

Projet Docker

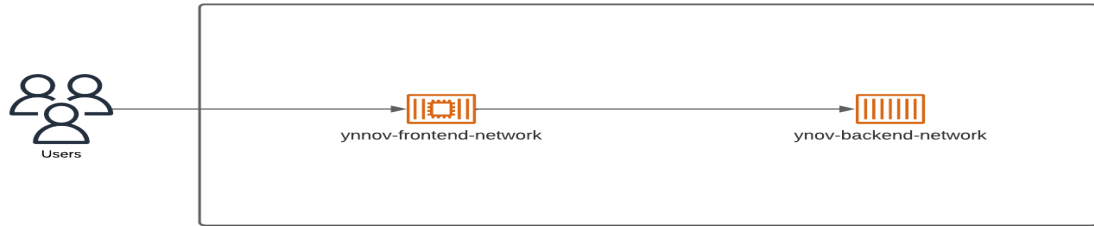
Fait le 07/12/2023

PAR : DIALLO Assane

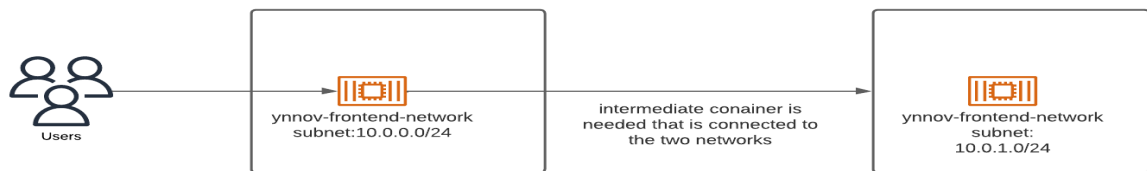
BARBICH Karima

Architecture Globale

FIRST TASK



TASK 2



Introduction

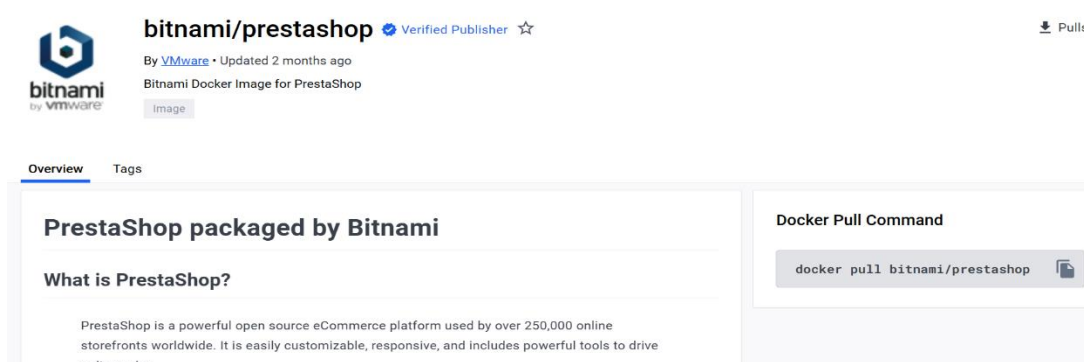
Ce projet consiste à orchestrer le déploiement d'une application e-commerce à l'aide de Docker, en mettant l'accent sur une architecture réseau robuste qui sépare les composants frontaux et dorsaux pour des raisons de sécurité et de performance. En utilisant l'image de PrestaShop disponible sur Docker Hub, nous mettrons en œuvre un environnement d'application réparti qui simule un cadre de production réel, tout en explorant les concepts avancés de réseautage tels que la communication inter-conteneurs et la persistance des données. Ce travail sert de fondation pratique pour comprendre comment les conteneurs Docker peuvent être utilisés pour construire et déployer des applications scalables et sécurisées.

Démarche suivie

1. Déployer une application e-commerce en utilisant l'image PrestaShop disponible sur Docker Hub, comprenant un site web frontal et une base de données pour les données persistantes.
2. S'assurer que les deux conteneurs peuvent communiquer entre eux en utilisant leurs noms de conteneurs.
3. Créer deux réseaux Docker distincts avec des CIDR spécifiés, en utilisant le paramètre `--subnet`.
4. Utiliser un conteneur intermédiaire, tel qu'un gateway ou router, pour connecter ces réseaux et configurer les tables de routage pour permettre la communication entre les conteneurs.
5. Vérifier que les conteneurs peuvent communiquer entre eux en utilisant leurs adresses IP, sans nécessairement déployer l'application.

1. Déploiement des conteneurs :

Récupération de l'image PrestaShop : Nous utilisons Docker Hub pour télécharger la dernière image de PrestaShop.



bitnami/prestashop Verified Publisher ☆

By [VMware](#) • Updated 2 months ago

Bitnami Docker Image for PrestaShop

Image

Overview Tags

PrestaShop packaged by Bitnami

What is PrestaShop?

PrestaShop is a powerful open source eCommerce platform used by over 250,000 online storefronts worldwide. It is easily customizable, responsive, and includes powerful tools to drive sales.

Docker Pull Command

```
docker pull bitnami/prestashop
```

```
Microsoft Windows [version 10.0.19045.3693]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\assan>docker pull bitnami/prestashop:latest
latest: Pulling from bitnami/prestashop
a6c5130f46d2: Pull complete
Digest: sha256:ee3e880973147ef236a971bcd2fb4cd829509ebb25b7c149edc6d5c8876fa30c
Status: Downloaded newer image for bitnami/prestashop:latest
docker.io/bitnami/prestashop:latest

What's Next?
View a summary of image vulnerabilities and recommendations - docker scout quickview bitnami/prestashop:latest

C:\Users\assan>
```

Création du réseau :

Nous avons commencé par créer un réseau Docker personnalisé nommé prestashop-net. Ensuite, nous avons listé tous les réseaux Docker disponibles sur notre système.

```
C:\Users\assan>docker network create prestashop-net
bb465f8e240d603f87c0662adb7df89e409ca9b763b7e9ad44a621ceba46fe65

C:\Users\assan>docker network ls
NETWORK ID        NAME                DRIVER              SCOPE
c85ba252395c      bridge              bridge              local
9bf46c3c7fe1      host                host                local
5e58b417cd7d      none                null                local
bb465f8e240d      prestashop-net      bridge              local
```

Création des conteneurs Prestashop et mysql :

```
C:\Users\assan>docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS        NAMES
C:\Users\assan>docker run -ti --name mysql --network prestashop-net -e MYSQL_ROOT_PASSWORD=admin -p 3307:3306 --privileged -d mysql:8.0.35-debian
635df568e3d428e06ca8de282fb44e629acb3ae7335663d3f71c0ef9e9596133
C:\Users\assan>docker run -ti --name prestashop --network prestashop-net -e DB_SERVER=mysql -p 8080:80 --privileged -d prestashop/prestashop
9b5bc788b19f41413147add495227a4f3591ace20d1661c934bbb5cf463bf570
C:\Users\assan>
```

Nous avons lancé deux conteneurs Docker. Le premier est un conteneur MySQL, configuré avec un mot de passe root et exposé sur le port 3307. Le deuxième est un conteneur PrestaShop, lié au conteneur MySQL via la variable d'environnement DB_SERVER. Ces deux conteneurs sont connectés au réseau prestashop-net que nous avons créé précédemment. Après avoir lancé ces commandes, notre application PrestaShop et notre base de données devraient être opérationnelles.

```
C:\Users\assan>docker network inspect prestashop-net
[
  {
    "Name": "prestashop-net",
    "Id": "bb465fbee240d603f87c0662adb7df89e409ca9b763b7e9ad44a621ceba46fe65",
    "Created": "2023-12-07T12:50:09.368575053Z",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": {},
      "Config": [
        {
          "Subnet": "192.168.16.0/20",
          "Gateway": "192.168.16.1"
        }
      ]
    },
    "Internal": false,
    "Attachable": false,
    "Ingress": false,
    "ConfigFrom": {
      "Network": ""
    },
    "ConfigOnly": false,
    "Containers": {
      "8f1f555290ed79baadea2ae1f08c2d82768ee687f61f740f145af2933aa7714d": {
        "Name": "prestashop",
        "EndpointID": "3ecd9f9fac7990ed0e5bca06afaf167563a3d87943188b6eea5b28e5fce40780",
        "MacAddress": "02:42:c0:a8:10:03",
        "IPv4Address": "192.168.16.3/20",
        "IPv6Address": ""
      },
      "f4b243ca5841849000e101a4a54f0459b83202b8277fbfe9c03f5d75279faacd": {
        "Name": "mysql",
        "EndpointID": "7e1ae0fa7f71b48b379a542ab576be208edb9788c41f6ce82f2be2a0aa523a064",
        "MacAddress": "02:42:c0:a8:10:02",
        "IPv4Address": "192.168.16.2/20",
        "IPv6Address": ""
      }
    },
    "Options": {},
    "Labels": {}
  }
]
C:\Users\assan>
```

Conteneur mysql

Mettre à jour la liste des paquets dans le conteneur :

```
C:\Users\assan>docker exec -it mysql bash
root@f4b243ca5841:/# apt update
get:1 http://deb.debian.org/debian bullseye InRelease [116 kB]
get:2 http://repo.mysql.com/apt/debian bullseye InRelease [17.9 kB]
get:3 http://deb.debian.org/debian-security bullseye-security InRelease [48.4 kB]
get:4 http://repo.mysql.com/apt/debian bullseye/mysql-8.0 amd64 Packages [12.6 kB]
get:5 http://deb.debian.org/debian bullseye-updates InRelease [44.1 kB]
get:6 http://deb.debian.org/debian bullseye/main amd64 Packages [8062 kB]
get:7 http://deb.debian.org/debian-security bullseye-security/main amd64 Packages [260 kB]
get:8 http://deb.debian.org/debian bullseye-updates/main amd64 Packages [17.7 kB]
Fetched 8578 kB in 8s (1065 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
root@f4b243ca5841:/#
```

Installation des outils réseau dans notre conteneur MySQL, y compris net-tools, iputils-ping, et iproute2 pour la configuration réseau :

```
root@f4b243ca5841:/# apt install net-tools inetutils-ping iproute2 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libatm1 libbbpf0 libbsd0 libcap2 libcap2-bin libelf1 libmd0 libmn10 libpam-cap libxtables12 netbase
Suggested packages:
  iproute2-doc
The following NEW packages will be installed:
  inetutils-ping iproute2 libatm1 libbbpf0 libbsd0 libcap2 libcap2-bin libelf1 libmd0 libmn10 libpam-cap
  libxtables12 net-tools netbase
0 upgraded, 14 newly installed, 0 to remove and 0 not upgraded.
Need to get 2045 kB of archives.
After this operation, 6391 kB of additional disk space will be used.
Get:1 http://deb.debian.org/debian bullseye/main amd64 libelf1 amd64 0.183-1 [165 kB]
Get:2 http://deb.debian.org/debian bullseye/main amd64 libbbpf0 amd64 1:0.3-2 [98.3 kB]
Get:3 http://deb.debian.org/debian bullseye/main amd64 libmd0 amd64 1.0.3-3 [28.0 kB]
Get:4 http://deb.debian.org/debian bullseye/main amd64 libbsd0 amd64 0.11.3-1+deb11u1 [108 kB]
Get:5 http://deb.debian.org/debian bullseye/main amd64 libcap2 amd64 1:2.44-1 [23.6 kB]
Get:6 http://deb.debian.org/debian bullseye/main amd64 libmn10 amd64 1.0.4-3 [12.5 kB]
Get:7 http://deb.debian.org/debian bullseye/main amd64 libxtables12 amd64 1.8.7-1 [45.1 kB]
Get:8 http://deb.debian.org/debian bullseye/main amd64 libcap2-bin amd64 1:2.44-1 [32.6 kB]
Get:9 http://deb.debian.org/debian bullseye/main amd64 iproute2 amd64 5.10.0-4 [930 kB]
Get:10 http://deb.debian.org/debian bullseye/main amd64 netbase all 6.3 [19.9 kB]
Get:11 http://deb.debian.org/debian bullseye/main amd64 inetutils-ping amd64 2:2.0-1+deb11u2 [245 kB]
Get:12 http://deb.debian.org/debian bullseye/main amd64 libatm1 amd64 1:2.5.1-4 [71.3 kB]
Get:13 http://deb.debian.org/debian bullseye/main amd64 libpam-cap amd64 1:2.44-1 [15.4 kB]
Get:14 http://deb.debian.org/debian bullseye/main amd64 net-tools amd64 1.60+git20181103.0eebece-1 [250 kB]
Fetched 2045 kB in 1s (2012 kB/s)
debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package libelf1:amd64.
(Reading database ... 9554 files and directories currently installed.)
Preparing to unpack .../00-libelf1_0.183-1_amd64.deb ...
Unpacking libelf1:amd64 (0.183-1) ...
Selecting previously unselected package libbbpf0:amd64.
Preparing to unpack .../01-libbbpf0_1%3a0.3-2_amd64.deb ...
```

Après l'installation des outils réseau, nous avons utilisé la commande **ifconfig** pour afficher la configuration actuelle des interfaces réseau de notre conteneur. Cela nous permet de voir l'adresse IP attribuée à notre conteneur.

```
root@f4b243ca5841:/#
root@f4b243ca5841:/#
root@f4b243ca5841:/# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.16.2 netmask 255.255.240.0 broadcast 192.168.31.255
    ether 02:42:c0:a8:10:02 txqueuelen 0 (Ethernet)
    RX packets 7759 bytes 11152486 (10.6 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 3812 bytes 255375 (249.3 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    loop txqueuelen 1000 (Local Loopback)
    RX packets 44 bytes 4268 (4.1 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 44 bytes 4268 (4.1 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@f4b243ca5841:/#
```

Nous avons ensuite utilisé la commande **ping** pour tester la connectivité réseau avec le conteneur PrestaShop, qui a résolu à l'adresse IP 192.168.16.3. Cela est essentiel pour confirmer que notre conteneur de base de données peut communiquer avec notre application PrestaShop.

```
root@f4b243ca5841:/#
root@f4b243ca5841:/# ping prestashop
PING prestashop (192.168.16.3): 56 data bytes
64 bytes from 192.168.16.3: icmp_seq=0 ttl=64 time=0.045 ms
64 bytes from 192.168.16.3: icmp_seq=1 ttl=64 time=0.122 ms
64 bytes from 192.168.16.3: icmp_seq=2 ttl=64 time=0.115 ms
64 bytes from 192.168.16.3: icmp_seq=3 ttl=64 time=0.113 ms
AC--- prestashop ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.045/0.099/0.122/0.031 ms
root@f4b243ca5841:/#
```

Conteneur Prestashop

Mettre à jour la liste des paquets dans le conteneur :

```
C:\Users\assan>docker exec -it prestashop bash
root@8f1f555290ed:/var/www/html# apt update
Get:1 http://deb.debian.org/debian bookworm InRelease [151 kB]
Get:2 http://deb.debian.org/debian bookworm-updates InRelease [52.1 kB]
Get:3 http://deb.debian.org/debian-security bookworm-security InRelease [48.0 kB]
Get:4 http://deb.debian.org/debian bookworm/main amd64 Packages [8780 kB]
Get:5 http://deb.debian.org/debian bookworm-updates/main amd64 Packages [6668 B]
Get:6 http://deb.debian.org/debian-security bookworm-security/main amd64 Packages [106 kB]
Fetched 9144 kB in 3s (3186 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
27 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@8f1f555290ed:/var/www/html#
```

Installation des outils de réseau supplémentaires dans votre conteneur PrestaShop, tels que **net-tools**, **iputils-ping**, et **iproute2** pour la configuration du réseau:

```
root@8f1f555290ed:/var/www/html#
root@8f1f555290ed:/var/www/html# apt install net-tools inetutils-ping iproute2 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libatm1 libbpf1 libcap2-bin libelf1 libmnl0 libpam-cap libxtables12 netbase
Suggested packages:
  iproute2-doc python3:any
The following NEW packages will be installed:
  inetutils-ping iproute2 libatm1 libbpf1 libcap2-bin libelf1 libmnl0 libpam-cap libxtables12 net-tools
  netbase
0 upgraded, 11 newly installed, 0 to remove and 27 not upgraded.
Need to get 1867 kB of archives.
After this operation, 6759 kB of additional disk space will be used.
Get:1 http://deb.debian.org/debian bookworm/main amd64 libelf1 amd64 0.188-2.1 [174 kB]
Get:2 http://deb.debian.org/debian bookworm/main amd64 libbpf1 amd64 1:1.1.0-1 [145 kB]
Get:3 http://deb.debian.org/debian bookworm/main amd64 libmnl0 amd64 1.0.4-3 [12.5 kB]
Get:4 http://deb.debian.org/debian bookworm/main amd64 libxtables12 amd64 1.8.9-2 [30.8 kB]
Get:5 http://deb.debian.org/debian bookworm/main amd64 libcap2-bin amd64 1:2.66-4 [34.7 kB]
Get:6 http://deb.debian.org/debian bookworm/main amd64 iproute2 amd64 6.1.0-3 [1046 kB]
Get:7 http://deb.debian.org/debian bookworm/main amd64 netbase all 6.4 [12.8 kB]
Get:8 http://deb.debian.org/debian bookworm/main amd64 inetutils-ping amd64 2:2.4-2+deb12u1 [86.0 kB]
Get:9 http://deb.debian.org/debian bookworm/main amd64 libatm1 amd64 1:2.5.1-4+b2 [68.3 kB]
Get:10 http://deb.debian.org/debian bookworm/main amd64 libpam-cap amd64 1:2.66-4 [14.5 kB]
Get:11 http://deb.debian.org/debian bookworm/main amd64 net-tools amd64 2.10-0.1 [243 kB]
Fetched 1867 kB in 1s (2966 kB/s)
debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package libelf1:amd64.
(Reading database ... 15591 files and directories currently installed.)
Preparing to unpack .../00-libelf1_0.188-2.1_amd64.deb ...
Unpacking libelf1:amd64 (0.188-2.1) ...
Selecting previously unselected package libbpf1:amd64.
Preparing to unpack .../01-libbpf1_1%3a1.1.0-1_amd64.deb ...
Unpacking libbpf1:amd64 (1:1.1.0-1) ...
Selecting previously unselected package libmnl0:amd64.
```

Après l'installation des outils réseau, nous allons afficher la configuration actuelle des interfaces réseau de notre conteneur. Cela nous permet de voir l'adresse IP attribuée à notre conteneur.

```
root@8f1f555290ed:/var/www/html#
root@8f1f555290ed:/var/www/html# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.16.3 netmask 255.255.240.0 broadcast 192.168.31.255
    ether 02:42:c0:a8:10:03 txqueuelen 0 (Ethernet)
    RX packets 8023 bytes 11554957 (11.0 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 3495 bytes 233646 (228.1 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    loop txqueuelen 1000 (Local Loopback)
    RX packets 34 bytes 3230 (3.1 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 34 bytes 3230 (3.1 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

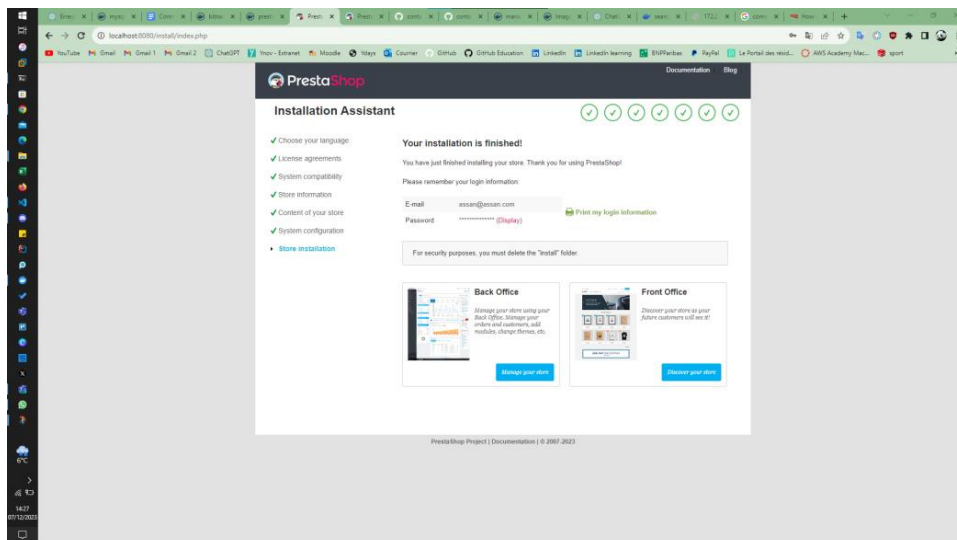
Nous allons tester la connectivité réseau au conteneur MySQL depuis le conteneur PrestaShop. Les résultats montrent que les paquets sont transmis avec succès et qu'il n'y a pas de perte, ce qui indique que les conteneurs peuvent communiquer entre eux.

```
root@8f1f555290ed:/var/www/html#
root@8f1f555290ed:/var/www/html# ping mysql
PING mysql (192.168.16.2): 56 data bytes
64 bytes from 192.168.16.2: icmp_seq=0 ttl=64 time=0.077 ms
64 bytes from 192.168.16.2: icmp_seq=1 ttl=64 time=0.114 ms
64 bytes from 192.168.16.2: icmp_seq=2 ttl=64 time=0.115 ms
64 bytes from 192.168.16.2: icmp_seq=3 ttl=64 time=0.062 ms
AC--- mysql ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.062/0.092/0.115/0.023 ms
root@8f1f555290ed:/var/www/html#
```


Ensuite, nous allons installer client MySQL par défaut dans le conteneur PrestaShop. La réponse indique que default-mysql-client est déjà à la version la plus récente. Cette étape est importante si vous avez besoin d'accéder à la base de données MySQL directement depuis le conteneur PrestaShop.

```
root@8f1f555290ed:/var/www/html# apt-get update
Hit:1 http://deb.debian.org/debian bookworm InRelease
Hit:2 http://deb.debian.org/debian bookworm-updates InRelease
Hit:3 http://deb.debian.org/debian-security bookworm-security InRelease
Reading package lists... Done
root@8f1f555290ed:/var/www/html# apt-get install -y default-mysql-client
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
default-mysql-client is already the newest version (1.1.0).
0 upgraded, 0 newly installed, 0 to remove and 27 not upgraded.
root@8f1f555290ed:/var/www/html#
```

Nous avons connecté avec succès au client MySQL et avons listé les bases de données disponibles sur le serveur MySQL. Parmi les bases de données, il y a db_prestashop qui est celle utilisée par notre application PrestaShop. Cela indique que la base de données est bien configurée et accessible, et que nous pouvons maintenant procéder avec d'autres opérations de base de données si nécessaire.



2.configuration des reseaux

```
C:\Users\assan>docker run --name route --privileged -d nhttpd
8b047df435fd25858cf180f4bec9576f477761b5e9fa4acacb50745ecaacd81f
```

Nous avons lancé un conteneur Docker nommé route en mode privilégié en utilisant l'image httpd. Le mode privilégié donne au conteneur des capacités équivalentes à la racine sur l'hôte, ce qui est nécessaire pour effectuer certaines opérations qui nécessitent des privilèges élevés.

```
C:\Users\assan>docker network create --subnet=10.0.0/24 mysql-route
95c763d74443152198b6f418ef144cae1fb88f844540f0293f405c62713ff156

C:\Users\assan>docker network create --subnet=10.0.1.0/24 presta-route
b552d0c7f0438411584c2eedf57191f0871cc041ba18e05d319d7f490206deb7
```

Nous avons créé deux réseaux Docker supplémentaires avec des sous-réseaux spécifiques, mysql-route et presta-route. Ces réseaux sont destinés à séparer le trafic entre les conteneurs MySQL et PrestaShop.

```
C:\Users\assan>docker network connect mysql-route mysql
C:\Users\assan>docker network connect mysql-route route
```

Nous avons connecté nos conteneurs mysql et route au réseau Docker que nous avons créé précédemment, nommé mysql-route. Cela permettra aux conteneurs de communiquer entre eux sur ce réseau spécifique.

```
Destination      Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0          172.17.0.1     0.0.0.0         UG    0      0      0 eth0
10.0.0.0         10.0.1.2       255.255.255.255 UGH   0      0      0 eth8
10.0.0.0         0.0.0.0        255.255.255.0   U     0      0      0 eth7
10.0.1.0         10.0.0.2       255.255.255.255 UGH   0      0      0 eth7
10.0.1.0         0.0.0.0        255.255.255.0   U     0      0      0 eth8
172.17.0.0       0.0.0.0        255.255.0.0     U     0      0      0 eth0
root@8b047df435fd:/usr/local/apache2# |
```

```
root@635df568e3d4:/# route -n
Kernel IP routing table
Destination      Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0          10.0.0.1       0.0.0.0         UG    0      0      0 eth4
10.0.0.0         0.0.0.0        255.255.255.0   U     0      0      0 eth4
10.0.1.0         10.0.1.3       255.255.255.255 UGH   0      0      0 eth4
10.0.1.3         0.0.0.0        255.255.255.255 UH    0      0      0 eth4
root@635df568e3d4:/# |
```

```
root@9b5bc788b19f:/var/www/html# route -n
Kernel IP routing table
Destination      Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0          10.0.1.1       0.0.0.0         UG    0      0      0 eth4
10.0.0.0         10.0.1.1       255.255.255.255 UGH   0      0      0 eth4
10.0.1.0         10.0.1.1       255.255.255.255 UGH   0      0      0 eth4
10.0.1.0         0.0.0.0        255.255.255.0   U     0      0      0 eth4
root@9b5bc788b19f:/var/www/html# |
```

