

## Partie 1 : Préparation de l'Environnement de Base sur Ubuntu

L'objectif de cette partie est d'installer les outils de développement fondamentaux.

1. **Ouvrir un Terminal :** La plupart des opérations se feront en ligne de commande. Ouvrez un terminal (raccourci : Ctrl+Alt+T).

2. **Mettre à jour le système :** Assurez-vous que votre système est à jour.

Bash

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

3. **Installer les Outils Essentiels (Python, Git, venv) :** Python et Git sont souvent pré-installés, mais nous vérifions et installons les paquets nécessaires.

Bash

```
sudo apt install -y python3 python3-pip python3-venv git
```

4. **Installer Visual Studio Code (VS Code) :** Nous l'installons via le gestionnaire de paquets snap pour une intégration simple.

Bash

```
sudo snap install --classic code
```

## Partie 2 : Configuration du Projet et de GitHub

L'objectif est de récupérer le projet depuis le dépôt centralisé et de le configurer localement.

1. **Configurer votre identité Git :** Git a besoin de savoir qui vous êtes. Remplacez "Votre Nom" et "[votre.email@example.com](mailto:votre.email@example.com)".

Bash

```
git config --global user.name "Votre Nom"  
git config --global user.email "votre.email@example.com"
```

2. **Cloner le Dépôt du Projet :** Créez une copie locale du projet depuis GitHub.

- Naviguez vers votre dossier personnel : cd ~
- Clonez le dépôt (remplacez Nathanael-Git par le nom d'utilisateur GitHub de votre coéquipier) :

Bash

```
git clone https://github.com/Nathanael-Git/projet-ia-reseau.git
```

3. **Ouvrir le Projet dans VS Code :**

- Entrez dans le dossier du projet : cd projet-ia-reseau
- Lancez VS Code dans ce dossier : code .

4. **Créer et Configurer l'Environnement Virtuel Python :**

- Dans le terminal de VS Code (Ctrl+Maj+ù), créez l'environnement :

Bash

```
python3 -m venv .venv
```

- Activez-le (vous devrez faire cela à chaque fois que vous travaillez sur le projet) :

Bash

```
source .venv/bin/activate
```

- Votre invite de commande doit maintenant être préfixée par (.venv).

## 5. Installer les bibliothèques Python nécessaires :

Bash

```
pip install pandas
```

## 6. Configurer l'Interpréteur Python dans VS Code :

- Ouvrez la palette de commandes : Ctrl+Shift+P.
- Tapez et sélectionnez Python: Select Interpreter.
- Choisissez l'option qui contient .venv dans son chemin.

# Partie 3 : Installation et Configuration de GNS3

C'est le cœur de notre laboratoire de simulation.

## 1. Ajouter le dépôt officiel de GNS3 :

Bash

```
sudo add-apt-repository ppa:gns3/ppa
```

Appuyez sur Entrée pour confirmer.

## 2. Installer GNS3 (GUI et Serveur) :

Bash

```
sudo apt update  
sudo apt install -y gns3-gui gns3-server
```

## 3. Configurer les Permissions durant l'installation :

- Une fenêtre apparaîtra dans le terminal vous demandant si les utilisateurs non-root peuvent lancer GNS3. Choisissez <Yes>.
- Une autre fenêtre demandera s'ils peuvent capturer des paquets. Choisissez <Yes>.

## 4. Ajouter votre utilisateur aux groupes requis : Ceci est crucial pour que GNS3 fonctionne correctement. Remplacez votre-username par votre propre nom d'utilisateur Ubuntu.

Bash

```
sudo usermod -aG ubridge,libvirt,kvm,wireshark votre-username
```

## 5. Redémarrer l'ordinateur : Un redémarrage complet est le moyen le plus sûr pour que toutes les nouvelles permissions de groupe soient appliquées.

# Partie 4 : Configuration du Laboratoire Virtuel GNS3

Nous allons maintenant importer l'image du commutateur et configurer la topologie.

1. **Obtenir l'image du commutateur :** Vous devez télécharger l'image d'un commutateur Cisco IOSvL2. Le fichier requis est au format .vmdk ou .qcow2. (Note pour Nathanael : vous devrez partager ce fichier avec votre ami).
2. **Lancer GNS3 et Importer l'image :**
  - Lancez GNS3 depuis le menu de vos applications.

- Allez dans **Edit > Preferences > Qemu VMs**.
- Cliquez sur **New**.
- **Name : Cisco IOSvL2 Switch**
- **RAM : 1024 MB**
- **Console type : telnet**
- **Disk image :** Naviguez jusqu'à votre fichier image (.vmdk ou .qcow2). Acceptez de le copier dans le répertoire par défaut.
- Cliquez sur **Finish**.

### 3. Ajuster le modèle de commutateur :

- Sélectionnez **Cisco IOSvL2 Switch** dans la liste et cliquez sur **Edit**.
- Onglet **General settings -> Category -> Switches**.
- Onglet **Network -> Adapters -> 16**.
- Cliquez sur **OK** pour fermer les fenêtres.

### 4. Créer la Topologie de Base :

- Dans GNS3, créez un nouveau projet (**File > New blank project**).
- Glissez-déposez **deux Cisco IOSvL2 Switch** (S1, S2).
- Glissez-déposez un nœud **NAT** depuis la section des "End devices".
- **Câblage :**
  - Connectez S1 (**GigabitEthernet0/1**) à S2 (**GigabitEthernet0/1**).
  - Connectez S2 (un port libre, ex: **GigabitEthernet0/2**) au nœud NAT.

## Partie 5 : Configuration Finale (Réseau et Collecte de Logs)

La dernière étape consiste à faire communiquer les appareils et à centraliser les logs.

### 1. Configurer le service **rsyslog** sur votre machine Ubuntu :

- Créez et éditez un fichier de configuration :
 

Bash

```
sudo nano /etc/rsyslog.d/20-gns3.conf
```
- Collez-y le contenu exact suivant :
 

Plain Text

```
:fromhost-ip, !isequal, "127.0.0.1"  /var/log/gns3/network-devices.log
& stop
```
- Créez le dossier de log et donnez-lui les bonnes permissions :
 

Bash

```
sudo mkdir -p /var/log/gns3
sudo chown syslog:adm /var/log/gns3
```
- Redémarrez le service :
 

Bash

```
sudo systemctl restart rsyslog
```

### 2. Identifier l'adresse IP de la passerelle GNS3 :

- Dans un terminal Ubuntu, trouvez l'IP de l'interface `virbr0` :

Bash

```
ip addr show virbr0
```

- Notez l'adresse IP (généralement `192.168.122.1`).

### 3. Configurer les Commutateurs Virtuels :

- Démarrez la topologie dans GNS3 (bouton "Play").
- Ouvrez la console de S1 et S2.
- Sur **chaque** commutateur, entrez les commandes suivantes (en remplaçant `192.168.122.1` par l'IP trouvée ci-dessus si elle est différente) :

Bash

```
conf t
```

```
! Donne une IP au switch pour communiquer
interface Vlan1
ip address 192.168.122.10 255.255.255.0 ! (Utilisez .11 pour S2)
no shutdown
exit

! Configure la destination des logs
logging host 192.168.122.1

! Active l'horodatage détaillé
service timestamps debug datetime msec
service timestamps log datetime msec

end
write memory ! Sauvegarde la configuration
```

### 4. Vérification Finale :

- Sur votre machine Ubuntu, surveillez le fichier de log :

Bash

```
tail -f /var/log/gns3/network-devices.log
```

- Sur la console de S1, entrez `conf t` puis `exit`. Une ligne de log doit apparaître dans le terminal de surveillance.

**Vous êtes maintenant prêt !** Votre environnement est identique à celui de votre coéquipier. Vous pouvez exécuter le script `log_analyzer.py` du projet pour commencer l'analyse.

Démarrer agentCréer

Dans quelle mesure Manus a-t-il répondu à votre question ?