

SAE 12

S'initier aux Réseaux Informatiques

Par Mohamad El Akhal El Bouzidi
Et Julien Poultier
Encadré par Mr. Mascarón

Table des matières :

- Introduction
- Prérequis nécessaires
- Schéma réseaux que nous allons réaliser et respecter dans ce projet
 - o Connectivité du Raspberry Pi
 - o Schéma réseaux de notre machine VM Ubuntu
- Configuration de base de notre machine VM Ubuntu
 - o Options à choisir V4RT
 - o Premier démarrage quoi faire ?
 - Adressage IP
 - IP statique
 - Masque Réseaux
 - Passerelle
 - DNS
 - On importe le proxy de l'IUT
 - Testons notre branchement
- Installation du serveur DHCP
 - o Installation du ISC-DHCP-SERVER
 - o Configuration du ISC-DHCP-SERVER
 - Modifions le fichier dhcpd.conf
 - Modification du ISC-dhcp-conf-server
 - Redémarrage du service ISC-DHCP-SERVER
- Préparation du Raspberry Pi
 - o Installons le système opératif du Raspberry Pi
 - Installation du Raspberry Pi Imager
 - Configuration du logiciel
 - Flashons l'OS dans la carte SD
 - o Branchement du Raspberry Pi à la baie
- Trouvons notre Raspberry Pi dans notre réseau
 - o Comment faire :
 - Nmap
 - o Commande à réaliser avec Nmap
- Configurer le IP packet forwarding
 - o Commandes
 - o Résultat
- Conclusion

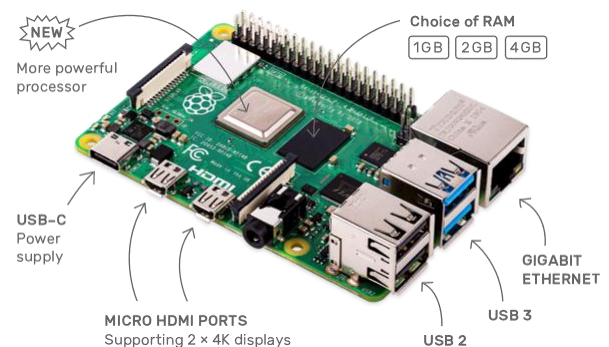
Introduction :

Ce projet vient du SAE 102 appelé : S'initier aux réseaux informatiques, le but de ce projet est plutôt simple : nous avons un ordinateur, en ce cas, un Raspberry Pi, cet ordinateur sera branché au réseau que nous allons créer et configurer grâce aux outils informatiques qu'on a dû trouver sur le net. Pour cet ordinateur nous n'aurons accès à aucune interface graphique quelle qu'elle soit, donc aucun écran, mais aussi aucun périphérique, souris, clavier, clef USB. Le seul équipement que l'on pourra insérer à l'intérieur de notre Raspberry Pi sera une carte SD, que nous allons devoir flasher avec le bon système opératif et la configurer pour nous donner un accès distant au RP (Raspberry Pi). L'hôte avec lequel on pourra contrôler le RP sera n'importe quel ordinateur qui est dans notre réseau. Pour réaliser ce réseau, on devra tout d'abord créer une machine virtuelle qui va être en quelque sorte notre centre de commande pour toutes les opérations que nous allons réaliser dans ce projet, on expliquera plus en détail dans ce document la configuration nécessaire pour faire marcher correctement cette machine virtuelle afin de mener à bien ce projet.

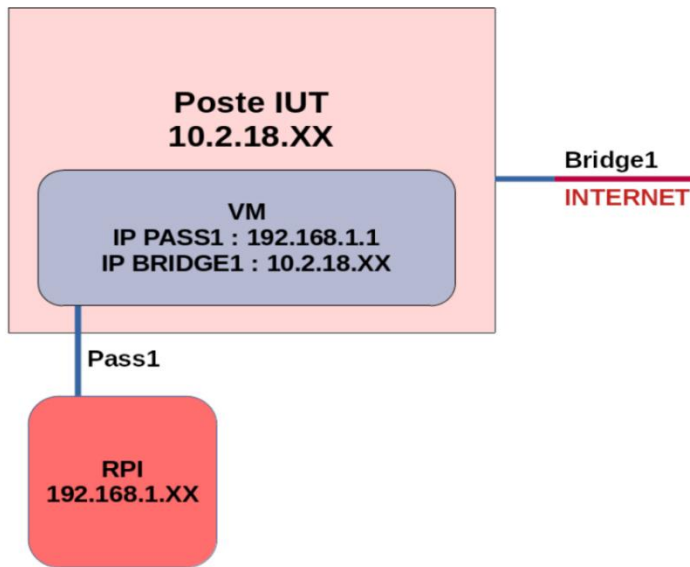
Pour réaliser ce projet des prérequis sont nécessaires, la manipulation correcte d'un terminal de commande est nécessaire, mais pas seulement. La connaissance des commandes basiques Shell (ls, cd, nano ...) sera nécessaire au vu du fait que l'on n'aura pas d'interface graphique à l'intérieur de notre RP. Savoir créer et configurer une VM (machine virtuelle), est aussi nécessaire, cependant nous écrirons étape par étape la configuration à mettre dans VI4RT. Et finalement savoir comment les adresses IP dans un réseau sont distribués à toutes les machines depuis un serveur DHCP, ceci aussi sera expliqué plus en détail plus tard dans ce document.

Entrons dans le vif du sujet, vous connaissez sûrement ce qu'est un Raspberry Pi n'est-ce pas ? Ne vous inquiétez pas si votre réponse fut négative on va vous expliquer très rapidement de quoi il s'agit. Un Raspberry Pi est tout simplement un ordinateur de petite taille, mais ce n'est pas parce que c'est petit que c'est inutile ou peu puissant, certains modèles de Raspberry sont extrêmement puissants et ils ont jusqu'à 8Go de ram. Ces petits ordinateurs sont extrêmement utiles pour les gens qui savent s'en servir correctement, on peut les utiliser comme serveurs pour héberger des sites web, centres multimédias et bien plus. Dans notre projet, cet ordinateur va nous servir pour accéder à la température mais aussi à l'humidité du milieu où il se trouve, grâce aux capteurs de température et d'humidité : le DHT11, il s'agira du dernier élément présent dans notre projet et nous expliquerons comment il fonctionne et comment nous allons l'utiliser.

Un RP a des entrées et des sorties comme nous pouvons le voir sur la photo ci-dessus, nous allons utiliser principalement son port gigabit Ethernet et son lecteur de carte SD.



Comme vous le savez, le but de ce projet est d'accéder à distance à notre RP grâce à internet. Pour cela il faut un réseau, et qui dit réseau dit schéma réseau. Voici donc notre schéma réseau que l'on a conçu :



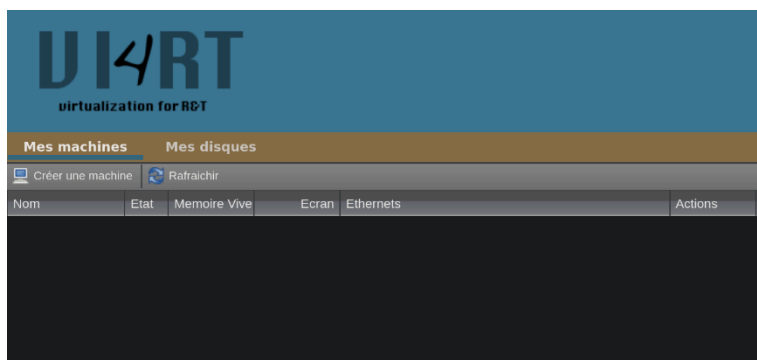
Ce réseau est constitué de 3 machines différentes :
 Notre Poste IUT, où nous allons créer la machine virtuelle qui va nous permettre de faire les opérations nécessaires.
 La VM : qui sera composée de deux cartes réseau différentes pour établir une connexion simultanée à Internet et au Raspberry Pi, cette VM là sera créée avec les cartes réseaux Pass1 et Bridge1, chacune d'entre elles aura une utilisation différente.
 Bridge1 permettra une connexion à internet c'est-à-dire avoir un accès à

l'extérieur de notre réseau, nous allons devoir la configurer avec une des adresses IP de la plage des IP de notre machine physique, nous expliquerons plus en détail comment le faire plus tard dans ce document. Pass1 sera la carte réseau qui fera la liaison entre notre RP et notre machine virtuelle, cette carte réseau, ne sera pas sur le réseau de l'IUT on peut alors se permettre d'écrire l'adresse IP qu'on veut, nous avons choisi 192.168.1.1, rappelons qu'une adresse IP machine ne sera jamais 192.168.1.0 ceci est réservé seulement pour la représentation des réseaux.

Ensuite nous n'allons absolument pas toucher à notre RP d'un point de vue réseaux car son adresse IP sera donnée automatiquement par DHCP grâce à notre VM.

Maintenant que nous avons vu un schéma de notre configuration, nous allons maintenant le concrétiser. Allez sur « vi4rt.univ-pau.fr » (sans les guillemets), bien sûr depuis un poste de l'IUT sinon vous aurez une erreur.

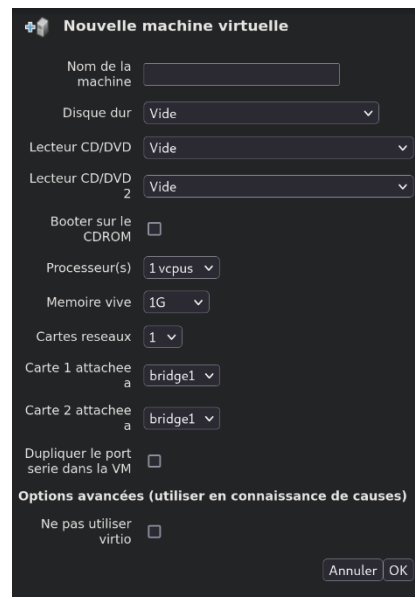
Vous allez tomber sur un page comme celle-ci :



Sur cette page vous allez vous rendre compte qu'il y a 2 onglets, « Mes machines » et « Mes disques », on ne va pas s'en servir de cette deuxième car elle ne nous intéresse pas, mais si vous voulez sauvegarder les changements que vous avez fait sur une VM pour les rendre persistantes n'hésitez pas à utiliser cet onglet.

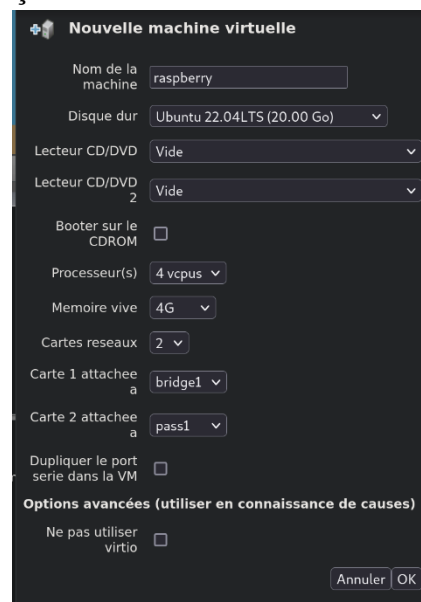
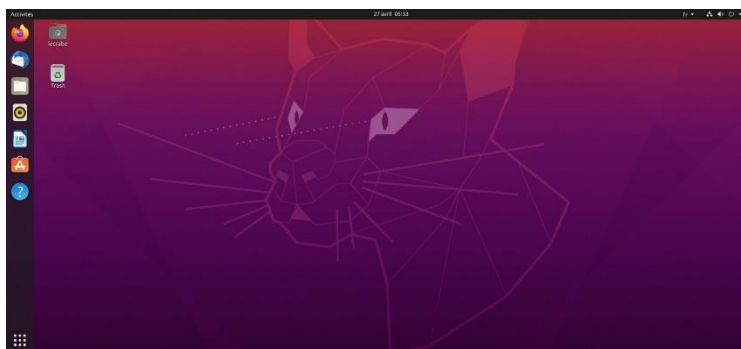
Bon, allons à la partie qui nous intéresse, « Mes machines », nous allons directement aller dans Créer Une machine :

Quand vous cliquerez dessus, vous allez tomber sur une page comme celle-ci. Une fois dedans vous allez mettre un nom à votre machine, ensuite vous allez choisir un disque dur, c'est-à-dire un système opératif pour la machine Virtuelle, depuis le temps qu'on a réalisé ce projet une nouvelle version Ubuntu est sortie, Ubuntu 22, n'hésitez surtout pas à la prendre, mais nous n'avons pas ce choix-ci au moment de la réalisation, donc nous avons pris la seule disponible : Ubuntu 18. Une fois valide vous mettez à votre préférence le nombre de cœurs CPU et de mémoire Vive que vous allez y mettre à votre VM. Nombre de cartes réseaux vous allez devoir choisir l'option 2 cartes réseaux, comme on vous a expliqué avant. En cartes attaches vous allez choisir en premier « Bridge1 » et en deuxième « Pass 1 », et c'est tout.



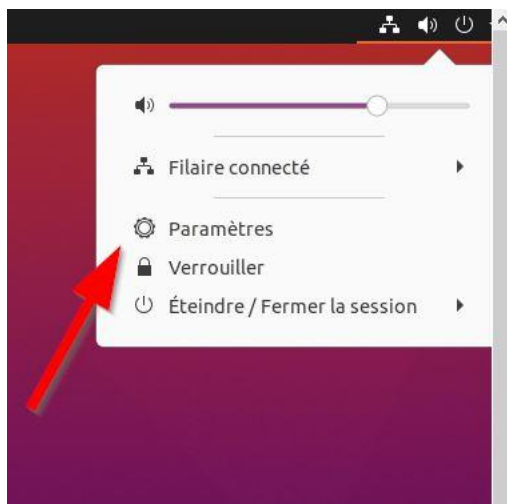
Vous allez voir un résultat qui va ressembler énormément à ça :
Actuellement il ne vous reste plus qu'appuyer sur le bouton ok, et attendre que votre machine démarre

Bon, maintenant que votre machine Ubuntu a démarré correctement, saisissez le mot de passe suivant pour y accéder : « tprzo.40 », une fois que vous saisissez ce mot de passe vous allez croiser cet écran :

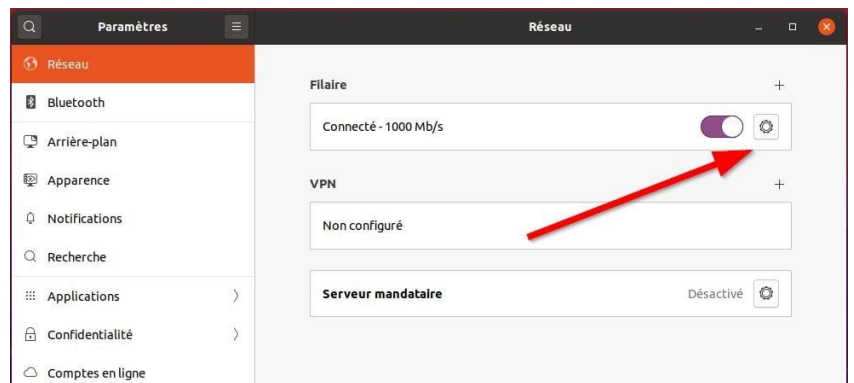


Une fois ici, vous avez passé l'étape la plus dure, allons maintenant configurer notre poste et ses 2 interfaces réseaux. Une qui va nous permettre une connexion à internet et l'autre qui va nous permettre d'établir la communication RPI, VM, Internet.

Sans trop tarder allez-y, cherchez ce petit bouton et allez sur les paramètres réseaux de votre VM :

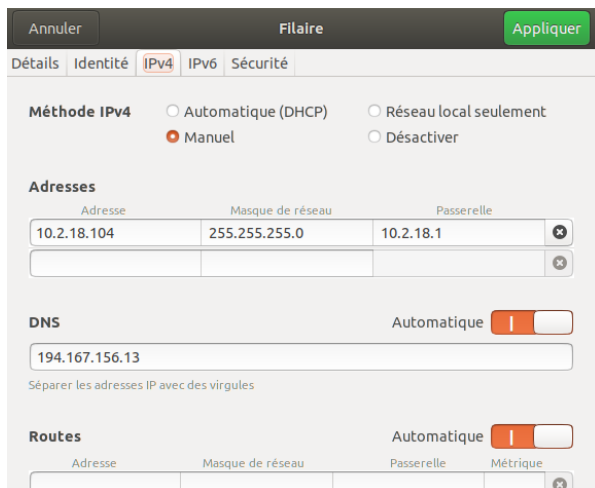


Vous allez y appuyer délicatement, une fois dedans vous allez voir une fenêtre comme celle-ci :



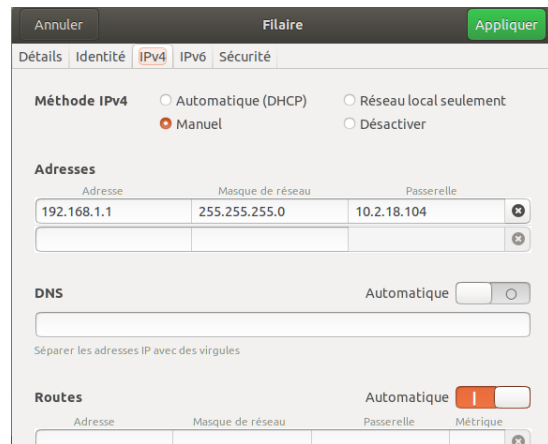
Vous allez devoir appuyer a nouveau toujours délicatement là où la flèche rouge vous pointe. Normalement vous n'allez pas voir juste une interface réseau comme sur la capture, mais 2 interfaces réseaux, une sera Bridge1 et l'autre Pass1, c'est exactement la même chose vous allez voir :

Une fois que vous êtes à l'intérieur des paramètres, cherchez un onglet IPV4, quand vous le trouverez appuyez dessus et vous allez tomber sur quelque chose qui ressemble à ça :



Normalement le champ automatique est déjà pré-saisi, donc vous allez appuyer sur « Manuel », une fois dessus vous allez mettre une des IP dans votre plage IP qui est écrite sur votre machine physique, pour moi ça a été : 10.2.18.104, pour vous ça sera, 10.2.18.XX, vous remplacerez XX avec votre plage IP, pour le masque on gardera toujours un /24 et en passerelle ça sera toujours 10.2.18.1, pour pouvoir accéder correctement à Internet. Dans le champ DNS vous pouvez mettre votre DNS préféré, mais en préférence à l'IUT saisissez : 194.167.156.13.

Bon ce n'était pas aussi compliqué que ce que vous avez cru au début, à partir de là on a fait qu'une des deux interfaces réseau, pour la deuxième allez sur le même menu que je vous ai dit au début et cherchez PASS1. Vous rechercherez de nouveau l'onglet IPV4 et vous tomberez sur ça. La configuration sera un peu différente que celle du BRIDGE1, en adresse IP, honnêtement vous pouvez mettre ce que vous voulez tant que c'est une adresse IP valide, nous avons choisi 192.168.1.1, au masque, nous garderons le masque /24, cependant en passerelle nous allons saisir



l'adresse IP que vous avez mis pour l'interface Bridge1, elle sera notre passerelle, donc vous mettez : 10.2.18.104 ou 10.2.18.XX, XX étant votre plage IP. Dans le DNS on a choisi de ne rien mettre car cela n'a aucun sens : cette interface ne sera pas connectée à Internet elle devra tout simplement rediriger les paquets IP depuis Internet jusqu'à notre RPI.

Avant de continuer, pour pouvoir communiquer avec le monde extérieur on doit importer le proxy de l'IUT. Ouvrez tout simplement votre terminal de commandes (ctrl + alt + T), et saisissez ces deux commandes :

```
export http_proxy=http://cache.univ-pau.fr:3128
export https_proxy=http://cache.univ-pau.fr:3128
```

Normalement, vous pouvez désormais accéder à internet depuis votre VM à votre aise, pour tester ceci, ouvrez à nouveau votre terminal de commandes et saisissez :

`ping google.fr`, comme ça vous pourrez tester votre connexion à internet, et si votre DNS marche bien. Si cette commande a abouti vous devez voir quelque chose comme ça :

```
momo@LAPTOP-TF41B99V:~$ ping google.fr
PING google.fr (142.250.179.99) 56(84) bytes of data.
64 bytes from par21s20-in-f3.1e100.net (142.250.179.99): icmp_seq=1 ttl=117 time=95.8 ms
64 bytes from par21s20-in-f3.1e100.net (142.250.179.99): icmp_seq=2 ttl=117 time=48.8 ms
```

Si vous l'avez, soyez heureux car ça marche !!

Maintenant on va bien se salir les mains car on va rentrer dans le vif de notre sujet, on va configurer le serveur DHCP. Mais attendez, vous ne savez pas ce que c'est non ?

Je vais vous expliquer :

Le protocole DHCP est un service réseau qui permet aux ordinateurs clients, dans ce cas notre RPI, d'obtenir automatiquement une configuration réseau sans avoir à la configurer manuellement. Les ordinateurs configurés pour utiliser DHCP reçoivent leur configuration réseau du serveur DHCP et l'utilisateur ne peut pas en contrôler les détails. Cela rend la configuration réseau transparente et automatique pour l'utilisateur.

Désormais, vous savez ce que DHCP signifie et à quoi il sert, maintenant je vais vous montrer comment faire pour l'installer :

Vous ouvrez votre terminal de commandes (ctrl + alt + T), vous allez saisir tout d'abord

```
sudo apt update -y && sudo upgrade -y
```


Suite à cette commande vous allez voir une grande suite de lignes qui vont apparaître en face de vous, mais n'ayez pas peur, là on est tout simplement en train de faire une mise à jour des paquets que l'on a et on les met à jour pour avoir toutes les dernières versions de notre système.

```
sudo apt install isc-dhcp-server -y
```

Avec cette commande on va installer le service qui va nous permettre de transformer notre machine en serveur DHCP.

Après une longue attente et plusieurs lignes écrites sur notre terminal de commandes, le service « isc-DHCP-server » est finalement installé et prêt à être configuré.

Allons le configurer, allons-y d'abord dans le fichier de configuration de base, celui où on va lui dire tous les paramètres réseaux à envoyer à nos dispositifs qui vont les demander. Pour le modifier on va utiliser l'éditeur de texte « nano », mais vous êtes libres d'utiliser l'éditeur de texte que vous préférez, (vi, vim, emacs, sublime...), normalement nano est déjà installé dans le système opératif Ubuntu. Mais rien ne vous empêche de faire la commande suivante :

```
sudo apt install nano -y
```

Une fois que vous êtes sûr que nano est bien installé sur votre système opératif, allons éditer ce .conf, pour le trouver, c'est simple il est toujours dans le chemin suivant : **/etc/dhcp** Et il est appelé **dhcpd.conf**, maintenant que vous savez où est-ce qu'il est, on va tout simplement dire à nano que nous allons le modifier avec la commande suivante

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

Le sudo est nécessaire, car nous allons modifier le fichier d'un service, une fois dedans vous allez voir un écran comme celui-ci apparaîtra :

Vous pouvez si vous le voulez, tout supprimer ce qu'il y a à l'intérieur car nous allons rajouter notre propre configuration. La configuration de laquelle je parle est la suivante :

```
GNU nano 6.2 /etc/dhcp/
dhcpd.conf
#
# Sample configuration file for ISC dhcpd
#
# Attention: If /etc/ltsp/dhcpd.conf exists, that will be used
# configuration file instead of this file.
#
# option definitions common to all supported networks...
default-lease-time 600;
max-lease-time 7200;
authoritative;
subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.100 192.168.1.200;
    option routers 192.168.1.1;
    option domain-name-servers 194.167.156.13;
}
# The ddns-updates-style parameter controls whether or not the
# attempt to do a DNS update when a lease is confirmed. We default to the
# behavior of the version 2 packages ('none', since DHCP v2 didn't
# have support for DDNS.)
ddns-update-style none;
# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
#authoritative;
```

```
GNU nano 6.2 /etc/d
dhcpd.conf
#
# Sample configuration file for ISC dhcpd
#
# Attention: If /etc/ltsp/dhcpd.conf exists, that will be used as
# configuration file instead of this file.
#
# option definitions common to all supported networks...
option domain-name "example.org";
option domain-name-servers ns1.example.org, ns2.example.org;

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

# The ddns-updates-style parameter controls whether or not the server will
# attempt to do a DNS update when a lease is confirmed. We default to the
# behavior of the version 2 packages ('none', since DHCP v2 didn't
# have support for DDNS.)
ddns-update-style none;

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
#authoritative;

# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
#log-facility local7;

# No service will be given on this subnet, but declaring it helps the
# DHCP server to understand the network topology.
```

Dans cette config qu'on a écrit après, default-lease-time, et Max-lease-time, qui signifient tous les deux, qu'après cette période on doit laisser partir l'IP et en fournir une nouvelle, c'est une configuration nécessaire. Bon après on a l'option *authoritative* cette option ne dit ni plus ni moins que peu importe la requête que le client va envoyer, envoie-lui une nouvelle adresse IP, je ne vais pas m'approfondir

dessus car ce n'est pas l'objectif du TP. Après ces 3 lignes, on a ce que notre serveur DHCP va envoyer aux appareils arrivants, tout d'abord il donnera des adresses IP qui seront comprises entre 192.168.1.100 et 192.168.1.200, deuxièmement il donnera aux appareils la route par défaut, qui en ce cas sera notre interface PASS 1 et finalement il enverra le DNS, on a décidé de mettre le DNS de l'IUT car on restera pour cet SAE que dans cet environnement.

Une fois tout ceci marqué et sauvegardé (rappel : pour sortir et sauvegarder depuis nano la commande est : ctrl + S, ctrl + X), nous allons dire à notre serveur DHCP quelle interface réseaux doit-il utiliser, car on en a 2, pour faire ceci nous devons aller au fichier suivant : */etc/default/isc-dhcp-server*, donc à nouveau il faudra l'ouvrir avec nano :

```
sudo nano /etc/default/isc-dhcp-server
```

Une fois dedans vous allez devoir le modifier d'une telle manière qu'il doit ressembler à ça :

```
Defaults for isc-dhcp-server (sourced by /etc/init.d/isc-dhcp-server)

# Path to dhcpd's config file (default: /etc/dhcp/dhcpd.conf).
#DHCPDv4_CONF=/etc/dhcp/dhcpd.conf
#DHCPDv6_CONF=/etc/dhcp/dhcpd6.conf

# Path to dhcpd's PID file (default: /var/run/dhcpd.pid).
#DHCPDv4_PID=/var/run/dhcpd.pid
#DHCPDv6_PID=/var/run/dhcpd6.pid

# Additional options to start dhcpd with.
# Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead
#OPTIONS=""

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

Vous mettrez en commentaire l'option *interfacesv6*, et à l'intérieur des guillemets d'*interfacesv4* vous allez mettre *ens4*. Mais pourquoi *ens4*, car elle est le nom de notre carte réseaux PASS 1, vous pouvez vérifier cela en saisissant la commande « ip a » et vous allez

remarquer que c'est bien celle-là.

Nous avons fini la configuration de notre serveur DHCP, pour l'activer, nous allons tout d'abord le démarrer puis on verra le statut :

Donc on saisit ces deux commandes :

```
sudo systemctl restart isc-dhcp-server
```

```
sudo systemctl status isc-dhcp-server
```

À l'issue de ces deux commandes vous devez voir quelque chose qui ressemble à ça :

```
supervisor@ubuntu:/etc/default$ nano /etc/default/isc-dhcp-server
supervisor@ubuntu:/etc/default$ sudo systemctl status isc-dhcp-server.service
● isc-dhcp-server.service - ISC DHCP IPv4 server
   Loaded: loaded (/lib/systemd/system/isc-dhcp-server.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2022-10-20 11:26:29 CEST; 13min ago
     Docs: man:dhcpd(8)
    Main PID: 2972 (dhcpd)
      Tasks: 1 (limit: 4915)
   CGroup: /system.slice/isc-dhcp-server.service
           └─2972 dhcpd -user dhcpd -group dhcpd -f -4 -pf /run/dhcp-server/dhcpd.pid -cf /etc/dhcp/dhcpd.conf ens4

oct. 20 11:26:30 ubuntu sh[2972]: Sending on Socket/fallback/fallback-net
oct. 20 11:26:30 ubuntu dhcpd[2972]: Sending on LPF/ens4/52:54:00:e8:c5:75/192.168.1.0/24
oct. 20 11:26:30 ubuntu dhcpd[2972]: Sending on Socket/fallback/fallback-net
oct. 20 11:26:30 ubuntu dhcpd[2972]: Server starting service.
oct. 20 11:28:11 ubuntu dhcpd[2972]: DHCPREQUEST for 192.168.1.100 from dc:a6:32:df:bc:f5 via ens4
oct. 20 11:28:11 ubuntu dhcpd[2972]: DHCPACK on 192.168.1.100 to dc:a6:32:df:bc:f5 (raspberrypi) via ens4
oct. 20 11:33:17 ubuntu dhcpd[2972]: DHCPREQUEST for 192.168.1.100 from dc:a6:32:df:bc:f5 (raspberrypi) via ens4
oct. 20 11:33:17 ubuntu dhcpd[2972]: DHCPACK on 192.168.1.100 to dc:a6:32:df:bc:f5 (raspberrypi) via ens4
oct. 20 11:38:17 ubuntu dhcpd[2972]: DHCPREQUEST for 192.168.1.100 from dc:a6:32:df:bc:f5 (raspberrypi) via ens4
oct. 20 11:38:17 ubuntu dhcpd[2972]: DHCPACK on 192.168.1.100 to dc:a6:32:df:bc:f5 (raspberrypi) via ens4
supervisor@ubuntu:/etc/default$
```

Ça marche ! Maintenant on a plus qu'à préparer notre Raspberry Pi et on aura fini le plus dur de l'SAE.

Si votre résultat ne ressemble pas à celui qui est donnée un peu plus haut, essayez de regarder les erreurs, habituellement ça vient d'une erreur de frappe dans le *dhcpd.conf*.

Maintenant allons configurer ce RP, tout d'abord pour le RP vous allez avoir besoin d'une carte SD et d'un adaptateur carte SD a USB, normalement cela vous est fourni a L'IUT. Pour flasher le système opératif dans la carte SD plusieurs méthodes sont possibles. Nous allons utiliser ici raspberrypi imager, qui va nous servir comme logiciel pour flasher la carte SD et donner vie à notre RP. Tout d'abord pour l'installer et le télécharger nous allons le faire dans un système Ubuntu, c'est-à-dire que si vous voulez flasher depuis un Windows, vous allez devoir juste télécharger un .exe pour le faire, depuis Ubuntu ce n'est pas aussi compliqué, on va saisir cette commande depuis notre terminal de commandes :

```
sudo snap install rpi-imager
```

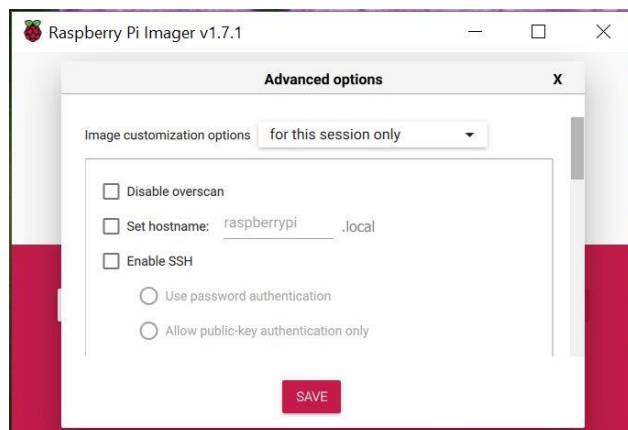
Et voila, maintenant lancez rpi-imager et vous allez avoir un écran comme le suivant :



En choose OS, vous allez chercher pour Raspberry Pi os 32-bit, une fois sélectionne, vous allez chercher dans storage votre carte SD que vous aurez insère dans votre adaptateur et branche dans un des ports du pc.

Une fois que tout est bon, vous allez cliquer sur le petit engrenage qui est apparu en bas a droite :

Cet engrenage vous ouvrira cet écran qui va vous demander si vous voulez activer certaines choses dans votre Raspberry Pi os, nous on utilisera que l'option enable ssh. Une fois coché, vous pouvez modifier le mot de passe et le login comme bon vous le voulez. Une fois que vous avez fini, vous allez tout simplement sortir de la et appuyer sur « Ecrire », la vous allez attendre jusqu'à, que le logiciel vous dise que tout s'est bien passée.



Prenez votre carte SD l'insérer sur l'emplacement spécifiques des cartes SD dans votre RP. Et voilà tout est bon, vous avez fini.

Une fois cela fait, vous allez prendre un câble RJ45 et brancher votre RP a votre carte réseaux PASS1, depuis la baie que vous avez juste devant vous, rappelez-vous qu'il faut la brancher sur la baie du poste, pas sur la baie de la salle. Si vous avez bien configure et que tout s'est bien passée.

Le serveur DHCP aura dû envoyer un configuration réseaux, au nouvel équipement qui est arrivé dans le réseau. D'une manière intuitive nous savons déjà l'adresse IP du RP, car c'est le premier équipement qui s'est branche dans notre réseau donc il aura la première adresse IP de la plage des adresses IP que nous avons saisi. Mais nous n'allons pas nous arrêter la, on va aller chercher quelle adresse IP est-ce que notre serveur DHCP a donnée a notre RP, pour réussir a trouver notre RP dans le réseau, nous allons utiliser la commande NMAP, qui permet entre autres d'identifier toutes les machines qui sont disponibles dans notre réseau.

Pour le faire nous allons tout d'abord installer nmap avec la commande suivante :

```
sudo apt install nmap -y
```

Une fois installé nous allons saisir cette commande pour trouver toutes les machines dans un réseau spécifique, dans ce cas le nôtre :

```
nmap -sV 192.168.1.0/24
```

Cette commande permet de voir toutes les machines connectées dans notre réseau mais en plus elle nous dit quel service dans ses ports est allumé, en ce cas nous allons chercher le port 22 qui sera celui des connexions ssh, car nous voulons nous connecter par ssh à notre Raspberry Pi. À l'issue de cette commande vous allez vous retrouver avec ça :

```
Starting Nmap 7.60 ( https://nmap.org ) at 2022-10-20 11:51 CEST
Nmap scan report for ubuntu (192.168.1.2)
Host is up (0.00036s latency).
All 1000 scanned ports on ubuntu (192.168.1.2) are closed

Nmap scan report for 192.168.1.100
Host is up (0.00068s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
```

Nous pouvons voir que l'adresse IP 192.168.1.100 a le port et donc le service ssh ouverts, donc on en déduit que c'est notre RP.

Maintenant nous allons tout simplement nous connecter à notre RP avec la commande suivante :

```
ssh pi@192.168.1.100
```

Il nous demandera quelque chose comme ça :

```
pi@192.168.1.100's password:
```

Le mot de passe par défaut si vous ne l'avez pas modifié est : *raspberrypi*

Donc saisissez ça et vous aurez accès par ligne de commande à votre Raspberry Pi.

Avant ça vous allez devoir écrire « yes » et voilà vous allez avoir un écran qui est exactement comme ça.

```
pi@192.168.1.100's password:
Linux raspberrypi 5.15.61-v7l+ #1579 SMP Fri Aug 26 11:13:03 BST 2022 armv7l

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 Sep 22 02:20:27 2022 from 192.168.1.2

Wi-Fi is currently blocked by rfkill.
Use raspi-config to set the country before use.

pi@raspberrypi:~ $
```

Vous allez vous en rendre compte très vite que bien que vous ayez très bien configuré votre serveur DHCP, vous n'avez pas accès à internet depuis votre RP. Cela est à cause que le réseau que vous venez de créer est un réseau caché. Vous allez tout simplement partager la connexion internet de votre VM au Raspberry Pi

Pour faire cela va activer tout d'abord le routage sur notre VM, pour l'activer saisissez cette commande dans votre VM :

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Ce que vous venez de faire est d'activer le IP forwarding dans votre VM, comme ça elle va faire fonction de router dans notre réseau, c'est-à-dire elle va diriger des paquets d'une interface à une autre.

Maintenant que nous avons fait ça, nous allons lui dire de rediriger les paquets vers notre interface réseau, PASS1. Pour le faire nous allons utiliser la commande *iptables* :

```
sudo iptables -t nat -A POSTROUTING -o ens3 -j MASQUERADE
```

Une fois que vous avez saisi cette commande dans votre VM, dirigez-vous vers votre Raspberry Pi et importez le proxy :

```
export http_proxy=http://cache.univ-pau.fr:3128  
export https_proxy=http://cache.univ-pau.fr:3128
```

Vous pourrez ping 8.8.8.8, c'est-à-dire le monde extérieur sans aucun problème.

Et voilà nous avons fini notre SAE 12.