Rapport final Pépinière

Creil-Society

Société 5

Pierre Famchon

Nicolas Edouard

Baptiste Duval

Michel Bauchart

Table des matières

1.Gant

Tout d'abord nous avons réalisé un gant pour nous permettre de nous organiser, lors du déroulement du projet.

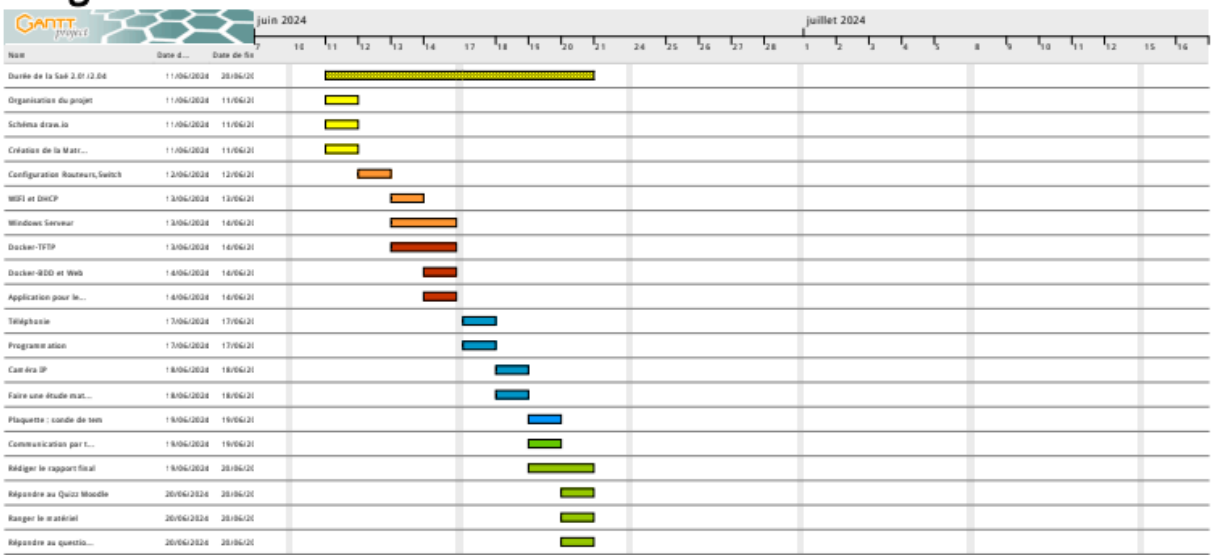
Untitled Gantt Project 11 juin 2024

Tâches 2

Nom	Date de début	Date de fin
Durée de la Saé 2.01/2.04	11/06/2024	20/06/2024
Organisation du projet	11/06/2024	11/06/2024
Schéma draw.io	11/06/2024	11/06/2024
Création de la Matrice RACI et du Diagramme de Gantt	11/06/2024	11/06/2024
Configuration Routeurs,Switch	12/06/2024	12/06/2024
WIFI et DHCP	13/06/2024	13/06/2024
Windows Serveur	13/06/2024	14/06/2024
Docker-TFTP	13/06/2024	14/06/2024
Docker-BDD et Web	14/06/2024	14/06/2024
Application pour les commerciaux	14/06/2024	14/06/2024
Téléphonie	17/06/2024	17/06/2024
Programmation	17/06/2024	17/06/2024
Caméra IP	18/06/2024	18/06/2024
Faire une étude mathématique à partir de la caméra IP	18/06/2024	18/06/2024
Plaquette : sonde de tem	19/06/2024	19/06/2024
Communication par téléphone en Anglais	19/06/2024	19/06/2024
Rédiger le rapport final	19/06/2024	20/06/2024
Répondre au Quizz Moodle	20/06/2024	20/06/2024
Ranger le matériel	20/06/2024	20/06/2024
Répondre au questions de la soutenance	20/06/2024	20/06/2024

Untitled Gantt Project 11 juin 2024

Diagramme de Gantt 3



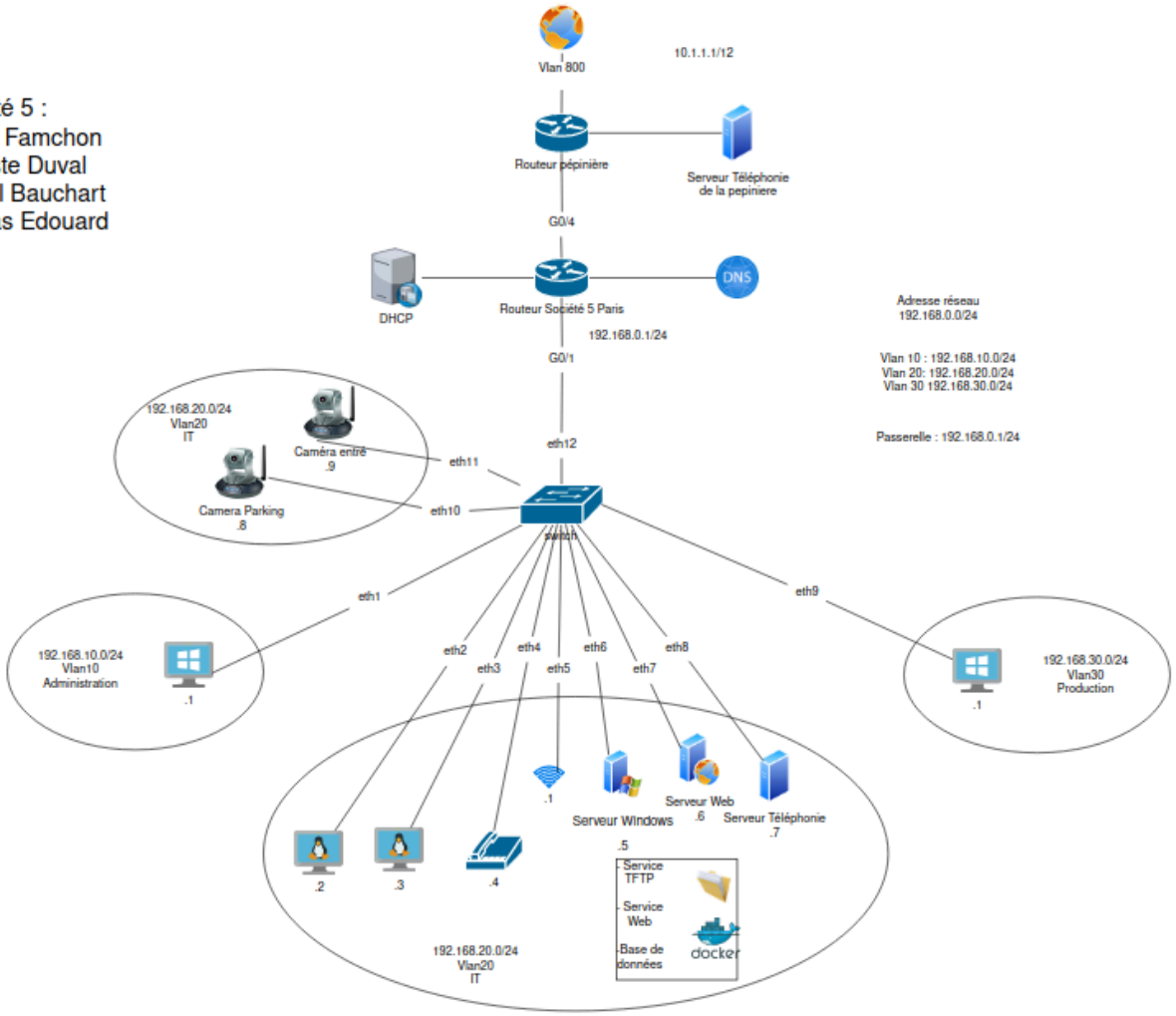
2.Matrice RACI

Nous avons donc répartis les tâches entre nous, sous forme d'une matrice RACI, nous sommes resté très fidèle à cette matrice même si lors d'un projet avec tant de tâche à prévoir tout ne se déroule pas comme prévu.

	A	B	C	D	E
1		Nicolas	Pierre	Baptiste	Michel
2	Gant provisionnel	R, A	R	I	C
3	Matrice RACI	I	R, A	I	C
4	Infrastructure réseau	C	C	R, A	R
5	Cablage	A C	I	C	R
6	Configuration Routeur	C	C	R, A	I
7	Configuration Switch	R, A	C	R	I
8	WIFI	I	I	I	R, A
9	DHCP	C	R, A	I	I
10	DNS	C	R, A	I	I
11	ACL	R	I	R, A	I
12	Windows-serveur	I	I	I	R, A
13	Docker-TFTP	C	R, A	I	I
14	Docker-BDD et web	C	R, A	I	I
15	Téléphonie	I	C	R	I
16	Programmation	R, A	C	R	I
17	Télécommunication	I	I	R	R A
18	Camera IP	R, A	R	C	C
19	Thermomètre	R, A	I	R	R
20	Faire une étude mathématique	I	I	I	R
21	Communication téléphonique	R, A	C	I	I
22	Rédiger le rapport final	C	R, A	C	C

3.Plan du réseau : plan draw.io

Société 5 :
Pierre Famchon
Baptiste Duval
Michel Bauchart
Nicolas Edouard



4.Câblage

Un tableau représentant le câblage de notre réseau à été mis en place, pour nous permettre de revenir facilement sur notre câblage réseau en cas de problème, mais aussi pour permettre une meilleure lisibilité, permettant de se retrouver à travers tous nos câbles et ceux des autres sociétés.

	Ports	Services (Vlan)	Int Client Switch
Pierre	04.2	Production(30)	Fa0/9
Nicolas	01.2	Administration(10)	Fa0/1
Baptiste	02.2	IT(20)	Fa0/2
Michel	03.2	IT(20)	Fa0/3

Équipements	Int sur Switch	IP
Caméra 1	Fa0/10	192.168.20.9
Caméra 2	Fa0/11	192.168.20.10
Wifi	Fa0/5	192.168.20.8
Serveur Windows	Fa0/6	192.168.20.5
Serveur Web	Fa0/7	192.168.20.6
Serveur Téléphonie	Fa0/8	192.168.20.7
Téléphone	Fa0/4	192.168.10.4

Interface Routeur	Utilisation
G0/1	trunk (aussi pour Fa0/12)
G0/4	Passerelle (pour vlan800)

5.Configuration du Switch

Après la partie câblage nous commençons dans une première partie par la configuration du Switch :

Dans un premier temps nous renommons notre switch en Paris avec la commande :

- hostname Creil

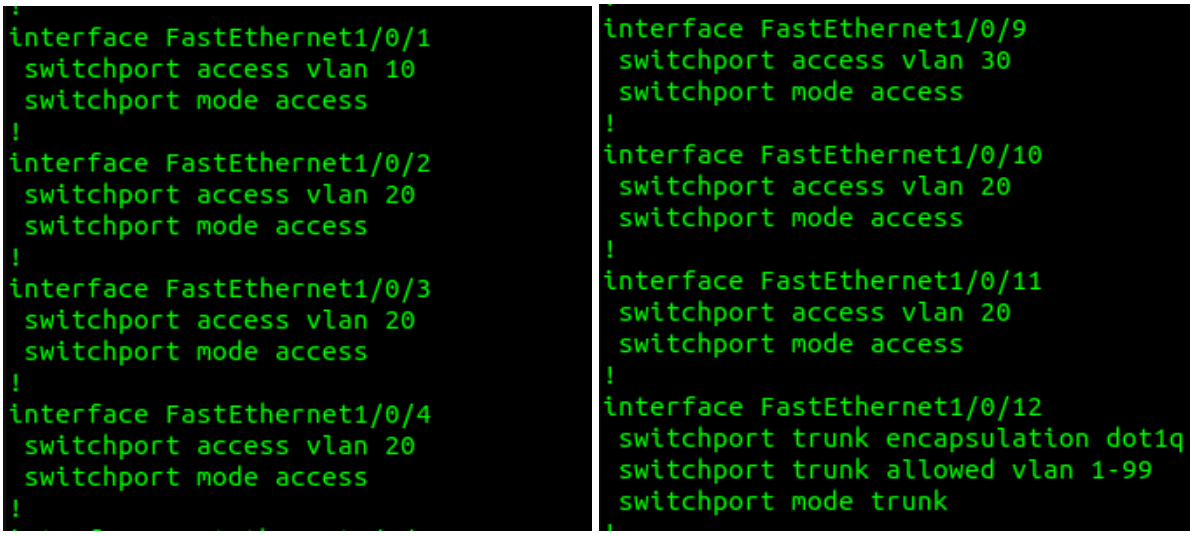
Ensuite dans un second temps nous mettons nos différents vlan : 10, 20, 30, sur différentes interfaces avec la commande :

- Interface FastEthernet0/X
  switchport access vlan XX
  switchport mode access

Puis nous créons un trunk pour relier les vlan en eux et leurs permettre de communiquer entre eux sur l'interface Fa1/0/12
Pour cela nous rajoutons le commande :

- switchport trunk encapsulation dot1q
- switchport trunk allowed vlan 1-99

Exemple juste en dessous :



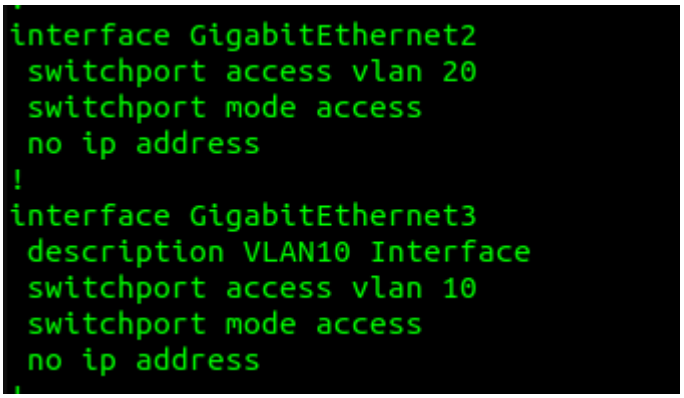
6.Configuration du Routeur

Ensuite nous avons fait la configuration de base de notre routeur (même s'il possède déjà la configuration du téléphone, que nous verrons plus tard) :

Tout d'abord nous avons configuré nos interfaces :

Nous reprenons la configuration des interfaces du Switch :

Interface GigabitEthernet 0, 2 et 3 :



Interface Vlan :

Ensuite nous configurons nos vlan en associant l'adresse réseau de chaque vlan : 10, 20, 30 :

interface vlanXX
ip address 192.168.XX.1 255.255.255.0

```
interface Vlan10
 ip address 192.168.10.1 255.255.255.0
 ip access-group 10 out
 ip nat inside
 ip virtual-reassembly in
!
interface Vlan20
 ip address 192.168.20.1 255.255.255.0
 ip access-group 20 out
 ip nat inside
 ip virtual-reassembly in
!
interface Vlan30
 ip address 192.168.30.1 255.255.255.0
 ip access-group 30 out
 ip nat inside
 ip virtual-reassembly in
```

Interface GigabitEthernet1 :

Puis nous créons un trunk sur l'interface GigabitEthernet1 qui regroupe tous nos vlan qui sont compris sur la plage 1-1005, pour permettre aux vlan de communiquer entre eux :

```
interface GigabitEthernet1
 switchport trunk allowed vlan 1-1005
 switchport mode trunk
 no ip address
```

Interface GigabitEthernet4 :

Et pour finir nous configurons l'adresse de l'interface GigabitEthernet4 en lui associant l'ip du routeur pépinière : 10.5.1.1 et son masque 255.240.0.0

```
interface GigabitEthernet4
 ip address 10.5.1.1 255.240.0.0
 ip nat outside
 ip virtual-reassembly in
 duplex auto
 speed auto
```

7.DHCP, DNS

Nous avons ensuite avec activer le DHCP, et le DNS-serveur, pour cela nous avons tout d'abord :

- exclu les adresses des serveurs windows, web et de la caméra, respectivement 20.5, 20.6, 20.12

- désigné le vlan sur lequel configurer le dhcp : ip dhcp pool vlanXX,
- puis l'adresse réseau de ce vlan avec : network 192.168.XX.0, le masque : 255.255.255.0

- puis sa route par défaut : default-router 192.168.XX.1

- et pour finir l'on active le dns serveur : 10.0.0.1 (dns-server pépinière), puis une fois le dns server de windows server déployé, nous mettons l'adresse ip du serveur windows pour pouvoir sortir du réseau et re-rentrer ensuite pour pouvoir, par exemple accéder à notre site web

```
ip dhcp excluded-address 192.168.20.5
ip dhcp excluded-address 192.168.20.6
ip dhcp excluded-address 192.168.20.12
!
ip dhcp pool vlan10
 network 192.168.10.0 255.255.255.0
 default-router 192.168.10.1
 dns-server 192.168.20.5
!
ip dhcp pool vlan20
 network 192.168.20.0 255.255.255.0
 default-router 192.168.20.1
 dns-server 192.168.20.5
!
ip dhcp pool vlan30
 network 192.168.30.0 255.255.255.0
 default-router 192.168.30.1
 dns-server 192.168.20.5
```

8.NAT

Une fois le DHCP et le serveur DNS mis en place, l'on ajoute le NAT pour avoir accès à internet.
L'on rajoute dans la configuration du routeur précédemment créé 3 choses :

- tout d'abord dans chaque interfaces de vlan il nous faut rajouter la commande ip nat inside et ip nat outside

```
interface GigabitEthernet4
 ip address 10.5.1.1 255.240.0.0
 ip nat outside
 ip virtual-reassembly in
 duplex auto
 speed auto
!
interface Vlan1
 no ip address
 shutdown
!
interface Vlan3
 no ip address
!
interface Vlan10
 ip address 192.168.10.1 255.255.255.0
 ip access-group 10 out
 ip nat inside
 ip virtual-reassembly in
```

- ensuite l'on fait le commande : ip nat inside source list NAT_VLANXX 1 interface GigabitEthernet4 overload et pour finir nous rajoutons une route vers le routeur pépinière : ip route 0.0.0.0 0.0.0.0 10.0.0.1

```
ip nat inside source static tcp 192.168.20.6 80 interface GigabitEthernet4 80
ip nat inside source list NAT_VLAN10 interface GigabitEthernet4 overload
ip nat inside source list NAT_VLAN20 interface GigabitEthernet4 overload
ip nat inside source list NAT_VLAN30 interface GigabitEthernet4 overload
ip route 0.0.0.0 0.0.0.0 10.0.0.1
ip ssh version 2
```

(Les acces-list : NAT_VlAN10 - 30 seront présentés dans la partie suivante)

9.ACL

Nous reprenons la configuration précédente et plus particulièrement les access-list déjà créer et utiliser pour le NAT

```
!
ip access-list extended NAT_VLAN10
 permit ip 192.168.10.0 0.0.0.255 any
ip access-list extended NAT_VLAN20
 permit ip 192.168.20.0 0.0.0.255 any
ip access-list extended NAT_VLAN30
 permit ip 192.168.30.0 0.0.0.255 any
```

Ensuite nous ajoutons des ACL pour permettre de gérer les communications entre les vlan 10, 20, 30.

Nous avons comme cahier des charges :

- de permettre au vlan 20 (IT) de communiquer avec tous les vlan, mais les autres vlan ne peuvent pas accéder au vlan 20 (IT),
- puis de permettre la disponibilité des services du vlan 20 (windows-serveur, serveur-web...) à tous les vlan,
- et pour finir les autres vlan 10 et 30 ne peuvent pas communiquer entre eux.

Tout d'abord nous commençons par créer nos 3 acces-list extended VLAN10_TO_VLAN30, VLAN30_TO_VLAN10, et VLAN20_TO_OTHER, qui vont répondre respectivement au 3 points du cahier des charges.

Pour les deux premières acl extended : VLAN10_TO_VLAN30 et VLAN30_TO_VLAN10

L'ordre est très important pour les acl on ajoute les commandes :

- deny icmp 192.168.10.0 0.0.0.255 192.168.30.0 0.0.0.255 :
  Cette commande permet de retirer l'autorisation au vlan 10 de communiquer avec le réseau du vlan 30 : 192.168.30.0.
- permit ip any host 192.168.20.5 :
  Autorise le vlan10 à communiquer avec le serveur Windows se trouvant dans le vlan20.
- permit ip any host 192.168.20.6 :
  Autorise le vlan10 à communiquer avec le serveur Web se trouvant dans le vlan20.
- deny icmp 192.168.10.0 0.0.0.255 192.168.20.0 0.0.0.255 :

Refuse du vlan10, l'accès au vlan 20.

- permit ip any any :
  Autorise tout autre type de communication.

```
ip access-list extended VLAN10_TO_VLAN30
 deny   icmp 192.168.10.0 0.0.0.255 192.168.30.0 0.0.0.255
 permit ip any host 192.168.20.5
 permit ip any host 192.168.20.6
 deny   icmp 192.168.10.0 0.0.0.255 192.168.20.0 0.0.0.255
 permit ip any any
```

```
ip access-list extended VLAN30_TO_VLAN10
 deny   icmp 192.168.30.0 0.0.0.255 192.168.10.0 0.0.0.255
 permit ip any host 192.168.20.5
 permit ip any host 192.168.20.6
 deny   icmp 192.168.30.0 0.0.0.255 192.168.20.0 0.0.0.255
 permit ip any any
```

Pour la dernière acl extended : VLAN20_OTHERS

L'ordre est encore une fois très important pour les acl on ajoute les commandes :

- permit icmp 192.168.20.0 0.0.0.255 192.168.10.0 0.0.0.255 :
  Permet d'autoriser le vlan 20 à communiquer avec le réseau du vlan 10 : 192.168.30.0.
- permit icmp 192.168.20.0 0.0.0.255 192.168.30.0 0.0.0.255 :
  Autorise du vlan20, l'accès au vlan30.
- permit ip any any :
  Autorise tout autre type de communication.

```
ip access-list extended VLAN20_TO_OTHERS
 permit icmp 192.168.20.0 0.0.0.255 192.168.10.0 0.0.0.255
 permit icmp 192.168.20.0 0.0.0.255 192.168.30.0 0.0.0.255
  permit ip any any
```

10.Wifi

Nous avons déployé notre deuxième service : le service WIFI :

Dans un premier temps nous avons commencer par configurer l'identifiant SSID et les paramètres pour le réseau invité

Cisco

Tout d'abord nous avons commencé par renommer la borne wifi :

- hostname ap

Nous avons donc continué en mettant en place une limite de taux de messages et un mot de passe secret chiffré pour l'utilisateur avec deux commandes :

- logging rate-limit console 9
- enable secret 5 1IKxr$xkmCdqrIuDCOj2mWce6h00

Puis nous avons configuré l'identifiant (SSID) en temps que "entreprise1" :

dot11 ssid entreprise1 :

- authentication open
- authentication key-management wpa version 2
- guest-mode
- wpa-psk ascii 7 1315051D0C18167A7B

dot11 ssid entreprise1 : permet de configurer le nom du Service Set Identifier (SSID), associé à une interface
auth key wpa ver 2 : permet de configurer les paramètres d'authentification d'un réseau sans fil en utilisant le protocole WPA2 (Wi-Fi Protected Access 2)
guest-mode : permet d'activer le mode invité sur un réseau sans fil.
wpa-psk ascii 7 1315051D0C18167A7B : permet définir la clé pré-partagée (Pre-Shared Key, PSK)

Ensuite nous avons configuré les paramètres invité :

dot11 guest :

- username Cisco password 7 123A0C041104
- bridge irb

dot11 guest : permet de configurer les paramètres pour le réseau invité
username Cisco password 7 123A0C041104 : permet de créer un nom d'utilisateur : cisco avec un mot de passe chiffré
bridge irb : permet d'activer le routage (IRB)

Dans un second temps nous avons configurer les interfaces Radio 0 et 1, GigaBitEthernet0, et l'interface BVI1 :

Nous avons donc commencé par configurer les deux première interface : Radio 0 et 1 :

interface Dot11Radio0 :

- no ip address
- encryption mode ciphers aes-ccm
- ssid entreprise1
- antenna gain 0
- station-role root
- bridge-group 1
- bridge-group 1 subscriber-loop-control
- bridge-group 1 spanning-disabled
- bridge-group 1 block-unknown-source
- no bridge-group 1 source-learning
- no bridge-group 1 unicast-flooding

int doit11Radio 0 : permet d'accéder à l'interface radio dot11Radio 0
no ip address : permet de supprimer l'adresse ip de l'interface
encryption mode ciphers aes : permet de configurer le mode de chiffrement
ssid entreprise1 : permet d'accéder à la configuration global
antenna gain 0 : permet de modifier le gain de l'antenne à 0
station-role root : permet de définir le rôle de l'interface en temps que point d'accès principal

Nous avons utiliser ses configurations pour le groupe 1 que nous allons reprendre pour chaque interface :

bridge-group 1 : permet d'associer interface au groupe 1
bridge-group 1 subscriber-loop-control : permet d'activer la boucle subscriber pour le groupe 1
bridge-group 1 spanning-disabled : permet de désactiver le protocole de spanning Tree pour le groupe 1
bridge-group 1 block-unknown-source : permet de bloquer les ource inconnues dans le groupe 1
no bridge-group 1 source-learning : permet de désactiver l'apprentissage source pour le groupe 1
no bridge-group 1 unicast-flooding : permet de désactiver la diffusion unicast pour le groupe 1

interface Dot11Radio1 :

- no ip address
- shutdown
- antenna gain 0
- peakdetect
- no dfs band block
- channel dfs
- station-role root
- bridge-group 1
- bridge-group 1 subscriber-loop-control
- bridge-group 1 spanning-disabled
- bridge-group 1 block-unknown-source
- no bridge-group 1 source-learning
- no bridge-group 1 unicast-flooding

shutdown : permet de basculer cette interface en mode arrêt
peakdetect : permet de détecter des pics de signal
no dfs band block : permet d'indiquer que la bande DFS est n'est pas bloquée
channel dfs : permet de configurer le canal de l'interface pour utiliser DFS

interface GigabitEthernet0 :

- no ip address
- duplex auto
- speed auto
- bridge-group 1
- bridge-group 1 spanning-disabled
- no bridge-group 1 source-learning

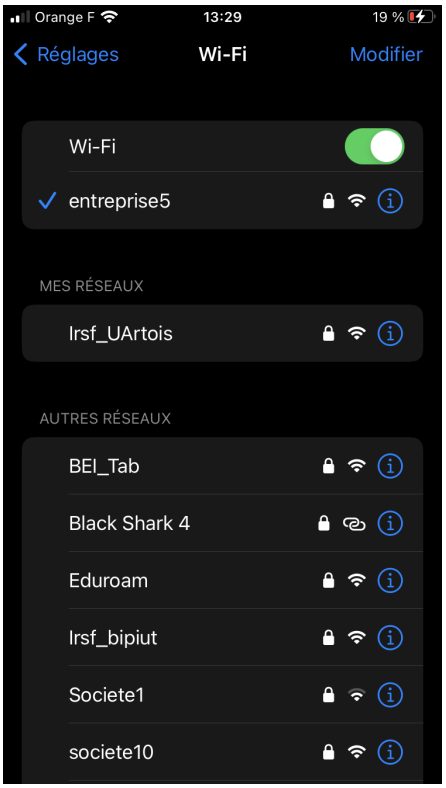
duplex auto : permet de configurer le mode duplex automatique, ce qui permet à l'interface de négocier automatiquement le mode de communication duplex avec l'appareil auquel elle est connecté
speed auto : permet de configurer la vitesse de liaison automatique, ce qui permet à l'interface de négocier la vitesse de transmission des données

interface BVI1 :

- ip address 192.168.60.60 255.255.255.0
- ipv6 address dhcp
- ipv6 address autoconfig
- ipv6 enable

ip address : permet de configurer l'adresse ip de l'interface BVI1
ipv6 address dhcp : permet de d'obtenir une adresse ipv6 dynamique
ipv6 address autoconfig : permet de configurer de manière automatique l'adresse ipv6
ipv6 enable : permet d'activer la prise en charge de l'ipv6

Une fois tout cela terminé, nous avons pu nous connecter au réseau WIFI depuis nos téléphone portable :


Orange F
13:29
19 %
Réglages
Wi-Fi
Modifier
Wi-Fi
entreprise5
MES RÉSEAUX
Irsf_UArtois
AUTRES RÉSEAUX
BEI_Tab
Black Shark 4
Eduroam
Irsf_bipiut
Societe1
societe10

11.Caméra IP

Pour configurer la caméra du modèle FE8391-V, il suffit de suivre ces étapes pour connecter et configurer la caméra.

Tout d'abord il faut brancher la caméra sur le switch de notre entreprise qui est configuré avec le protocole DHCP pour que lorsque la caméra soit branchée sur notre commutateur celle-ci est une adresse IP et soit trouvable sur le réseau.

Pour cela nous avons utilisé la commande arp-scan avec la démarche suivante :

```
administrateur@rt-mv:~$ sudo apt-get install arp-scan
```

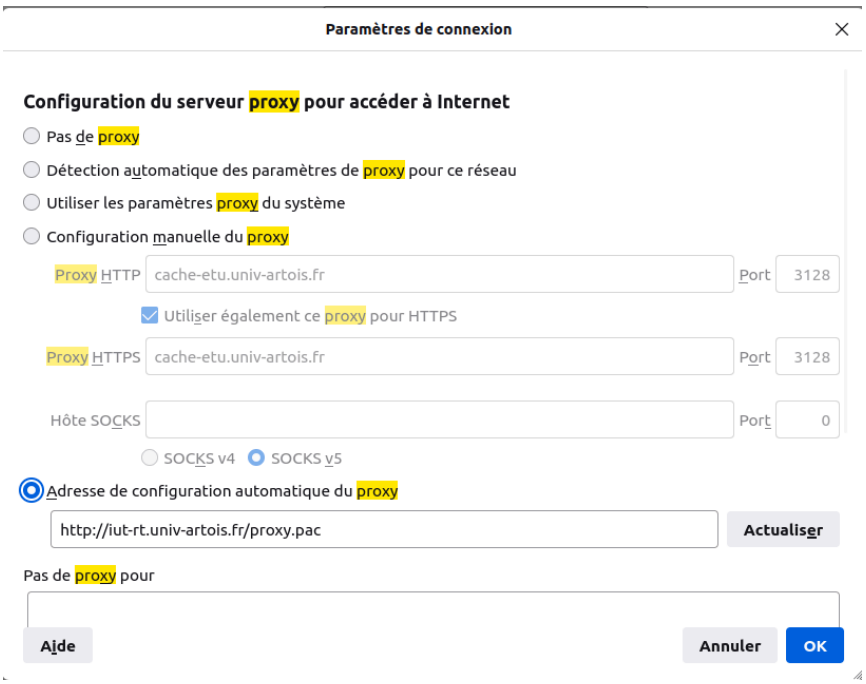
Puis on a besoin d'exécuter cette commande pour analyser toute les adresses et appareils connecté sur le réseau du VLAN 20 :

```
administrateur@rt-mv:~$ sudo arp-scan --interface=enp0s3 --localnet
[sudo] Mot de passe de administrateur :
Interface: enp0s3, type: EN10MB, MAC: 08:00:27:70:82:f0, IPv4: 192.168.20.10
Starting arp-scan 1.9.7 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.20.1    3c:51:0e:03:03:d0       Cisco Systems, Inc
192.168.20.3    08:00:27:00:02:12       PCS Systemtechnik GmbH
192.168.20.5    08:00:27:00:02:11       PCS Systemtechnik GmbH
192.168.20.8    64:9e:f3:b3:7f:b0       Cisco Systems, Inc
192.168.20.12   00:02:d1:57:4e:77       Vivotek, Inc.

23 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.9.7: 256 hosts scanned in 2.053 seconds (124.70 hosts/sec). 5
responded
```
n

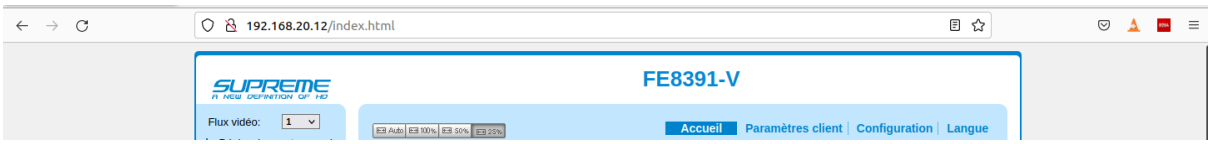
Comme on peut le voir ci-dessous, on a la caméra de l'entreprise Vivotek qui a eu une adresse IP automatiquement distribuée.

Avec cette adresse on peut se connecter sur l'interface Web de la caméra mais d'abord on a besoin de changer le proxy configuré de base :



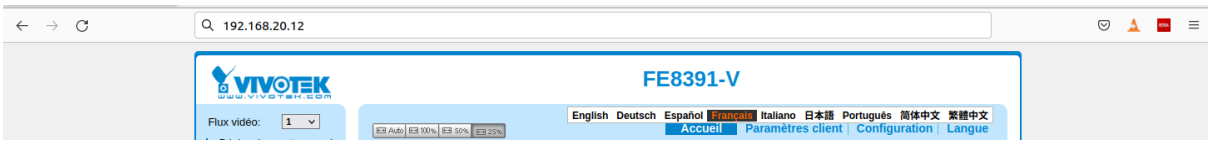
Ici nous allons utiliser le proxy.pac de l'IUT pour pouvoir accéder à l'interface Web car sinon elle est bloquée par le réseau de l'artois.

Une fois cet étape réalisé, on peut s'intéresser à l'interface Web de la caméra et la configurer :



Lorsque l'on se rend sur l'adresse IP de la caméra nous arrivons directement sur son interface Web, maintenant il faut s'intéresser à la configuration pour récupérer l'image de la caméra.

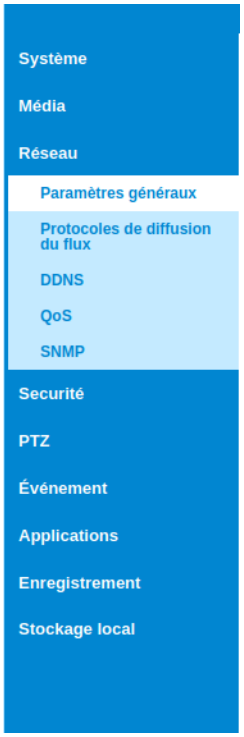
D'abord on va se rendre dans la langue et mettre en français pour que l'on comprenne mieux les menus de configuration :



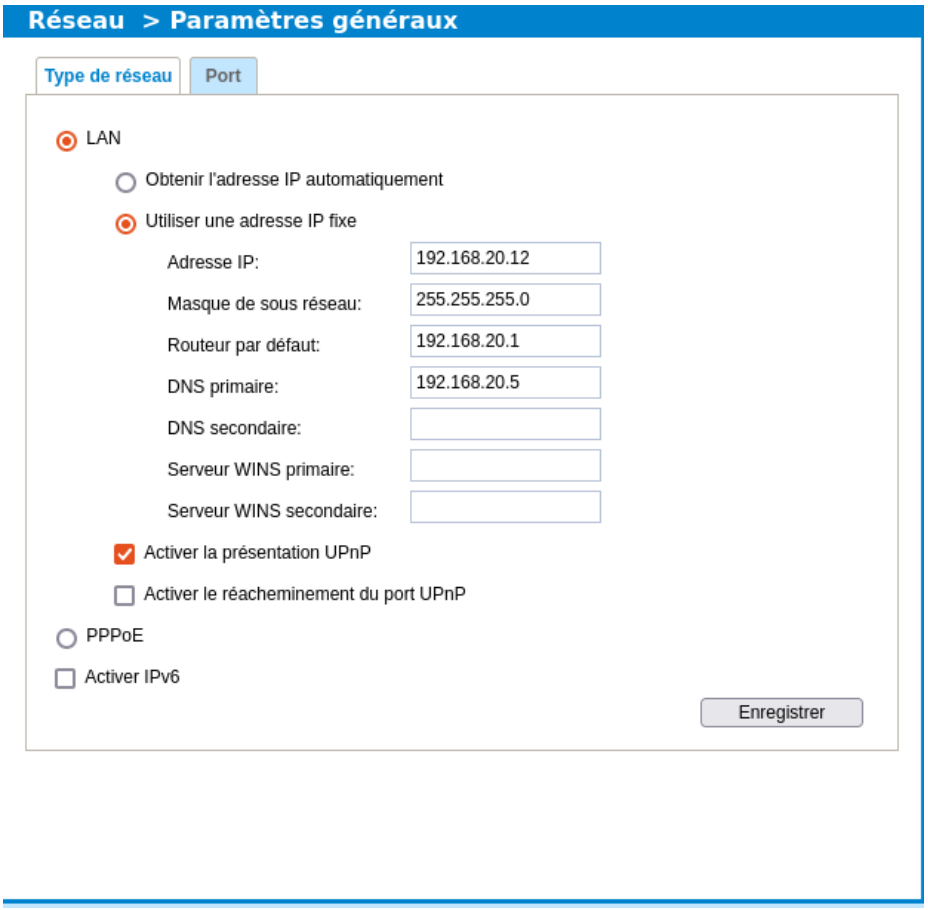
Comme on le voit en cliquant sur langue on a toutes ces langues à disposition.

Ensuite on va pouvoir s'intéresser à la configuration, on va se rendre dans l'onglet Configuration puis nous allons donner une adresse IP fixe à la caméra pour que l'interface Web ne soit pas trouvable et qu'il fasse refaire une recherche sur le réseau à chaque fois.

Pour cela dans l'onglet configuration on va se rendre dans " Réseau " puis aller dans " Paramètres généraux " :

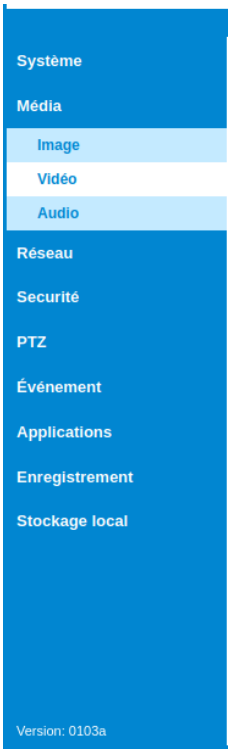


Une fois rendu ici, nous avons ces informations que l'on peut compléter pour adresser une IP fixe à la caméra pour que en plus de ça elle reste dans le réseau qui lui est attribué, dans notre cas elle est situé dans le VLAN 20 service IT :

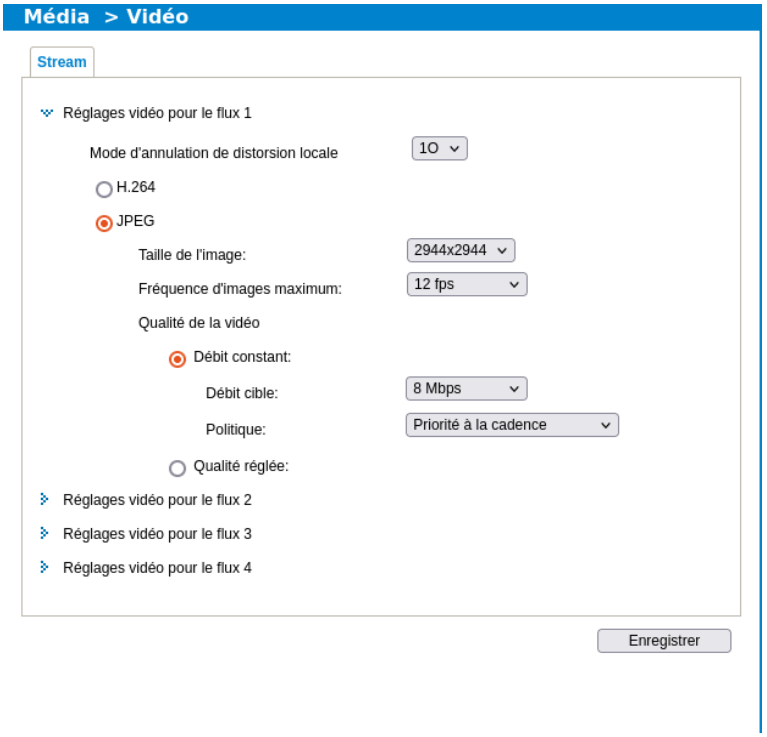


Avec la configuration suivante, maintenant l'interface Web sera toujours sur l'adresse " 192.168.20.12 " et en plus on peut communiquer avec si on le souhaite avec un ping par exemple.

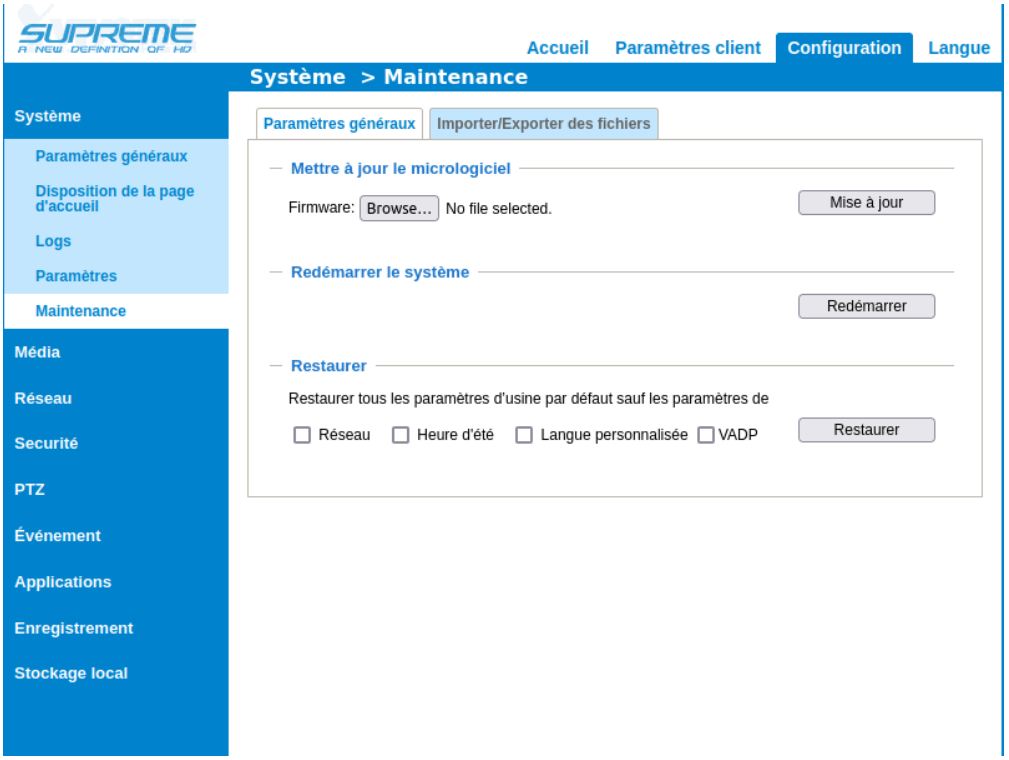
Une fois ceci réalisé, on va pouvoir s'intéresser à la configuration de l'image de la caméra et pour cela on va se rendre dans l'onglet " Média " puis
" Vidéo" :



Une fois rendu dans ces onglets on va régler le flux vidéo de la caméra :

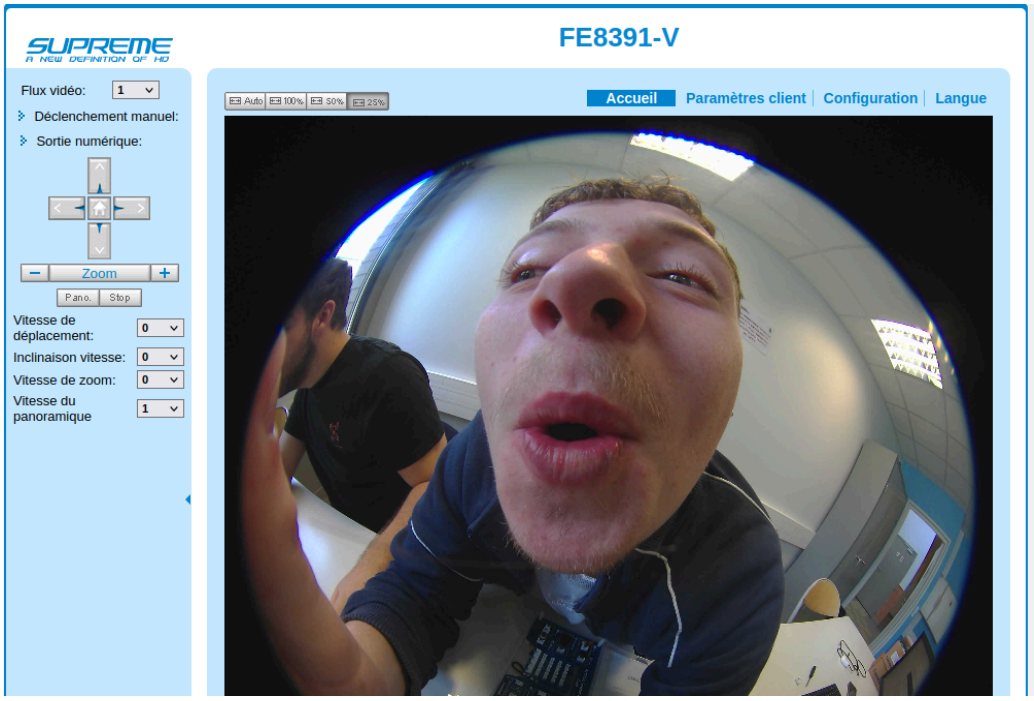


Une fois configuré avec ces paramètres, le flux de la caméra s'affichera sur la page principal de la caméra mais il faut d'abord redémarrer la caméra. Pour cela on va se rendre dans l'onglet " Système " puis " dans " Maintenance " :



Une fois dans cet onglet on va pouvoir redémarrer la caméra et en se rendant dans l'onglet " Accueil ".

On peut voir directement le flux vidéo de la caméra :



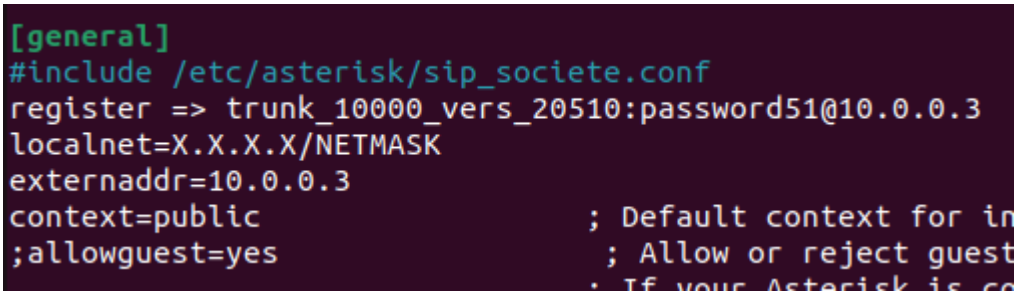
12. Serveur-téléphone

Dans la suite de ce projet, nous avons commencé par déployer notre premier service, le serveur téléphonique.
Nous avions 4 fichiers de configuration à modifier avant de pouvoir accéder à la page Fanvil du téléphone :

- sip.conf
- sip_societe.conf
- extensions
- extensions_societe.conf

sip.conf :

- register => trunk_10000_vers_20510:password51@10.0.0.3 : indique à Asterisk de s'enregistrer auprès du serveur SIP de la pépinière (les identifiants se trouvent dans sip_societe.conf).
- localnet=X.X.X.X/NETMASK : indique le réseau local.
- externaddr=10.0.0.3 : indique l'adresse IP externe du serveur Asterisk.
- context=public : contexte est un ensemble de règles dans le fichier extensions.conf et plus précisement dans la section [public].



sip_societe.conf :

<u>On commence par créer un user :</u>

- type = friend : permet à cet utilisateur d'initier et de recevoir des appels.
- username = user51 : nom d'utilisateur qu'on utilisera pour s'authentifier auprès du serveur Asterisk.
- secret = 5555 : mot de passe de user51 pour s'identifier.
- host = dynamic : indique que user51 est dynamique et de s'enregistrer auprès du serveur Asterisk.
- context = societe5 : indique que les règles d'appel de user51, sont dans le fichier extensions.conf, dans la section [societe5].
- directmedia = no : indique que tous les flux de médias passeront par

le serveur Asterisk.

Ensuite on fait un trunk pour connecter deux systèmes de téléphonie IP :

- host = 10.0.0.3 : indique l'adresse IP du serveur SIP avec lequel ce tronc doit communiquer.
- allow = ulaw : indique le codec utilisé pour effectuer la compression ou la décompression de fichiers multimédias comme les chansons ou les vidéos.
- insecure = port,invite : permet des connexions sans authentification stricte.
- nat = yes : indique à Asterisk d'activer des réglages spécifiques
- qualify = yes : permet de vérifier régulièrement la disponibilité du trunk.

```
[user51]
type = friend
username = user51
secret = 5555
host = dynamic
context = societe5
directmedia = no

[trunk_20510_vers_10000]
type = friend
secret = password51
context = societe5
host = 10.0.0.3
allow = ulaw
insecure = port,invite
nat = yes
qualify = yes
```

extensions.conf :

ce code définit un contexte public et inclut les extensions et règles du contexte societe5 ainsi que celles définies dans le fichier /etc/asterisk/extensions_societe.conf

```
[public]
include => societe5
#include /etc/asterisk/extensions_societe.conf
```

extensions_societe.conf :

ce code définit comment les appels sont routés dans le context 'societe5'.

- exten=>20510,1,dial(SIP/user51) : cette ligne spécifie que lorsqu'un appel est dirigé vers l'extension 20510, Asterisk doit essayer de joindre SIP/user51.
- exten=>_10005,1,dial(SIP/trunk_20110_vers_10000/${EXTEN}) signifie que lorsque l'extension 10005 est composée, elle exécutera l'action dial(SIP/trunk_20110_vers_10000/${EXTEN}).

  Cela suppose qu'il existe une configuration appropriée pour le tronc SIP trunk_20110_vers_10000 et les utilisateurs ou les numéros de destination correspondants. Et de même pour les autres lignes.

```
[societe5]
exten=>20510,1,dial(SIP/user51)

exten=>_10005,1,dial(SIP/trunk_20510_vers_10000/${EXTEN})
exten=>_10004,1,dial(SIP/trunk_20510_vers_10000/${EXTEN})
exten=>_10003,1,dial(SIP/trunk_20510_vers_10000/${EXTEN})
exten=>_10002,1,dial(SIP/trunk_20510_vers_10000/${EXTEN})
exten=>_10001,1,dial(SIP/trunk_20510_vers_10000/${EXTEN})
exten=>_10000,1,dial(SIP/trunk_20510_vers_10000/${EXTEN})

exten=>_20XXX,1,dial(SIP/trunk_20510_vers_10000/${EXTEN})
```

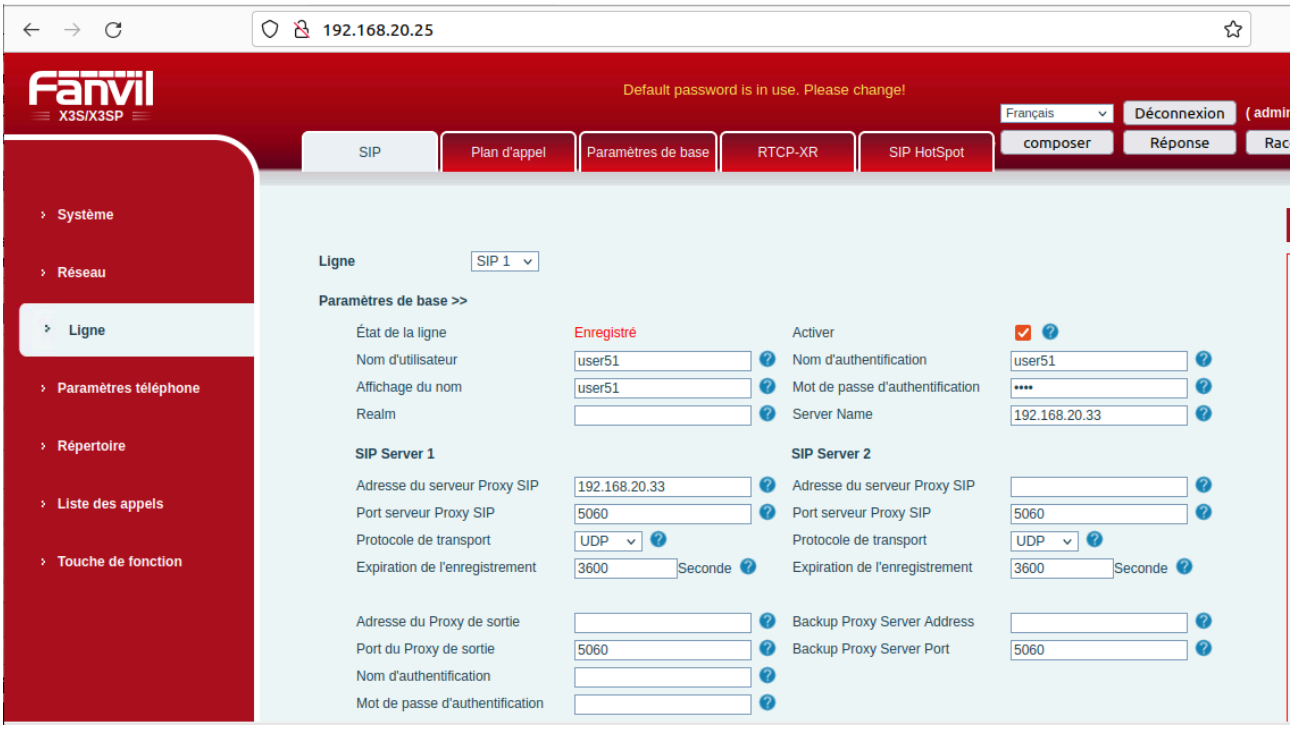
Ensuite on configure le fanvil : 192.168.20.25

On ajoute 3 choses dans notre fanvil : user, password et adresse du serveur proxy SIP (Serveur asterisk).

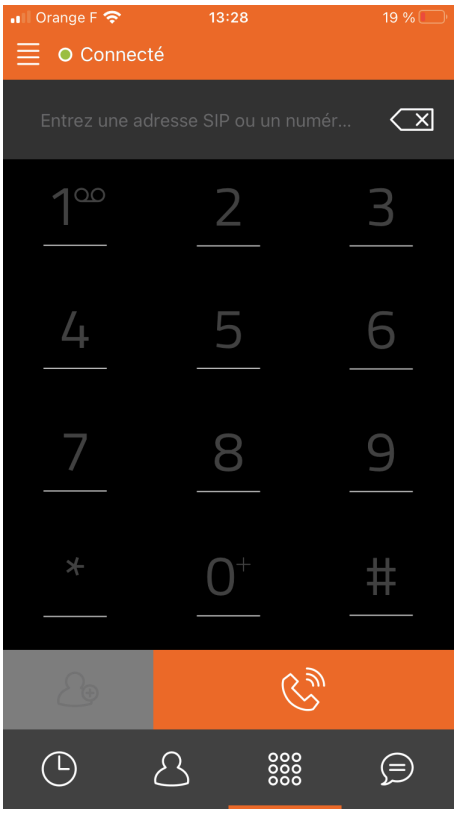
- tout d'abord nous ajoutons notre user 51, préalablement configuré dans le fichier sip_societe.conf.

- ensuite nous rajoutons le password : 5555, lui aussi dans le fichier sip_societe.conf.

- et pour finir on ajoute adresse IP du serveur proxy SIP (Serveur asterisk) : 192.168.20.33



On rajoute un user52 dans le fichier sip_societe.conf. En se connectant au wifi de notre societe5, et en installant linphone nous nous connectons au compte SIP et notre softphone fonctionne :



Prioriser les communications téléphonique sur le routeur :

On commence par créer 3 class-map : DATA, VOICE et SIGNALING

- match ip dscp default : indique que des paquets DSCP de la classe DATA, ont une valeur par défaut à 0, cela sert aux **données générales**.

- match ip dscp ef : les paquets de la classe VOICE, ont une valeur DSCP à 46 (Expedited Forwarding, valeur DSCP 46), cela sert pour le **trafic de voix**.

- match protocole sip : les paquets de la classe SIGNALING, utilise le protocole SIP (Session Initiation Protocol), cela sert à la **signalisation des appels**.

```
!
class-map match-any DATA
 match ip dscp default
class-map match-any VOICE
 match ip dscp ef
class-map match-any SIGNALING
 match protocol sip
!
```

Ensuite on vient réutiliser les class-map créé plus haut et on crée 1 policy-map : VOICE_POLICY

- class VOICE : applique les règles de la classe VOICE (idem pour les autres classes).

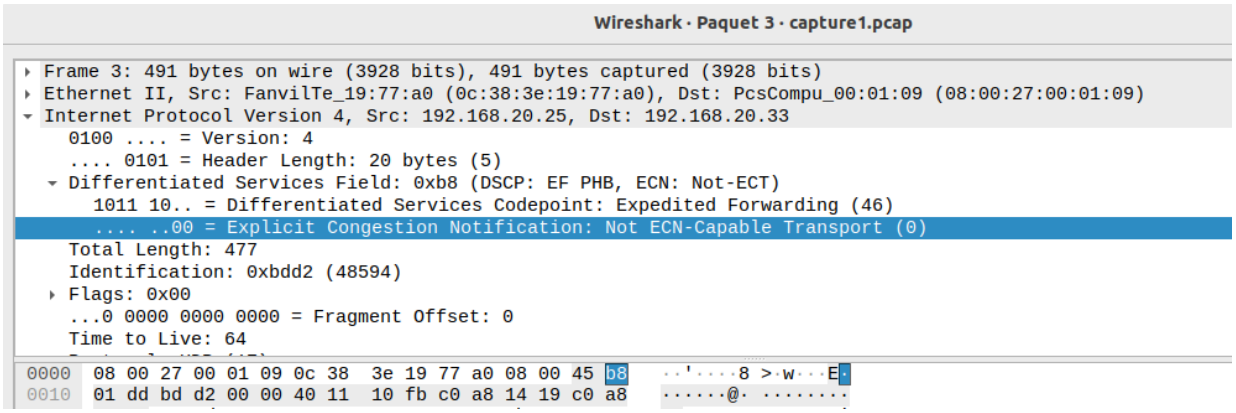
- priority percent 50 : réserve 50% de la bande passante pour le trafic voix avec une priorité élevée.

- bandwidth percent 10 : réserve 10% de la bande passante pour le trafic de signalisation SIP.

- bandwidth percent 40 : réserve 40% de la bande passante pour le trafic de données générales.

```
policy-map VOICE_POLICY
 class VOICE
  priority percent 50
 class SIGNALING
  bandwidth percent 10
 class DATA
  bandwidth percent 40
```

On remarque que le Differentiated Services Field est à b8 soit 46, nous pouvons donc constater que cela fonctionne, le trafic voix est prioritaire.



13. Docker-BDD-Web serveur

Pour pouvoir réaliser le serveur web nous avons dut mettre en place un docker, ce docker nous a servi à lancer deux services.
On a lancer les deux services dans un docker-compose.

docker-compose.yml:
Il y a le service web :

```
build: .
ports:
  - "80:80"
volumes:
  - .:/web
environment:
  - FLASK_APP=app.py
  - FLASK_ENV=development
  - DB_USER=admin
  - DB_PASS=Progtr00*
  - DB_HOST=db
  - DB_NAME=pepiniere
```

ça lancer donc le web sur le port 80, et j'ai fait le lien avec la base de donnée pour voir récupérer les informations à prendre pour les services ci-dessous

Nos Services

ID	Nom	Description	Prix	Catégorie
1	Service de dveloppement web	Cration de sites web et d'applications	1000.00	Dveloppement
2	Service de design graphique	Conception de logos et de supports visuels	500.00	Design
3	Service de marketing numrique	Stratgies de marketing en ligne	800.00	Marketing
4	Service de conseil en affaires	Conseils pour optimiser la performance des entreprises	1200.00	Conseil
5	Service d'hbergement web	Hbergement scuris pour les sites web	200.00	Technologie

ces services sont affiché grâce ac cela dans la page html:

```
<section id="services" class="wrapper style3">
  <div class="inner">
    <div class="container">
      <h1>Nos Services</h1>
      <table class="table">
        <thead>
          <tr>
            <th>ID</th>
            <th>Nom</th>
            <th>Description</th>
            <th>Prix</th>
            <th>Catégorie</th>
          </tr>
        </thead>
        <tbody>
          {% for service in services %}
          <tr>
            <td>{{ service[0] }}</td> <!-- ID -->
            <td>{{ service[1] }}</td> <!-- Name -->
            <td>{{ service[2] }}</td> <!-- Description -->
            <td>{{ service[3] }}</td> <!-- Price -->
            <td>{{ service[4] }}</td> <!-- Category -->
          </tr>
          {% endfor %}
        </tbody>
      </table>
    </div>
  </div>
```

Ici nous récupérons donc les services qui sont affiché dans le site intérnet.

Le docker-compose lance aussi mysql avec les informations de notre base de donnée pour pouvoir se connecté:

```
db:
  image: mysql:8.0
  command: --default-authentication-plugin=mysql_native_password
```

```
environment:
  - MYSQL_ROOT_PASSWORD=progtr00
  - MYSQL_USER=admin
  - MYSQL_PASSWORD=Progtr00*
  - MYSQL_DATABASE=pepiniere
ports:
  - "3306:3306"
volumes:
  - mysql-data:/var/lib/mysql
  - ./database.sql:/docker-entrypoint-initdb.d/database.sql
```

La base de donnée s'appelle donc pépinière, pour pouvoir le mettre dans mon docker j'ai fait la commande docker exec -it web_db_1 /bin/bash, je me connecte dans mysql et dans ma base de donnée pour ça .

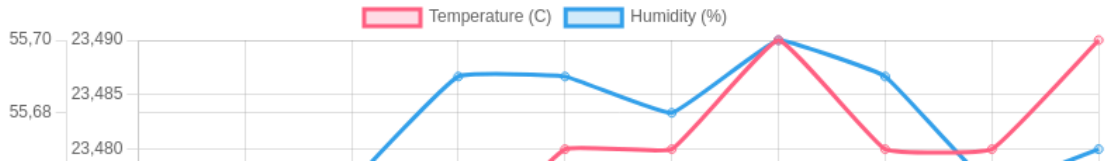
Comme nous avons dû faire une page pour afficher une caméra et la base de donnée de la température.
On a donc mit dans le menu les pages:

Pour aller sur le site de température je clique sur le bouton "aller a temp", et on arrive sur cette page:

societe5.com/temperature_data

Arduino Data

Temperature (C)	Humidity (%)	Timestamp
23.49	55.67	2024-06-19 11:10:18
23.48	55.66	2024-06-19 11:09:48
23.48	55.69	2024-06-19 11:09:18
23.49	55.7	2024-06-19 11:08:48
23.48	55.68	2024-06-19 11:08:18
23.48	55.69	2024-06-19 11:07:48
23.47	55.69	2024-06-19 11:07:18
23.45	55.66	2024-06-19 11:06:48
23.47	55.6	2024-06-19 11:06:18
23.47	55.58	2024-06-19 11:05:48



C

14. Docker-TFP

sudo apt update
sudo apt install tftpd-hpa tftp-hpa

sudo nano /etc/default/tftpd-hpa

TFTP_USERNAME="tftp"
TFTP_DIRECTORY="/var/lib/tftpboot"
TFTP_ADDRESS="0.0.0.0:69"
TFTP_OPTIONS="--secure"

sudo mkdir -p /var/lib/tftpboot
sudo chown -R tftp:tftp /var/lib/tftpboot
sudo chmod -R 777 /var/lib/tftpboot

sudo systemctl restart tftpd-hpa
sudo systemctl enable tftpd-hpa

sudo systemctl status tftpd-hpa

Installer le client TFTP :

sudo apt update
sudo apt install tftp-hpa

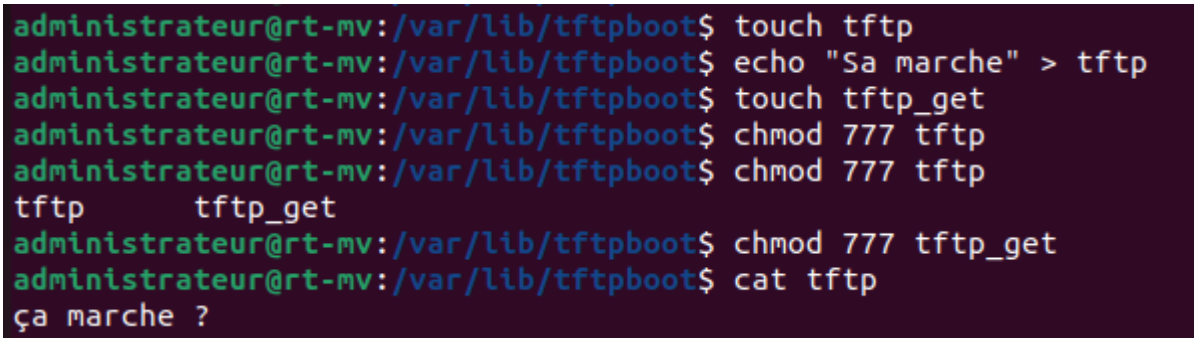
Pour que cette étape se réalise il faut d'abord créer un fichier du même nom dans /var/lib/tftpboot :

echo "Bonjour mon petit loup!" > example.txt
tftp [IP_du_serveur]
tftp> put example.txt
tftp> quit

Pour celle ci il faut créer un dossier peut importe le nom :

tftp [IP_du_serveur]
tftp> get example.txt
tftp> quit

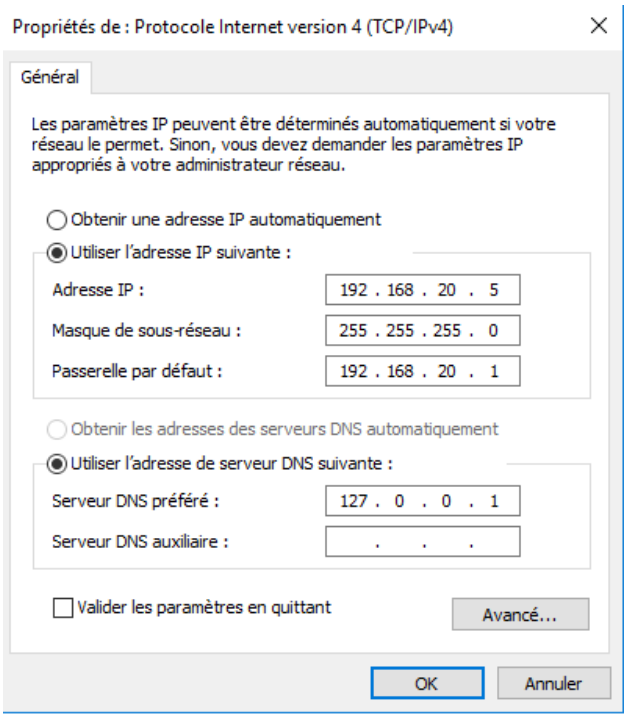
Serveur



client

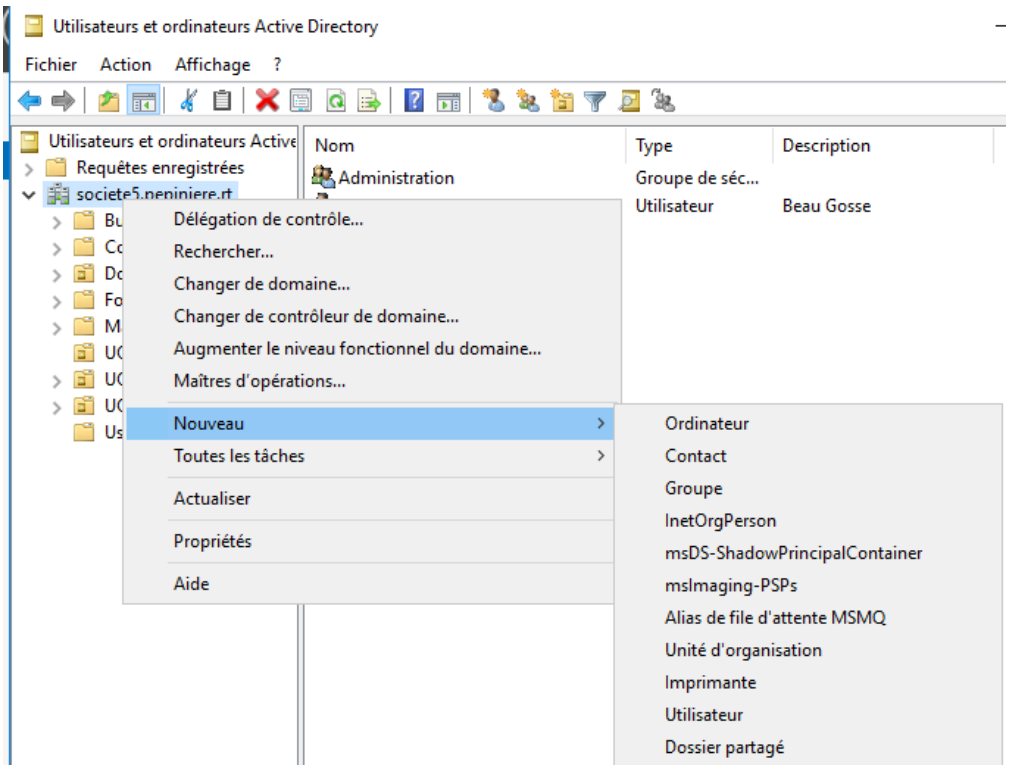
15. Windows Serveur

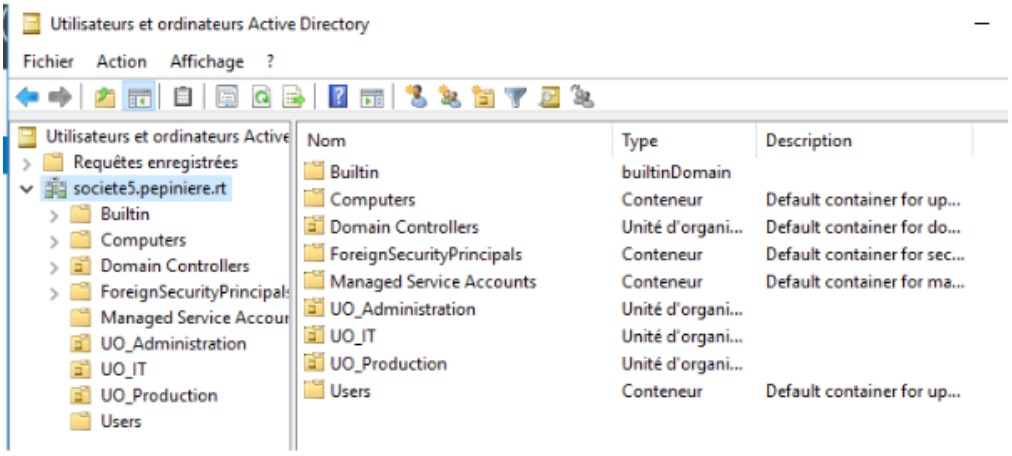
attribution d'une IP fixe ainsi que du DNS de la pépinière dans les paramètre réseau :



ajout du DNS nommé societe5.pepiniere.rt avec la fonctionnalités AD DS :

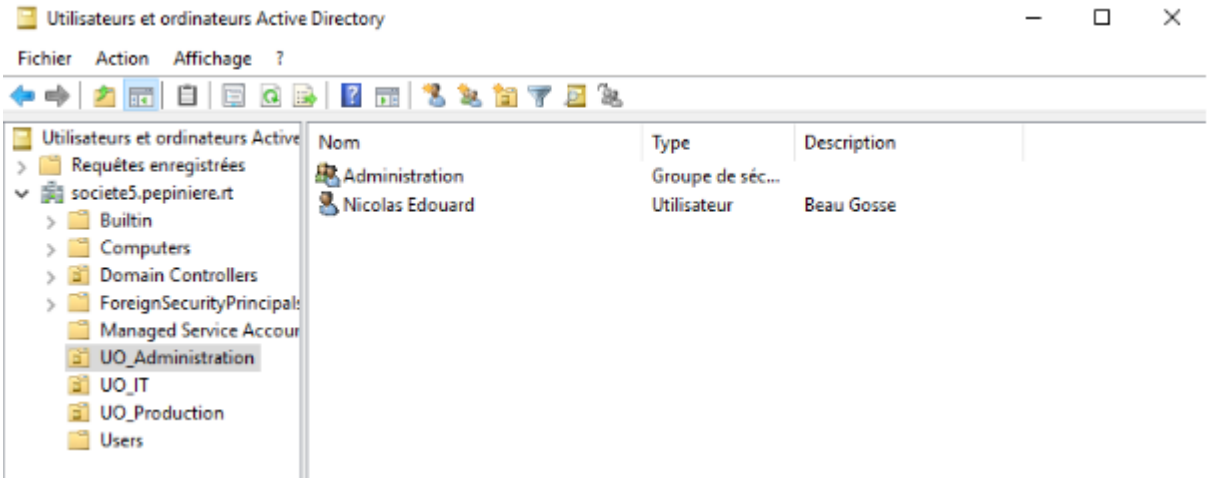
création des unitées d'organisation (UO) par groupe (Administration, IT, Production) dans utilisateurs et ordinateurs Active Directory :



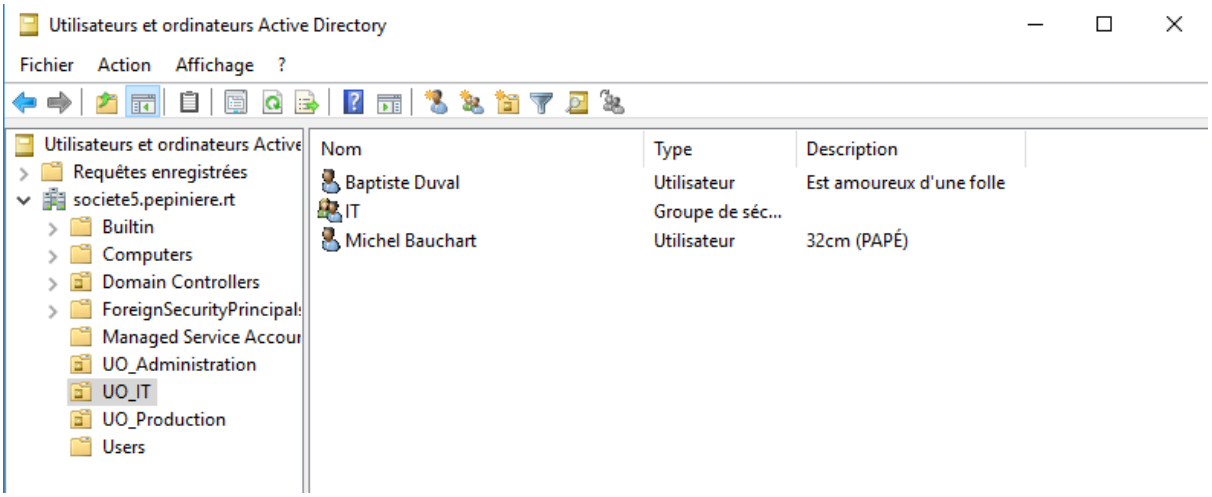


création des groupes dans chaque UO et création des utilisateurs dans leur groupes

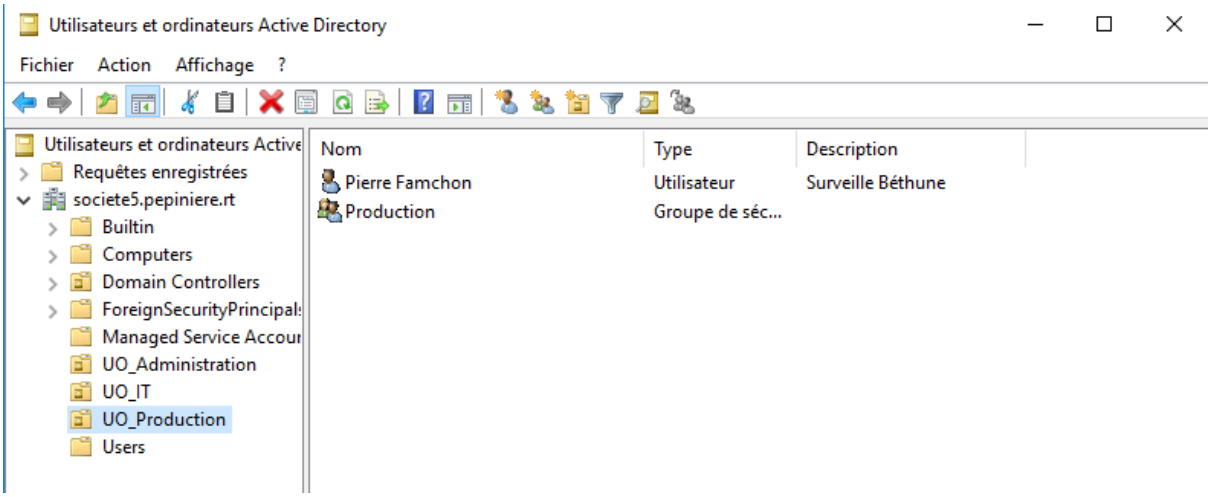
Dans UO_Administration



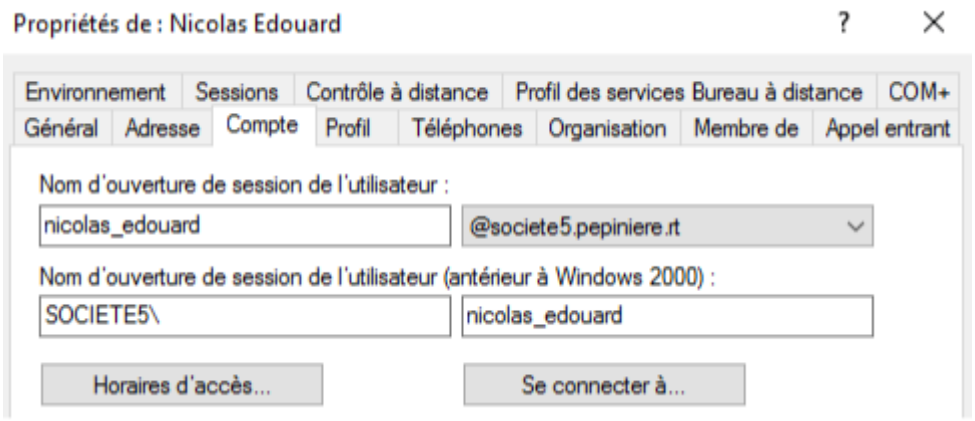
Dans UO_IT



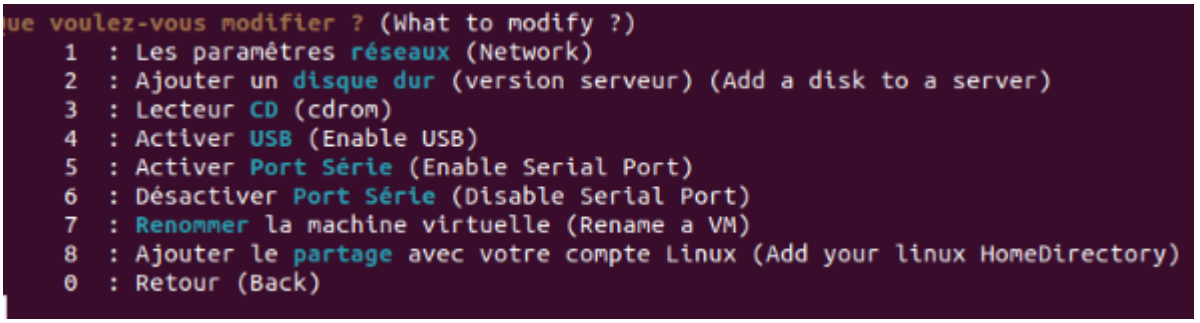
Dans UO_Production



exemple d’un identifiant d’utilisateur

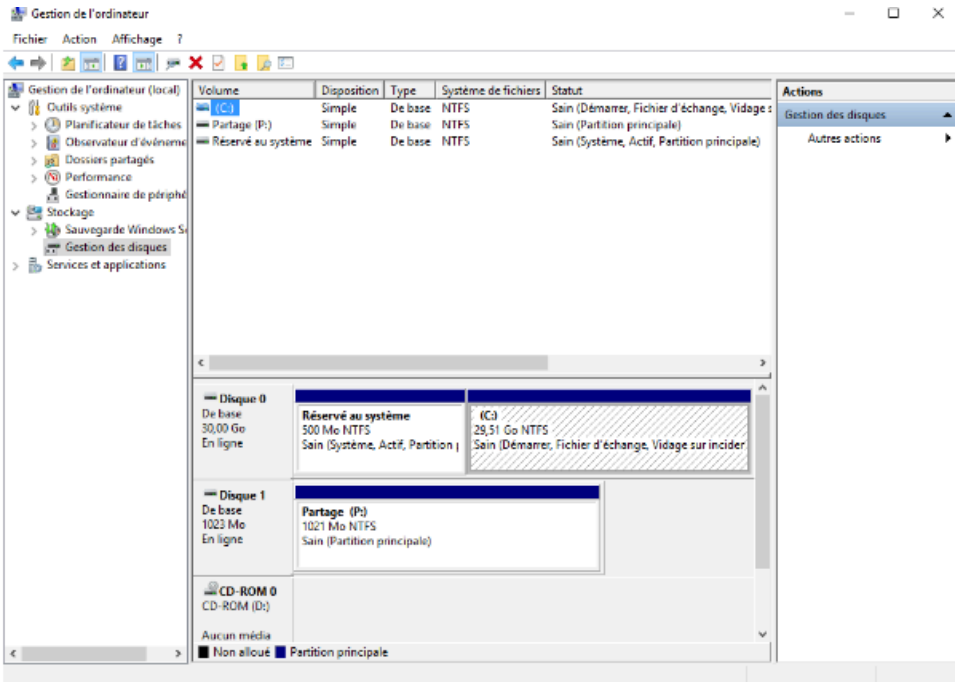


ajout d’un disque:
éteindre la machine.
Avec le script MachinesVirtuelles modifier la machine pour ajouter un disque :



la 2eme option

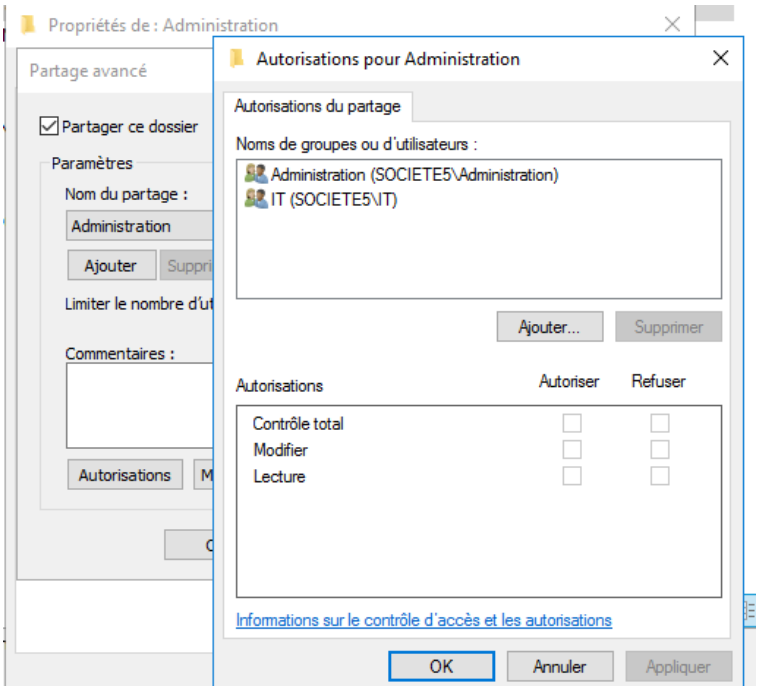
Dans la gestion de l’ordinateur puis dans le gestion des disques on crée la partition Partage



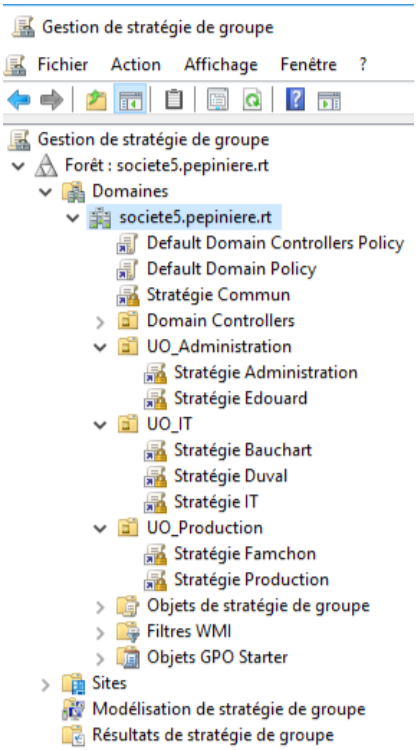
création des dossier partager (1 par utilisateur, 1 par groupe et 1 commun)

Administration	11/06/2024 13:59	Dossier de fichiers
Bauchart M	14/06/2024 11:34	Dossier de fichiers
Commun	11/06/2024 14:00	Dossier de fichiers
Duval B	13/06/2024 14:28	Dossier de fichiers
Edouard N	11/06/2024 15:26	Dossier de fichiers
Famchon P	11/06/2024 14:00	Dossier de fichiers
IT	13/06/2024 14:26	Dossier de fichiers
Production	11/06/2024 14:00	Dossier de fichiers

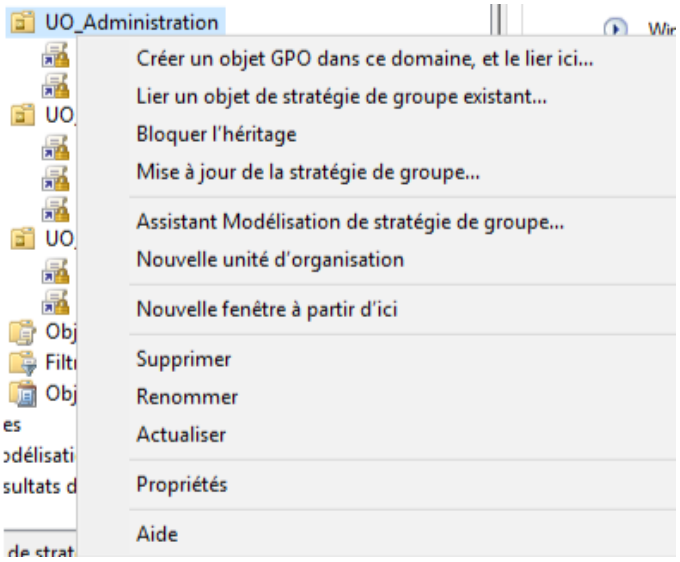
exemple d'un partage de dossier dans les propriétés du dossier partage avancé on coche la case pour partager ce dossier puis dans autorisation on ajoute les groupes ou utilisateur selon le dossier partagé



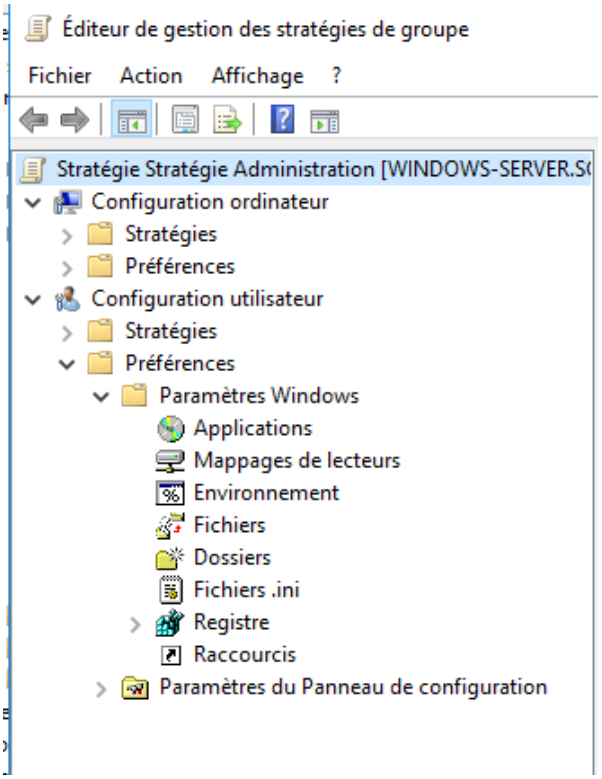
mise en place d'une gestion des stratégies de groupe (GPO) pour faciliter l'accès au dossier partager des utilisateurs dans Gestion de stratégie de groupe on va mettre les GPO selon le besoin d'accès de chaque utilisateur :



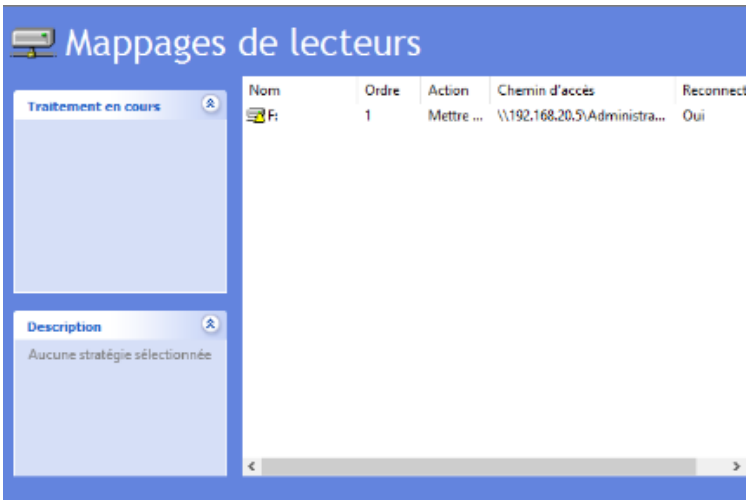
Clique droit sur UO dans lequel on veut ajouter la GPO puis Créer un objet GPO dans ce domaine, et le lier ici...



une fois créée on la modifie puis on va dans Préférences > Paramètres Windows > Mappages de lecteurs

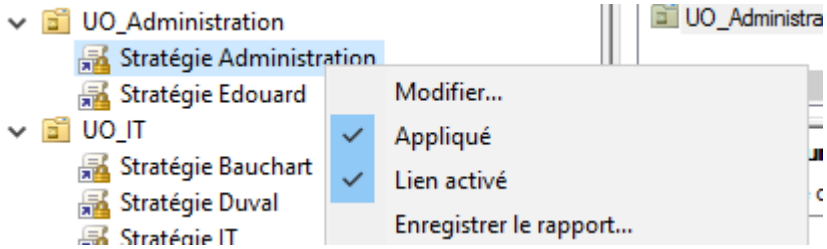


dans le mappages on en crée un nouveau



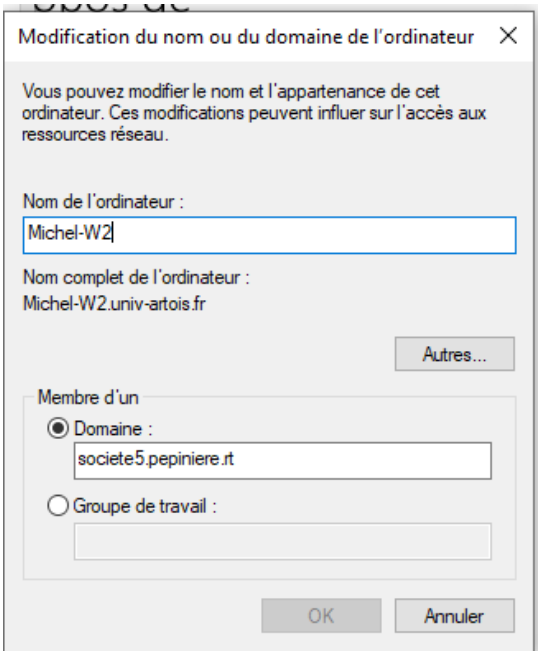
dans l'emplacement il faut spécifier le chemin du dossier dans le serveur avec \\@IP du serveur puis on lui attribue une lettre non utilisé sur les machines cliente puis on peut valider

Il faut maintenant appliquer notre GPO pour ca il faut faire un clique droit dessus puis Appliqué

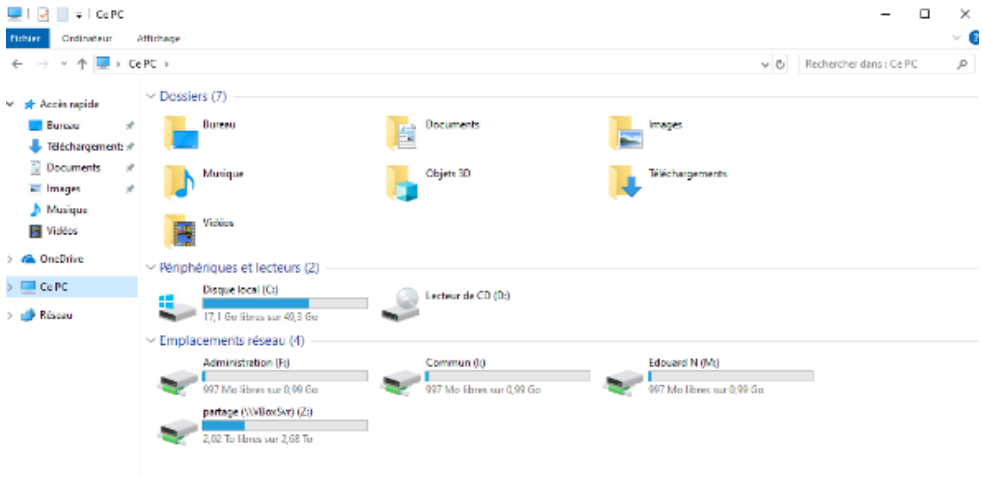


sur une machine windows, ubuntu, MAC ajout au domain
Sur Windows le DHCP attribue l'ip et en dns l'ip du windows server
on va ajouter la machine au domaine

Dans paramètre > Système > à propos de > Renommer ce PC (avancé) > Modifier



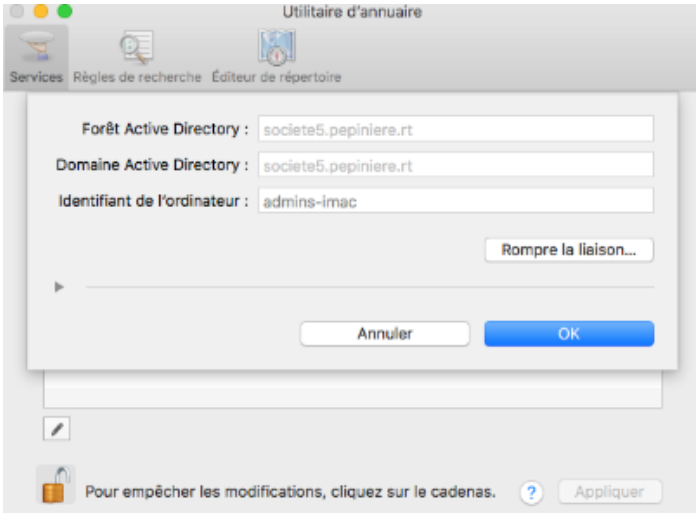
Puis les dossier partagé sont appliqué grâce au GPO pour vérifier on va dans les dossiers puis Ce PC et les dossiers apparaisse



Sur ubuntu :

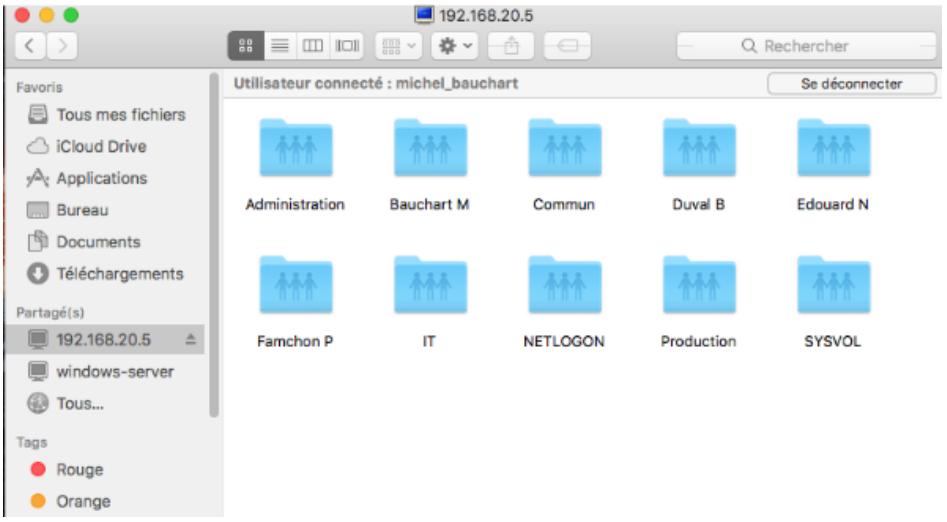
Client-Ubuntu
https://drive.google.com/open?id=14ekOK9gz1tknfhUClNUFYliCTv-ZvP6lmIoTuegkuGE&usp=drive_copy

Sur MAC dans l'utilitaire d'annuaire on peut rejoindre le domain dans l'active directory

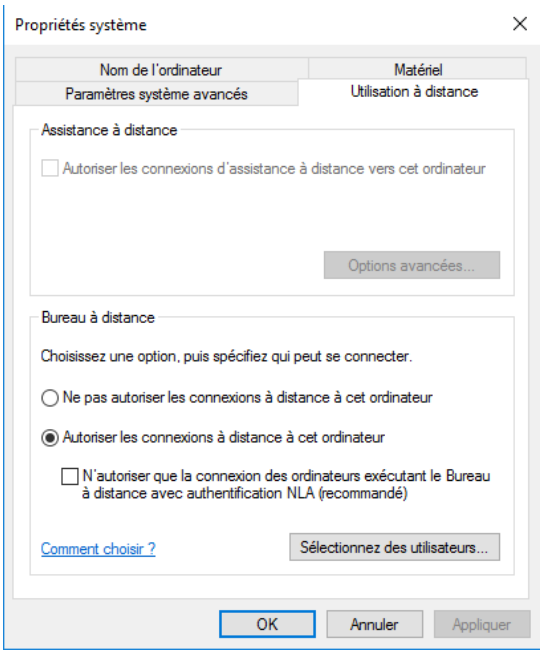


Pour le partage de fichier il faut chercher notre serveur dans la barre de recherche

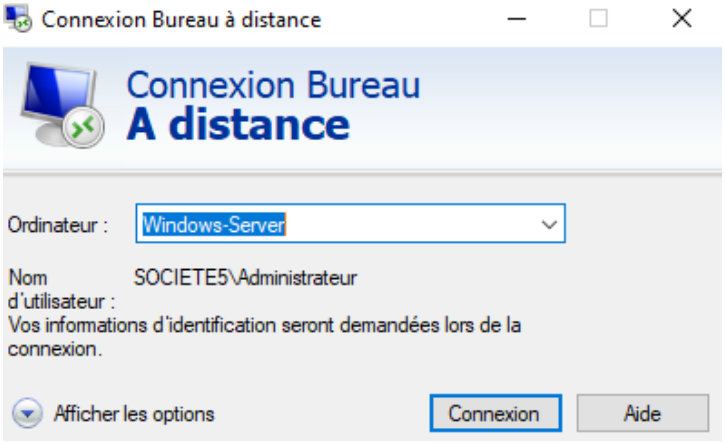
et dans la catégorie partagé on voit notre serveur il suffit de cliquer dessus pour avoir les fichier partagé



Pour le Bureau à distance il faut l'activer sur windows server

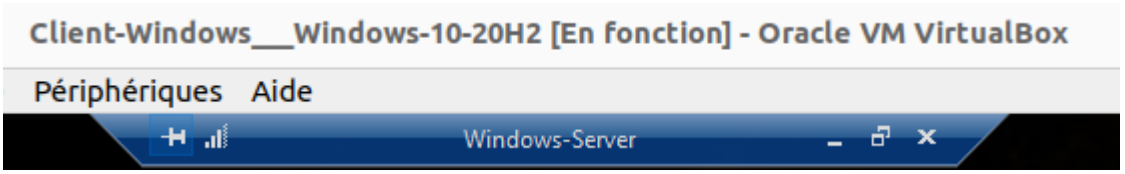


sur le client on ouvre le bureau a distance et on met le nom de notre serveur



on se connecte en administrateur

cette barre nous confirme que l'on est connecté

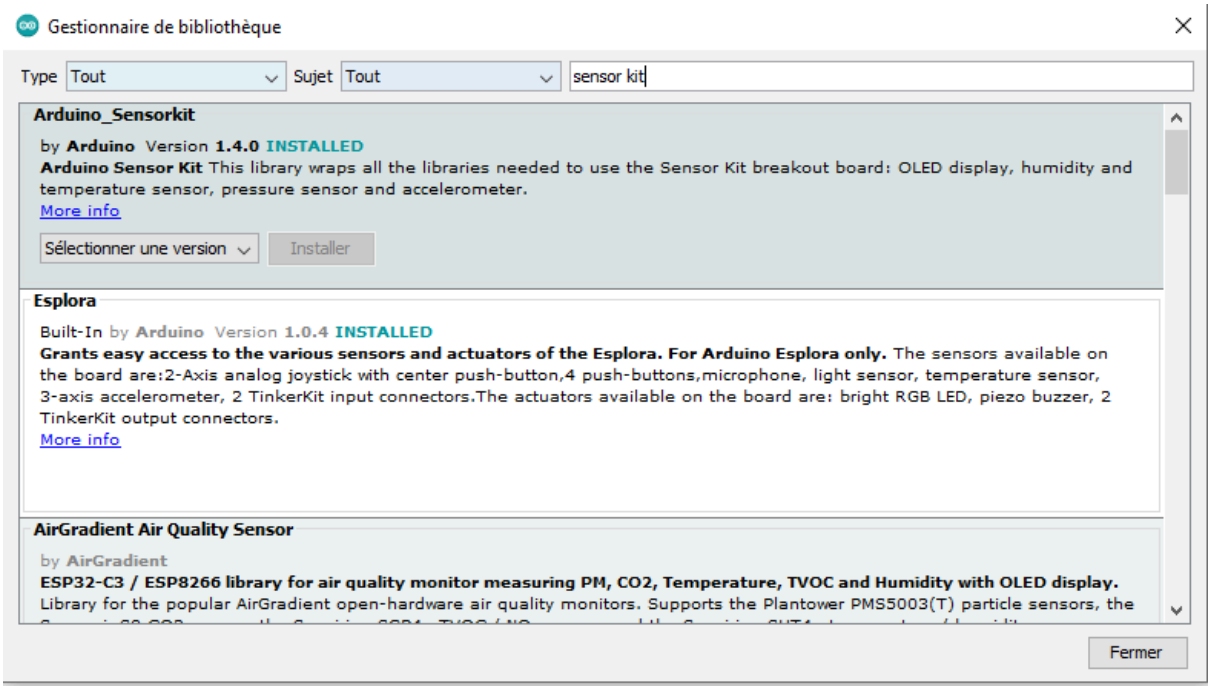


Pour pouvoir accéder au site avec societe5.com dans outil > DNS il faut mettre aucun mises à jour dynamiques
dans les propriétés de societe5.pepiniere.rt

16. Plaquette Arduino

Pour la configuration de la maquette arduino avec le module
“ Temperature & Humidity “, on va d’abord se connecter sur une session Windows.

Maintenant on a juste à se rendre dans l’onglet “ Croquis “ puis cliquer dans
“ Gestionnaire de bibliothèque “




```
#include "Arduino_SensorKit.h"

// Uncomment the line below if using DHT20 (black sensor)
#define Environment Environment_I2C

#include "SPI.h"
#include "Ethernet.h"

// Définir l'adresse MAC, l'adresse IP, la passerelle et le masque de sous-réseau
byte mac[] = { 0xA8, 0x61, 0x0A, 0xAE, 0xDF, 0x50 };
IPAddress ip(192, 168, 30, 3);
```

```
IPAddress gateway(192, 168, 30, 1); // Assurez-vous que cela correspond à votre configuration réseau
IPAddress subnet(255, 255, 255, 0); // /24 masque de sous-réseau
IPAddress dns(192, 168, 20, 5); // Serveur DNS, par exemple Google DNS

// Initialiser l'Ethernet client
EthernetClient client;

void setup() {
  // Uncomment the line below if using DHT20 (black sensor)
  Wire.begin();

  // Uncomment the line below if you're connecting your DHT20 to a pin different than 3
  // Environment.setPin(4);

  Ethernet.begin(mac, ip, dns, gateway, subnet);
  Serial.begin(9600);
  Environment.begin();
  delay(1000);
  Serial.println("Ethernet ready");

  // Vérifier et afficher l'adresse IP attribuée
  Serial.print("Assigned IP: ");
  Serial.println(Ethernet.localIP());
}

void loop() {
  Serial.print("Temperature = ");
  Serial.print(Environment.readTemperature()); // print temperature
  Serial.println(" C");
  Serial.print("Humidity = ");
  Serial.print(Environment.readHumidity()); // print humidity
  Serial.println(" %");
  delay(2000);
}
```

```
Ethernet ready
Assigned IP: 192.168.30.3
Temperature = 23.33 C
Humidity = 57.62 %
Temperature = 23.35 C
Humidity = 57.61 %
```

17. Etude Mathématique

1- Le PoE
Rappeler le principe du PoE et donner son avantage principal

Le PoE est un câble Ethernet qui fait à la fois transfert de données et alimentation électrique en 48V ce qui permet d'éliminer le besoin d'une source d'alimentation séparée pour les dispositifs comme les caméras IP, les téléphones VoIP, les bornes wifi ce qui simplifie l'installation des dispositifs réseau.

2- Normes PoE
Rappeler les différentes normes de PoE et donner leurs caractéristiques.

IEEE 802.3af (PoE)

- Puissance maximale fournie : 15.4 W par port
- Tension : 44-57 V
- Courant maximal : 350 mA

IEEE 802.3at (PoE+)

- Puissance maximale fournie : 25.5 W par port
- Tension : 50-57 V
- Courant maximal : 600 mA

3- Switch cisco2960
Donner les caractéristiques électriques et PoE du switch cisco2960
Sachant que le switch dispose de 48 ports, quelle est la puissance maximale PoE disponible par port ?

Le switch Cisco 2960 est un switch de couche 2 qui prend en charge le PoE

- Nombre de ports : 48 ports
- Puissance totale disponible pour le PoE : 740W

Puissance maximale par port = $\frac{740\ W}{48\ ports} \approx 15.42\ W\ par\ port$

4- Caractéristiques de la caméra FD8181V de Vivotek
Quelle est la norme PoE supportée par la caméra ?

la norme supportée par la caméra **IEEE 802.3af (PoE)**.

Quelle est la consommation électrique maximale ?

Elle est de 6.5 Watt

Quel est le nombre théorique maximal de caméras connectables au switch ?

Puissance du switch = 740 W

Consommation d'une caméra = 6.5 W

Nombre maximal de caméras = $\frac{740}{6.5} = 113\ caméras$

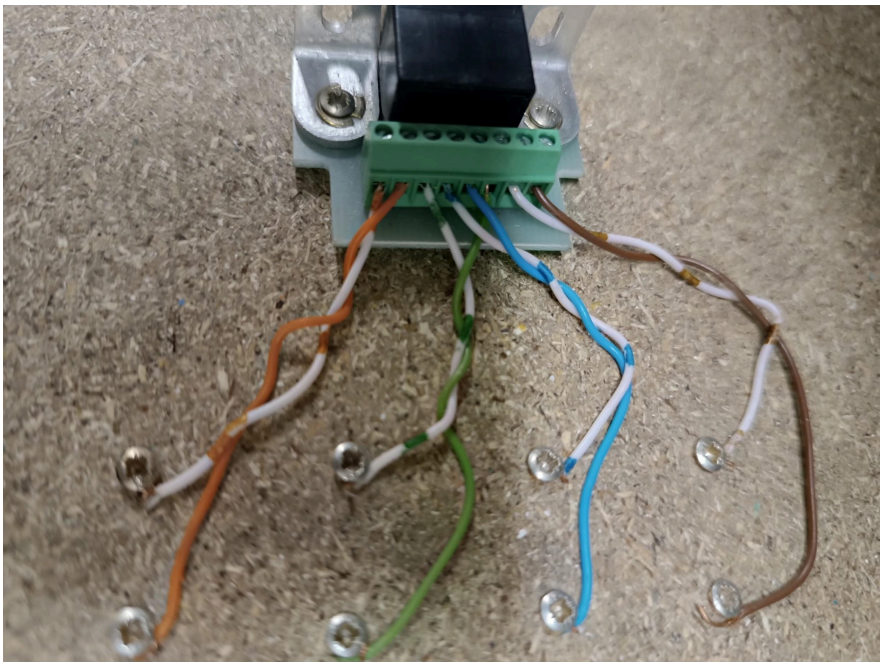
5- Caractéristiques PoE
Mesures sur le switch
Mesure à vide :

Mesurer la tension PoE à vide (sans brancher la caméra) entre les différentes broches du connecteur RJ45. Conclusion.

La tension est nul car le switch ne donne pas de tension a une sortie sans branchement à un équipement

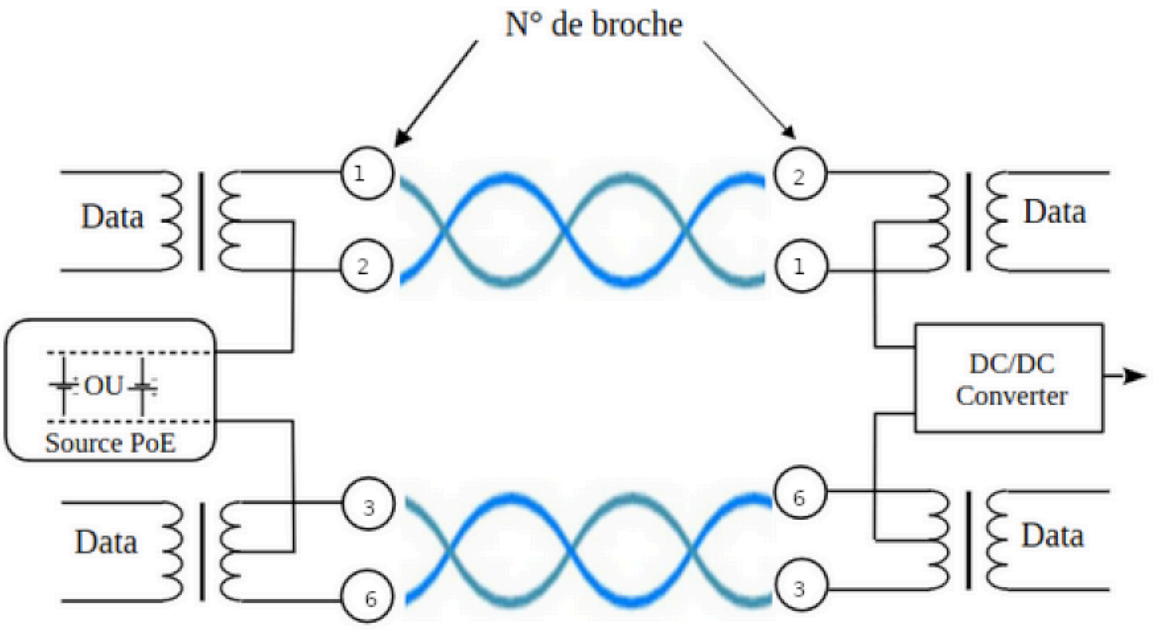
Mesure en charge :
Relier la caméra au switch à l'aide des câbles et fils fournis.
Mesure la tension disponible entre les différentes broches du connecteur RJ45.

entre les bornes 1 et 3 nous mesurons 50 V ce qui est normal
Cela est pareil pour les bornes 1 et 6 , 2 et 6 , 2 et 3. (fils marron et vert)



Vos relevés sont-ils conformes au câblage normalisé ?

Nos relevés sont conformes car comme vérifié auparavant la valeur max de la caméra est de 50 V
Compléter le schéma ci-dessous (sens de la source PoE et numéros de broches) et expliquer le principe de fonctionnement de la transmission de l'énergie à travers le câble de données.



Compte-tenu de la puissance absorbée par la caméra, calculer le courant I circulant dans le câble.

P = U * I

Tension(U) = 50 V
Puissance = 6.5 W

$I = \frac{P}{U}$

avec P = 6.5 W
et U = 50 V

$I = \frac{6.5}{50}$ = 0.13 A

6- Longueur maximale du câble

Le câble de liaison entre la caméra et le switch est du même type que l'échantillon de 30 mètres caractérisé lors des TP du module R105. Le fonctionnement de la caméra est assurée si la tension à ses bornes est au minimum de 48 V.

En supposant que le courant soit toujours égal à I, valeur calculée à la question 5, déterminer la longueur maximale de la ligne entre le switch et la caméra (voir module R105).

U = R * I

Résistance du câble = R = 10.3 Ω
Intensité = 0.13 A

U = 10.3 * 0.13 = 1.3 V

Sachant qu'on a 48 V

48 + 1.3 = 49.3 V

Il nous reste 0.7 V de marge avant de ne plus pouvoir utiliser la caméra

Pour 1 mètre de câble :

$\frac{10.3}{30}$ = 0.33 Ohms

U = R * I

Résistance pour 1 mètre de câble = 0.33 Ohms
Intensité = 0.13 A

U = 0.33 * 0.13 = 0.0429 V

Par déduction on a essayer différentes longueur de câble, ici l'idéal de la longueur du câble est de 46.5 m :

46.5 * 0.0429 = 1.99 V

au delà de 46.5 mètres de câble la tension n'est plus suffisante pour alimenter la caméra

7- Délai de propagation

Compte-tenu de la distance Lmax, quel est le délai de propagation des données entre la caméra et le switch.
Quelles sont les conséquences pratiques de cette information ?

Délai de propagation = $\frac{Longueur\ du\ câble}{Vitesse\ de\ propagation\ de\ l'électricité}$

Délai de propagation = $\frac{46.5}{2*10^8}$ = $23.2 * 10^{-6}$ m/s

18. Programmation

Le but de notre programme est d'écouter les conversations d'une autre entreprise et d'en créer un fichier audio.
En soit il s'agit de passer d'un pcap à un raw,wav,mp3.

Dans un premier temps nous allons récupérer le fichier à l'aide de plusieurs commande :

>ssh VotreLoginEtudiant'@172.31.25.40
par exemple
>ssh michel_bauchart@172.31.25.40

Puis tcpdump -i eth1 -w output.pcap '(udp port 5060 or tcp port 5060 or (udp portrange 10000-20000))'

ce qui va faire en sorte d'avoir que les trames sip et rtp

pour tout écouter sauf ce qui viens de 10.0.0.6 et l'écrire dans un fichier pcap

Une fois le fichier obtenue nous pouvons nous rendre sur wireshark
et si il n'est pas installé :

apt install wireshark

dessus dans téléphonie nous pouvons voir appel voip et voir l'ensemble des appels ayant eu lieu durant l'enregistrement. nous pouvons même déjà l'écouter mais notre but est de le récupérer avec un script python.
Il n'y a qu'un appel dans notre pcap mais si votre pcap en contient plusieur nous vous invitons à ne garder que les fichier en lien avec l'appelle choisis depuis wireshark dans le cas contraire vous obtiendrez à la fin un long vocal avec tous les appelle à la suite.

une fois votre nouveau pcap créer par exemple output.pcap voici le programme python. Vous pouvez aussi le trouver dans code.py

Vous récupérerez alors un fichier en . raw de votre appel.
importez le dans audacity pour l'écouter avec ces paramètres.
encodage : U law
mode bouddhisme par default
taux d'échantillonnage 10000

Voilà vous pouvez à présent écouter l'audio

```
from scapy.all import *
import struct
import wave

# Fonction pour extraire les paquets RTP de la capture
def extract_rtp_packets(pcap_file):
    packets = rdpcap(pcap_file)
    rtp_packets = []
    ip_pairs = set()  # Pour stocker les paires d'adresses IP

    for packet in packets:
        if packet.haslayer(UDP):
            udp_payload = bytes(packet[UDP].payload)
            if len(udp_payload) > 12:
                version = udp_payload[0] >> 6
                payload_type = (udp_payload[1] & 0x7F)
```

```
                sequence_number = struct.unpack('>H',
udp_payload[2:4])[0]
                timestamp = struct.unpack('>I',
udp_payload[4:8])[0]
                if version == 2:
                    rtp_packets.append((sequence_number, timestamp,
udp_payload[12:], payload_type))
                    ip_pairs.add((packet[IP].src, packet[IP].dst))

    return rtp_packets, ip_pairs

# Fonction pour sauvegarder les données audio dans un fichier WAV
def save_to_wav(rtp_packets, output_file, codec):
    wav_file = wave.open(output_file, 'wb')
    wav_file.setnchannels(1)

    if codec == 'PCMU':
        wav_file.setsampwidth(2)
        wav_file.setframerate(8000)
        for packet in rtp_packets:
            for byte in packet[2]:
                sample = ulaw2linear(byte)
                wav_file.writeframes(struct.pack('<h', sample))

    elif codec == 'PCMA':
        wav_file.setsampwidth(2)
        wav_file.setframerate(8000)
        for packet in rtp_packets:
            for byte in packet[2]:
                sample = alaw2linear(byte)
                wav_file.writeframes(struct.pack('<h', sample))

    elif codec == 'G722':
        wav_file.setsampwidth(2)
        wav_file.setframerate(16000)
        for packet in rtp_packets:
            for i in range(0, len(packet[2]), 2):
                sample = struct.unpack('>h', packet[2][i:i+2])[0]
```

```
                wav_file.writeframes(struct.pack('<h', sample))

    wav_file.close()

# Fonction de conversion PCMU (u-law) en linéaire
def ulaw2linear(ulawbyte):
    exp_lut = [0, 132, 396, 924, 1980, 4092, 8316, 16764]
    ulawbyte = ~ulawbyte & 0xff
    sign = (ulawbyte & 0x80)
    exponent = (ulawbyte >> 4) & 0x07
    mantissa = ulawbyte & 0x0F
    sample = exp_lut[exponent] + (mantissa << (exponent + 3))
    if sign != 0:
        sample = -sample
    return sample

# Fonction de conversion PCMA (a-law) en linéaire
def alaw2linear(alawbyte):
    alawbyte ^= 0x55
    sign = alawbyte & 0x80
    exponent = (alawbyte & 0x70) >> 4
    mantissa = alawbyte & 0x0f
    sample = (mantissa << 4) + 8
    if exponent != 0:
        sample += 0x100
    if exponent > 1:
        sample <<= (exponent - 1)
    if sign != 0:
        sample = -sample
    return sample

# Chemin vers le fichier pcap
pcap_file = 'output.pcap'
# Chemin vers le fichier de sortie WAV
output_wav_file = 'output.wav'
# Codec utilisé (PCMU, PCMA, G722)
codec = 'PCMU'  # Changez cela en fonction du codec utilisé dans
vos paquets RTP
```

```
# Extraction des paquets RTP
rtp_packets, ip_pairs = extract_rtp_packets(pcap_file)
# Tri des paquets RTP par numéro de séquence pour l'ordre correct
rtp_packets.sort(key=lambda x: x[0])
# Sauvegarde des données audio dans un fichier WAV
save_to_wav(rtp_packets, output_wav_file, codec)

# Affichage des adresses IP des conversations interceptées
print("Conversations téléphoniques interceptées entre les adresses
IP :")
for src_ip, dst_ip in ip_pairs:
   print(f"{src_ip} <-> {dst_ip}")

print(f"Audio extrait et sauvegardé dans {output_wav_file}")
```