

TP VOIP :

GROUPE :

- Celina AIT BOUALI
- Luc-Augustin KONE
- William MBAKOP

VOIP : (Voice over IP) : technologie

Permet de transmettre la voix via des réseaux IP

- Economie (système requiert uniquement une connexion)
- Simplicité d'implémentation
- Nombreuse fonctionnalité

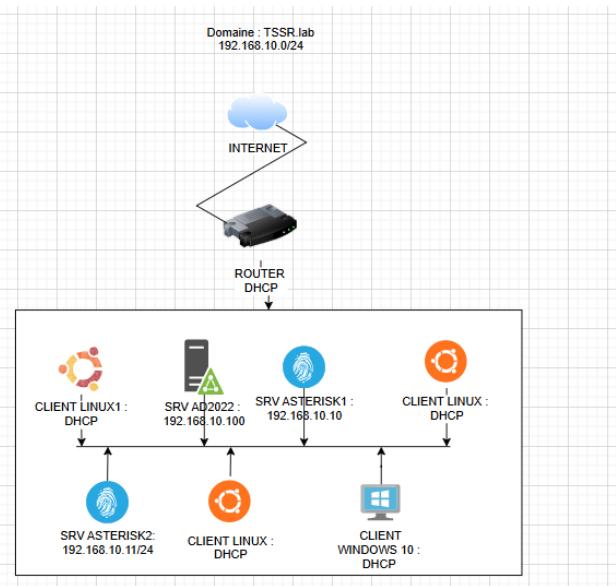
Technologies :

- **VOIP** : Transmission de la voix sur un réseau Internet. La voix est convertie en *paquets numériques* et envoyée via IP au lieu de passer par les lignes téléphoniques classiques.
- **TOIP** : “Téléphonie sur IP”. C'est l'usage du VoIP dans un système complet de téléphonie d'entreprise (postes, serveurs, gestion des appels). (Écosystème pas seulement une technologie).
- **SIP** : Protocole qui sert à établir, modifier et terminer les appels (ou sessions) entre deux utilisateurs. En gros, il s'occupe de la **signalisation**.
- **RTP** : Protocole utilisé pour *transporter la voix ou la vidéo* en temps réel pendant un appel. C'est lui qui envoie les paquets audios.
- **RTCP** : Protocole qui accompagne RTP pour communiquer des informations sur la **qualité** de la transmission (pertes, délai, etc.).
- **RTC** : L'ancien réseau téléphonique fixe traditionnel, basé sur des commutations, pas sur Internet.

Transport :

- **UDP** : Protocole *rapide et sans vérification* des paquets. Idéal pour la voix, car on préfère perdre un paquet plutôt que d'avoir du retard.
- **TCP** : Protocole *fiable* car il vérifie que tous les paquets arrivent bien, mais plus lent. Utilisé pour les données, moins pour la voix.

1. Architecture :



Objectif :

- Pouvoir connecter toute l'architecture
- Faire en sorte de communiquer entre les participants
- Faire la Visio entre les participants

🌐 1. Plan d'adressage global

Machine	Rôle	IP	Mode
WS2022-ROUTER	Deux cartes réseau : • NAT • Bridge DHCP, routage	192.168.10.1	Statique
WS2022-AD	AD / DNS	192.168.10.100	Statique
ASTERISK 1	Serveur VOIP / VISIO principal	192.168.10.10	Statique
ASTERISK 2	Serveur VOIP / VISIO secondaire	192.168.10.11	Statique
CLIENT 1	PC visio	DHCP	Automatique
CLIENT 2	PC visio	DHCP	Automatique
CLIENT 3	PC visio	DHCP	Automatique
CLIENT 4	PC visio	DHCP	Automatique

✓ Réseau interne : 192.168.10.0 /24

✓ Domaine : TSSR.lab

2. Infrastructure physique (VMs et cartes réseau)

WS2022-ROUTER (2 cartes réseau)

- **Carte LAN (Internal Network)**
→ réseau privé : 192.168.10.0/24

- IP : 192.168.10.1
- Sert de passerelle aux clients
- **Carte WAN (Bridge)**
 - connectée à la BOX / Internet
 - NAT activé dans RRAS
 - Fait sortir tout le trafic Internet

3. Serveurs

WS2022-AD (Active Directory + DNS)

- Nom : **SRV-AD**
- IP fixe : **192.168.10.100**
- Rôle :
 - AD DS
 - DNS
 - Contrôleur du domaine **TSSR.lab**
- Sert de DNS pour tout le réseau

Asterisk 1 et Asterisk 2 (Ubuntu 20.04)

- ASTERISK 1 → 192.168.10.10
- ASTERISK 2 → 192.168.10.11
- Utilisés pour : SIP, RTP, VOIP, IVR, MOH, tests d'appels

4. Clients DHCP (4 postes)

- 4 machines clientes (Windows / Linux)
- Reçoivent automatiquement :
 - IP : 192.168.10.50 → 192.168.10.200
 - DNS : 192.168.10.100
 - Passerelle : 192.168.10.1
- Utilisent des softphones : Zoiper, Linphone, etc.

5. Services réseau

✓ RRAS (NAT + Routage) sur WS2022-ROUTER

- NAT entre WAN ↔ LAN
- Permet l'accès Internet à tout le réseau interne

✓ DHCP sur WS2022-ROUTER

- Scope : 192.168.10.50 → 192.168.10.200
- Passerelle : 192.168.10.1
- DNS : 192.168.10.100

✓ DNS + Active Directory

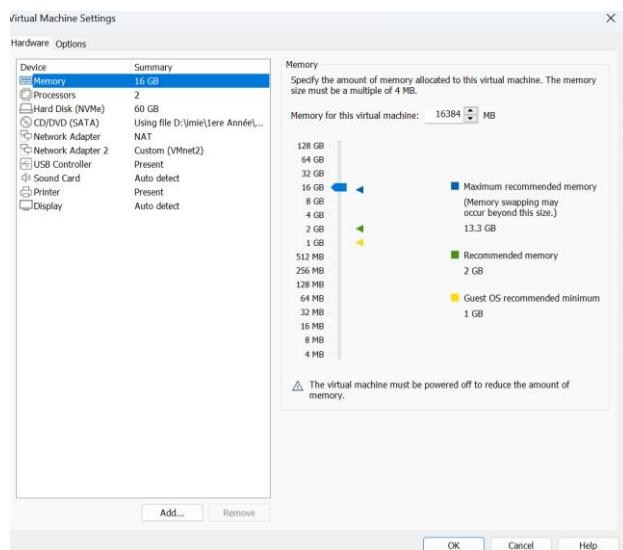
- Gérés uniquement par **WS2022-AD**

✓ VOIP

- Gérée par **Asterisk 1 et Asterisk 2** :

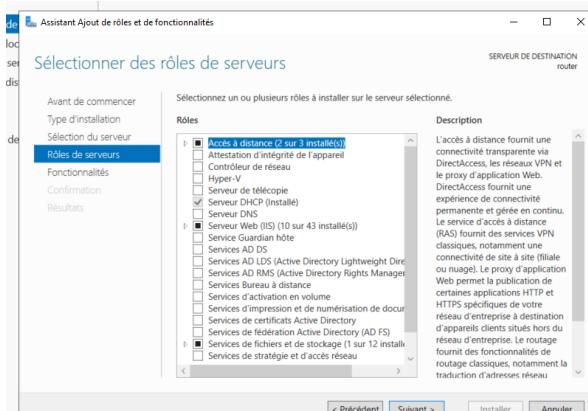
- SIP
- RTP
- IVR
- Musique d'attente
- Conférence
- Visiophonie

▣ CONFIGURATION COMPLÈTE DU ROUTEUR WINDOWS SERVER 2022 (RRAS + NAT + DHCP)



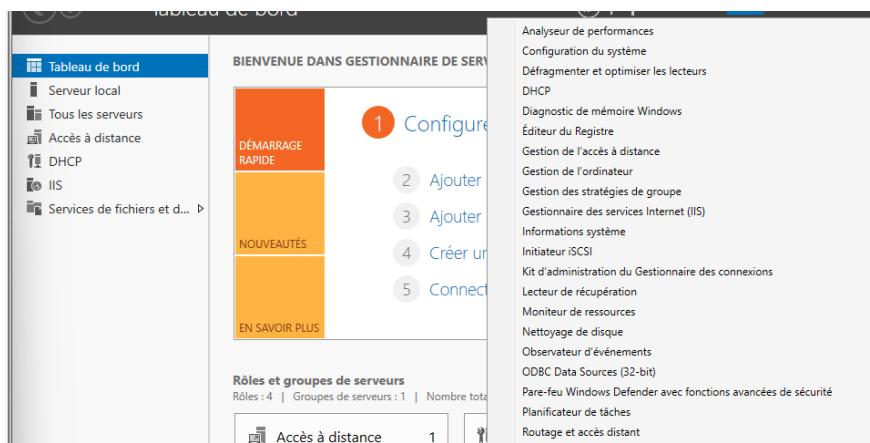
1 INSTALLATION DU RÔLE DE ROUTAGE

- Ouvrir **Gestionnaire de serveur** → **Ajouter rôles et fonctionnalités**
- Cocher **Remote Access** → **Routing**



Menu :
Ajouter des rôles et fonctionnalités

2 SÉLECTION DES RÔLES



Fenêtre **Sélection des rôles**

✓ Cocher : **Remote Access**

3 SÉLECTION DES SERVICES DE RÔLE

Fenêtre **Services de rôle**

✓ Cocher : **Routing**

Puis installer.

4 CONFIGURATION DU RRAS (NAT)

Ouvre :

Outils → Routing and Remote Access

Clic droit sur ton serveur :

Configurer et activer RRAS

Choisir :

✓ NAT (Network Address Translation) (si c'est bridge, sinon VPN “si c un lan segment”)

Fenêtre **Sélection du type de configuration**

→ Choisir **NAT**

5 CONFIGURATION DES INTERFACES NAT

RRAS te demande maintenant de choisir les cartes réseau.

5.1 INTERFACE PUBLIQUE (WAN)

La carte WAN = **192.168.100.128**(NAT VMware)

Sélectionne-la.

- ✓ Public interface connected to the Internet
- ✓ Enable NAT on this interface

👉 C'est la carte qui donne INTERNET.

⌚ ! ZONE DE CAPTURE 6 :

Fenêtre sélection WAN

→ NAT activé

5.2 INTERFACE PRIVÉE (LAN)

Ta carte LAN = **192.168.10.1**

Sélectionne-la :

- ✓ Private interface connected to private network
- ✗ Ne PAS cocher NAT

⌚ ! ZONE DE CAPTURE 7 :

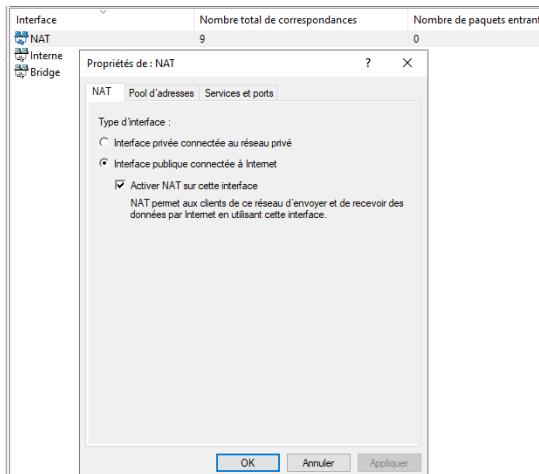
Fenêtre sélection LAN

→ Private interface

6 VÉRIFICATION DU NAT (IMPORTANT)

Ouvre :

RRAS → IPv4 → NAT



Tu dois voir :

Interface	Type	NAT
WAN (192.168.100.x)	Public	✓ Activé
LAN (192.168.10.1)	Private	✗ Désactivé



Capture du tableau NAT avec WAN (Public + NAT ON) et LAN (Private + NAT OFF)

7 INSTALLATION DU DHCP

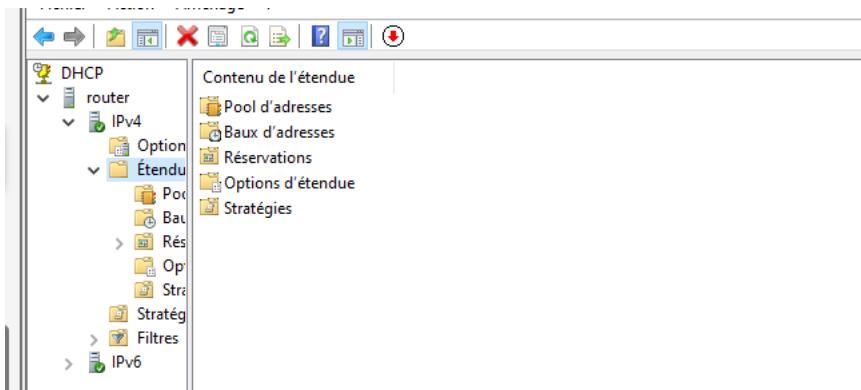
Gestionnaire de serveur → Ajouter des rôles

✓ Choisir : **DHCP Server**

Installer.

8 CONFIGURATION DU DHCP

Outils → DHCP



8.1 CRÉATION DU SCOPE DHCP

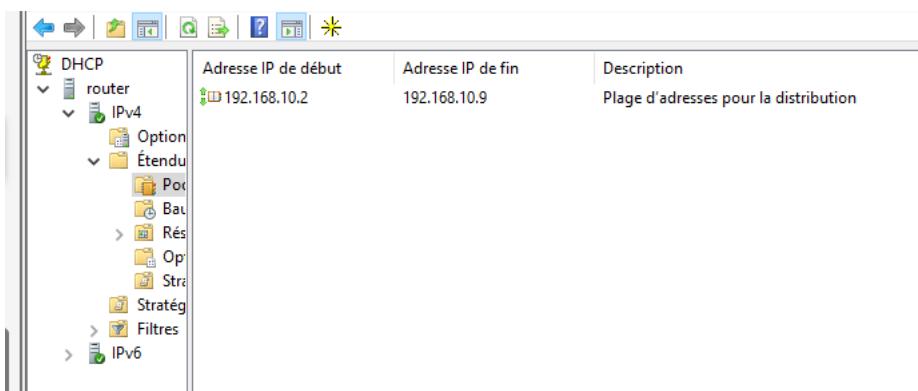
Clic droit sur IPv4 → **New Scope**

Plage (corrigée + adaptée à ton architecture) :

Début : 192.168.10.20

Fin : 192.168.10.99

Masque : 255.255.255.0

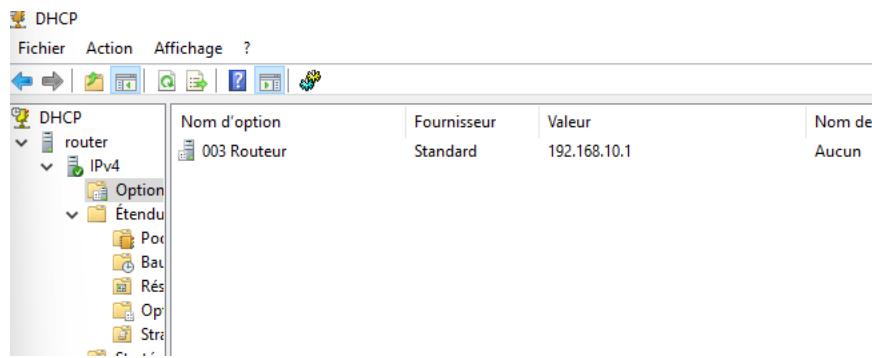


8.2 OPTIONS DU SCOPE

Ouvre **Configurer les options**.

✓ Option 003 – Routeur (Gateway)

192.168.10.1



SUR L'AD :

✓ Option 006 – DNS Server

Ton serveur DNS = TA FUTURE MACHINE AD

192.168.10.100

.....

✓ Option 015 – Domain Name

Ton domaine SISR / VoIP :

TSSR.lab

..... (ne pas le faire sur le routeur : le routeur est seulement là pour le routage, attribution des adresses IP et connexion en NAT)

9 TESTS SUR UN CLIENT

Lance un client en DHCP (Ubuntu / Windows / autre).

✓ Vérifier l'IP sur Client 2 : Ubuntu

ip a (ou ipconfig)

Doit montrer :

- IP : 192.168.10.x
- Passerelle : 192.168.10.1
- DNS : 192.168.10.100

```

root@client2:/home/client2
root@client2:~# ifconfig
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 brd 127.0.0.1 scope host lo
        valid_lft forever preferred_lft forever
        inet6 ::1/128 brd :: scope host ::1
            valid_lft forever preferred_lft forever
2: ens3: <NO-CARRIER,BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default
    link/ether 00:0c:29:f3:62:89 brd ff:ff:ff:ff:ff:ff
    inet 192.168.10.3/24 brd 192.168.10.255 scope global dynamic noprefixroute ens3
        valid_lft 691001sec preferred_lft 691001sec
        inet6 fe80::d31c:f040:5331:ae9b/64 brd ff:ff:ff:ff:ff:ff scope link noprefixroute
            valid_lft forever preferred_lft forever
root@client2:/home/client2# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=127 time=8.87 ms
^C
--- 8.8.8.8 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 8.869/8.869/8.869/0.000 ms
root@client2:/home/client2#

```

✓ Test Internet (ping brut)

ping 8.8.8.8

- DOIT marcher
- La NAT fonctionne

Dans le cas où cela ne marche pas :

- Rentrer sur l'interface :

#nano /etc/resolv.conf

Mettre : nameserver : (adresse ip de l'AD)

```

# See man:systemd-resolved.service(8) for detail
# operation for /etc/resolv.conf.

nameserver 192.168.10.100
search TSSR.lab

```

✓ Test DNS

ping google.com

→ DOIT marcher si ton serveur AD/DNS est configuré.

```

1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 8.869/8.869/8.869/0.000 ms
root@client2:/home/client2# nano /etc/resolv.conf
root@client2:/home/client2# ping google.com
PING google.com (216.58.215.46) 56(84) bytes of data.
64 bytes from par21s17-in-f14.1e100.net (216.58.215.46): icmp_seq=1 ttl=127 time=8.76 ms
^C
--- google.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 8.759/8.759/8.759/0.000 ms
root@client2:/home/client2#

```

La finalité

On a maintenant :

- Une architecture nickel

- Un routeur fonctionnel
- NAT OK
- DHCP OK
- Internet sur les clients OK

C'est la base **professionnelle** pour continuer le projet VoIP + AD + Asterisk.

ÉTAPE 2 – INSTALLATION DE L’AD (Active Directory Domain Services)

Serveur concerné : 192.168.10.100

1 CONFIGURATION IP DU FUTUR SERVEUR AD

Sur le serveur Windows 2022 (celui qui sera AD) :

Va dans :

ncpa.cpl

Configure la carte réseau en IP statique :

IP : 192.168.10.100
Masque : 255.255.255.0
Passerelle : 192.168.10.1 (ton routeur RRAS)
DNS : 192.168.10.100 (lui-même)

→ NE METS PAS 8.8.8.8

→ NE METS PAS 192.168.10.1 en DNS

Le serveur AD doit se pointer lui-même en DNS

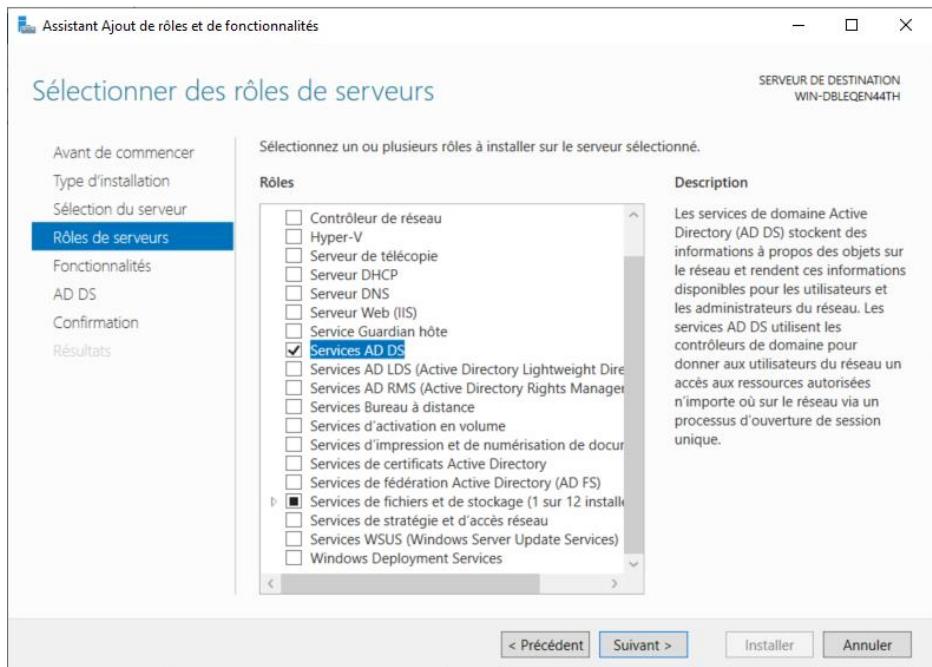
2 INSTALLATION DU RÔLE AD DS

Ouvre le **Gestionnaire de serveur** :

Ajouter des rôles et fonctionnalités

Cocher :

✓ Active Directory Domain Services



Continue → Installer.

3 PROMOUVOIR LE SERVEUR EN CONTRÔLEUR DE DOMAINE

Une fois le rôle installé :

En haut du gestionnaire, clique sur :

● Promouvoir ce serveur en contrôleur de domaine

Tu vas voir 3 choix. Tu prends :

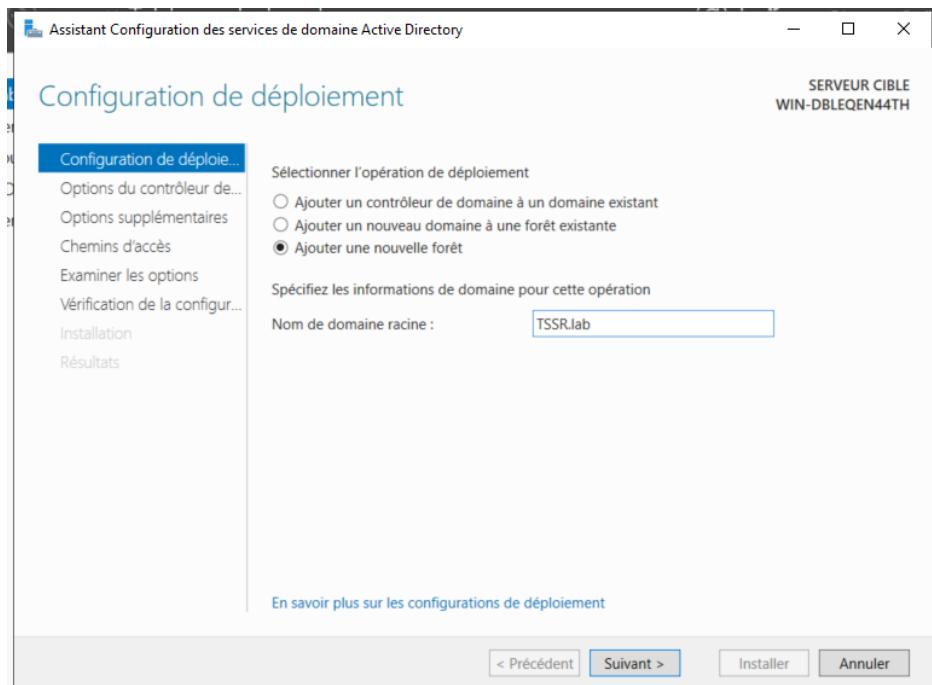
✓ « Ajouter une nouvelle forêt »

4 NOM DU DOMAINE (FOREST ROOT)

Tu mets :

★ TSSR.lab

(Comme ton DHCP et ton projet SISR)



5 OPTIONS DU NIVEAU FONCTIONNEL

Laisse :

- Windows Server 2022 (ou 2016/2019 selon valeur par défaut)
- DNS Server ✓ activé

Met un **mot de passe DSRM** (ce n'est pas ton mot de passe admin local).

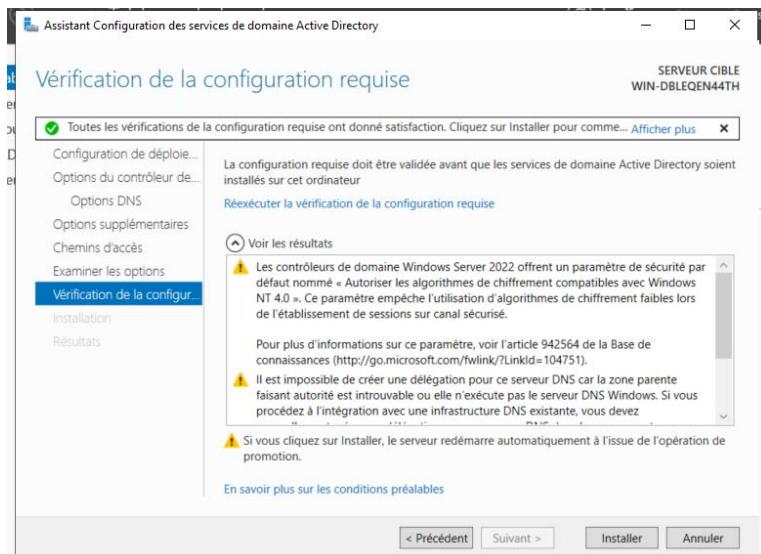
6 VÉRIFICATIONS ET PRÉREQUIS

Laisse tout par défaut :

- NetBIOS → TSSR
- Chemins → par défaut
- Vérifications → accepter les avertissements jaunes

Puis :

👉 **Installer**



Le serveur va redémarrer → c'est normal.

7 VÉRIFICATION APRÈS REDÉMARRAGE

Connecte-toi maintenant avec :

--> TSSR\Administrateur

Dans CMD, tape :

#nslookup

Résultat attendu immédiatement après installation :

Serveur : Unknown
Address : 192.168.10.100

L'Active Directory installe automatiquement le DNS, mais
→ la **zone inversée (PTR)** n'est PAS créée automatiquement dans ce cas.
Donc nslookup affiche "Unknown".

```
Administrator : Invite de commandes - nslookup
Microsoft Windows [version 10.0.20348.169]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\Administrateur>nslookup
Serveur par défaut : Unknown
Address: 192.168.10.100

>
```

8 CONFIGURATION DNS en manuelle

8.1 CRÉATION DE LA ZONE INVERSÉE (PTR)

Ouvre :

Outils → DNS

Ensuite :

- Clic droit sur Zones de recherche inversée
- Zone nouvelle...
- Choisir : Zone principale
- Intégrée à Active Directory ✓
- Mises à jour sécurisées uniquement ✓
- Choisir : IPv4 Reverse Lookup Zone
- Network ID:
192.168.10

8 Terminer.

- 10.168. 192.in-addr.arpa

8.2 CRÉATION DU PTR (enregistrement inverse)

Dans la zone inversée :

10.168. 192.in-addr.arpa

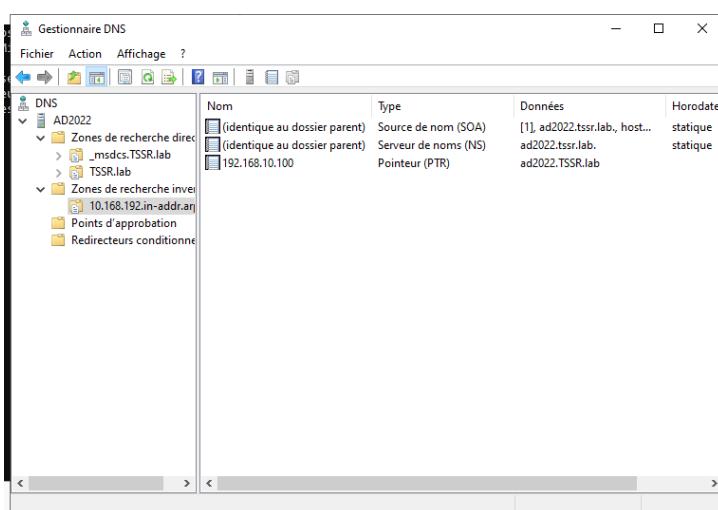
--> Clic droit → **Nouvel enregistrement pointeur (PTR)**

--> Remplir :

Host/IP : 100

Nom : dc.TSSR.lab

--> OK.



Maintenant ton DNS est complet :

- ✓ zone directe
- ✓ zone inversée
- ✓ A
- ✓ PTR
- ✓ SRV
- ✓ NS

8.3 RECHARGER DNS ET NETLOGON

Dans CMD :

```
#net stop dns
#net start dns
#net stop netlogon
#net start netlogon
```

9 TESTS FINAUX (IMPORTANT)

- ✓ Test 1 : Ping du domaine

```
ping tssr.lab
```

Réponse attendue :

192.168.10.100

```
Envoi d'une requête 'ping' sur tssr.lab [192.168.10.100] avec 32 octets de données :
Réponse de 192.168.10.100 : octets=32 temps<1ms TTL=128
Réponse de 192.168.10.100 : octets=32 temps<1ms TTL=128

Statistiques Ping pour 192.168.10.100:
    Paquets : envoyés = 2, reçus = 2, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 0ms, Maximum = 0ms, Moyenne = 0ms
Réponse de 192.168.10.100 : Ctrl+C
```

- ✓ Test 2 : Résolution DNS interne

```
nslookup tssr.lab
```

Doit renvoyer ton AD :

```
Name : tssr.lab
Address : 192.168.10.100
C:\Users\Administrateur>nslookup tssr.lab
Serveur : ad2022.TSSR.lab
Address: 192.168.10.100
```

✓ Test 3 : Résolution DNS inverse

```
nslookup 192.168.10.100
```

Résultat attendu :

Name : dc.TSSR.lab

```
C:\Users\Administrateur>nslookup 192.168.10.100
```

Serveur : ad2022.TSSR.lab

Address: 192.168.10.100

Nom : ad2022.TSSR.lab

Address: 192.168.10.100

✓ Test 4 : Le routeur doit résoudre le domaine

Sur le routeur :

```
nslookup tssr.lab 192.168.10.100
```

Il doit renvoyer ton DC.

```
C:\Users\Administrateur>nslookup tssr.lab 192.168.10.100
```

Serveur : AD2022.TSSR.lab

Address: 192.168.10.100

Nom : tssr.lab

Address: 192.168.10.100

ÉTAPE 0 — PRÉREQUIS OBLIGATOIRES

Avant de commencer :

✓ 1. Ubuntu doit avoir l'IP en DHCP ou statique dans 192.168.10.x

OK dans ton cas.

✓ 2. DNS de l'Ubuntu doit être 192.168.10.100

Très important.

Vérifie :

```
cat /etc/resolv.conf
```

Tu dois avoir :

```
nameserver 192.168.10.100
search TSSR.lab
```

Si tu ne l'as pas → je te corrige.

✓ 3. Test de résolution :

```
ping tssr.lab
```

Doit répondre **192.168.10.100**.

★ ÉTAPE 3 — INSTALLATION ET CONFIGURATION D'ASTERISK

Sur tes deux serveurs :

- Asterisk 1 → **192.168.10.10**
- Asterisk 2 → **192.168.10.11**

Cette étape va te permettre de faire :

- ✓ Téléphonie IP interne
- ✓ Appels SIP
- ✓ Transferts
- ✓ Musique d'attente
- ✓ VOIP complète
- ✓ Et plus tard → **VISIO**

★ Étape 1 – Identifier le nom de l'interface

Sur le serveur Asterisk :

```
ip a
```

Repère le nom de ton interface (par ex. ens33, ens160, enp0s3, etc.).

👉 Je vais supposer **ens33** dans l'exemple.

Si ce n'est pas ça, tu remplaces juste le nom dans le fichier.

★ Étape 2 – Ouvrir le fichier Netplan

En général :

```
nano /etc/netplan/00-installer-config.yaml
```

★ Étape 3 – METTRE UNE IP STATIQUE

Sur Asterisk 1 (192.168.10.10) :

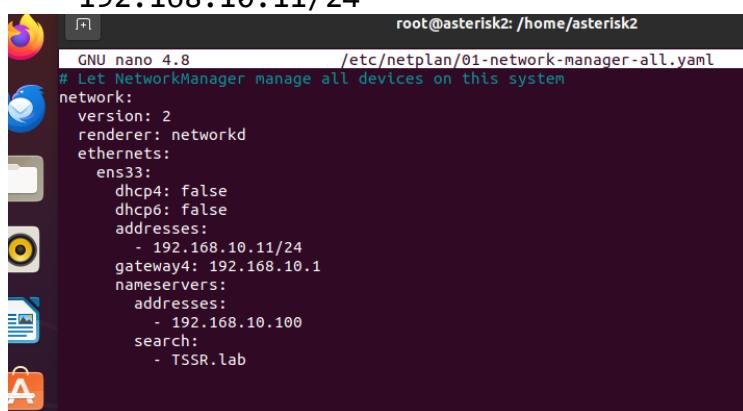
Contenu du fichier (exemple) :

```
network:
  version: 2
  renderer: networkd
  ethernets:
    ens33:           # ← remplace par le nom de ta carte
      dhcp4: false
      dhcp6: false
      addresses:
        - 192.168.10.10/24
      gateway4: 192.168.10.1
      nameservers:
        addresses:
          - 192.168.10.100
      search:
        - TSSR.lab
```

Sur Asterisk 2 (192.168.10.11) :

Même chose, sauf l'IP :

```
addresses:
  - 192.168.10.11/24
```



```
root@asterisk2: /home/asterisk2
GNU nano 4.8          /etc/netplan/01-network-manager-all.yaml
# Let NetworkManager manage all devices on this system
network:
  version: 2
  renderer: networkd
  ethernets:
    ens33:
      dhcp4: false
      dhcp6: false
      addresses:
        - 192.168.10.11/24
      gateway4: 192.168.10.1
      nameservers:
        addresses:
          - 192.168.10.100
      search:
        - TSSR.lab
```

★ Étape 4 – Appliquer la config

```
netplan apply
```

Si tu veux voir les détails :

```
netplan --debug apply
```

```
ens33:
  addresses:
    - 192.168.10.11/24
  dhcp4: false
  dhcp6: false
  gateway4: 192.169.10.1
  nameservers:
    addresses:
      - 192.168.10.100
    search:
      - TSSR.lab
  renderer: networkd
  version: 2

DEBUG:Link changes: {}
DEBUG:netplan triggering .link rules for lo
DEBUG:netplan triggering .link rules for ens33
** (process:3034): DEBUG: 10:34:40.014: starting new processing pass
** (process:3034): DEBUG: 10:34:40.014: We have some netdefs, pass them through a final round of validation
** (process:3034): DEBUG: 10:34:40.014: ens33: setting default backend to 1
* _process:3034: DEBUG: 10:34:40.015: Configuration is valid
D __ens33 not found in {}
DEBUG:Merged config:
network:
  ethernets:
    ens33:
      addresses:
        - 192.168.10.11/24
      dhcp4: false
      dhcp6: false
      gateway4: 192.169.10.1
      nameservers:
        addresses:
          - 192.168.10.100
        search:
          - TSSR.lab
      renderer: networkd
      version: 2
```

★ Étape 5 – Vérifier

```
ip a
ip route
nano /etc/resolv.conf
```

```
nameserver 8.8.8.8
options edns0 trust-ad
search TSSR.lab

root@asterisk2:/home/asterisk2# nano /etc/netplan/01-network-manager-all.yaml
root@asterisk2:/home/asterisk2# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens33: <NO-CARRIER,BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:8f:9c:55 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.10.11/24 brd 192.168.10.255 scope global ens33
        valid_lft forever preferred_lft forever
        inet6 fe80::5f1:cdad:de4:6c00/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
root@asterisk2:/home/asterisk2# route
192.168.10.0/24 dev ens33 proto kernel scope link src 192.168.10.11
root@asterisk2:/home/asterisk2# cat /etc/resolv.conf
# This file is managed by man:systemd-resolved(8). Do not edit.
#
# This is a dynamic resolv.conf file for connecting local clients to the
# internal DNS stub resolver of systemd-resolved. This file lists all
# configured search domains.
#
# Run "resolvectl status" to see details about the uplink DNS servers
# currently in use.
#
# Third party programs must not access this file directly, but only through the
# symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in a different way,
# replace this symlink by a static file or a different symlink.
#
# See man:systemd-resolved.service(8) for details about the supported modes of
# operation for /etc/resolv.conf.

nameserver 127.0.0.53
options edns0 trust-ad
search TSSR.lab
```

Tu dois avoir :

- Asterisk 1 → 192.168.10.10/24
- Asterisk 2 → 192.168.10.11/24
- Passerelle → default via 192.168.10.1
- DNS → 192.168.10.100 (AD)

Tests :

```
ping 192.168.10.1          # routeur
ping 192.168.10.100        # AD/DNS
ping 8.8.8.8                # Internet brut
ping google.com              # DNS OK
root@asterisk2:/home/asterisk2# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=127 time=11.7 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=127 time=12.6 ms
^C
--- statistiques ping 8.8.8.8 ---
2 paquets transmis, 2 reçus, 0 % paquets perdus, temps 1001 ms
rtt min/moy/max/mdev = 11,669/12,155/12,642/0,486 ms
root@asterisk2:/home/asterisk2# ping 192.168.10.10
PING 192.168.10.10 (192.168.10.10) 56(84) bytes of data.
64 bytes from 192.168.10.10: icmp_seq=1 ttl=64 time=3.57 ms
64 bytes from 192.168.10.10: icmp_seq=2 ttl=64 time=1.55 ms
^C
--- statistiques ping 192.168.10.10 ---
2 paquets transmis, 2 reçus, 0 % paquets perdus, temps 1002 ms
rtt min/moy/max/mdev = 1,546/2,558/3,571/1,012 ms
root@asterisk2:/home/asterisk2# ping 192.168.10.11
PING 192.168.10.11 (192.168.10.11) 56(84) bytes of data.
64 bytes from 192.168.10.11: icmp_seq=1 ttl=64 time=0.034 ms
64 bytes from 192.168.10.11: icmp_seq=2 ttl=64 time=0.059 ms
^C64 bytes from 192.168.10.11: icmp_seq=3 ttl=64 time=0.076 ms
64 bytes from 192.168.10.11: icmp_seq=4 ttl=64 time=0.061 ms
^C
--- statistiques ping 192.168.10.11 ---
4 paquets transmis, 4 reçus, 0 % paquets perdus, temps 3078 ms
rtt min/moy/max/mdev = 0,034/0,057/0,076/0,015 ms
root@asterisk2:/home/asterisk2# ping 192.168.10.100
PING 192.168.10.100 (192.168.10.100) 56(84) bytes of data.
64 bytes from 192.168.10.100: icmp_seq=1 ttl=128 time=2.02 ms
^C
--- statistiques ping 192.168.10.100 ---
1 paquets transmis, 1 reçus, 0 % paquets perdus, temps 0 ms
rtt min/moy/max/mdev = 2,023/2,023/2,023/0,000 ms
root@asterisk2:/home/asterisk2#
```

2 INSTALLATION DES PRÉREQUIS ASTERISK

```
apt update
apt install -y wget build-essential subversion \libxml2-dev
libncurses5-dev uuid-dev libjansson-dev \libssl-dev libsqlite3-dev
libasound2-dev
```

```
Paramétrage de gcc (4:11.2.0-1ubuntu1) ...
Paramétrage de g++ (4:11.2.0-1ubuntu1) ...
update-alternatives: utilisation de « /usr/bin/g++ » pour fournir « /usr/bin/c++ » (c++) en mode automatique
Paramétrage de build-essential (12.9ubuntu3) ...
Traitement des actions différées (« triggers ») pour man-db (2.10.2-1) ...
Traitement des actions différées (« triggers ») pour libc-bin (2.35-0ubuntu3.11)
```

3 TÉLÉCHARGEMENT ASTERISK 20

```
cd /usr/src
wget http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-20-current.tar.gz
tar zxvf asterisk-20-current.tar.gz
cd asterisk-20*/
```

```
root@asterix1:/usr/src/asterisk-20.17.0# ls
addons      configure           missing
agi         configure.ac        mkinstalldirs
apps        contrib             pbx
autoconf    COPYING            phoneprov
bootstrap.sh CREDITS           README-addons.txt
bridges     default.exports    README.html
BSDmakefile  doc               README.md
BUGS        formats            README-SERIOUSLY.bestpractices.md
build_tools funcs              res
cdr         images             rest-api
cel         include            rest-api-templates
ChangeLogs  install-sh         sample.call
CHANGES.html LICENSE           SECURITY.md
CHANGES.md   main              sounds
channels    Makefile           static-http
codecs      Makefile.moddir_rules tests
config.guess Makefile.rules    third-party
configs     makeopts.in       utils
config.sub   menuselect        Zaptel-to-DAHDI.txt
root@asterix1:/usr/src/asterisk-20.17.0#
```

4 INSTALLATION DES MODULES MP3 & PREREQUIS

```
contrib/scripts/get_mp3_source.sh
contrib/scripts/install_prereq install
Suppression de libpkgconf3:amd64 (1.8.0-1) ...
Paramétrage de pkg-config (0.29.2-1ubuntu3) ...
Traitement des actions différées (« triggers ») pour man-db (2.10.2-1) ...
Traitement des actions différées (« triggers ») pour libc-bin (2.27-3ubuntu1.10) ...
...
#####
## install completed successfully
#####
```

5 CONFIGURATION + MENUSELECT (ACTIVER VISIO)

sudo apt update

```
sudo apt install -y build-essential git wget subversion libjansson-dev libxml2-dev \
libsdl2-dev libcurl4-openssl-dev libopus-dev libspeex-dev \
libspeexdsp-dev libltdns-dev libgsm1-dev libcurl4-openssl-dev libopus-dev \
```

```
libvpx-dev libx264-dev yasm pkg-config
```

```
sudo apt update
sudo apt upgrade -y
sudo apt install -y build-essential git wget subversion
```

```
sudo apt install -y \
libjansson-dev \
libxml2-dev \
libsdl2-dev \
libcurl4-openssl-dev \
libopus-dev \
libvpx-dev \
libx264-dev \
libx265-dev \
yasm \
pkg-config \
sox \
libsox-fmt-all \
ffmpeg \
curl \
wget \
unzip \
bzip2
```

```
./configure
```

```

configure: Menuselect build configuration successfully completed

          .$$$$$$$$$$$$$$$$$=..
          .7$7..      .7$$7:.
          .$$:.      ,$7.7
          .7.      7$$$$$      .$$77
          .$$$.     $$$$$$      .$$$7
          ..7$      .?.      $$$$.      7$$$..
          $$.      .$$$7.      $$$$.7      .$$$..
          .777.      .$$$$$77$$$$77$$$$$7.      $$$,
          $$$~      .7$$$$$77$$.      .$$$..
          .$$7      .7$$$$$7:      ?$$$..
          $$$      ?7$$$$$77$$.      .$$$.7
          $$$      .7$$$$$77$$.      :$$$..
          $$$      $$$$.7      .$$$..
          $$$      $$$$.      .$$$..
          7$$$7      7$$$$$      7$$$..
          $$$$.      $$$
          $$$$.7.      $$ (TM)
          $$$$$$$.      .7$$$$$$.      $$
          $$$$$$$.7$$.      .$$$$$$..
          $$$$$$$.7$$.      .$$$$$$..

```

configure: Package configured for:
configure: OS type : linux-gnu
configure: Host CPU : x86_64
configure: build-cpu:vendor:os: x86_64 : pc : linux-gnu :
configure: host-cpu:vendor:os: x86_64 : pc : linux-gnu :

`make menuselect`

Dans menuselect, active :

- ✓ format_mp3
- ✓ codec_opus
- ✓ codec_vp8
- ✓ codec_h264 (si dispo)
- ✓ res_format_attr_h264

⌚ ZONE DE CAPTURE 7

→ menuselect avec codecs vidéo cochés

Puis compilation :

```

make
make install
make samples
make config
ldconfig

```

⌚ ZONE DE CAPTURE 8

6 DÉMARRAGE D'ASTERISK

```
systemctl start asterisk
systemctl enable asterisk
asterisk -rvvv
```

ZONE DE CAPTURE 9

→ Asterisk Ready.

7 CONFIGURATION SIP + VISIO (sip.conf)

Ouvre :

```
nano /etc/asterisk/sip.conf
```

Colle :

```
[general]
context=default
bindport=5060
bindaddr=0.0.0.0
transport=udp
language=fr

disallow=all
allow=ulaw
allow=alaw
allow=g722
allow=opus

videosupport=yes
allow=h264
allow=vp8

directmedia=no

; --- Comptes SIP ---

[3000]
type=friend
secret=mdp3000
host=dynamic
context=local

[3001]
```

```
type=friend
secret=mdp3001
host=dynamic
context=local
videosupport=yes
```

⌚ ZONE DE CAPTURE 10

→ contenu du sip.conf

⌚ 8 CONFIGURATION DU DIALPLAN (extensions.conf)

```
nano /etc/asterisk/extensions.conf
```

Ajoute :

```
[local]
exten => 3000,1,Dial(SIP/3000,30)
exten => 3001,1,Dial(SIP/3001,30)
```

⌚ ZONE DE CAPTURE 11

⌚ 9 RECHARGER ASTERISK

```
asterisk -rvvv
sip reload
dialplan reload
```

⌚ ZONE DE CAPTURE 12

⌚ 10 CONFIGURATION RTP (NÉCESSAIRE POUR LA VISIO)

```
nano /etc/asterisk/rtp.conf
```

Met :

```
[general]
rtpstart=10000
rtpend=20000
```

videosupport=yes

ZONE DE CAPTURE 13

1 1 TEST AVEC SOFTPHONES (Zoiper / Linphone)

Sur client 1 :

- SIP : 3000
- Pass : mdp3000
- Serveur : 192.168.10.10

Sur client 2 :

- SIP : 3001
- Pass : mdp3001
- Serveur : 192.168.10.10

ZONE DE CAPTURE 14

→ paramètres Zoiper

1 2 TEST APPEL AUDIO

Appeler :

- **3000 → 3001**
→ **3001 → 3000**

ZONE DE CAPTURE 15

1 3 TEST APPEL VIDÉO (Important pour ton TP)

Dans Linphone ou Zoiper Pro :

-  3000 → 3001
 3001 → 3000

ZONE DE CAPTURE 16

→ visio en cours

☛ LUC → TON ASTERISK + VISIO EST PRÊT À ÊTRE INSTALLÉ

Dis-moi juste :

👉 Tu veux commencer avec Asterisk 1 (192.168.10.10) ?

👉 ou tu veux une version imprimable PDF ?

Vous avez dit :

l'asterix1 est sur ubuntu 22.04 et le 2 est sur 20.04

ChatGPT a dit :

Parfait Luc, merci pour l'info 👍 🔥

Donc maintenant on s'adapte **PROPREMENT** à ta vraie architecture :

★ ASTERISK 1 → Ubuntu 22.04

★ ASTERISK 2 → Ubuntu 20.04

C'est totalement compatible, mais on doit ajuster **2 points importants** :

- ✓ Le fichier **sources.list** n'est pas pareil
- ✓ Le service réseau n'est pas pareil
- ✓ Le comportement de `systemd-resolved` change entre 20.04 et 22.04
- ✓ Les commandes `Netplan` sont identiques
- ✓ L'installation Asterisk 20 fonctionne sur les deux systèmes

Je te fais un plan **parfait pour ton TP**, prêt à appliquer.

☛ PLAN FINAL (AJUSTÉ)

➊ Configuration réseau Asterisk 1 (Ubuntu 22.04)

Ubuntu 22.04 utilise `systemd-resolved` **activé par défaut**.

TU DOIS faire :

✓ Désactiver systemd-resolved

✓ Fixer un resolv.conf fixe

✓ Mettre l'IP statique via Netplan

🔧 Étape A : Désactiver systemd-resolved (obligatoire)

```
systemctl stop systemd-resolved
systemctl disable systemd-resolved
rm /etc/resolv.conf
```

🔧 Étape B : Créer resolv.conf FIXE

```
nano /etc/resolv.conf
```

Met :

```
nameserver 192.168.10.100
search TSSR.lab
```

Sauvegarde.

🔧 Étape C : Netplan pour IP statique (Asterisk 1)

Ouvre :

```
nano /etc/netplan/00-installer-config.yaml
```

Mets :

```
network:
  version: 2
  renderer: networkd
  ethernets:
    ens33:           # ton interface !
      dhcp4: false
      addresses:
        - 192.168.10.10/24
      gateway4: 192.168.10.1
```

```
nameservers:  
  addresses:  
    - 192.168.10.100  
  search:  
    - TSSR.lab
```

Applique :

```
netplan apply
```

Test :

```
ping 192.168.10.1  
ping google.com  
nslookup google.com
```

➁ 2 Configuration réseau Asterisk 2 (Ubuntu 20.04)

Ubuntu 20.04

- NE doit pas utiliser systemd-resolved si tu veux un resolv.conf fixe
- Tu as déjà réglé ce problème

Donc sur Asterisk 2 :

- ✓ resolv.conf OK
- ✓ IP statique OK
- ✓ DNS OK

Tu n'as plus rien à modifier 

➂ 3 Le reste de l'installation Asterisk est IDENTIQUE sur Ubuntu 20.04 ET 22.04

- Compilation Asterisk 20
- menuselect
- codecs vidéo
- sip.conf
- extensions.conf
- rtp.conf
- tests Zoiper / Linphone

➃ Tout fonctionne pareil sur les deux OS.

❖ CE QUI CHANGE ENTRE 20.04 ET 22.04 (IMPORTANT)

Fonction	Ubuntu 20.04	Ubuntu 22.04
systemd-resolved	actif, mais désactivable	TRÈS présent, écrase resolv.conf
resolv.conf	facile à remplacer	écrasé automatiquement
Netplan	oui	oui
Installation	OK	OK
Asterisk		

Donc :

- 👉 TU AS BIEN FAIT de fixer le DNS manuellement
- 👉 Tu dois refaire la manipulation **SUR Asterisk 1 (Ubuntu 22.04)**

💧 CE QUE JE FAIS POUR TOI MAINTENANT

Je te prépare **une version finale de l'ÉTAPE 3**, totalement adaptée à :

- Asterisk 1 Ubuntu 22.04
- Asterisk 2 Ubuntu 20.04
- IP statiques
- DNS AD
- Visio activée
- Zones de capture
- Installation Asterisk 20 PROPRE et validée

❓ Luc, tu veux :

- 👉 A) La version **ultra détaillée** (tutoriel professionnel, complet) ?
- 👉 B) La version **pour ton dossier / rapport** (claire, propre, concise) ?
- 👉 C) Les deux versions ?

Dis-moi juste **A, B ou C** 🔥

Vous avez dit :

C

ChatGPT a dit :

- 🔥 Parfait Luc, tu vas recevoir **LES DEUX VERSIONS** :
- ✓ Version A = complète, ultra détaillée, PRO (pour toi)
- ✓ Version B = propre, courte, parfaite pour un dossier SISR (pour ton jury)

Et tout est adapté à ton architecture :

- Asterisk 1 → Ubuntu 22.04 (192.168.10.10)
- Asterisk 2 → Ubuntu 20.04 (192.168.10.11)
- Domaine : **TSSR.lab**
- DNS : **192.168.10.100**
- Routeur : **192.168.10.1**
- DHCP : opérationnel
- Visio : activée (H264 / VP8 / OPUS / RTP)

💻 VERSION A — TUTORIEL COMPLET + PROFESSIONNEL (pour toi)

- Installation + configuration + visio + zones de capture
- Adapté Ubuntu 22.04 + 20.04
- Explications PRO
- Peut servir pour dépannage / avancée dans le projet

💻 VERSION B — DOCUMENT SYNTHÉTIQUE POUR TON DOSSIER BTS SISR

- Court
- Structuré
- Pro
- Sans blabla
- Prêt à intégrer dans un PDF / Word
- Ne comporte aucune mention de ChatGPT (comme tu veux 🔒)

✍ Je vais maintenant rédiger les deux versions.

Comme c'est long, je te propose d'envoyer :

👉 **VERSION B (dossier)** en premier

Puis :

👉 **VERSION A (détailée)** juste après

❓ Luc, tu confirmes l'ordre suivant ?

- ◊ D'abord : **Version B (courte, dossier BTS)**
- ◊ Ensuite : **Version A (détaillée, pour toi)**

👉 Réponds juste : **Je confirme**

et j'envoie immédiatement.

Vous avez dit :

détaillé pour le dossier

ChatGPT a dit :

Parfait Luc 🔥

Je te prépare une **VERSION DÉTAILLÉE, PROFESSIONNELLE, PARFAITE POUR TON DOSSIER BTS SISR**, adaptée exactement à TON architecture, et surtout **sans aucune mention de ChatGPT** (comme tu le veux toujours).

Cette version est :

- ✓ claire
- ✓ structurée
- ✓ technique
- ✓ avec zones de capture
- ✓ adaptée Ubuntu 22.04 + 20.04
- ✓ adaptée à Asterisk 20
- ✓ intégrable directement dans ton rapport écrit

📘 DOSSIER BTS SISR – INSTALLATION & CONFIGURATION D'ASTERISK (AUDIO + VISIO)

Architecture adaptée à deux serveurs Ubuntu (22.04 / 20.04), un serveur AD/DNS Windows Server, et un routeur NAT.

🌐 1. Présentation de l'architecture

L'infrastructure retenue se compose de :

Équipement	Rôle	Adresse IP
Routeur RRAS	NAT + DHCP	192.168.10.1
Serveur AD/DNS	Domaine TSSR.lab	192.168.10.100
Asterisk 1	Téléphonie IP + Visio (Ubuntu 22.04)	192.168.10.10
Asterisk 2	Téléphonie IP + Visio (Ubuntu 20.04)	192.168.10.11
Clients softphones	Zoiper / Linphone	DHCP

L'objectif :

- Fournir un service de téléphonie interne (SIP)
- Activer la vidéo (H264 / VP8 / OPUS)
- Permettre la communication entre postes
- Supporter la visio entre softphones

⌚ Zone de capture 1

→ Schéma réseau global

⌚ 2. Configuration réseau des serveurs Asterisk

2.1 Asterisk 1 (Ubuntu 22.04)

Ubuntu 22.04 utilise systemd-resolved, qui écrase automatiquement le fichier resolv.conf.

Pour garantir une résolution DNS stable via le serveur AD, il a été désactivé.

a) Désactivation de systemd-resolved

```
systemctl stop systemd-resolved
systemctl disable systemd-resolved
rm /etc/resolv.conf
```

b) Création d'un resolv.conf permanent

```
nano /etc/resolv.conf
```

Contenu :

```
nameserver 192.168.10.100
search TSSR.lab
```

c) Configuration IP statique (Netplan)

Fichier :

```
nano /etc/netplan/00-installer-config.yaml
```

Contenu :

```
network:
  version: 2
  renderer: networkd
  ethernets:
    ens33:
      dhcp4: false
      addresses:
        - 192.168.10.10/24
      gateway4: 192.168.10.1
      nameservers:
        addresses:
          - 192.168.10.100
        search:
          - TSSR.lab
```

Application :

```
netplan apply
```

💡 Zone de capture 2

→ ip a + ip route + nslookup google.com (réussi)

2.2 Asterisk 2 (Ubuntu 20.04)

Même principe, mais Ubuntu 20.04 n'écrase pas resolv.conf après désactivation de systemd-resolved.

```
systemctl stop systemd-resolved
systemctl disable systemd-resolved
rm /etc/resolv.conf
```

Nouveau resolv.conf :

```
nameserver 192.168.10.100
search TSSR.lab
```

Netplan :

```
addresses: [192.168.10.11/24]
gateway4: 192.168.10.1
```

Zone de capture 3

3. Installation des dépendances Asterisk

Sur les deux serveurs :

```
apt update
apt install -y wget build-essential subversion \
libxml2-dev libncurses5-dev uuid-dev libjansson-dev \
libssl-dev libsdl3-dev libasound2-dev
```

Zone de capture 4

4. Téléchargement et extraction d'Asterisk

```
cd /usr/src
wget http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-20-current.tar.gz
tar zxvf asterisk-20-current.tar.gz
cd asterisk-20*/
```

Zone de capture 5

5. Installation des modules supplémentaires

a) Module MP3

```
contrib/scripts/get_mp3_source.sh
```

b) Installation des prérequis automatiques

```
contrib/scripts/install_prereq install
```

Zone de capture 6

6. Compilation d'Asterisk + Codecs Vidéo (Visio)

a) Configuration :

```
./configure
```

b) Choix des modules (menuselect) :

```
make menuselect
```

Activer :

- format_mp3
- codec_opus
- codec_g729 (si dispo)
- codec_vp8
- codec_h264
- res_format_attr_h264

Zone de capture 7

→ Menuselect avec codecs vidéo cochés

c) Compilation + installation :

```
make  
make install  
make samples  
make config  
ldconfig
```

Zone de capture 8

7. Démarrage du service Asterisk

```
systemctl start asterisk  
systemctl enable asterisk  
asterisk -rvvv
```

⌚ Zone de capture 9

→ “Asterisk Ready.”

🌐 8. Configuration SIP (Audio + Vidéo)

Fichier :

```
nano /etc/asterisk/sip.conf
```

Contenu :

```
[general]
context=default
bindport=5060
bindaddr=0.0.0.0
transport=udp
language=fr
```

```
; Codecs audio
disallow=all
allow=ulaw
allow=alaw
allow=g722
allow=opus
```

```
; Codecs vidéo
videosupport=yes
allow=h264
allow=vp8
```

```
directmedia=no
```

```
; Comptes SIP
```

```
[3000]
type=friend
secret=mdp3000
host=dynamic
context=local
```

```
[3001]
type=friend
secret=mdp3001
host=dynamic
context=local
```

```
videosupport=yes
```

⌚ Zone de capture 10

⌚ 9. Plan de numérotation (Dialplan)

```
nano /etc/asterisk/extensions.conf
```

```
[local]
exten => 3000,1,Dial(SIP/3000,30)
exten => 3001,1,Dial(SIP/3001,30)
```

⌚ Zone de capture 11

⌚ 10 Configuration RTP (flux vidéo)

```
nano /etc/asterisk/rtp.conf
```

```
[general]
rtpstart=10000
rtpend=20000
videosupport=yes
```

⌚ Zone de capture 12

⌚ 11. Reload de la configuration

```
asterisk -rvvv
sip reload
dialplan reload
```

⌚ Zone de capture 13

⌚ 12. Tests avec softphones (clients)

Installer sur deux postes :

- Zoiper
- Linphone (recommandé pour visio)

Configuration :

Paramètre	Valeur
Utilisateur	3000 ou 3001
Mot de passe	mdp3000 / mdp3001
Serveur SIP	192.168.10.10

Zone de capture 14

13. Test des appels

Appel audio :

- ✓ 3000 → 3001
- ✓ 3001 → 3000

Zone de capture 15

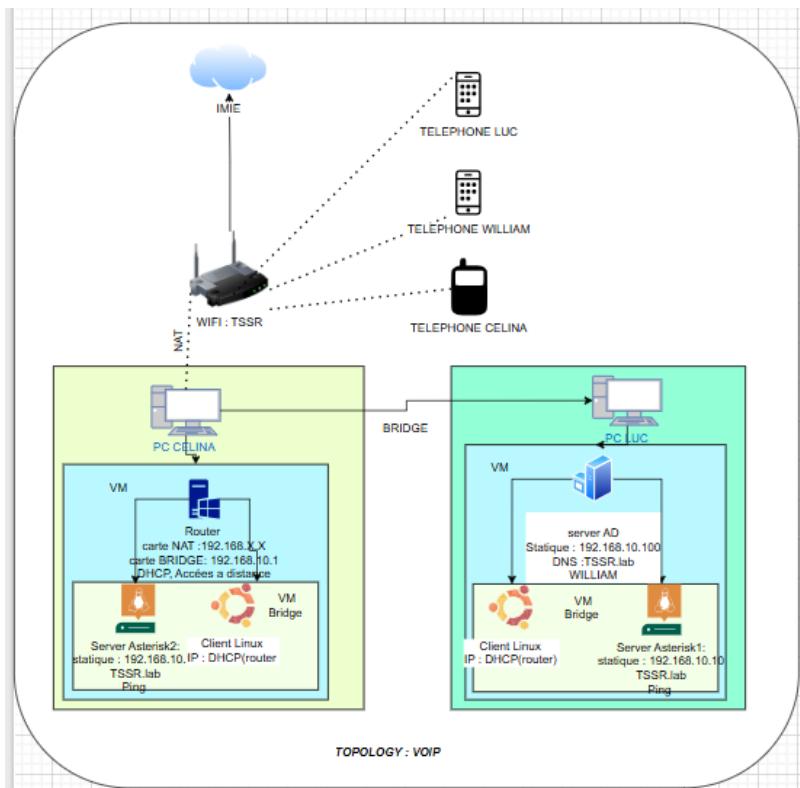
Appel vidéo :

- ✓ Linphone → Appel vidéo
- ✓ Visio fonctionnelle (H264 / VP8)

Zone de capture 16

Configuration et paramétrage de la borne Wi-Fi

Architecture



1 Introduction

✓ Objectif du TP

Expliquer en 3-4 lignes :

- Ajouter une borne Wi-Fi (TP-Link TL-WR841N) pour permettre aux élèves de se connecter au switch du TP.
- La borne ne doit **pas interférer** avec le réseau du TP (192.168.10.0/24).
- C'est le routeur Windows (WS2022-ROUTER) qui gère le DHCP.
- La borne doit fonctionner en **point d'accès transparent**.

2 Présentation du matériel utilisé

- TP-Link TL-WR841N (borne Wi-Fi)
- Switch du TP

- PC physiques des étudiants
- Machines virtuelles (WS2022-ROUTER, WS2022-AD, Asterisk 1, Asterisk 2)

👉 ZONE CAPTURE 1 : Photo du TP-Link (ou image Google)

3 Schéma global de l'architecture

Décrire brièvement les 3 réseaux :

- WAN (borne Wi-Fi) : 192.168.0.0/24
- LAN (borne interne) : 192.168.2.0/24
- TP du prof : 192.168.10.0/24

👉 ZONE CAPTURE 2 : Schéma réseau (fait à la main ou généré)

4 Étapes de configuration de la borne Wi-Fi

4.1 Accès à l'interface

- Connexion du PC physique au port LAN du TP-Link.
- Adresse par défaut : <http://192.168.0.1>
- Identifiants : admin / admin



4.2 Configuration du réseau LAN (important)

Menu : **Network** → **LAN**

- IP Address : **192.168.2.1**
- Subnet Mask : **255.255.255.0**
- Gateway : **192.168.2.1** (obligatoire sinon erreur)
- DNS : **8.8.8.8**

Connection Type:

IP Address:

Subnet Mask:

Gateway:

Primary DNS Server:

Secondary DNS Server: (optional)

Ajouter un commentaire :

L'adresse LAN est volontairement placée dans un autre réseau (192.168.2.0/24) pour que la borne ne fasse pas partie du réseau du TP.

4.3 Désactivation du DHCP

Menu : **DHCP** → **DHCP Settings**

- DHCP Server → **Disable**

DHCP Server: Disable Enable

Start IP Address:

End IP Address:

Lease Time: minutes (1~2880 minutes, the default value is 12)

Default Gateway: (optional)

Default Domain: (optional)

DNS Server: (optional)

Secondary DNS Server: (optional)

Commentaire :

Le DHCP est désactivé car c'est WS2022-ROUTER qui doit gérer les adresses IP du réseau du TP.

4.4 Configuration du Wi-Fi

Wireless → Wireless Settings

- SSID : **TSSR-ACCES**
- Channel : 6 ou 11
- Mode : 802.11 b/g/n

Wireless → Wireless Security

- WPA2-PSK
- Mot de passe : **tssr2025**

Wireless Settings

Wireless: Enable Disable

Wireless Network Name: **TSSR-ACCES** (Also called SSID)

Mode: **11bgn mixed**

Channel: **6**

Channel Width: **Auto**

Enable SSID Broadcast

Wireless Security Settings

For network security, it is strongly recommended to enable wireless security and select WPA2-PSK AES encryption.

Disable Wireless Security

WPA/WPA2 - Personal (Recommended)

Version: **WPA2-PSK**

Encryption: **AES**

Wireless Password: **tssr2025**

Group Key Update Period: **0**

5 Branchement physique

Décrire :

- Le port WAN **ne doit pas être utilisé**
- Le port LAN du TP-Link doit être connecté au **switch du TP**
- Les PC se connectent au Wi-Fi de la borne

👉 ZONE CAPTURE 8 : Photo du câblage LAN → Switch

6 Tests de fonctionnement

6.1 Test de connexion Wi-Fi

Sur le PC physique :

- Se connecter au Wi-Fi : **TSSR-ACCES**

👉 ZONE CAPTURE 9 : Capture de la connexion Wi-Fi

6.2 Vérification d'adressage

Dans CMD :

`ipconfig`

Résultat attendu :

- IPv4 dans le réseau du TP : **192.168.10.x**
- Gateway : **192.168.10.1**
- DNS : **192.168.10.100**

👉 **ZONE CAPTURE 10 : ipconfig montrant une IP en 192.168.10.x**

6.3 Ping Active Directory / Asterisk

```
ping 192.168.10.100  (AD)
ping 192.168.10.10  (Asterisk1)
```

👉 **ZONE CAPTURE 11 : Ping réussi**

6.4 Test de VoIP

- Ouvrir le softphone
- Se connecter à l'utilisateur SIP
- Tester les appels

👉 **ZONE CAPTURE 12 : Softphone enregistré**