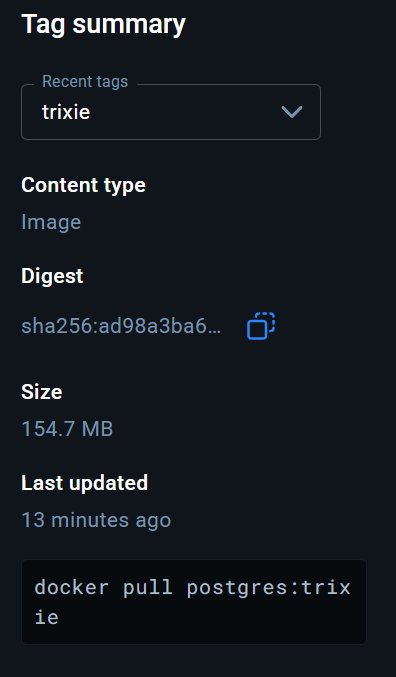
Si on veux utiliser un Network on toi le cree en premier   
  
Premier étape aller sur le site <https://hub.docker.com/explore> et sélectionner l’image que vous voulez puis il faut trouver sa et le coper dans PowerCelle admin



Puis on fait :   
docker image ls  
On voie toute nos image   
  
Pour démarre un conteneur on fait :   
docker run et\_le\_nom\_de\_l’image

S’il a une erreur il faut rajouter des champs :   
docker run -d -e POSTGRES\_PASSWORD=pass postgres

Pour rajouter un nom   
docker run -d --name PG1 -e POSTGRES\_PASSWORD=pass postgres

On fait :   
docker ps  
Cela nous permet de voir tout les container démarrais

On peux voir tout les conainer avec :

Docker ps -a

dans le powerCell quon a lance le conteneur on fait :   
ctrl+c

pour le stop l’arrete

ou  
  
docker stop CONTAINER ID ou NAMES / docker stop 65df09ad9b79  
du conteneur pour le stop

pour supprimer un conteneur :  
docker rm CONTAINER ID

pour supprimer tout les conteneur :

docker ps -–filter status=exited -d   
on recupere tout les CONTAINER ID   
Puis on fait :   
docker rm $( docker ps –filter status=exited -d )  
On supprimer tout les conteneur

on fait :   
docker -it le\_nom\_du\_conteure

pour ouvrir un terminal interactive on et dans le conteneur

pour rentre dans le conteneur :

docker exec -it PG22 psql -U postgres PG22 et le nom de mon conteneur

psql -U postgres ce propre a Postgres

pour quitter le conteneur :   
exit

pour revenir dans le conteneur:   
docker exec -it PG1 bash PG1 et le nom de mon conteneur

Pour cree un volume :  
docker volume create psqlenset le\_nom\_quon\_veux

Le volume et un stockage qui se trouve a l’extérieure du conteneur

Pour voir tout les volume :   
docker volume ls

Pour cree un conteneur avec un volume :   
docker run -d --name PG2 -v psqlenset:/var/lib/postgresql/data -e POSTGRES\_PASSWORD=pass postgres

Pour rajouter un Pord

docker run -d --name PG23 -p 5432:5432 -v psqlenset:/var/lib/postgresql/data -e POSTGRES\_PASSWORD=pass postgres

Pour rajouter réseau

docker run -d --name PG24 -p 5432:5432 -v psqlenset:/var/lib/postgresql/data --network ensetnet -e POSTGRES\_PASSWORD=pass postgres

docker run -d \  
 --name mysql-db \  
 -e MYSQL\_ROOT\_PASSWORD=rootpass \  
 -e MYSQL\_DATABASE=animaux \  
 -e MYSQL\_USER=benjamin \  
 -e MYSQL\_PASSWORD=passben \  
 -v mysql-data:/var/lib/mysql \  
 -v "$(pwd)/init.sql:/docker-entrypoint-initdb.d/init.sql" \  
 --network dev-net \  
 mysql:latest

-e MYSQL\_DATABASE=animaux Crée automatiquement une base de données nommée animaux à l'initialisation du

conteneur.

-e MYSQL\_USER=benjamin Crée un utilisateur MySQL nommé benjamin.   
-e MYSQL\_PASSWORD=passben Définit le mot de passe de l’utilisateur benjamin à

passben

Pour voir Un json de notre contenur

Docker inspect le\_nom\_du\_contneur

/var/lib/postgresql/data a cherche dans la dock le bon emplacement

Pour voir tout se qui et active :

Docker ps -q

Pour arret tout en meme temps :

Docker stop $(docker ps -q)

Va tout supprimer

Docker rm $(docker ps -a -q)

Pour supprimer une image :   
docker rmi et\_le\_nom\_de\_image

On toi d’apport supprimer le conteneur puis l’image

Pour voir toute les image

Docker image ls

Pour afficher l’ensemble des réseau

docker network ls

Pour créer un nouceau réseau :

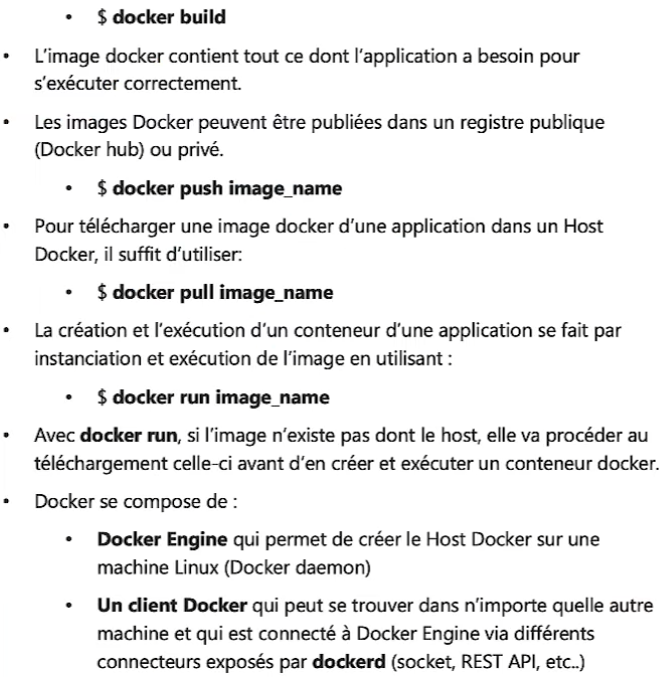
docker network create --driver bridge ensetnet ensetnet le nom quon veux

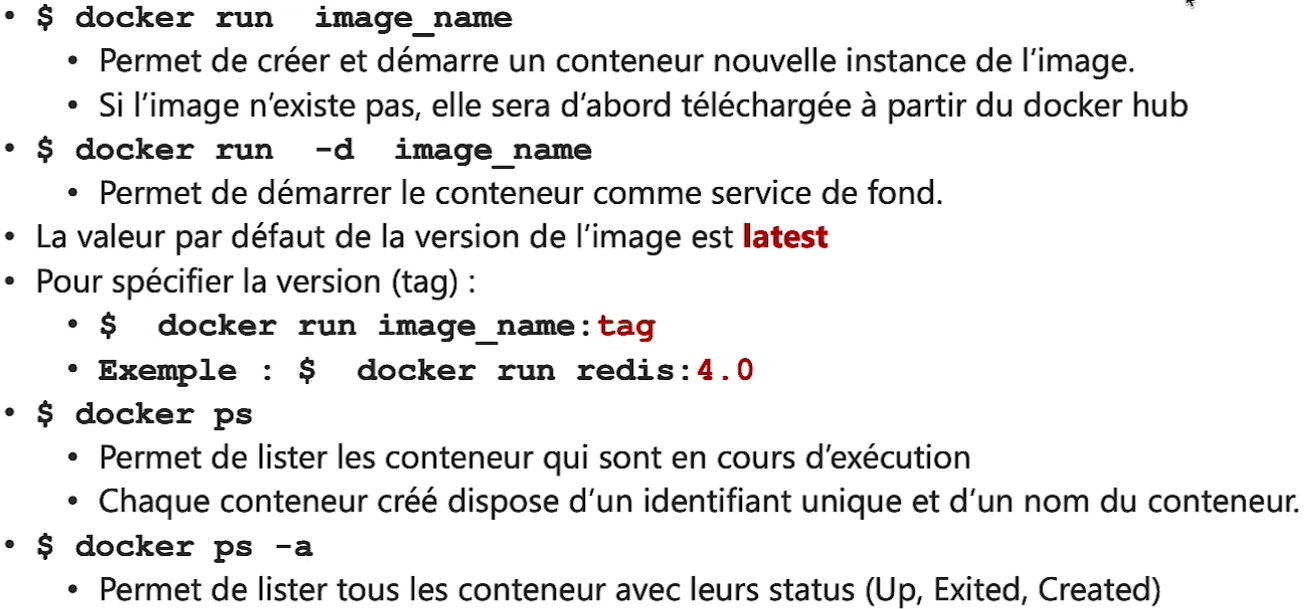
Pour info :

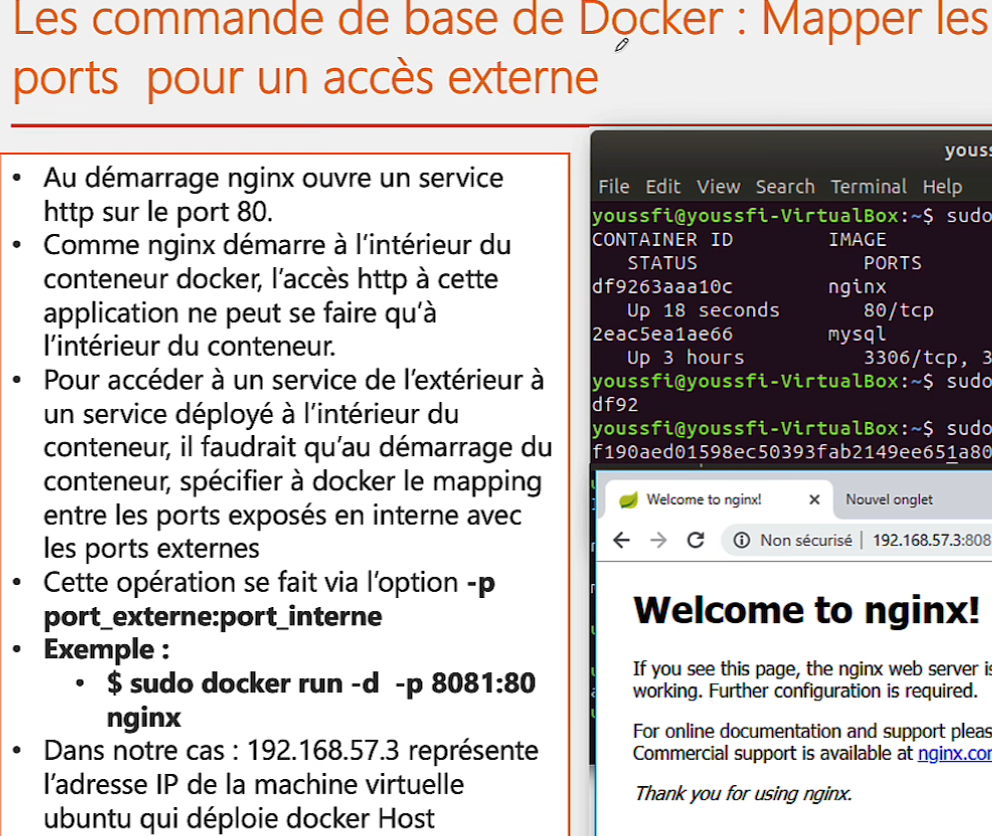
clear

pour avoir rien dans le terminale

<ctrl> + <l> (c'est un raccourci clavier pour envoyer le signal de nettoyage au terminal qui le supporte)

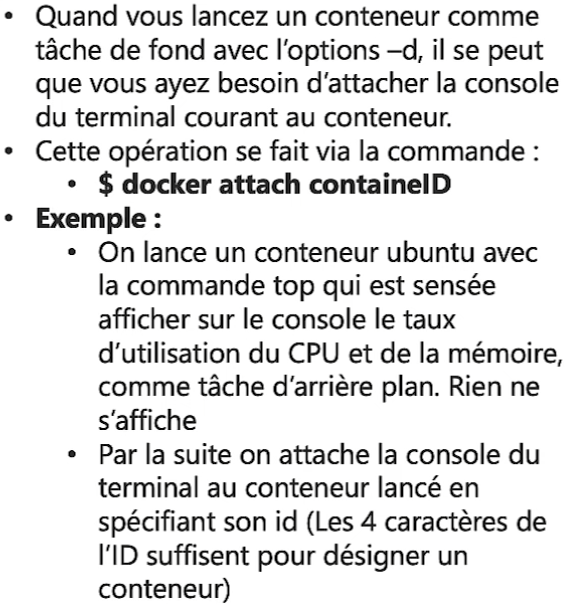


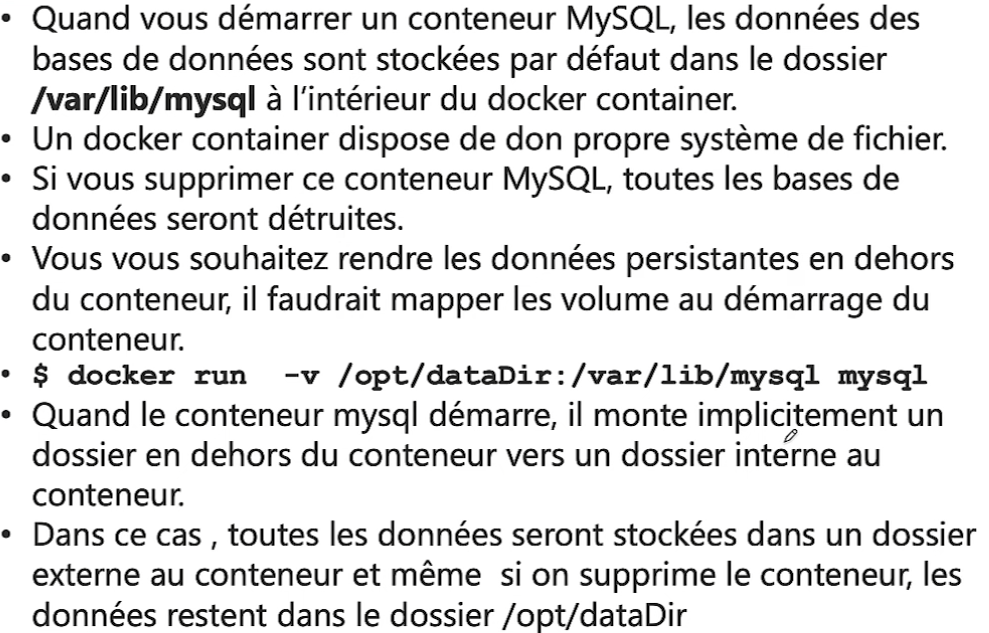


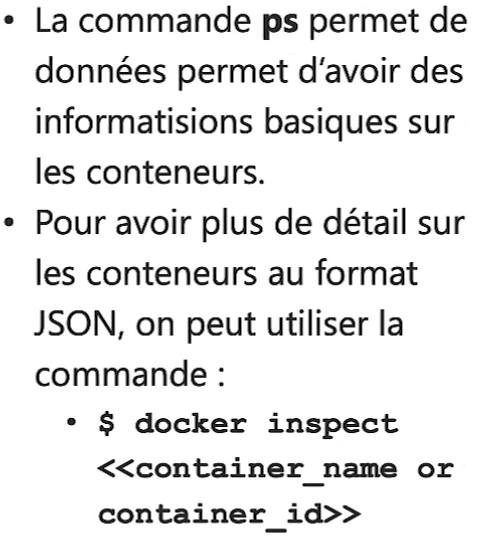


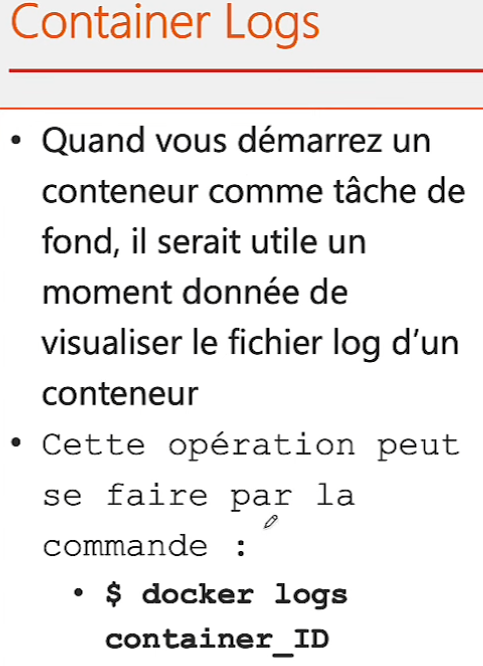


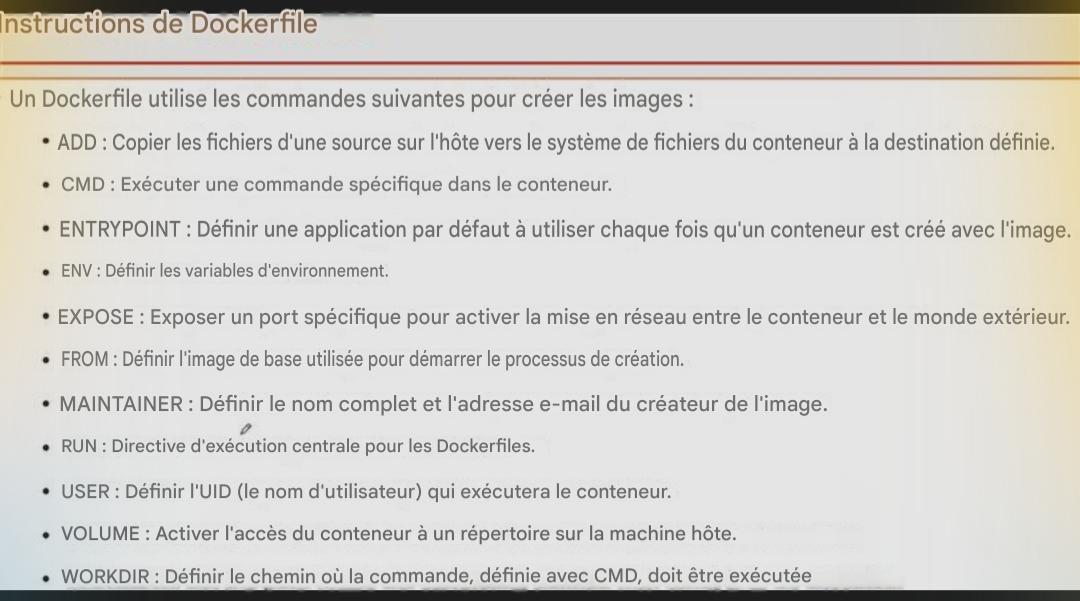


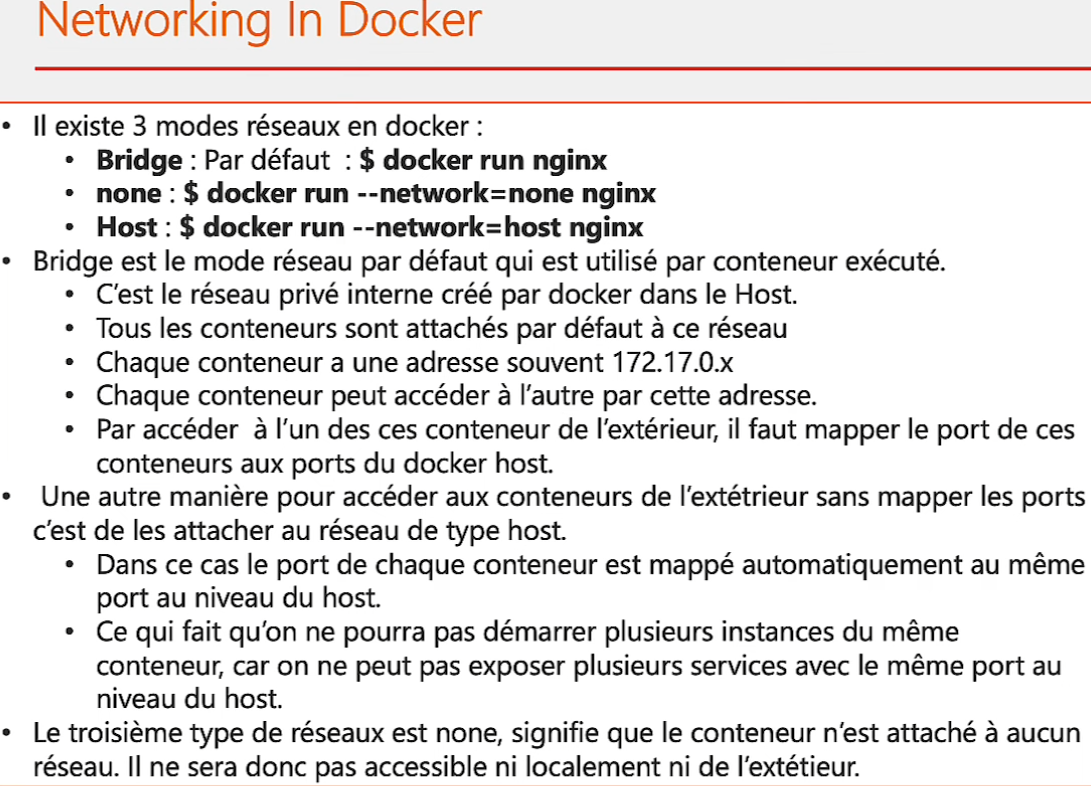


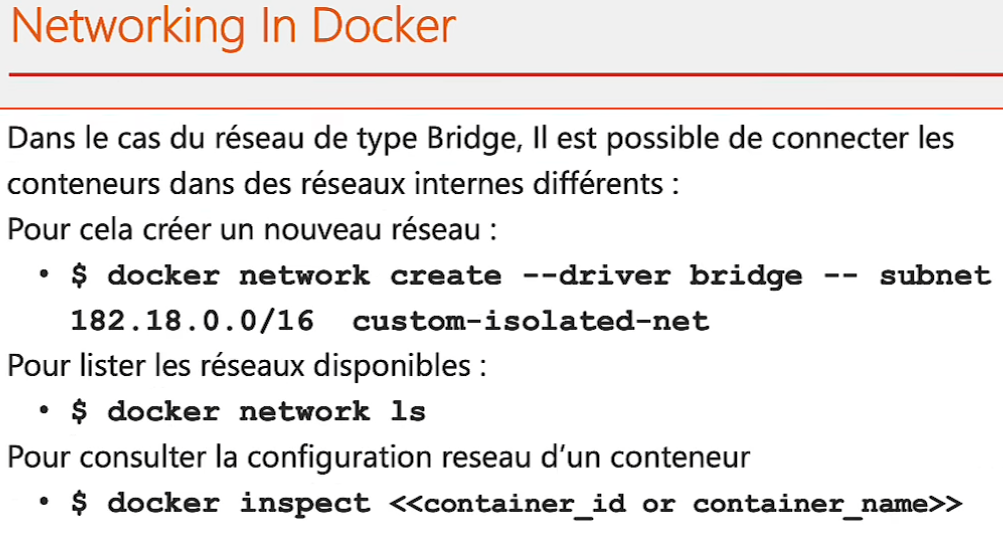


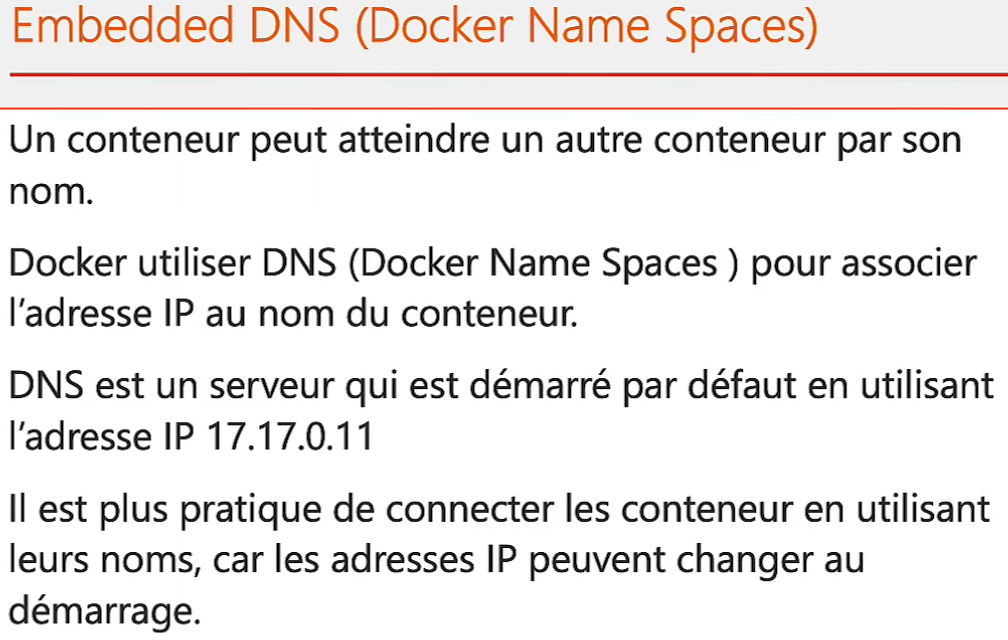


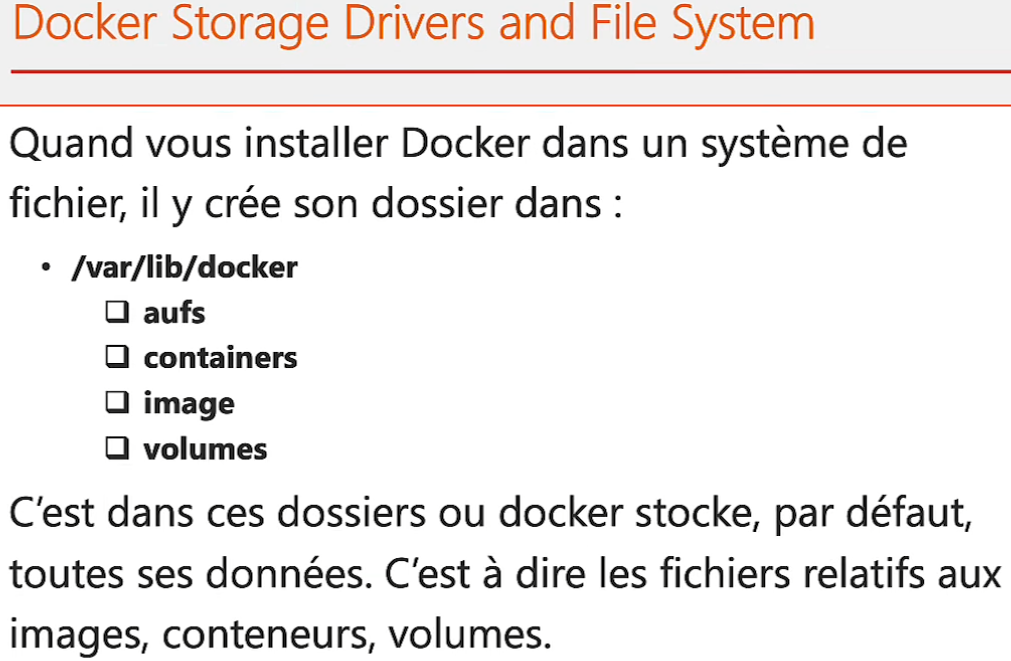


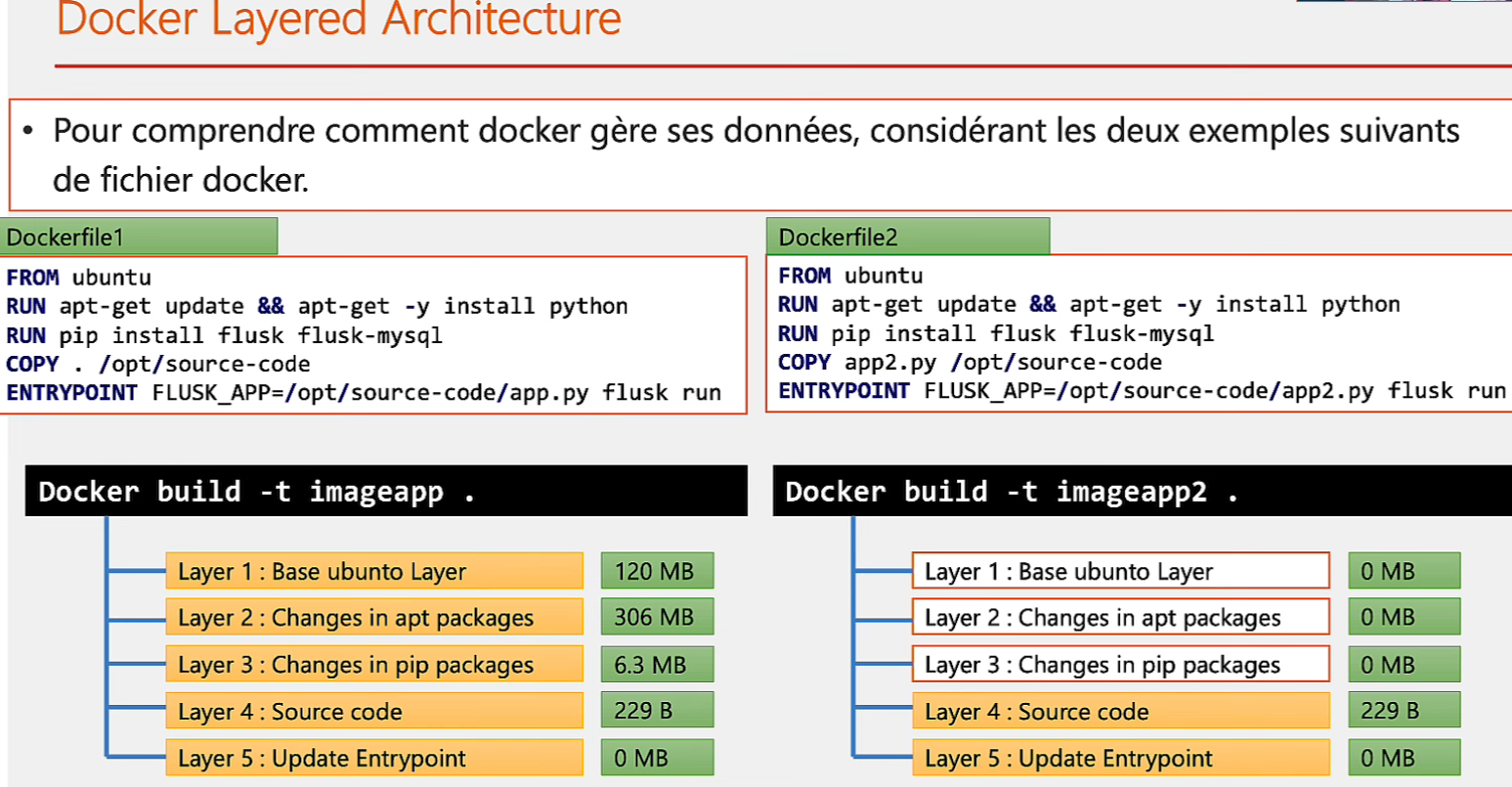


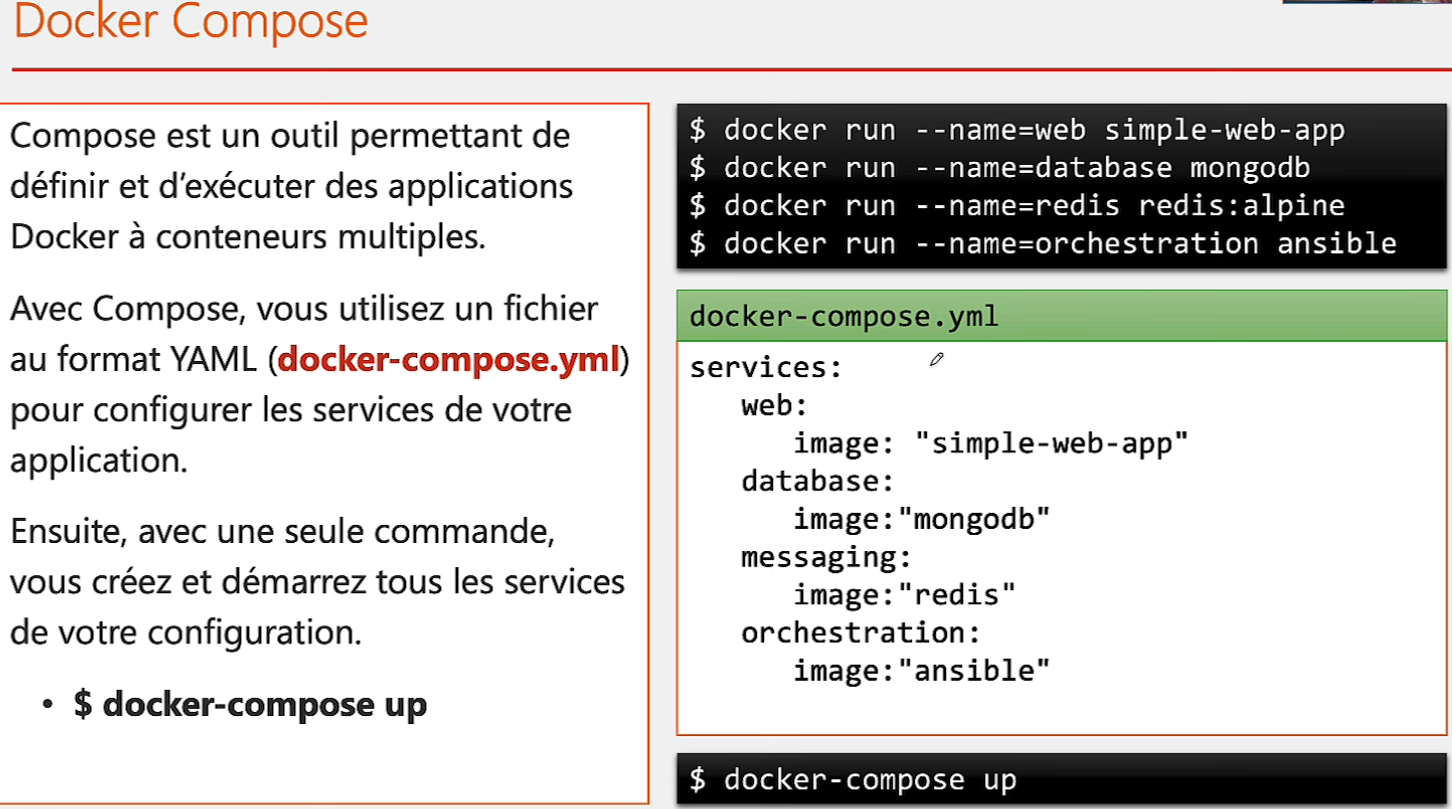












sudo -i

Pour avoir le root

Docker aide :

Commande

docker run -it alpine /bin/sh

Memo   
-it = mode interactif + terminal

Accéder à la documentation d’une commande

| **Commande anglaise** | **Traduction française** | **Explication** |
| --- | --- | --- |
| man commande | man commande | Affiche le manuel (documentation) de la commande |
| commande -h | commande -h | Affiche une aide rapide (souvent abrégée) |
| commande --help | commande --help | Affiche une aide complète avec les options disponibles |

Exemple : man ls → affiche le manuel de la commande ls

**Combiner des commandes avec un pipe (**|**)**

| **Commande** | **Traduction** | **Résultat** |
| --- | --- | --- |
| `echo toto | sed 's/o/i'` | Affiche "toto", puis remplace les "o" par "i" → titi |
| `ls -l | grep ".php"` | Liste les fichiers et filtre ceux qui contiennent .php |
| `cat fichier.txt | wc -l` | Compte le nombre de lignes dans le fichier |
| `docker ps | grep apache` | Affiche les conteneurs Docker contenant "apache" |

Commandes utiles dans Linux / WSL / Docker

| **Commande** | **Traduction / Utilité** |
| --- | --- |
| ls | Liste les fichiers |
| cd dossier | Change de dossier |
| pwd | Affiche le chemin actuel |
| mkdir nom | Crée un dossier |
| rm fichier | Supprime un fichier |
| cp source destination | Copie un fichier |
| mv source destination | Déplace ou renomme un fichier |
| touch fichier.txt | Crée un fichier vide |
| nano fichier.txt | Édite un fichier en mode texte |
| chmod +x script.sh | Rend un script exécutable |
| ./script.sh | Exécute un script |

**docker run --rm alpine sh -c "find / | wc -l"**

Cette commande lance un conteneur Alpine Linux, exécute une commande à l’intérieur, puis le supprime automatiquement après exécution.

| **Partie** | **Traduction** | **Rôle** |
| --- | --- | --- |
| docker | Docker | Le programme principal |
| run | Exécuter | Lance un nouveau conteneur |
| --rm | Supprimer après usage | Supprime le conteneur une fois terminé (nettoyage automatique) |
| alpine | Image Alpine | Utilise l’image Alpine Linux (très légère) |
| sh | Shell | Lance le shell sh dans le conteneur |
| -c | Commande | Permet d’exécuter une commande en chaîne |
| `"find / | wc -l"` | Trouver tous les fichiers et les compter |

**Que fait find / | wc -l exactement ?**

* find / → explore **tous les fichiers et dossiers** à partir de la racine /
* | → envoie le résultat à la commande suivante
* wc -l → **compte le nombre de lignes** (donc le nombre d’éléments trouvés)

Résultat : le **nombre total de fichiers et dossiers** dans le conteneur Alpine

Si vous souhaitez ne compter que les fichiers (et pas les dossiers), vous pouvez affiner la commande :

docker run --rm alpine sh -c "find / -type f | wc -l"

Voici les informations disponibles dans les sources concernant **bash** et **sh** :

* **Bash** (Bourne Again SHell) est une implémentation complète du shell POSIX du projet GNU. Il offre l’édition de ligne de commande interactive, le contrôle des tâches, des fonctionnalités similaires à csh comme l’historique et l’expansion des accolades, et de nombreuses autres fonctionnalités avancées. Dans l’image Docker officielle, Bash est installé à /usr/local/bin/bash et non à /bin/bash. Il est recommandé d’utiliser le shebang #!/usr/bin/env bash pour garantir la portabilité dans les scripts [**What is Bash?**](https://hub.docker.com/_/bash#what-is-bash) [**How to use this image**](https://hub.docker.com/_/bash#how-to-use-this-image).

Cette commande bash -cpermet d'invoquer l'interpréteur de commandes Bash et d'exécuter la chaîne de commande qui suit. Dans le contexte de Docker, elle est souvent utilisée dans les Dockerfiles ou avec docker runet docker execpour exécuter des commandes shell complexes, notamment lorsque des fonctionnalités telles que la substitution de variables, le piping ou le chaînage de commandes sont nécessaires.

* **sh** est le shell par défaut sur Linux, utilisé par Docker pour exécuter les instructions de type shell dans les Dockerfile (par exemple, RUN echo hello). Par défaut, la forme shell des instructions Dockerfile utilise ["/bin/sh", "-c"] comme shell [**SHELL**](https://docs.docker.com/reference/dockerfile/#shell).

Commandes Docker de base

| **Commande** | **Traduction / Utilité** |
| --- | --- |
| docker --version | Affiche la version de Docker |
| docker ps | Liste les conteneurs en cours |
| docker images | Liste les images Docker |
| docker run image | Lance un conteneur |
| docker run -it image bash | Lance un conteneur en mode interactif |
| docker exec -it nom bash | Ouvre un terminal dans un conteneur |
| docker stop nom | Arrête un conteneur |
| docker rm nom | Supprime un conteneur |
| docker rmi image | Supprime une image |
| docker-compose up | Lance les services définis dans docker-compose.yml |
| docker-compose down | Arrête et supprime les services |

Cat > avec le nom du fichier puis le code et fermer avec CTRL+d pour ferme< ctrl>+d