docker-debian13

```
MARK RODERICK
```

🏗️ C'est quoi Docker ?

Docker est une plateforme qui permet de **"conteneuriser"** des applications. Un conteneur est un paquet léger, autonome et exécutable qui contient tout ce dont une application a besoin pour fonctionner :

- Le code.
- Le système d'exploitation (une version très légère).
- Les bibliothèques (libraries).
- Les variables d'environnement.

Le but ultime : "It works on my machine"

Le but principal de Docker est de résoudre le célèbre problème : *"Mais ça marchait sur mon ordi !"*. En enfermant l'application dans un conteneur, vous garantissez qu'elle fonctionnera de la même manière sur votre PC, sur le serveur de test ou dans le Cloud.

⚖️ Docker vs Machines Virtuelles (VM)

C'est la question classique. Pourquoi ne pas simplement utiliser une VM (comme VirtualBox ou VMware) ?

Caractéristique	Machine Virtuelle (VM)	Conteneur Docker
Poids	Lourd (plusieurs Go par VM)	Très léger (quelques Mo)
Démarrage	Lent (plusieurs minutes)	Instantané (quelques secondes)
Architecture	Embarque un OS complet (noyau inclus)	Partage le noyau de l'hôte
Performance	Perte de performance (couche d'émulation)	Performances quasi-natives

---

🛠️ Les 3 concepts clés à retenir

Pour parler Docker, il faut connaître ces trois termes :

1. **Le Dockerfile :** C'est la "recette de cuisine". Un fichier texte simple qui liste les instructions pour construire votre environnement (ex: "Installe Linux, installe Python, copie mon script").
2. **L'Image :** C'est le "plat congelé". C'est le résultat de la compilation du Dockerfile. Une image est figée, on ne peut pas la modifier une fois créée.
3. **Le Conteneur :** C'est le "plat réchauffé et prêt à manger". C'est l'instance vivante de votre image. Vous pouvez lancer 10 conteneurs identiques à partir d'une seule image.

🚀 Pourquoi tout le monde l'utilise ?

- **Isolation :** Vous pouvez faire tourner deux versions de PHP (ou de n'importe quel outil) sur le même serveur sans qu'elles ne se battent entre elles.
- **Scalabilité :** Si votre site web reçoit trop de monde, vous pouvez lancer 50 conteneurs identiques en quelques secondes pour absorber le trafic.
- **Nettoyage facile :** Marre d'un projet ? Supprimez le conteneur, et votre ordinateur est comme neuf. Aucune trace de fichiers d'installation ne traîne dans votre système.

> **En résumé :** Docker transforme l'informatique en Lego. On assemble des briques standardisées, isolées et ultra-rapides à déployer.

Installation docker

Étape 1 — Préparer la VM Debian

Mettre à jour le système

apt update && apt upgrade -y

```
sio@debian13:~$ su -
Mot de passe :
root@debian13:~# apt update && apt upgrade -y
Ign : 1 http://deb.debian.org/debian trixie InRelease
Ign : 2 http://security.debian.org/debian-security trixie-security InRelease
Ign : 2 http://security.debian.org/debian-security trixie-security InRelease
Ign : 1 http://deb.debian.org/debian trixie InRelease
Ign : 1 http://deb.debian.org/debian trixie InRelease
Ign : 2 http://security.debian.org/debian-security trixie-security InRelease
Err : 2 http://security.debian.org/debian-security trixie-security InRelease
  Erreur temporaire de résolution de « security.debian.org »
Err : 1 http://deb.debian.org/debian trixie InRelease
  Erreur temporaire de résolution de « deb.debian.org »
Tous les paquets sont à jour.
Attention : Impossible de récupérer http://deb.debian.org/debian/dists/trixie/InRelease  Erreur temp
oraire de résolution de « deb.debian.org »
Attention : Impossible de récupérer http://security.debian.org/debian-security/dists/trixie-security
/InRelease  Erreur temporaire de résolution de « security.debian.org »
Attention : Le téléchargement de certains fichiers d'index a échoué, ils ont été ignorés, ou les anc
iens ont été utilisés à la place.
Sommaire :
  Mise à niveau de : 0. Installation de : 0Supprimé : 0. Non mis à jour : 0
root@debian13:~#
```

Installer les outils de base

apt install -y curl git openssh-server

```
root@debian13:~# apt install -y curl git openssh-server
openssh-server est déjà la version la plus récente (1:10.0p1-7).
openssh-server passé en « installé manuellement ».
Installation de :
  curl  git

Installation de dépendances :
  git-man  liberror-perl  patch

Paquets suggérés :
  git-doc    git-gui  gitweb   git-mediawiki  ed
  git-email  gitk     git-cvs  git-svn        diffutils-doc

Sommaire :
  Mise à niveau de : 0. Installation de : 5Supprimé : 0. Non mis à jour : 120
Taille du téléchargement : 11,5 MB
  Espace nécessaire : 53,5 MB / 1 204 MB disponible

Réception de : 1 http://deb.debian.org/debian trixie/main amd64 curl amd64 8.14.
1-2+deb13u2 [270 kB]
Réception de : 2 http://deb.debian.org/debian trixie/main amd64 liberror-perl al
l 0.17030-1 [26,9 kB]
```

```
Sélection du paquet curl précédemment désélectionné.
(lecture de la base de données... 119051 fichiers et répertoires déjà installés.
)
Préparation du dépaquetage de .../curl_8.14.1-2+deb13u2_amd64.deb ...
Dépaquetage de curl (8.14.1-2+deb13u2) ...
Sélection du paquet liberror-perl précédemment désélectionné.
Préparation du dépaquetage de .../liberror-perl_0.17030-1_all.deb ...
Dépaquetage de liberror-perl (0.17030-1) ...
Sélection du paquet git-man précédemment désélectionné.
Préparation du dépaquetage de .../git-man_1%3a2.47.3-0+deb13u1_all.deb ...
Dépaquetage de git-man (1:2.47.3-0+deb13u1) ...
Sélection du paquet git précédemment désélectionné.
Préparation du dépaquetage de .../git_1%3a2.47.3-0+deb13u1_amd64.deb ...
Dépaquetage de git (1:2.47.3-0+deb13u1) ...
Sélection du paquet patch précédemment désélectionné.
Préparation du dépaquetage de .../archives/patch_2.8-2_amd64.deb ...
Dépaquetage de patch (2.8-2) ...
Paramétrage de liberror-perl (0.17030-1) ...
Paramétrage de patch (2.8-2) ...
Paramétrage de git-man (1:2.47.3-0+deb13u1) ...
Paramétrage de curl (8.14.1-2+deb13u2) ...
Paramétrage de git (1:2.47.3-0+deb13u1) ...
Traitement des actions différées (« triggers ») pour man-db (2.13.1-1) ...
root@debian13:~#
```

Vérifier l'adresse IP de la VM

ip addr show eth0

```
root@debian13:~# ip addr show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group defa
ult qlen 1000
    link/ether 00:15:5d:03:36:94 brd ff:ff:ff:ff:ff:ff
    altname enx00155d033694
    inet 10.30.5.27/16 brd 10.30.255.255 scope global dynamic noprefixroute eth0
       valid_lft 690915sec preferred_lft 690915sec
    inet6 fe80::ac2d:d619:4f47:19b2/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
root@debian13:~#
```

Étape 2 — Installer Docker

Installation automatique via le script officiel

curl -fsSL https://get.docker.com | sh

```
root@debian13:~# curl -fsSL https://get.docker.com | sh
# Executing docker install script, commit: f381ee68b32e515bb4dc034b339266aff1fbc
460
+ sh -c apt-get -qq update >/dev/null
+ sh -c DEBIAN_FRONTEND=noninteractive apt-get -y -qq install ca-certificates cu
rl >/dev/null
+ sh -c install -m 0755 -d /etc/apt/keyrings
+ sh -c curl -fsSL "https://download.docker.com/linux/debian/gpg" -o /etc/apt/ke
yrings/docker.asc
+ sh -c chmod a+r /etc/apt/keyrings/docker.asc
+ sh -c echo "deb [arch=amd64 signed-by=/etc/apt/keyrings/docker.asc] https://do
wnload.docker.com/linux/debian trixie stable" > /etc/apt/sources.list.d/docker.l
ist
+ sh -c apt-get -qq update >/dev/null
+ sh -c DEBIAN_FRONTEND=noninteractive apt-get -y -qq install docker-ce docker-c
e-cli containerd.io docker-compose-plugin docker-ce-rootless-extras docker-build
x-plugin docker-model-plugin >/dev/null
Using systemd to manage Docker service
+ sh -c systemctl enable --now docker.service
  UNIT                         LOAD   ACTIVE SUB       DESCRIPTION
  proc-sys-fs-binfmt_misc.a… loaded active running   Arbitrary Executable File …
```

```
 docker-init:
  Version:          0.19.0
  GitCommit:        de40ad0

================================================================================
To run Docker as a non-privileged user, consider setting up the
Docker daemon in rootless mode for your user:

    dockerd-rootless-setuptool.sh install

Visit https://docs.docker.com/go/rootless/ to learn about rootless mode.


To run the Docker daemon as a fully privileged service, but granting non-root
users access, refer to https://docs.docker.com/go/daemon-access/

WARNING: Access to the remote API on a privileged Docker daemon is equivalent
         to root access on the host. Refer to the 'Docker daemon attack surface'
         documentation for details: https://docs.docker.com/go/attack-surface/

================================================================================

root@debian13:~#
```

Démarrer et activer Docker

- systemctl start docker
- systemctl enable docker

```
root@debian13:~# systemctl start docker
root@debian13:~# systemctl enable docker
Synchronizing state of docker.service with SysV service script with /usr/lib/sys
temd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable docker
root@debian13:~#
```

Vérifier l'installation

- docker --version
- docker ps

```
root@debian13:~# docker --version
Docker version 29.3.0, build 5927d80
root@debian13:~# docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
root@debian13:~#
```

Étape 3 — Installer Docker Compose

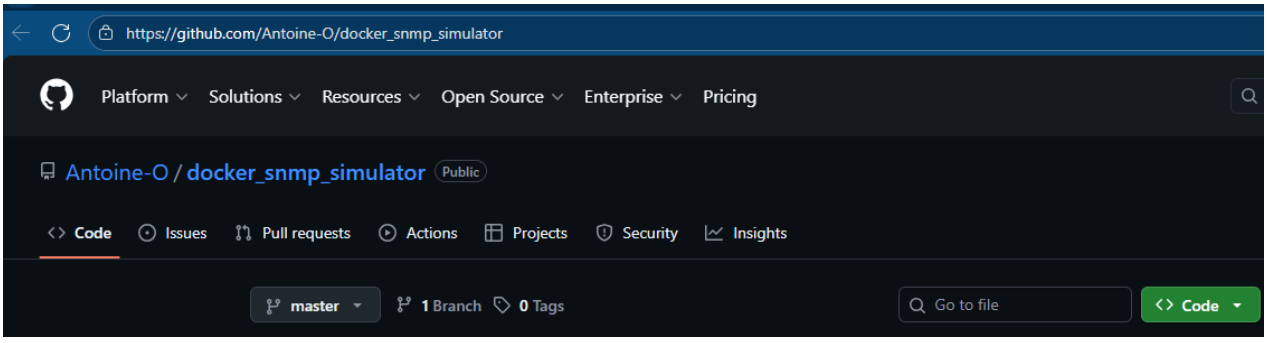
- apt install -y docker-compose-plugin
- docker compose version

```
root@debian13:~# apt install -y docker-compose-plugin
docker-compose-plugin est déjà la version la plus récente (5.1.0-1~debian.13~tri
xie).
Sommaire :
  Mise à niveau de : 0. Installation de : 0Supprimé : 0. Non mis à jour : 120
root@debian13:~# docker compose version
Docker Compose version v5.1.0
root@debian13:~#
```

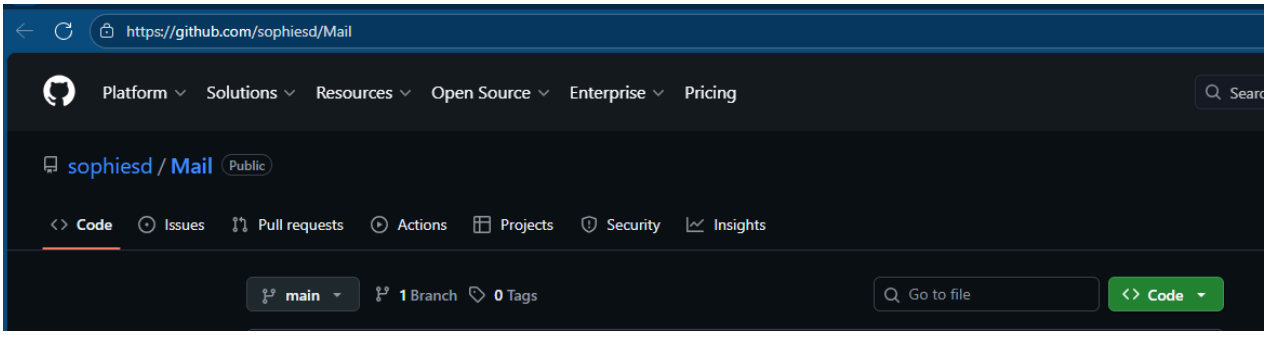
Étape 4 — Récupérer un projet

les projets à prend pour ce projet

- git clone https://github.com/Antoine-O/docker_snmp_simulator.git
  -



- git clone https://github.com/sophiesd/Mail.git
  -



```
root@debian13:~# git clone https://github.com/Antoine-O/docker_snmp_simulator.gi
t
Clonage dans 'docker_snmp_simulator'...
remote: Enumerating objects: 45, done.
remote: Counting objects: 100% (45/45), done.
remote: Compressing objects: 100% (39/39), done.
remote: Total 45 (delta 11), reused 32 (delta 4), pack-reused 0 (from 0)
Réception d'objets: 100% (45/45), 22.68 Kio | 1.13 Mio/s, fait.
Résolution des deltas: 100% (11/11), fait.
root@debian13:~# git clone https://github.com/sophiesd/Mail.git
Clonage dans 'Mail'...
remote: Enumerating objects: 89, done.
remote: Counting objects: 100% (89/89), done.
remote: Compressing objects: 100% (73/73), done.
remote: Total 89 (delta 13), reused 72 (delta 5), pack-reused 0 (from 0)
Réception d'objets: 100% (89/89), 37.27 Kio | 1.96 Mio/s, fait.
Résolution des deltas: 100% (13/13), fait.
root@debian13:~#
```

1. cd /home/docker_snmp_simulator

mv /root/docker_snmp_simulator /home/

placer le dossier dans home

```
root@debian13:~# mv /root/docker_snmp_simulator /home/
root@debian13:~#
```

2. nano docker-compose.yml

> fichier de configuration

dOCKER COMPOSE

Commandes utiles

🐳 Docker / Docker Compose Commands

📌 Voir l’état des conteneurs

```
docker compose ps
```

📌 Voir les logs d’un conteneur

```
docker compose logs network_switch_1
```

📌 Suivre les logs en temps réel

```
docker compose logs -f network_switch_1
```

📌 Sauvegarder les logs dans un fichier

```
docker compose logs > /home/logs_all.txt
```

📌 Arrêter les conteneurs (conserve la configuration)

```
docker compose stop
```

📌 Détruire les conteneurs (utile si on modifie le YAML)

```
docker compose down
```

📌 Reconstruire et relancer après modification

```
docker compose up -d --build
```

📌 Voir les statistiques CPU/RAM des conteneurs

```
docker stats
```

📌 Voir l’état d’un service spécifique

```
docker compose ps dns
```

📌 Utiliser un fichier compose spécifique

```
docker compose -f docker-compose-dkim.yml ps
```

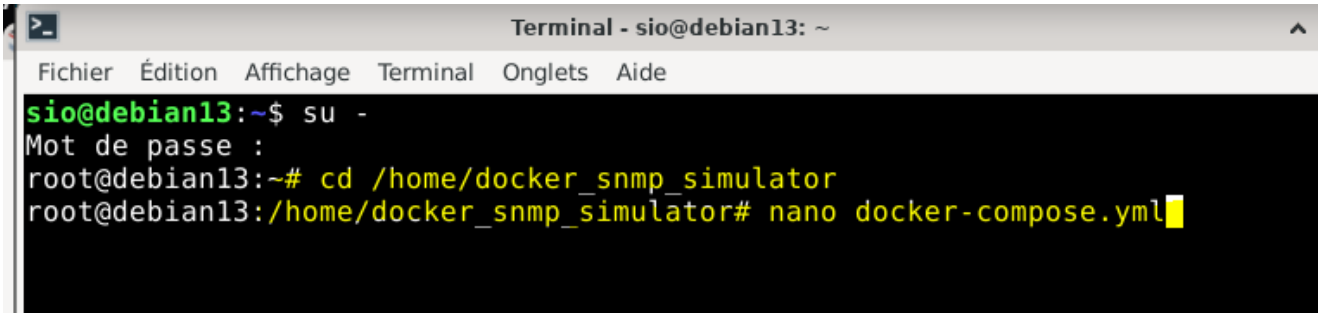
📌 Voir les derniers logs d’un conteneur Docker classique

```
docker logs mail_postfix --tail 10
```

Paramétrage réseau

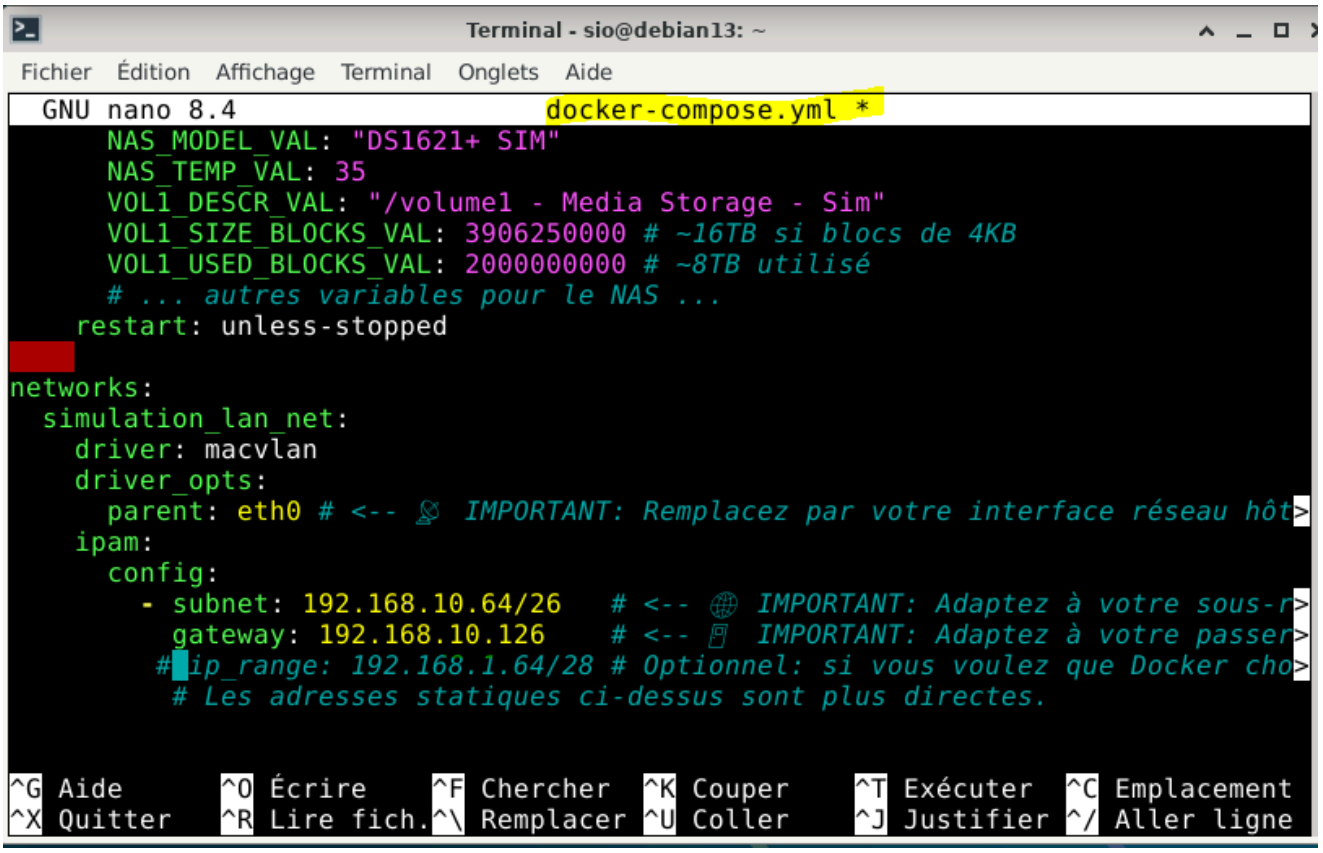
Nom vlan	Numero vlan	IP Vlan	Masque du vlans	1ere adresse disponible	Derniere adresse disponible	ombr d'hot
Vlan client 1	111	192.168.10.0	/26	192.168.10.1	192.168.10.62	60
Vlan client 2	112	192.168.10.64	/26	192.168.10.65	192.168.10.126	52
Srv prod	114	192.168.10.160	/28	192.168.10.161	192.168.10.174	14
srv it	115	192.168.10.176	/29	192.168.10.177	192.168.10.182	6
vlan wifi	113	192.168.10.128	/27	192.168.10.129	192.168.10.158	30
vlan srv authentification	116	192.168.10.184	/29	192.168.10.185	192.168.10.190	6
vlan redondance	117	192.168.10.192	/30	192.168.10.193	192.168.10.194	2

pour y accéder le fichier yml docker compose

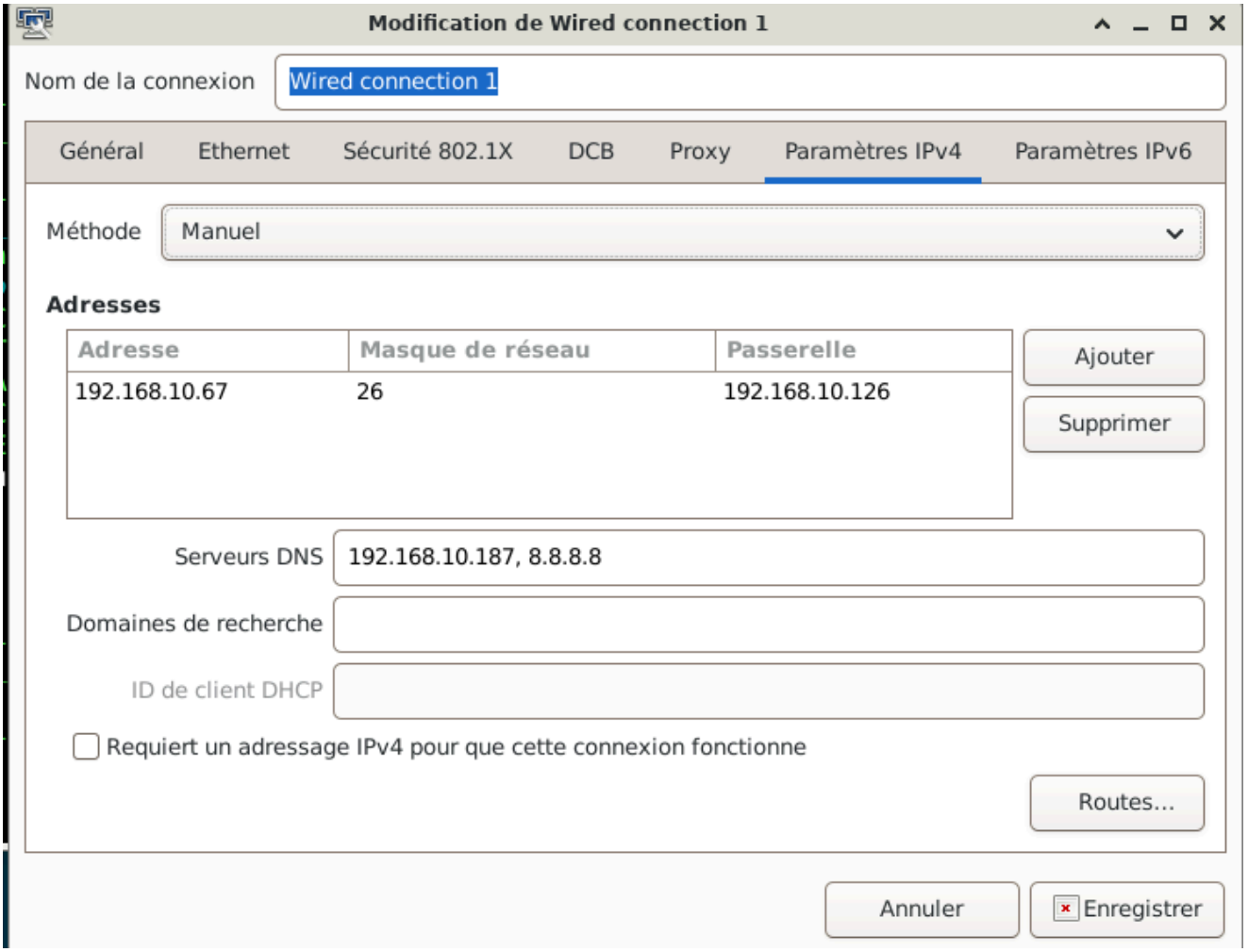


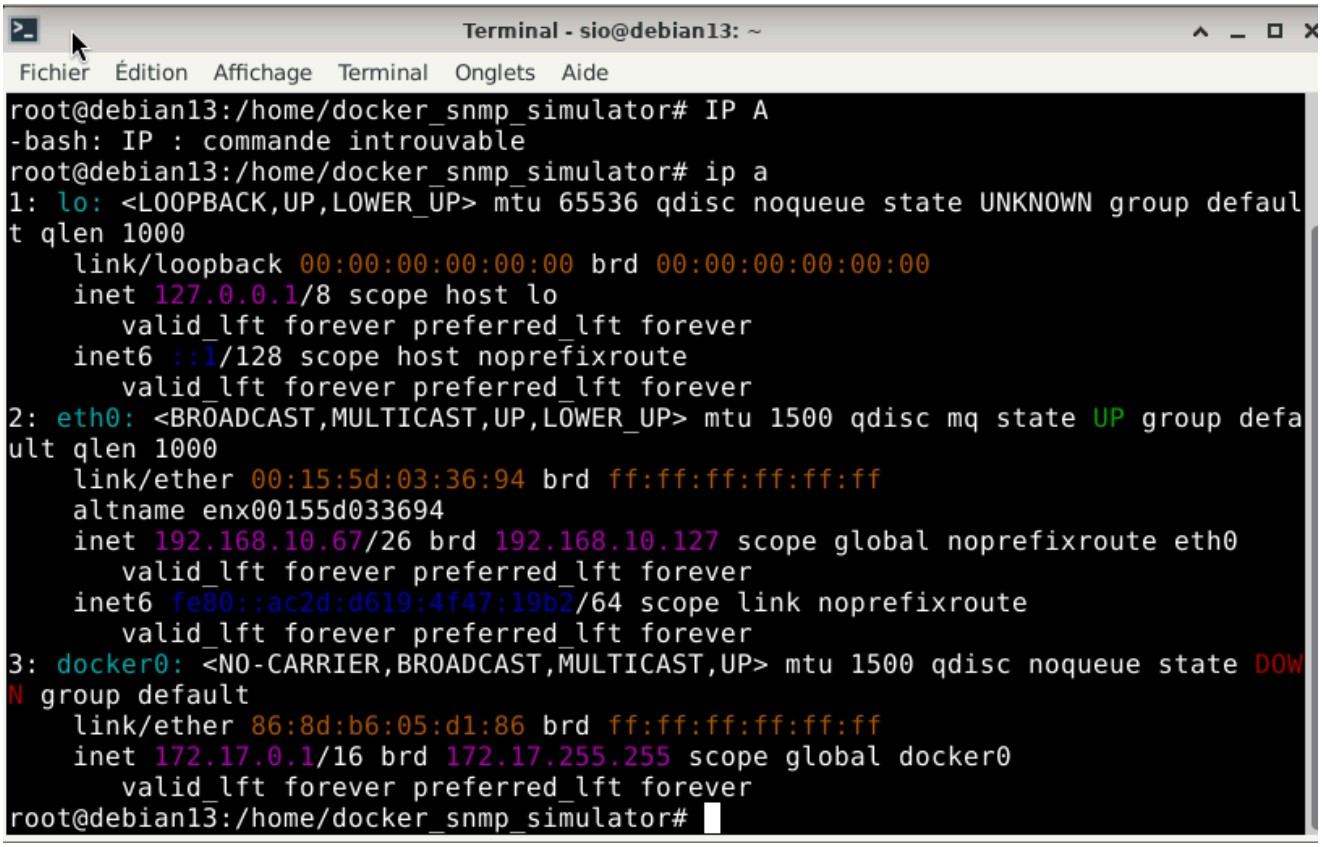
Changement du ip après une modfication du fichier yml

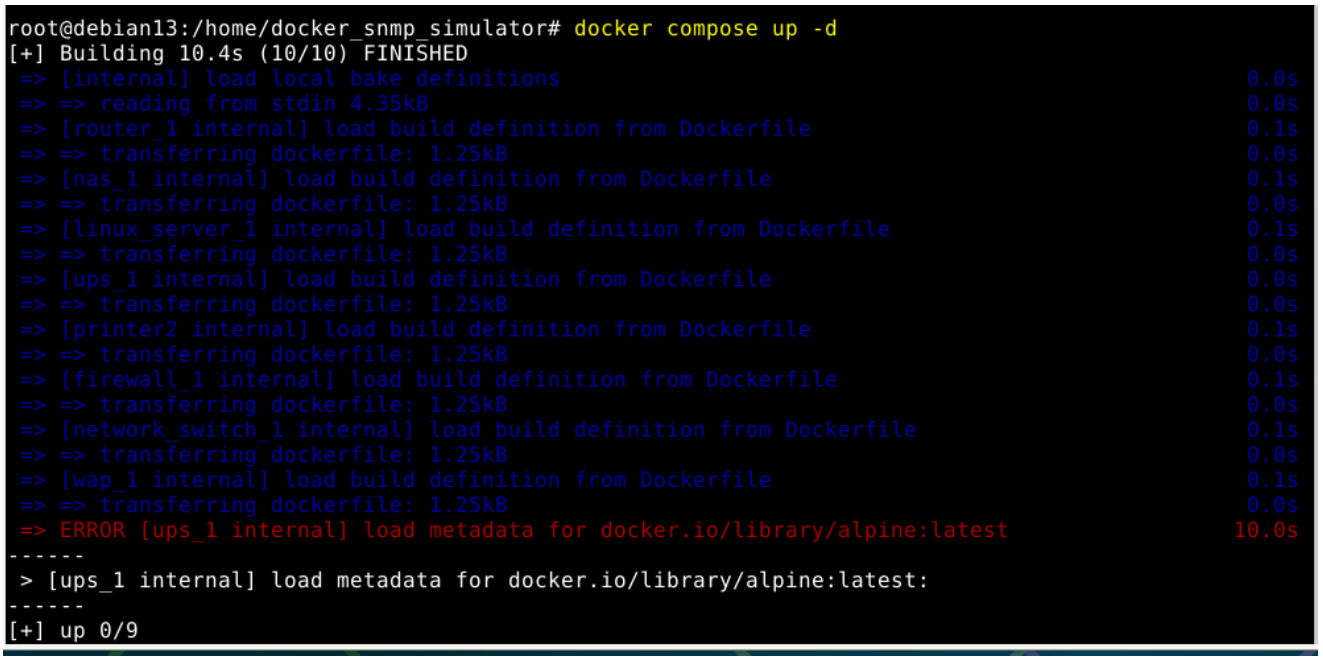
la machine est id dans vlan 112 > LAN
IP: 192.168.10.64/26
GATEWAY: 192.168.10.126



On modifie l'adress ip sur la machine




Terminal - sio@debian13: ~
Fichier Édition Affichage Terminal Onglets Aide
root@debian13:/home/docker_snmp_simulator# IP A
-bash: IP : commande introuvable
root@debian13:/home/docker_snmp_simulator# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t 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 noprefixroute
valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group defa
ult qlen 1000
link/ether 00:15:5d:03:36:94 brd ff:ff:ff:ff:ff:ff
altname enx00155d033694
inet 192.168.10.67/26 brd 192.168.10.127 scope global noprefixroute eth0
valid_lft forever preferred_lft forever
inet6 fe80::ac2d:d619:4f47:19b2/64 scope link noprefixroute
valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOW
N group default
link/ether 86:8d:b6:05:d1:86 brd ff:ff:ff:ff:ff:ff
inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
valid_lft forever preferred_lft forever
root@debian13:/home/docker_snmp_simulator#


root@debian13:/home/docker_snmp_simulator# docker compose up -d
[+] Building 10.4s (10/10) FINISHED
=> [internal] load local bake definitions 0.0s
=> => reading from stdin 4.35kB 0.0s
=> [router_1 internal] load build definition from Dockerfile 0.1s
=> => transferring dockerfile: 1.25kB 0.0s
=> [nas_1 internal] load build definition from Dockerfile 0.1s
=> => transferring dockerfile: 1.25kB 0.0s
=> [linux_server_1 internal] load build definition from Dockerfile 0.1s
=> => transferring dockerfile: 1.25kB 0.0s
=> [ups_1 internal] load build definition from Dockerfile 0.0s
=> => transferring dockerfile: 1.25kB 0.0s
=> [printer2 internal] load build definition from Dockerfile 0.1s
=> => transferring dockerfile: 1.25kB 0.0s
=> [firewall_1 internal] load build definition from Dockerfile 0.1s
=> => transferring dockerfile: 1.25kB 0.0s
=> [network_switch_1 internal] load build definition from Dockerfile 0.1s
=> => transferring dockerfile: 1.25kB 0.0s
=> [wap_1 internal] load build definition from Dockerfile 0.1s
=> => transferring dockerfile: 1.25kB 0.0s
=> ERROR [ups_1 internal] load metadata for docker.io/library/alpine:latest 10.0s
------
> [ups_1 internal] load metadata for docker.io/library/alpine:latest:

[+] up 0/9

```
 ⠿ Image docker_snmp_simulator-ups_1            Building                                    10.4s
 ⠿ Image docker_snmp_simulator-nas_1            Building                                    10.4s
 ⠿ Image docker_snmp_simulator-linux_server_1   Building                                    10.4s
 ⠿ Image docker_snmp_simulator-printer2         Building                                    10.4s
 ⠿ Image docker_snmp_simulator-wap_1            Building                                    10.4s
 ⠿ Image docker_snmp_simulator-network_switch_1 Building                                    10.4s
 ⠿ Image docker_snmp_simulator-printer1         Building                                    10.4s
 ⠿ Image docker_snmp_simulator-router_1         Building                                    10.4s
Dockerfile:2

--------------------

   1 |     # Utiliser une image de base Alpine Linux légère

   2 | >>> FROM alpine:latest

   3 |

   4 |     # Mettre à jour le cache des paquets et installer net-snmp et les outils associés

--------------------

target printer2: failed to solve: alpine:latest: failed to resolve source metadata for docker.io/li
brary/alpine:latest: failed to do request: Head "https://registry-1.docker.io/v2/library/alpine/man
ifests/latest": dial tcp: lookup registry-1.docker.io on 192.168.10.187:53: read udp 192.168.10.67:
47837->192.168.10.187:53: i/o timeout

root@debian13:/home/docker_snmp_simulator# nano docker-compose.yml
root@debian13:/home/docker_snmp_simulator#
```