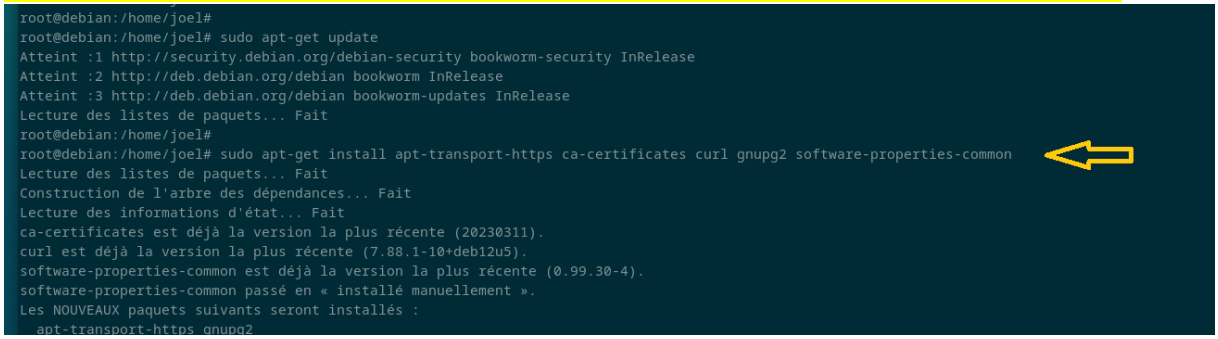
Documentation Serveur Base de données

1-L'Installation de Docker

D'abord on met à jour les paquets avec : sudo apt-get update

Puis on installe les paquets nécessaires pour utiliser le dépôt APT via HTTPS avec :

sudo apt-get install apt-transport-https ca-certificates curl software-properties-common

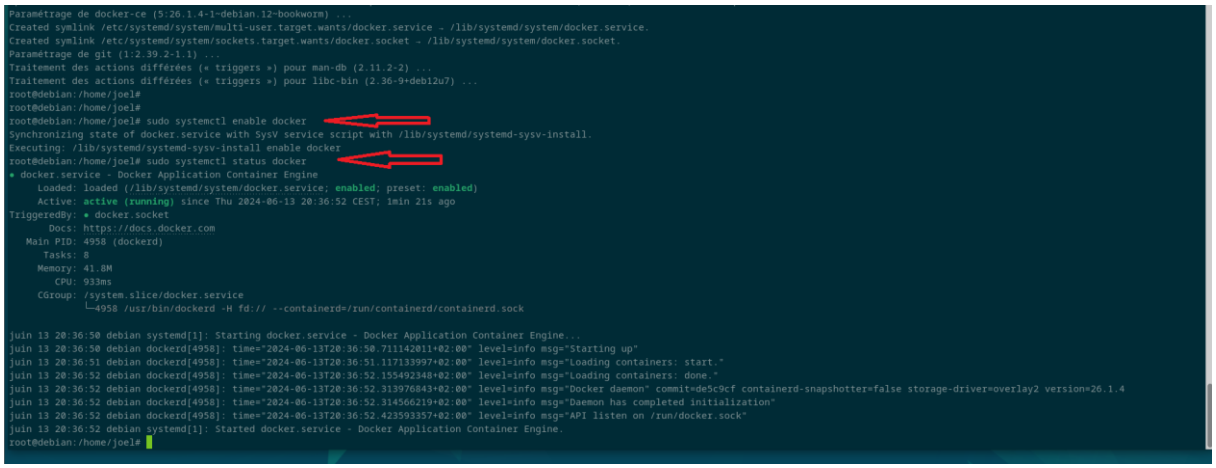


Enfin on installe Docker avec : sudo apt-get install docker-ce docker-ce-cli containerd.io

Pour verifier que Docker est bien présent sur notre serveur on va utiliser :

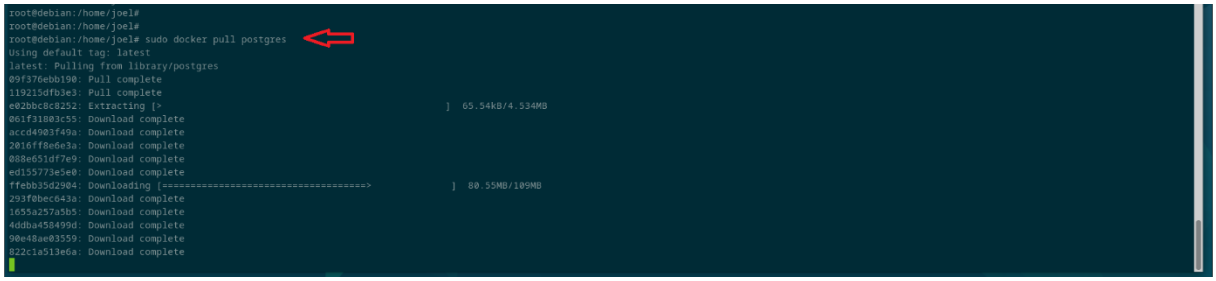
Docker run hello-world ça nous retourne Hello from Docker! Donc tout va bien.

On va programmer le démarrage automatique de Docker chaque fois le serveur se rallume :



2-Création et installation du conteneur PostgreSQL :

Avec sudo docker pull postgres on va télécharger l'image de postgreql dans docker



Puis on va créer et lancer un conteneur tout en créant un utilisateur en lui assignant un mot de passe :

Dans notre cas l'utilisateur est 'joel' et le mot de passe est 'parynov' et le conteneur est 'yplanningDB'

On va aussi programmer le démarrage automatique de notre conteneur :

3- Création de la base de données :

On se connecte à notre conteneur : sudo docker exec -it yplanningDB bash

Puis on se connecte à Docker : psql -U joel -d postgres

Une fois connecté à Docker on crée notre base de données :

```
joel=# Create DATABASE yplanning;
CREATE DATABASE
joel=#
joel=#
```

Sur notre bureau nous avons pris le soin de créer notre script de base de données nous allons copier le script dans notre conteneur :

```
root@debian:/home/joel# sudo docker cp Bureau/script.sql yplanningDB:/script.sql
Successfully copied 3.58kB to yplanningDB:/script.sql
root@debian:/home/joel#
```

Nous allons nous connecter et exécuter notre script

La sortie ‘CREATE TABLE’ nous prouve que notre script existe bel et bien ;

Puis à la fin on se deconnecte

```
root@debian:/home/joel#
root@debian:/home/joel# docker exec -it yplanningDB bash
root@646d9b951036:/# psql -U joel -d yplanning
psql (16.3 (Debian 16.3-1.pgdg120+1))
Type "help" for help.

yplanning=# \i /script.sql
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
yplanning=# \q
root@646d9b951036:/# exit
exit
root@debian:/home/joel#
```

4 : Mise en place des sauvegardes automatisée :

Nous allons mettre en place les sauvegardes automatisées :

D’abord on fait notre script de sauvegarde

```
#!/bin/bash

# Variables
BACKUP_DIR="/backups"
BACKUP_FILE="yplan_backup_$(date +\%Y\%m\%d\%H\%M\%S).sql"
CONTAINER_NAME="yplanningDB"
DB_USER="joel"
DB_NAME="yplanning"

# Créer le répertoire de sauvegarde s'il n'existe pas
mkdir -p ${BACKUP_DIR}

# Exécuter la commande pg_dump dans le conteneur Docker
sudo docker exec ${CONTAINER_NAME} pg_dump -U ${DB_USER} ${DB_NAME} > ${BACKUP_DIR}/${BACKUP_FILE}

# Optionnel : supprimer les sauvegardes anciennes (plus de 7 jours)
find ${BACKUP_DIR} -type f -name "*.sql" -mtime +7 -exec rm {} \;

echo "Sauvegarde effectuée: ${BACKUP_DIR}/${BACKUP_FILE}"
```

Puis on lui rajoute les droits d’exécution au script :

```
root@debian:/home/joel# chmod +x Bureau/backup.sh
root@debian:/home/joel#
```

Avec Crontab on va automatiser l’éxecution du script de sauvegarde en utilisant la commande :

sudo crontab -e

ensuite on va rajouter une entrée cron pour exécuter le script tous les jours à 1h du matin :

Edit this file to introduce tasks to be run by cron.
#
Each task to run has to be defined through a single line
indicating with different fields when the task will be run
and what command to run for the task
#
To define the time you can provide concrete values for
minute (m), hour (h), day of month (dom), month (mon),
and day of week (dow) or use '*' in these fields (for 'any').
#
Notice that tasks will be started based on the cron's system
daemon's notion of time and timezones.
#
Output of the crontab jobs (including errors) is sent through
email to the user the crontab file belongs to (unless redirected).
#
For example, you can run a backup of all your user accounts
at 5 a.m every week with:
0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
For more information see the manual pages of crontab(5) and cron(8)
#
m h dom mon dow   command

@ 1 * * * /Bureau/backup.sh