès extérieur pour les VM
    - 2.4.2.1 Un peu de réseau
    - 2.4.2.2 Configuration du proxy
    - 2.4.2.3 Mise à jour
    - 2.4.2.4 Installation d'outils
  - 2.5 Quelques trucs en plus
- 3. TP02
  - 1 Dernière configurations sur la VM
    - 1.1 Changement du nom de machine
    - 1.2 Installation et configuration de la commande `sudo`
    - 1.3 Configuration de la synchronisation d'horloge
  - 2 Installation et configuration basique d'un serveur de base de données
    - 2.1 Installer postgresql
    - 2.2 Connexion à postgresql
    - 2.3 Créer un nouvel utilisateur
    - 2.4 Créer une nouvelle base de données
    - 3.5 Manipuler une base de données
- Procédure déploiement d'application. SAE 3.03
  - 1. Table des matières
- 2. TP01
  - 2.1 Un peu de vocabulaire et de convention
  - 2.2 Connexion à distance
    - 2.2.1 Première connexion à la machine de virtualisation
    - 2.2.2 Faciliter la connexion
  - 2.3 Créer et gérer des machines virtuelles
    - 2.3.1 Création d'une machine virtuelle
    - 2.3.2 Démarrer la machine virtuelle
    - 2.3.3 Arrêt et suppression de la machine virtuelle
    - 2.3.4 Obtenir des informations sur la machine virtuelle
    - 2.3.5 Quelques informations sur le réseau et la VM
    - 2.3.6 Utilisation de la machine virtuelle
      - 2.3.6.1 Solution 1 : Console virtuelle
      - 2.3.6.2 Solution 2 : Connexion SSH
    - 2.3.7 Changement de la configuration réseau
  - 2.4 Configurer et mettre à jour la machine virtuelle
    - 2.4.1 Connexion root et SSH
    - 2.4.2 Accès extérieur pour les VM
      - 2.4.2.1 Un peu de réseau
      - 2.4.2.2 Configuration du proxy

- 2.4.2.3 Mise à jour
  - 2.4.2.4 Installation d'outils
- 2.5 Quelques trucs en plus
- 3. TP02
  - 1 Dernière configurations sur la VM
    - 1.1 Changement du nom de machine
    - 1.2 Installation et configuration de la commande `sudo`
    - 1.3 Configuration de la synchronisation d'horloge
  - 2 Installation et configuration basique d'un serveur de base de données
    - 2.1 Installer postgresql
    - 2.2 Connexion à postgresql
    - 2.3 Créer un nouvel utilisateur
    - 2.4 Créer une nouvelle base de données
    - 3.5 Manipuler une base de données
- Procédure déploiement d'application. SAE 3.03
  - 1. Table des matières
- 2. TP01
  - 2.1 Un peu de vocabulaire et de convention
  - 2.2 Connexion à distance
    - 2.2.1 Première connexion à la machine de virtualisation
    - 2.2.2 Faciliter la connexion
  - 2.3 Créer et gérer des machines virtuelles
    - 2.3.1 Création d'une machine virtuelle
    - 2.3.2 Démarrer la machine virtuelle
    - 2.3.3 Arrêt et suppression de la machine virtuelle
    - 2.3.4 Obtenir des informations sur la machine virtuelle
    - 2.3.5 Quelques informations sur le réseau et la VM
    - 2.3.6 Utilisation de la machine virtuelle
      - 2.3.6.1 Solution 1 : Console virtuelle
      - 2.3.6.2 Solution 2 : Connexion SSH
    - 2.3.7 Changement de la configuration réseau
  - 2.4 Configurer et mettre à jour la machine virtuelle
    - 2.4.1 Connexion root et SSH
    - 2.4.2 Accès extérieur pour les VM
      - 2.4.2.1 Un peu de réseau
      - 2.4.2.2 Configuration du proxy
      - 2.4.2.3 Mise à jour
      - 2.4.2.4 Installation d'outils
  - 2.5 Quelques trucs en plus

- 3. TP02
  - 1 Dernière configurations sur la VM
    - 1.1 Changement du nom de machine
    - 1.2 Installation et configuration de la commande `sudo`
    - 1.3 Configuration de la synchronisation d'horloge
  - 2 Installation et configuration basique d'un serveur de base de données
    - 2.1 Installer postgresql
    - 2.2 Connexion à postgresql
    - 2.3 Créer un nouvel utilisateur
    - 2.4 Créer une nouvelle base de données
    - 3.5 Manipuler une base de données
- Procédure déploiement d'application. SAE 3.03
  - 1. Table des matières
- 2. TP01
  - 2.1 Un peu de vocabulaire et de convention
  - 2.2 Connexion à distance
    - 2.2.1 Première connexion à la machine de virtualisation
    - 2.2.2 Faciliter la connexion
  - 2.3 Créer et gérer des machines virtuelles
    - 2.3.1 Création d'une machine virtuelle
    - 2.3.2 Démarrer la machine virtuelle
    - 2.3.3 Arrêt et suppression de la machine virtuelle
    - 2.3.4 Obtenir des informations sur la machine virtuelle
    - 2.3.5 Quelques informations sur le réseau et la VM
    - 2.3.6 Utilisation de la machine virtuelle
      - 2.3.6.1 Solution 1 : Console virtuelle
      - 2.3.6.2 Solution 2 : Connexion SSH
    - 2.3.7 Changement de la configuration réseau
  - 2.4 Configurer et mettre à jour la machine virtuelle
    - 2.4.1 Connexion root et SSH
    - 2.4.2 Accès extérieur pour les VM
      - 2.4.2.1 Un peu de réseau
      - 2.4.2.2 Configuration du proxy
      - 2.4.2.3 Mise à jour
      - 2.4.2.4 Installation d'outils
  - 2.5 Quelques trucs en plus
- 3. TP02
  - 1 Dernière configurations sur la VM
    - 1.1 Changement du nom de machine

- 1.2 Installation et configuration de la commande `sudo`
- 1.3 Configuration de la synchronisation d'horloge
- 2 Installation et configuration basique d'un serveur de base de données
  - 2.1 Installer postgresql
  - 2.2 Connexion à postgresql
  - 2.3 Créer un nouvel utilisateur
  - 2.4 Créer une nouvelle base de données
  - 3.5 Manipuler une base de données

## 2. TP01

### 2.1 Un peu de vocabulaire et de convention

#### Info

Si nécessaire, le prompt indiquera également précisément 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 phy 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:

```
(phy)utilisateur@hevea19:~$ ls
```

### 2.2 Connexion à distance

#### 2.2.1 Première connexion à la machine de virtualisation

#### Info

La machine physique = hevea19

La machine de virtualisation = acajou12

Fingerprint	Machine	Algo
-------------	---------	------

Fingerprint	Machine	Algo
3072 SHA256:1pXDryxLNIVsKPTkEIAona46yIf/peCZ88qTFbxvHH4	<a href="#">acajou12.iutinfo.fr</a>	(RSA)
256 SHA256:CrB3/QEaXDvn0Qp1dQapmYSejef1Zh45oEl4Ozuk1hU	<a href="#">acajou12.iutinfo.fr</a>	(ECDSA)
256 SHA256:4QYLsyCTuLFa2QCk+jqi4MBPuqpO4yD+FULHAcXj7IU	<a href="#">acajou12.iutinfo.fr</a>	(ED25519)

```
(phy)utilisateur@hevea19:~$ ssh acajou12.iutinfo.fr
```

Effectuer les validations :

```
The authenticity of host 'acajou12.iutinfo.fr (172.18.49.39)' can't be established.  
ECDSA key fingerprint is SHA256:CrB3/QEaXDvn0Qp1dQapmYSejef1Zh45oEl4Ozuk1hU.  
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

Votre client SSH ajoutera alors le serveur dans le fichier `$HOME/.ssh/known_hosts` pour indiquer que vous lui faites confiance.

```
yes
```

```
utilisateur@acajou12.iutinfo.fr's password: [Entrer le mdp]
```

```
Connection closed by 172.18.49.39 port 22
```

### Info

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.

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.

## 2.2.2 Faciliter la connexion

- Fabriquer une paire de clés

La commande demande :

- un nom de fichier : vous pouvez laisser le nom de fichier par défaut. Notez le. Il est bon de le connaître
- 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.

```
(phy)utilisateur@hevea19:~$ ssh-keygen
```

Generating public/private rsa key pair.

Enter file **in** which to save the key (/home/infoetu/utilisateur/.ssh/id\_rsa):

Ici, j'entre la passphrase choisie (voir Notes smartphone)

Enter passphrase (empty **for** no passphrase):

Enter same passphrase again:

Your identification has been saved **in** /home/infoetu/utilisateur/.ssh/id\_rsa

Your public key has been saved **in** /home/infoetu/utilisateur/.ssh/id\_rsa.pub

The key fingerprint is:

SHA256:zw56Ut+qEh0QaWg42E7gAJZHNYzztuvGPn0QbnH5Frc utilisateur@hevea19

The key's randomart image is:

```
+---[RSA 3072]-----+
|==oo+++.o          |
|=.+=+o.=           |
|+.oo. . .          |
| . o o.o . .        |
| . o.S.. o .        |
| ..+oo o E          |
| . +oo.+            |
| =oo.+ .            |
| +oo+o.o.           |
+-----[SHA256]-----+
```

- 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\_keys3 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 AAAAC3NzaC1lZDI1NTE5AAAAIGHaJ0p3Vx34PVVet6ZzTii60Sd3H15CZTiqqx37f8hp user@machir
```



## Info

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

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

- Solution 1 : le faire manuellement
- Solution 2 : utiliser la commande `ssh-copy-id`

On utilisera les commandes sur la machine de virtualisation

```
```bash
ssh acajou12.iutinfo.fr
```
```

```
```bash
man ssh-copy-id
```
```

```
```bash
ssh-copy-id -i ~/.ssh/id_rsa.pub utilisateur@acajou12
```
```

```
```bash
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/infoetu/utilisateur/.
The authenticity of host 'acajou12 (172.18.49.39)' can't be established.
ECDSA key fingerprint is SHA256:CrB3/QEaXDvn0Qp1dQapmYSejef1Zh45oEl40zuk1hU.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any tl
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it
utilisateur@acajou12's password:
```

Number of key(s) added: 1

```
Now try logging into the machine, with:  "ssh 'utilisateur@acajou12'"
and check to make sure that only the key(s) you wanted were added.
```
```

- 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`) ;
- si la clé publique est présente dans le fichier `authorized_keys`, le serveur génère un challenge aléatoire et le transmet au client ;
- le client renvoie une signature du challenge qu'il effectue avec sa clé privée ;
- le serveur vérifie la signature à l'aide de la clé publique ;

- le client est identifié, la connexion est autorisée.

On reesaye la connexion à la machine de virtualisation :

```
utilisateur@hevea19:~$ ssh acajou12.iutinfo.fr
Linux acajou12 5.10.0-17-amd64 #1 SMP Debian 5.10.136-1 (2022-08-13) 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.
Last login: Thu Nov 17 09:03:29 2022 from 172.18.49.100
utilisateur@acajou12:~$
```

## 2.3 Créer et gérer des machines virtuelles

### Attention

Vous devez exécuter toutes les commandes de cette section sur votre machine de virtualisation.

Executer le script vmiut :

```
(virtu)utilisateur@acajou12:~$ source /home/public/vm/vm.env
```

### Attention

Vous devrez utiliser la commande `source` dans chaque nouveau shell

Pour manipuler les machines virtuelles utilisez vmiut (cf '`vmiut --help`').

Les machines virtuelles seront stockées sur acajou12 dans `/usr/local/virtual_machine/infoetu/utuli`

### 2.3.1 Création d'une machine virtuelle

```
(virtu)utilisateur@acajou12:~$ vmiut creer matrix
```

```

Virtual machine 'matrix' is created and registered.
UUID: 5c1daebf-bb94-4988-add4-11052b12e2e8
Settings file: '/usr/local/virtual_machine/infoetu/utilisateur/matrix/matrix.vbox'
0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%
Clone medium created in format 'VMDK'. UUID: 47653a3f-bda0-41a0-8eb2-6ae1429665e1
# Paramètres vmiut
MACHINE=matrix
VBOXES=/usr/local/virtual_machine/infoetu/utilisateur
RESEAU=vmnet8
MEMOIRE=1024
VRDEPORT=5000-5050
MODELE=/home/public/vm/disque-5Go-bullseye.vdi
HOST=acajou12

# Paramètres VirtualBox
name=matrix
UUID=5c1daebf-bb94-4988-add4-11052b12e2e8
path=/usr/local/virtual_machine/infoetu/utilisateur/matrix
memory=1024
etat=poweroff
vrdeport=-1
mac=08:00:27:f4:d3:c6

```

```
(virtu)utilisateur@acajou12:~$ vmiut lister
```

```

"Win10" {dc87e177-eb41-457c-9fcc-84c0ca9f38ca}
"matrix" {5c1daebf-bb94-4988-add4-11052b12e2e8}

```

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

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

alors soit:

- \* Vous ne vous trouvez pas sur la machine de virtualisation ;
- \* Les fichiers de votre machine virtuelle ont été effacés.

## 2.3.2 Démarrer la machine virtuelle

```
(virtu)utilisateur@acajou12:~$ vmiut demarrer matrix
```

```

Waiting for VM "matrix" to power on...
VM "matrix" has been successfully started.

```

## 2.3.3 Arrêt et suppression de la machine virtuelle

### A faire

Arrêter la vm à la fin de chaque séance

```
(virtu)utilisateur@acajou12:~$ vmiut arreter matrix
```

### Attention

Ne pas utiliser si on ne veut pas la supprimer

```
(virtu)utilisateur@acajou12:~$ vmiut supprimer matrix
```

### Astuce

Si vous souhaitez les sauvegarder, vous pouvez copier le répertoire  
/usr/local/virtual\_machine/login/NOMDEMACHINE

## 2.3.4 Obtenir des informations sur la machine virtuelle

```
(virtu)utilisateur@acajou12:~$ vmiut info matrix
```

```
# Paramètres vmiut
MACHINE=matrix
VBOXES=/usr/local/virtual_machine/infoetu/utilisateur
RESEAU=vmnet8
MEMOIRE=1024
VRDEPORT=5000-5050
MODELE=/home/public/vm/disque-5Go-bullseye.vdi
HOST=acajou12

# Paramètres VirtualBox
name=matrix
UUID=5c1daebf-bb94-4988-add4-11052b12e2e8
path=/usr/local/virtual_machine/infoetu/utilisateur/matrix
memory=1024
etat=running
vrdeport=5000
mac=08:00:27:f4:d3:c6
ip-possible=192.168.194.25
```

Si ip-possible=  
-> Il faut attendre quelques minutes et relancer la commande

## 2.3.5 Quelques informations sur le réseau et la VM

| Machine  | Adresse                        |
|--|--------------------------------|
| Machine de virtualisation                        | 192.168.194.1                  |
| Routeur, DNS                                     | 192.168.194.2                  |
| Adresses dynamiques (attribuées automatiquement) | 192.168.194.25-192.168.194.128 |

### Info

La machine virtuelle a été créée à partir d'un modèle.

Voici les caractéristiques du modèle:

Distribution: Debian GNU/Linux 11 (bullseye)

Utilisateur standard: user, mot de passe: user

Administrateur: root, mot de passe: root

empreinte de la clé SSH: SHA256:SUHhxVJVZFiBQ6/koNbZfA9reKHyzIrvPgJvOEJ8zuE

## 2.3.6 Utilisation de la machine virtuelle

- Solution 1 : Utiliser une console virtuelle: simule un clavier et un écran qui serait connectés physiquement à la machine virtuelle
- Solution 2 : Se connecter en SSH.

### 2.3.6.1 Solution 1 : Console virtuelle

```
(virtu)utilisateur@acajou12:~$ vmiut console matrix
```

ERROR: Failed to open display:

#### Info

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

#### Solution

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 :

```
(virtu)utilisateur@acajou12:~$ vmiut arreter matrix
```

Sortir de la machine de virtualisation

```
(virtu)utilisateur@acajou12:~$ CTRL + D
```

```
(phy)utilisateur@hevea19:~$ ssh -X acajou12.iutinfo.fr
```

#### Info

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

## Ne pas oublier

```
(virtu)utilisateur@acajou12:~$ source /home/public/vm/vm.env
```

```
(virtu)utilisateur@acajou12:~$ vmiut demarrer matrix
```

```
(virtu)utilisateur@acajou12:~$ 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.

```
vm$ debian login : root
password : root
```

```
vm$ ip addr show
```

```
root@debian:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:f4:d3:c6 brd ff:ff:ff:ff:ff:ff
    inet 192.168.194.25/24 brd 192.168.194.255 scope global dynamic enp0s3
        valid_lft 1566sec preferred_lft 1566sec
    inet6 fe80::a00:27ff:fef4:d3c6/64 scope link
        valid_lft forever preferred_lft forever
```

```
vm$ ip addr route
```

```
root@debian:~# ip route
default via 192.168.194.2 dev enp0s3
169.254.0.0/16 dev enp0s3 scope link metric 1000
192.168.194.0/24 dev enp0s3 proto kernel scope link src 192.168.194.25
```

## 2.3.6.2 Solution 2 : Connexion SSH

```
(virtu)utilisateur@acajou12:~$ ssh user@192.168.194.25
```

```
The authenticity of host '192.168.194.25 (192.168.194.25)' can't be established.  
ECDSA key fingerprint is SHA256:SUHxVJVZFibQ6/koNbZfA9reKHyzIrvPgJv0EJ8zuE.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '192.168.194.25' (ECDSA) to the list of known hosts.  
user@192.168.194.25's password: user
```

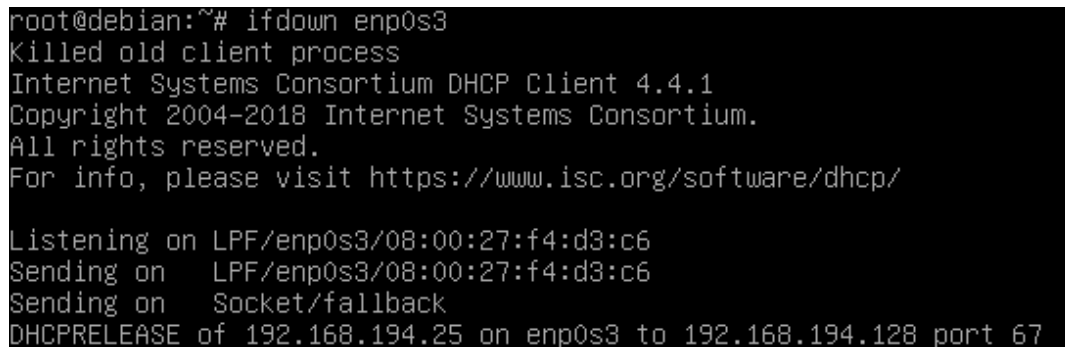
## 2.3.7 Changement de la configuration réseau

Votre machine virtuelle sera un serveur, hébergeant un service. Il est préférable qu'elle ait donc toujours la même adresse IP. Nous allons choisir l'adresse 192.168.194.3

```
(virtu)utilisateur@acajou12:~$ CTRL + D
```

```
(virtu)utilisateur@acajou12:~$ vmiut console matrix
```

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



```
root@debian:~# ifdown enp0s3  
Killed old client process  
Internet Systems Consortium DHCP Client 4.4.1  
Copyright 2004-2018 Internet Systems Consortium.  
All rights reserved.  
For info, please visit https://www.isc.org/software/dhcp/  
  
Listening on LPF/enp0s3/08:00:27:f4:d3:c6  
Sending on   LPF/enp0s3/08:00:27:f4:d3:c6  
Sending on   Socket/fallback  
DHCPRELEASE of 192.168.194.25 on enp0s3 to 192.168.194.128 port 67
```

Modifiez les fichiers `/etc/network/interfaces` et `/etc/resolv.conf` de façon à ce que la VM ait l'adresse statique 192.168.194.3 et qu'elle utilise le routeur 192.168.194.2 et serveur DNS 192.168.194.2.

Adresse statique et routeur :

```
root@vm# nano /etc/network/interfaces
```



```

GNU nano 5.4 /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug enp0s3
iface enp0s3 inet static
    address 192.168.194.3/24
    gateway 192.168.194.2

```

DNS (nameserver):

```
root@vm# nano /etc/resolv.conf
```

```

GNU nano 5.4
domain localdomain
search localdomain
nameserver 192.168.194.2
_

```

Redémarrer l'interface reseau

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

```

root@debian:~# ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:f4:d3:c6 brd ff:ff:ff:ff:ff:ff
    inet 192.168.194.3/24 brd 192.168.194.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fef4:d3c6/64 scope link
        valid_lft forever preferred_lft forever
root@debian:~# ip route show
default via 192.168.194.2 dev enp0s3 onlink
169.254.0.0/16 dev enp0s3 scope link metric 1000
192.168.194.0/24 dev enp0s3 proto kernel scope link src 192.168.194.3

```

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

```
root@vm# reboot
```

## 2.4 Configurer et mettre à jour la machine virtuelle

### 2.4.1 Connexion root et SSH

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

Quelle commande avez vous utilisée ?

```
```bash
(virtu)utilisateur@acajou12:~$ ssh root@192.168.194.3
```
```

Que se passe-t'il ?

```
(virtu)utilisateur@acajou12:~$ ssh root@192.168.194.3
The authenticity of host '192.168.194.3 (192.168.194.3)' can't be established.
ECDSA key fingerprint is SHA256:SUHhxVJVZFiBQ6/koNbZfA9reKHyzIrvPgJv0EJ8zuE.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.194.3' (ECDSA) to the list of known hosts.
root@192.168.194.3's password:
Permission denied, please try again.
root@192.168.194.3's password:
```

Pourquoi ? Il faut se connecter avec user

```
(virtu)utilisateur@acajou12:~$ ssh user@192.168.194.3
```

```
user@192.168.194.3's password: user
```

Linux debian 5.10.0-9-amd64 #1 SMP Debian 5.10.70-1 (2021-09-30) 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.

Last login: Thu Nov 17 11:09:18 2022 from 192.168.194.1

### Info

`su -l` : --login ou `-l ->` Exécute un script de connexion pour passer à un nom d'utilisateur

```
su -l
password : root
```

## 2.4.2 Accès extérieur pour les VM

### 2.4.2.1 Un peu de réseau

#### Info

Pour que vos VM aient accès à l'extérieur de ce réseau, elles passent par le routeur 192.168.194.2. 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

```
user@vm $ wget https://www.framasoft.org
```

### Erreur

Ca ne fonctionne pas encore car il faut configurer le proxy !

### 2.4.2.2 Configuration du proxy

```
nano /etc/environment
```

```
HTTP_PROXY=http://cache.univ-lille.fr:3128  
HTTPS_PROXY=http://cache.univ-lille.fr:3128  
http_proxy=http://cache.univ-lille.fr:3128  
https_proxy=http://cache.univ-lille.fr:3128  
NO_PROXY=localhost,192.168.194.0/24,172.18.48.0/22
```

```
(vm)root@debian:~# wget https://www.framasoft.org
```

Maintenant ça fonctionne !

### 2.4.2.3 Mise à jour

```
(vm)root@debian: apt update && apt full-upgrade
```

```
(vm)root@debian: reboot
```

### 2.4.2.4 Installation d'outils

```
(vm)root@debian: apt-get install vim
```

## 2.5 Quelques trucs en plus

```
(phy)utilisateur@hevea19:~$ codium $HOME/.ssh/config
```

```
Host virt
HostName acajou12.iutinfo.fr
ForwardAgent yes
```

```
Host vm
  HostName 192.168.194.3
  ForwardAgent yes
  User user
```

```
Host vmjump
  User user
  HostName 192.168.194.3
  ProxyJump virt
```

```
(phy)utilisateur@hevea19:~$ ssh virt
```

```
(phy)utilisateur@hevea19:~$ : ssh vmjump
```

- alias virt = utilise acajou12.iutinfo.fr
- alias vm : utilise user par défaut
- alias vmjump = la machine de virtualisation puis, automatiquement à la VM, sans autre manipulations
- ForwardAgent yes : transfère de votre agent SSH quand vous vous connectez sur la machine de virt

# 3. TP02

## 1 Dernière configurations sur la VM

### 1.1 Changement du nom de machine

```
(phy)utilisateur@hevea19:~$ ssh vm
```

```
(virtu)utilisateur@acajou12:~$ vmiut demarrer matrix  
ssh -X vm
```

#### Remarque

Utiliser l'utilisateur root !

Changer debian par matrix dans le fichier hostname puis reboot pour voir les changements

```
(vm)root@debian:~# nano /etc/hostname  
(vm)root@matrix:~# reboot
```

Changer matrix par matrix aussi dans le fichier hosts (Domain name system)

```
(vm)root@matrix:~# nano /etc/hosts  
(vm)root@matrix:~# reboot
```

```
(vm)root@matrix:~# ping matrix
```

### 1.2 Installation et configuration de la commande `sudo`

```
(vm)root@matrix:~# apt-get install sudo
```

```
(vm)root@matrix:~# nano /etc/sudoers
```

```
(vm)root@matrix:~# usermod -aG sudo user
```

User est bien ajouté au groupe sudo

```
(vm)user@matrix:~# groups
```

## 1.3 Configuration de la synchronisation d'horloge

```
(vm)user@matrix:~# date  
lun. 21 nov. 2022 15:18:38 CET
```

```
(virtu)utilisateur@acajou12:~$ date  
lun. 21 nov. 2022 14:18:38 CET
```

```
(phy)utilisateur@hevea19:~$ date  
lun. 21 nov. 2022 14:18:38 CET
```

```
(vm)user@matrix:~# nano /etc/systemd/timesyncd.conf
```

Décommenter les lignes et ajouter [NTP=ntp.univ-lille.fr](http://ntp.univ-lille.fr)

```
matrix
GNU nano 5.4 /etc/systemd/timesyncd.conf
# This file is part of systemd.
#
# systemd is free software; you can redistribute it and/or modify it
# under the terms of the GNU Lesser General Public License as published by
# the Free Software Foundation; either version 2.1 of the License, or
# (at your option) any later version.
#
# Entries in this file show the compile time defaults.
# You can change settings by editing this file.
# Defaults can be restored by simply deleting this file.
# See timesyncd.conf(5) for details.

[Time]
NTP=ntp.univ-lille.fr
FallbackNTP=0.debian.pool.ntp.org 1.debian.pool.ntp.org 2.debian.pool.ntp.org 3.debian.pool.ntp.org
RootDistanceMaxSec=5
PollIntervalMinSec=32
PollIntervalMaxSec=2048

root@matrix:~#
```

```
(vm)user@matrix:~# timedatectl set-ntp true
(vm)user@matrix:~# timedatectl status
(vm)user@matrix:~# systemctl restart systemd-timesyncd.service
```

### Note

Il faut attendre quelques secondes pour afficher la bonne heure au lancement de la vm

## 2 Installation et configuration basique d'un serveur de base de données

### 2.1 Installer postgresql

```
(vm)user@matrix:~# apt-get install postgresql
```



## Note

`systemd` est un gestionnaire de systèmes d'initialisation et de systèmes qui est devenu la nouvelle norme pour les distributions de Linux.

`systemctl` est l'outil de gestion essentiel pour contrôler le système d'initialisation

Liste de commandes `systemctl` :

```
(vm)user@matrix:~# systemctl start application.service
(vm)user@matrix:~# systemctl stop application.service
(vm)user@matrix:~# systemctl restart application.service
(vm)user@matrix:~# systemctl reload application.service
(vm)user@matrix:~# systemctl reload-or-restart application.service
(vm)user@matrix:~# systemctl enable application.service
(vm)user@matrix:~# systemctl disable application.service
(vm)user@matrix:~# systemctl status application.service
(vm)user@matrix:~# systemctl is-active application.service
(vm)user@matrix:~# systemctl is-enabled application.service
(vm)user@matrix:~# systemctl is-failed application.service
```

```
(vm)user@matrix:~# systemctl status postgresql.service
```

## 2.2 Connexion à postgresql

```
(vm)user@matrix:~# sudo -u postgres -i
```

## 2.3 Créer un nouvel utilisateur

```
(vm)postgres@matrix:~# createuser -d -r -l -P matrix
Saisir le mot de passe pour le nouveau rôle : matrix
Saisir le mot de passe à nouveau : matrix
```

## Note

\*-d = --createdb (Permet au nouvel utilisateur de créer une base de données)

- -r = --createrole (permet au nouvel utilisateur de créer un rôle)

\*-l = --login (Permet au nouvel utilisateur de se connecter , option par défaut )

- -P = --pwprompt (Permet à l'utilisateur de créer un mot de passe qui sera demandé à la connexion)

## 2.4 Créer une nouvelle base de données

```
(vm)postgres@matrix:~# createdb -O matrix matrix
```

### Note

- -O = --owner (permet de spécifier le propriétaire de la base de données)

### Changer d'utilisateur

Pour les commandes suivantes, exécuter les commandes avec l'utilisateur user et non pas postgres

```
(vm)postgres@matrix:~# psql -h localhost -U matrix -d matrix -c '\x' -c 'create table test(id Inte
```

### Note

- -h = --host (L'endroit où est installé postgres)
- -U = --username (Nom du compte utilisateur avec lequel on se connecte à la base de données)
- -d = --dbname (Nom de la base de données)
- -c = --command ('\x' indique l'exécution d'une requête sql)

```
(vm)postgres@matrix:~# psql -h localhost -U matrix -d matrix
matrix => \d
```

## 3.5 Manipuler une base de données

```
(vm)postgres@matrix:~#
matrix => insert into test values(1);
```

```
(vm)postgres@matrix:~#  
matrix => select * from test;
```