1 - sudo apt update

2 - sudo apt upgrade

3 - sudo apt install apt-transport-https ca-certificates curl gnupg2 software-properties-common -y

4 - curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

5 - echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/debian $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

6 - sudo apt update

7 - sudo apt install docker-ce

==========

CONTENU DU FICHIER Dockerfile

```
FROM ubuntu
RUN apt-get update
RUN apt-get install net-tools -y
RUN apt-get install curl -y
RUN apt-get install apache2 -y
RUN apt-get install apache2-utils -y
RUN apt-get clean
EXPOSE 80 443
CMD echo "ServerName localhost" >> /etc/apache2/apache2.conf
&& /etc/init.d/apache2 restart && bash
```

==========

```
docker search mysql
docker login -u username
docker tag ID username/apache_ready:V1.0
docker image push username/apache_ready:V1.0
```

======== INSTALLATION DE DOCKER COMPOSE =========

Téléchargement de Docker Compose

curl -L "https://github.com/docker/compose/releases/download/v2.22.0/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

Rendre Exécutable Le Programme

chmod +x /usr/local/bin/docker-compose

Vérification de l'installation

sudo docker-compose -v

Création du Fichier docker-compose.yml

touch docker-compose.yml

CONTENU DU FICHIER

```
version: '3.8'

services:
  db:
    image: mysql:5.7
    environment:
      MYSQL_ROOT_PASSWORD: passe
      MYSQL_DATABASE: exampledb
    volumes:
      - db_data:/var/lib/mysql

  web:
    build: .
    volumes:
      - ./app:/var/www/html
    ports:
      - "80:80"
    depends_on:
      - db
    environment:
      MYSQL_ROOT_PASSWORD: example
      MYSQL_DATABASE: exampledb
      DB_HOST: db
```

```
volumes:
  db_data:
```

CODE DOCKERFILE

```
FROM ubuntu:16.04
RUN apt-get update && \
apt-get install -y apache2 php libapache2-mod-php mysql-client php7.0-mysql
php-mysqli
RUN sed -i 's/;extension=php_mysqli.dll/ extension=php_mysqli.dll/'
/etc/php/7.0/apache2/php.ini
RUN sed -i 's/;extension=php_pdo_mysql.dll/extension=php_pdo_mysql.dll/'
/etc/php/7.0/apache2/php.ini

ADD ./host.conf /etc/apache2/sites-enabled/000-default.conf
ADD ./index.php /var/www/html/index.php

ENTRYPOINT ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]
```

CODE INDEX.PHP

```
<?php
$host = 'db';
$user = 'root';
$pass = getenv('MYSQL_ROOT_PASSWORD');
$db = getenv('MYSQL_DATABASE');
$conn = new mysqli($host, $user, $pass, $db);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "SHOW DATABASES";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    echo "<h1>Liste des bases de donnees:</h1>";
    echo "<ul>";
    while($row = $result->fetch_assoc()) {
        echo "<li>" . $row['Database'] . "</li>";
    }
```

```
    echo "</ul>";
} else {
    echo "0 results";
}
$conn->close();
?>
```

Contenu du fichier host.conf

```
<VirtualHost *:80>
ServerAdmin webmaster@localhost
DocumentRoot /var/www/html
</VirtualHost>
```

COMMANDE D'EXÉCUTION

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

Docker Swarm

Initialiser Docker Swarm **(Création du Manager)**

```
docker swarm init --advertise-addr [ip-du-manager]
```

Ajouter un nœud au Swarm (Ajout d'un worker)

```
docker swarm join --token [le token] [ip-du-manager]:2377
```

Informations du clusters

```
docker node ls
```

Déploiement d'un serveur web (NGINX) dans le cluster

```
docker service create --name mon-service-nginx --publish 80:80 nginx
```

Visualiser l'état du service (par le manager)

docker service ls

Modification du nombre de réplicas (3 réplicas)

docker service scale mon-servdeice-nginx=3

Déploiement de nos services basés sur swarm avec docker stack

Création du fichier De déploiement : docker-compose.yml

```
version: "3.9"

services:
  mon-service-nginx:
    image: nginx:alpine
    ports:
      - "80:80"
    deploy:
      replicas: 3
      restart_policy:
        condition: on-failure
```

Lister les satck

docker stack ls

Création du stack

docker stack deploy -c docker-compose.yml mon-stack