

# SAE 102 :

## S'initier aux réseaux informatiques

<b>0.0 - Introduction:</b>	<b>2</b>
<b>1.0 - Compréhension du projet :</b>	<b>2</b>
<b>1.1 - Configuration de l'OS</b>	<b>3</b>
<b>1.2 - Configuration de la Machine Virtuelle</b>	<b>4</b>
<b>2 - Début du projet</b>	<b>5</b>
<b>2.1 - DHCP</b>	<b>5</b>
<b>2.2 - SSH</b>	<b>7</b>
<b>2.3 - IP FORWARDING</b>	<b>8</b>
<b>3.0 - Bonus</b>	<b>9</b>
	<b>1</b>

## 0.0 - Introduction:

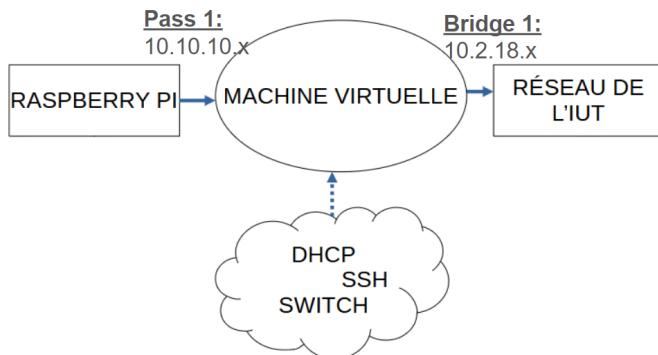
Ce SAE est un projet orienté sur le nano-ordinateur Raspberry pi, il a pour but la prise en main de cet appareil mais aussi découvrir de nouvelles choses au niveau des réseaux et des commandes linux. De plus, on cherchera à comprendre comment configurer à distance le Raspberry Pi, grâce à notamment une connexion SSH.



## 1.0 - Compréhension du projet :

Afin de mieux comprendre comment réaliser ce SAE il faut d'abord réaliser un schéma de notre installation et définir les différents rôles et objectifs de chacune des machines et outils que nous utiliserons. Déjà nous aurons besoin d'un ordinateur avec lequel on créera une machine virtuelle qui aura de multiples fonctions, nous aurons aussi besoin d'un raspberry pi qui devra être configuré et enfin de nombreux câbles différents pour réaliser les différentes connexions à faire au fur et à mesure de notre projet.

Ici on voit les différentes connexions qui existent entre les outils, le nuage montre les configurations à faire. Donc la machine virtuelle se voit attribuer un rôle de serveur DHCP, de switch enfin celui de routeur pour pouvoir avoir accès à internet et donc le donner au raspberry pi. De plus, notre machine virtuelle sera utilisée pour configurer à distance notre raspberry pi à l'aide d'une communication SSH. On voit donc que la machine virtuelle joue presque le même rôle qu'une box internet.

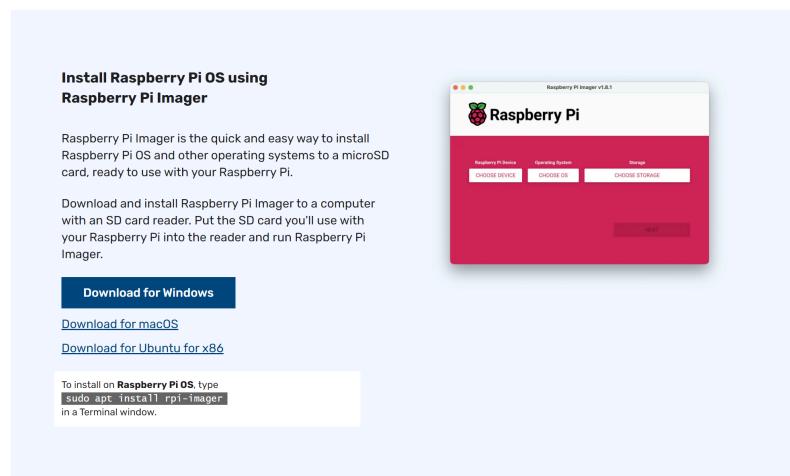


Ici on comprend mieux notre installation à noter que dans ce projet la machine virtuelle sert de DHCP, de switch et de passerelle, on peut considérer notre machine comme une box internet car elle remplit plusieurs fonctions qui permettent au réseau local d'accéder à internet ou juste de communiquer avec notre machine virtuelle.

## 1.1 - Configuration de l'OS

La première étape nécessaire au bon fonctionnement de notre raspberry pi est l'installation de son OS (operating system -> système d'exploitation).

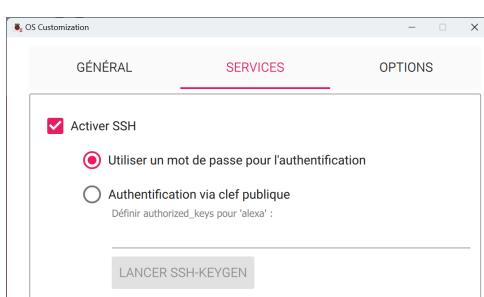
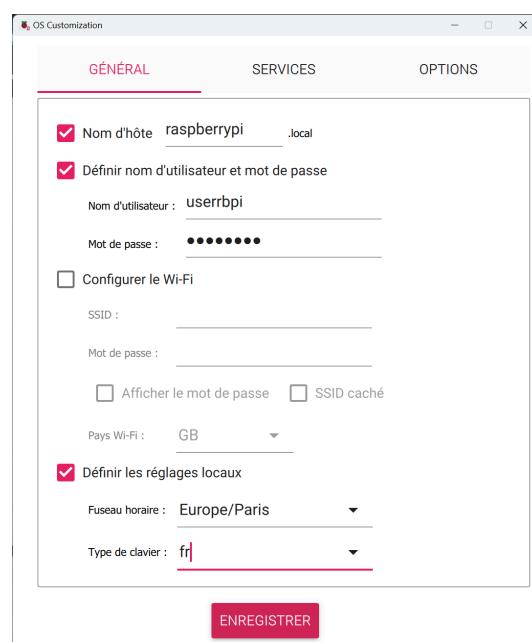
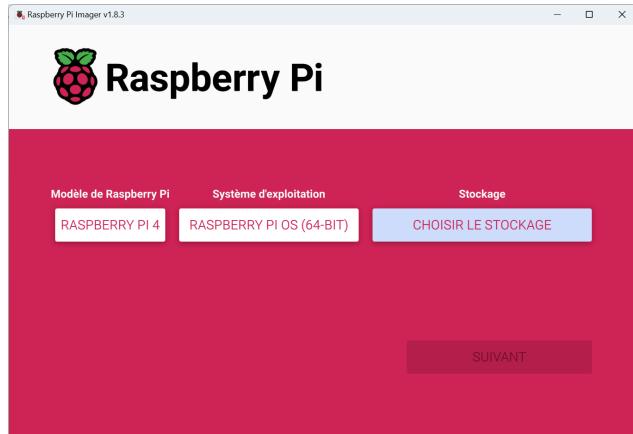
Ici nous décidons donc d'installer l'os Raspberry pi. Lorsque nous cliquons sur download on va recevoir un fichier en .exe qu'il faudra exécuter et/ou utiliser la commande **sudo apt install tpi-imager**.



Ensuite un écran avec de multiples configuration apparaîtra afin de personnaliser l'OS comme vous en avez besoin.

Ici on voit que l'on doit choisir le modèle de notre raspberry pi, la version de l'os et enfin le stockage ou l'os va être mis, dans notre cas on choisira notre carte sd à insérer dans le raspberry pi.

Dans le 3<sup>ème</sup> screen on peut voir d'autres paramètres utiles à la configuration, les paramètres nécessaires sont : un nom (ici userrbpi), un mot de passe (tpzro), la langue (français = fr) et enfin chose importante qui sera utile plus tard dans le projet c'est d'activer la fonction SSH dans l'onglet "services".



## 1.2 - Configuration de la Machine Virtuelle

Cette partie est primordiale à l'accomplissement de notre projet. Ici il s'agit de créer une machine virtuelle qui pourra répondre aux exigences de notre projet.

(ubuntu 22, 2 vcpu, 4go ram, 2 cartes réseau -pass1 -bridge1)

La première étape lors de l'allumage de notre machine virtuelle est sa configuration. En effet il faut faire en sorte qu'elle puisse accéder à internet pour la redistribuer ensuite au Raspberry Pi. Pour ce faire, on modifie manuellement les paramètres réseau de la machine virtuelle, du moins pour la carte réseau 1 (ens3) : bridge 1, car elle correspond à la liaison entre le routeur de l'IUT et notre machine.

On utilisera l'ip appartenant à la plage ip affichée sur le boîtier de notre pc. Dans notre cas on utilise : 10.2.18.44, avec un masque 255.255.255.0, une passerelle 10.2.18. et un DNS 194.167.156.13.

Ensuite il faut aussi modifier manuellement la seconde carte réseau (ens4) : pass 1, qui correspond au réseau entre notre machine virtuelle et notre raspberry pi. Ici on mettra comme adresse IP une ip qui n'a pas le même host id que celle de bridge 1 mais qui aura le même host id que le Raspberry pi, pour être sûr de ne pas se tromper. On utilisera donc l'host id : 10.10.10.x avec comme masque de sous réseau 255.255.255.0 et pas de passerelle ni de DNS. Ici on utilise comme ip 10.10.10.3.

Processeur(s)	2 vcpus
Mémoire vive	4G
Cartes réseaux	2
Carte 1 attachée à	bridge1
Carte 2 attachée à	pass1

Filiaire

Annuler	Appliquer			
Détails	Identité	IPv4	IPv6	Sécurité
<b>Méthode IPv4</b>				
<input type="radio"/> Automatique (DHCP) <input type="radio"/> Réseau local seulement				
<input checked="" type="radio"/> Manuel <input type="radio"/> Désactiver				
<input type="radio"/> Partagée avec d'autres ordinateurs				
<b>Adresses</b>				
Adresse	Masque de réseau	Passerelle		
10.2.18.44	255.255.255.0	10.2.18.1		
<b>DNS</b>				
Automatique <input checked="" type="checkbox"/>				
194.167.156.13				
Séparer les adresses IP avec des virgules				
<b>Routes</b>				
Automatique <input checked="" type="checkbox"/>				

Filiaire

Annuler	Appliquer			
Détails	Identité	IPv4	IPv6	Sécurité
<b>Méthode IPv4</b>				
<input type="radio"/> Automatique (DHCP) <input type="radio"/> Réseau local seulement				
<input checked="" type="radio"/> Manuel <input type="radio"/> Désactiver				
<input type="radio"/> Partagée avec d'autres ordinateurs				
<b>Adresses</b>				
Adresse	Masque de réseau	Passerelle		
10.10.10.3	255.255.255.0			
<b>DNS</b>				
Automatique <input checked="" type="checkbox"/>				

Une fois toutes les configurations faites on va pouvoir démarrer notre projet, cependant il faut être sûr que notre machine virtuelle ait accès à internet. Pour tester cela, on essaye de pinguer la passerelle depuis la machine virtuelle.

Ici le ping fonctionne, on en conclut que la connexion c'est bien passé, la machine virtuelle est bien connectée au réseau de l'IUT.

```
supervisor@ubuntu2204:~$ ping 10.2.18.1
PING 10.2.18.1 (10.2.18.1) 56(84) bytes of data.
64 bytes from 10.2.18.1: icmp_seq=1 ttl=254 time=0.690 ms
64 bytes from 10.2.18.1: icmp_seq=2 ttl=254 time=1.11 ms
64 bytes from 10.2.18.1: icmp_seq=3 ttl=254 time=1.12 ms
64 bytes from 10.2.18.1: icmp_seq=4 ttl=254 time=0.901 ms
64 bytes from 10.2.18.1: icmp_seq=5 ttl=254 time=1.17 ms
^C
--- 10.2.18.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 0.690/0.999/1.174/0.180 ms
```

## 2 - Début du projet

Maintenant que notre machine virtuelle et raspberry pi ont des configurations basiques, on peut commencer réellement notre projet.

### 2.1 - DHCP

La première étape suivant la configuration des machines va être la configuration du DHCP afin que le raspberry pi puisse avoir accès à une ip et donc pouvoir communiquer avec notre machine virtuelle et par la suite internet.

On va donc devoir faire en sorte que notre machine virtuelle devienne un serveur DHCP pour qu'elle puisse remplir sa tâche. Pour ce faire il faudra installer l'apt DHCP :

```
~$ sudo apt install isc-dhcp-server
```

Cette commande nous permet d'installer les fichiers permettant de configurer un serveur DHCP. Il faut bien confirmer le téléchargement des fichiers avec o.

A présent il faut donc modifier les fichiers pour qu'ils répondent aux besoins de notre projet, on utilisera alors un éditeur (nano/gedit/vi) de texte pour modifier ledit fichier.

```
~$ sudo nano /etc/dhcp/dhcpd.conf
```

Maintenant que nous avons accès au fichier il faudra modifier plusieurs lignes afin de pouvoir configurer notre serveur DHCP.

La première chose à modifier est le DNS, il faut en effet préciser le DNS dans le serveur DHCP car il en a besoin pour le transmettre aux appareils du réseau. Dans notre cas le DNS sera 194.167.156.13.

```
# option definitions common to all supported networks...
#option domain-name "example.org";
option domain-name-servers 194.167.156.13;

default-lease-time 600;
max-lease-time 7200;
```

Ensuite à la fin du fichier il faudra ajouter les paramètres permettant de donner les ip (plage ip,netmask), ainsi que l'ip de la passerelle par laquelle les appareils du réseau passeront pour accéder à internet. Ici la passerelle sera notre machine virtuelle.

```
subnet 10.10.10.0 netmask 255.255.255.0 {
    option subnet-mask 255.255.255.0;
    range dynamic-bootp 10.10.10.10 10.10.10.20;
    option routers 10.10.10.3;
}
```

Ensuite il faut préciser sur quel réseau le DHCP doit jouer son rôle, pour faire cela nous utiliserons :

**~\$ sudo nano /etc/default/isc-dhcp-server**

Ici on lui demande d'être actif pour le réseau ens4 qui comme vu précédemment correspond à notre pass 1. Interfaces correspond au type d'adresse IP, ici on est sur de l'ipv4 alors on utilise le champ **INTERFACESv4** pour y mettre notre carte réseau (ens4) afin que le DHCP sache où attribuer les ip.

```
# On what interfaces should the DHCP server (dhcpd) serve DHCP requests
#           Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACESv4="ens4"
INTERFACESv6=""
```

Une fois la configuration de notre DHCP faite il faut s'assurer de son bon fonctionnement on utilisera alors des commandes pour vérifier cela :

**~\$ sudo systemctl restart isc-dhcp-server**

Cette commande permet à notre machine de redémarrer son serveur DHCP. La seconde quant à elle nous permet de vérifier l'état de notre DHCP :

**~\$ sudo systemctl status isc-dhcp-server**

```
supervisor@ubuntu2204:~$ sudo systemctl restart isc-dhcp-server
supervisor@ubuntu2204:~$ sudo systemctl status isc-dhcp-server
● isc-dhcp-server.service - ISC DHCP IPv4 server
   Loaded: loaded (/lib/systemd/system/isc-dhcp-server.service; enabled; vendor
   Active: active (running) since Tue 2023-12-19 09:03:15 CET; 9s ago
     Docs: man:dhcpd(8)
 Main PID: 3374 (dhcpd)
   Tasks: 4 (limit: 4625)
   Memory: 4.5M
      CPU: 12ms
      CGroup: /system.slice/isc-dhcp-server.service
              └─3374 dhcpcd -user dhcpcd -group dhcpcd -f -4 -pf /run/dhcp-server/dhcpcd.pid

déc. 19 09:03:15 ubuntu2204 sh[3374]: PID file: /run/dhcp-server/dhcpcd.pid
déc. 19 09:03:15 ubuntu2204 dhcpcd[3374]: Wrote 0 leases to leases file.
déc. 19 09:03:15 ubuntu2204 sh[3374]: Wrote 0 leases to leases file.
déc. 19 09:03:15 ubuntu2204 dhcpcd[3374]: Listening on LPF/ens4/52:54:00:5a:ff:2>
déc. 19 09:03:15 ubuntu2204 sh[3374]: Listening on LPF/ens4/52:54:00:5a:ff:21/1>
déc. 19 09:03:15 ubuntu2204 dhcpcd[3374]: Sending on   LPF/ens4/52:54:00:5a:ff:2>
déc. 19 09:03:15 ubuntu2204 sh[3374]: Sending on   LPF/ens4/52:54:00:5a:ff:21/1>
déc. 19 09:03:15 ubuntu2204 dhcpcd[3374]: Sending on   Socket/fallback/fallback->
déc. 19 09:03:15 ubuntu2204 sh[3374]: Sending on   Socket/fallback/fallback-net.
déc. 19 09:03:15 ubuntu2204 dhcpcd[3374]: Server starting service.
1 lines, 1.21/21 (END)
```

Si le serveur DHCP s'allume en rouge alors cela veut dire qu'il ne fonctionnera pas.

**~\$ sudo journalctl -xe** nous indiquera donc l'origine du problème.

## 2.2 - SSH

A présent il s'agit de configurer la communication SSH, ici on veut pouvoir accéder à notre raspberry pi à distance et donc le configurer. La première étape est de pouvoir connaître l'adresse IP attribuée par notre DHCP au raspberry pi.

La première commande nous est nécessaire est celle qui va nous permettre de connaître l'adresse ip de notre Raspberry Pi. Cette commande s'appelle Nmap. Pour l'installer il faut d'abord faire cette commande :

```
~$ sudo apt install nmap
```

Une fois la commande exécutée on a accès aux différentes commandes de Nmap.

La plus importante et celle qui va nous servir pour ce projet est :

```
~$ sudo nmap 10.10.10.*
```

Cette commande est importante car elle nous permet d'obtenir les informations via un scan des réseaux avec le host id indiqué dans la commande (ici 10.10.10.x). En effet, elle passe par les ports de notre machine virtuelle et atteint le réseau de pass1. On peut donc maintenant obtenir l'ip du Raspberry pi.

```
Nmap scan report for 10.10.10.10
```

Maintenant que nous avons pu obtenir l'adresse ip de notre Raspberry Pi, il est temps d'essayer de se connecter à distance. La commande qui nous permettra cela est :

```
~$ sudo ssh userrbpi@ 10.10.10.10
```

(avec le format : ~\$sudo ssh [nom\_utilisateur\_raspberrypi]@[ip\_raspberrypi] )

*Password : tprzo*

Désormais la connexion SSH avec le Raspberry Pi est établie, nous pouvons donc le configurer à distance.

## **2.3 - IP FORWARDING**

A présent nous allons mettre en place l'ip forwarding (qui est le routage des paquets ip) qui va permettre à notre machine virtuelle de jouer un nouveau rôle. En effet, notre machine va aussi avoir un rôle de routeur. La configuration que nous allons lui appliquer va lui demander de rediriger tous les paquets réseau reçu par sa carte ens4 (réseau 10.10.10.0) sur son routeur passerelle c'est-à- dire celui de l'iut et donc donner accès à internet à ce réseau.

La première commande à faire est celle qui nous permettra l'accès au fichier pour pouvoir activer le routage des paquets ip :

```
~$ sudo nano /etc/sysctl.conf
```

Puis on décommente la ligne avec : net.ipv4.ip\_forward=1

```
# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1
```

Ensuite la commande suivante permet de confirmer la manœuvre :

```
~$sudo sysctl -p
```

La commande suivante indique que les paquets arrivant de la carte réseau ens4 sont redirigés sur la carte réseau ens3 :

```
~$ iptables -t nat -A POSTROUTING -o ens3 -j MASQUERADE
```

Enfin la commande suivante doit être faite pour enregistrer la configuration :

```
~$ iptables-save > /etc/iptables_rules.save
```

Maintenant que toutes les configurations sont faites nous pouvons désormais nous reconnecter sur notre Raspberry pi avec le SSH et vérifier que les ping marchent.

```
user@raspberrypi:~ $ ping 10.2.18.1
PING 10.2.18.1 (10.2.18.1) 56(84) bytes of data.
64 bytes from 10.2.18.1: icmp_seq=1 ttl=253 time=0.668 ms
64 bytes from 10.2.18.1: icmp_seq=2 ttl=253 time=1.56 ms
64 bytes from 10.2.18.1: icmp_seq=3 ttl=253 time=1.92 ms
64 bytes from 10.2.18.1: icmp_seq=4 ttl=253 time=1.77 ms
^C
--- 10.2.18.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3023ms
rtt min/avg/max/mdev = 0.668/1.479/1.923/0.485 ms
```

**Et voilà ce qui conclut notre SAE 1.2 sur la configuration à distance d'un Raspberry pi.**

## **3.0 - Bonus**

Cette catégorie est là uniquement pour montrer les différentes erreurs rencontrées afin de les éviter.

- L'erreur la plus courante est la faute de frappe, en effet vérifier les entrées que nous faisons est primordiale pour éviter les problèmes.
- La deuxième est l'oubli de sudo, ce qui ne nous donnera pas accès aux droits admin et donc nous empêchera d'exécuter certaines commandes.
- La troisième, est la configuration de la VM, il faut faire attention aux cartes réseau.
- La suivante est qu'il faut faire attention car après avoir branché notre Raspberry pi sur le réseau il y a un temps d'attente avant qu'il ait une ip attribuée par le DHCP.
- Enfin il faut aussi faire attention à ne pas oublier le nom d'utilisateur et le mot de passe défini sur le Raspberry Pi.