

TP2 : Analyse TCPDUMP + Docker

Partie 1 TCPDUMP

- 1) Installer tcpdump (sudo apt install tcpdump)
- 2) Effectuer une capture de trafic sur toutes les interfaces (sudo tcpdump)
- 3) Spécifier une interface pour la capture de trafic
- 4) Spécifier un protocole
- 5) Enregistrer la capture dans un fichier .pcap
- 6) Afficher le contenu du fichier .pcap
- 7) Analyser cette capture en utilisant Wireshark

Partie 2 : Docker

Objectifs :

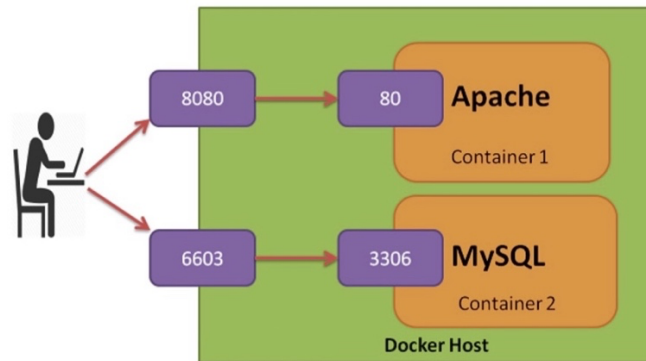
- La création des VM/Container Docker
- Création de réseaux virtuels
- Accès à distance aux containers : Tunneling, Mapping de ports
- Exposition des applications containerisées.

1) Installer Docker

- <https://docs.docker.com/engine/install/ubuntu/>
- Créer un container à partir d'une image Ubuntu.
- Vérifier que le container a une adresse IP valide et qu'il est connecté au bridge Docker0
- Faire un ping vers 8.8.8.8

- 2)** La conteneurisation avec Docker est devenue très populaire. Elle a permis d'héberger plusieurs applications de façon indépendante. Par défaut, les conteneurs Docker utilisent le réseau interne comme déjà vu dans la partie 1. Chaque conteneur Docker possède sa propre adresse IP accessible depuis la machine hôte.

Comme vous avez déjà vu dans la partie 1 les adresses IP internes des containers ne peuvent pas être utilisées pour accéder aux conteneurs à partir de l'extérieur contrairement à l'adresse IP de la machine hôte qui est visible par les machines du même LAN (et même de l'extérieur). L'objectif de cette partie est de rendre votre application hébergée sur le container accessible depuis l'extérieur. Il s'agit d'autoriser les utilisateurs externes d'accéder à votre application depuis l'Internet.



- 1) Créer un container Docker
- 2) Héberger une application dans ce container
- 3) Vérifier l'accès à l'application à partir de votre PC
- 4) Configurer et vérifier l'accès au container à partir d'un autre PC du même LAN.

Quelques commandes :

```
docker run -it -p 8080:80
```

```
docker run -it -p 6603:3066
```

3) Création d'un conteneur avec Dockerfile

Créer une application web conteneurisée à l'aide d'un **Dockerfile**.

- 1) Créez un fichier Dockerfile pour héberger une application web simple (par exemple, un serveur HTTP avec nginx ou un serveur Python Flask).
- 2) Utilisez le **mapping de ports** pour rendre l'application accessible depuis l'hôte (et le LAN).
- 3) Lancez le conteneur en utilisant l'image générée à partir du Dockerfile.

Exemple de Dockerfile :

```
FROM python:3.9-slim
WORKDIR /app
COPY . /app
RUN pip install -r requirements.txt
EXPOSE 5000
CMD ["python", "app.py"]
```

Questions :

1. Quel est le rôle de la commande EXPOSE dans un Dockerfile ?
2. Pourquoi utilise-t-on CMD à la fin du Dockerfile ?
3. Lancez votre conteneur avec le mapping de ports pour exposer l'application web :

```
docker build -t mon_app .
```

```
docker run -d -p 5000:5000 mon_app
```

4. Vérifiez que l'application est accessible depuis votre navigateur.