Projet Bulles docker 2



**Rapport de projet**

Zeqiri Amir – CIN2B

ETML, Sébeillon – N510B

40 Périodes

Maître : M. Maître

Table des matières

[1 Description du projet dans son ensemble 3](#_Toc192587348)

[1.1 Titre 3](#_Toc192587349)

[1.2 Sujet 3](#_Toc192587350)

[2 Analyse & Conception 3](#_Toc192587351)

[3 Réalisation 3](#_Toc192587352)

[3.1 Dockerfile 3](#_Toc192587353)

[3.2 Dockerfile 2 4](#_Toc192587354)

[3.3 Docker Compose 5](#_Toc192587355)

[4 Tests 5](#_Toc192587356)

[5 Problèmes rencontrés 6](#_Toc192587357)

[5.1 Création d’une image avec dockerfile 6](#_Toc192587358)

[5.2 Erreur de build avec le docker-compose (Netcat) 6](#_Toc192587359)

[6 Sources & Aides 6](#_Toc192587360)

[7 Conclusion 6](#_Toc192587361)

# Description du projet dans son ensemble

## Titre

P\_Bulles DockerNetworkToolkit

## Sujet

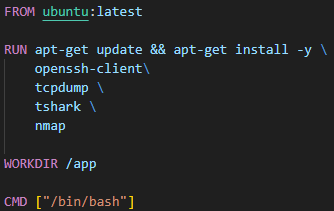
Bienvenue dans le projet "P\_Bulle DockerNetworkToolkit". Ce projet vise à vous initier aux concepts de la virtualisation par conteneur, en vous permettant de créer, gérer et orchestrer des conteneurs Docker. À la fin de ce projet, vous serez en mesure de comprendre les principes de base de Docker, de créer des conteneurs individuels et de les orchestrer en utilisant Docker Compose

# Analyse & Conception

Dans ce projet je vais avoir besoin de deux container Docker, le premier servira de boîte à outils comportant Tshark, tcpdump, Nmap et SSH comme outils. Le deuxième et dernier container servira tester ces outils-là. Tous les tests seront détaillés dans le [point 4](#_Tests) du rapport

# Réalisation

## Dockerfile



**1. Base de l'image**

dockerfile

CopierModifier

FROM ubuntu:latest

* On utilise l’image officielle **Ubuntu** en dernière version comme base.

**2. Installation des paquets**

dockerfile

CopierModifier

RUN apt-get update && apt-get install -y \

openssh-client\

tcpdump \

tshark \

nmap

* apt-get update : Met à jour la liste des paquets disponibles.
* apt-get install -y : Installe les outils suivants :
  + **openssh-client** : Permet d'utiliser SSH pour se connecter à d'autres machines.
  + **tcpdump** : Outil pour capturer et analyser le trafic réseau.
  + **tshark** : Version en ligne de commande de Wireshark (analyseur de paquets réseau).
  + **nmap** : Scanner de ports pour analyser les réseaux.

**3. Définition du répertoire de travail**

dockerfile

CopierModifier

WORKDIR /app

* Définit **/app** comme répertoire de travail par défaut à l’intérieur du conteneur.

**4. Commande de démarrage**

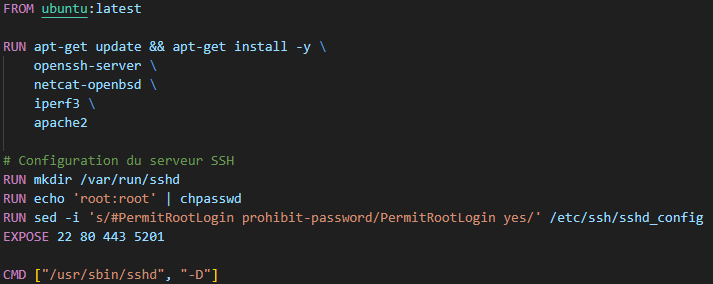
dockerfile

CopierModifier

CMD ["/bin/bash"]

* Lorsque le conteneur démarre, il ouvre un shell interactif **bash**.

## Dockerfile 2



**1. Base de l'image**

dockerfile

CopierModifier

FROM ubuntu:latest

* Comme pour le premier Dockerfile, on utilise **Ubuntu:latest** comme image de base.

**2. Installation des paquets**

dockerfile

CopierModifier

RUN apt-get update && apt-get install -y \

openssh-server \

netcat-openbsd \

iperf3 \

apache2

* **openssh-server** : Permet d'accéder au conteneur via SSH.
* **netcat-openbsd** : Outil de test réseau pour écouter et envoyer des connexions.
* **iperf3** : Utilisé pour tester la bande passante et la performance réseau.
* **apache2** : Serveur web pour tester la connectivité HTTP.

**3. Configuration du serveur SSH**

dockerfile

CopierModifier

RUN mkdir /var/run/sshd

RUN echo 'root:root' | chpasswd

RUN sed -i 's/#PermitRootLogin prohibit-password/PermitRootLogin yes/' /etc/ssh/sshd\_config

* **mkdir /var/run/sshd** : Crée le répertoire nécessaire pour exécuter le serveur SSH.
* **echo 'root:root' | chpasswd** : Définit le mot de passe root en **"root"**.
* **sed -i 's/#PermitRootLogin prohibit-password/PermitRootLogin yes/' /etc/ssh/sshd\_config** : Modifie la config pour autoriser la connexion en root via SSH.

**4. Exposition des ports**

dockerfile

CopierModifier

EXPOSE 22 80 443 5201

* **22** → SSH
* **80** → HTTP (Apache2)
* **443** → HTTPS
* **5201** → Iperf3 (test de bande passante)

**5. Commande de démarrage**

dockerfile

CopierModifier

CMD ["/usr/sbin/sshd", "-D"]

* Lance le serveur **SSH** en mode daemon pour rester actif.

## Docker Compose

Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

* **Définition des services**

yaml

CopierModifier

services:

* On définit les **services** qui seront déployés sous Docker.
* **Service 1 : toolkit**

yaml

CopierModifier

toolkit:

build:

context: .

dockerfile: toolkit.dockerfile

container\_name: toolkit

networks:

- toolkit\_network

tty: true # Permet d'avoir un terminal interactif

depends\_on:

- toolkit2

* **build:** : Spécifie la construction du conteneur avec toolkit.dockerfile.
* **container\_name:** : Définit le nom du conteneur (toolkit).
* **networks:** : Connecte ce conteneur au réseau toolkit\_network.
* **tty: true** : Permet d’avoir un terminal interactif (/bin/bash).
* **depends\_on:** : Indique que toolkit **dépend de toolkit2** (il ne démarrera qu’après que toolkit2 soit lancé).
* **Service 2 : toolkit2**

yaml

CopierModifier

toolkit2:

build:

context: .

dockerfile: toolkit2.dockerfile

container\_name: toolkit2

networks:

- toolkit\_network

ports:

- "22:22" # SSH

- "80:80" # HTTP

- "443:443" # HTTPS

- "5201:5201" # iperf3

tty: true

* **build:** : Spécifie la construction avec toolkit2.dockerfile.
* **container\_name:** : Définit le nom du conteneur (toolkit2).
* **networks:** : Connecte ce conteneur au réseau toolkit\_network.
* **ports:** : Redirige les ports du conteneur vers la machine hôte :
  + 22:22 → SSH
  + 80:80 → HTTP
  + 443:443 → HTTPS
  + 5201:5201 → Iperf3
* **tty: true** : Active un terminal interactif.
* **Définition du réseau**

yaml

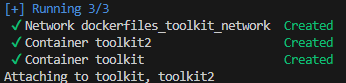
CopierModifier

networks:

toolkit\_network:

driver: bridge

* **toolkit\_network:** : Définit un **réseau Docker** pour connecter les deux conteneurs.
* **driver: bridge** : Utilise un **réseau en bridge**, permettant aux conteneurs de communiquer entre eux.

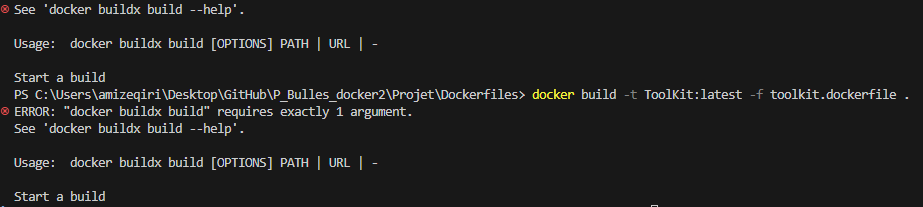


# Tests

Tests

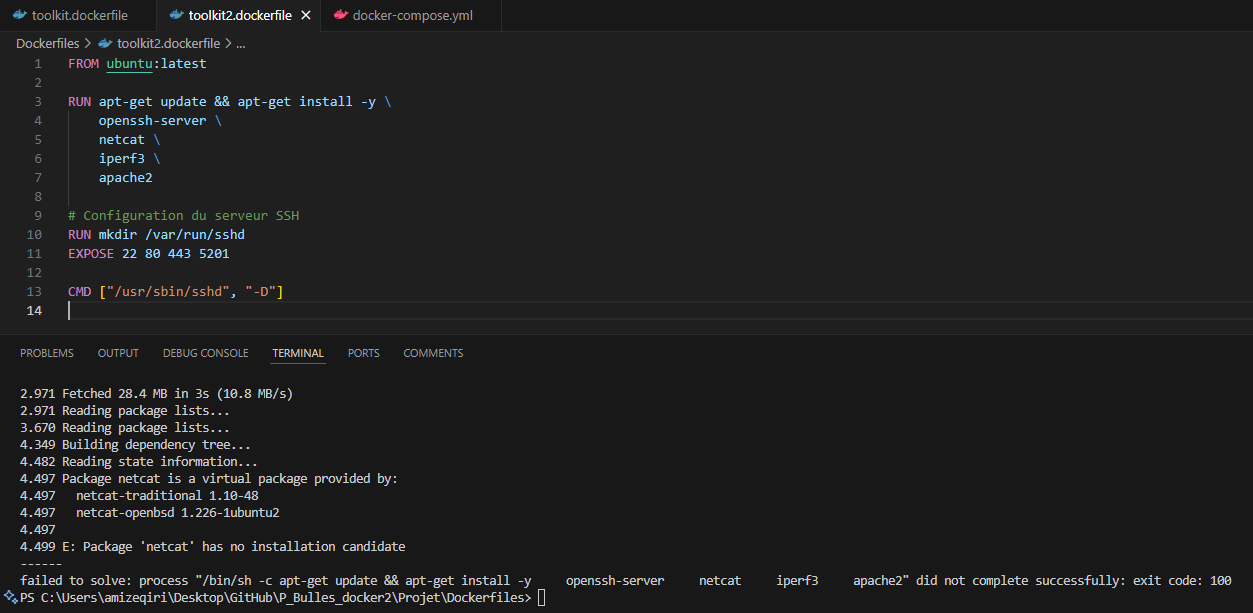
# Problèmes rencontrés

## Création d’une image avec dockerfile



Je n’arrivais pas à créer une image tirée de mon fichier dockerfile pour le premier container,

## Erreur de build avec le docker-compose (Netcat)



# Sources & Aides

[Lien](file:///C:\Users\amizeqiri\Desktop\GitHub\P_Bulles_docker2\Lien) : Lien

# Conclusion

Conclusion