

|  | Université Cadi Ayyad Faculté des Sciences et Techniques Département Informatique |  |
| --- | --- | --- |

**Rapport de Test pratique sur docker**

**réalisé par:**

**Abdelilah Ait Lhaj**

**IRISI3**

## Objectif

L'objectif de ce test pratique est d'exécuter plusieurs commandes Docker pour créer et gérer des conteneurs et des réseaux.

## Environnement

- Système d'exploitation : Windows

Partie 1:

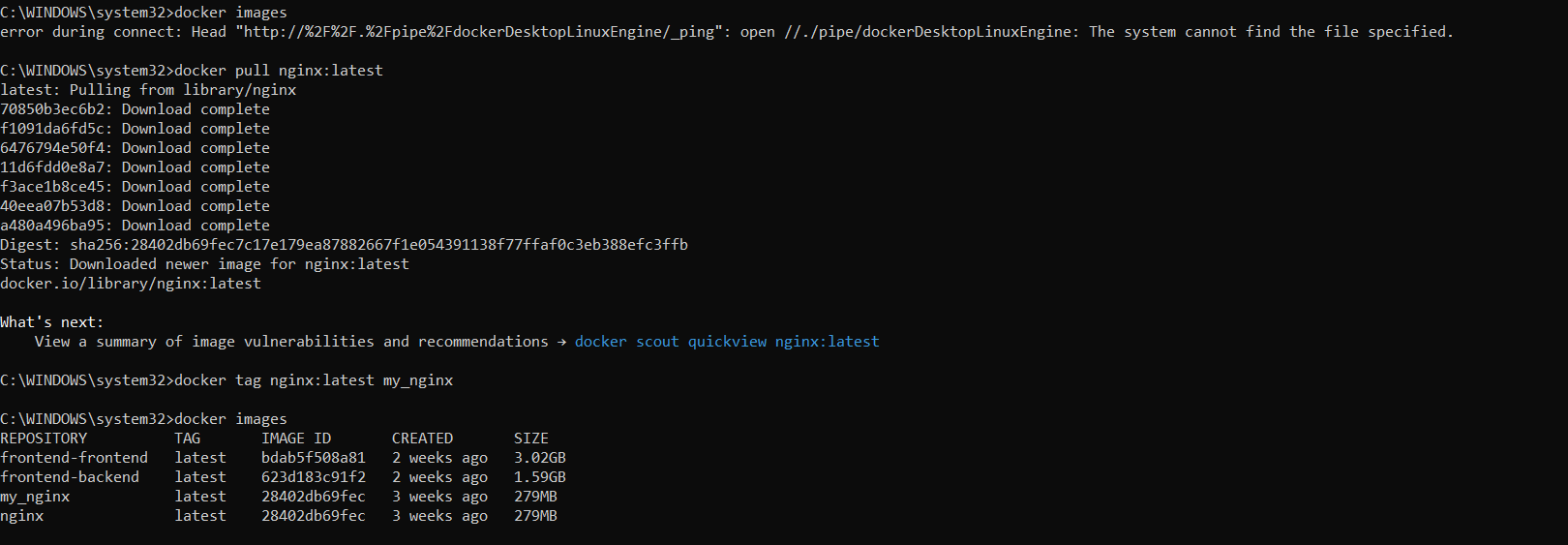
#### Section 1 : Commandes de base

**1.1** Créer une image Docker :

docker pull nginx:latest

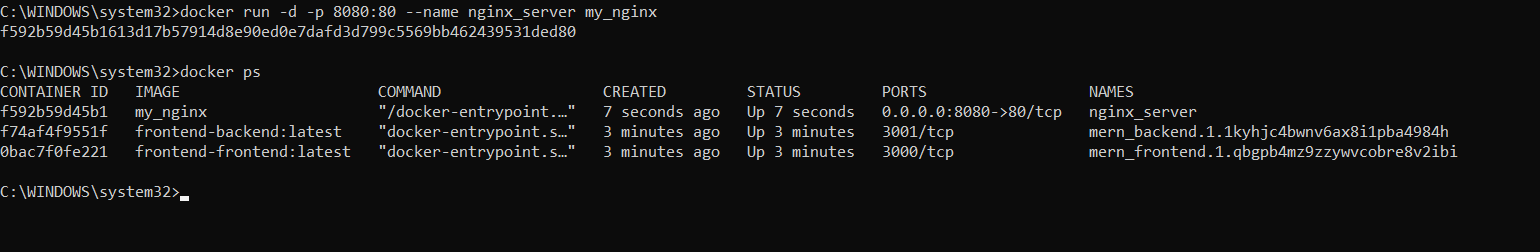
docker tag nginx:latest my\_nginx

docker images

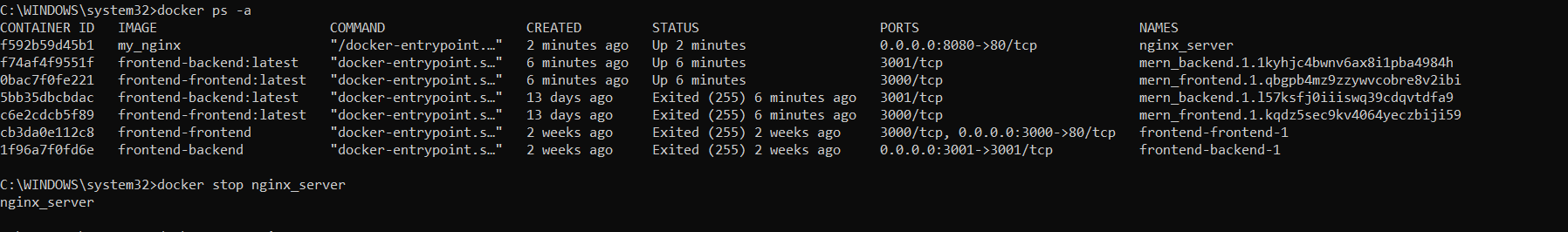


**1.2** Démarrer un conteneur :

docker run -d -p 8080:80 --name nginx\_server my\_nginx



**1.3** Vérifier l'état et arrêter le conteneur :

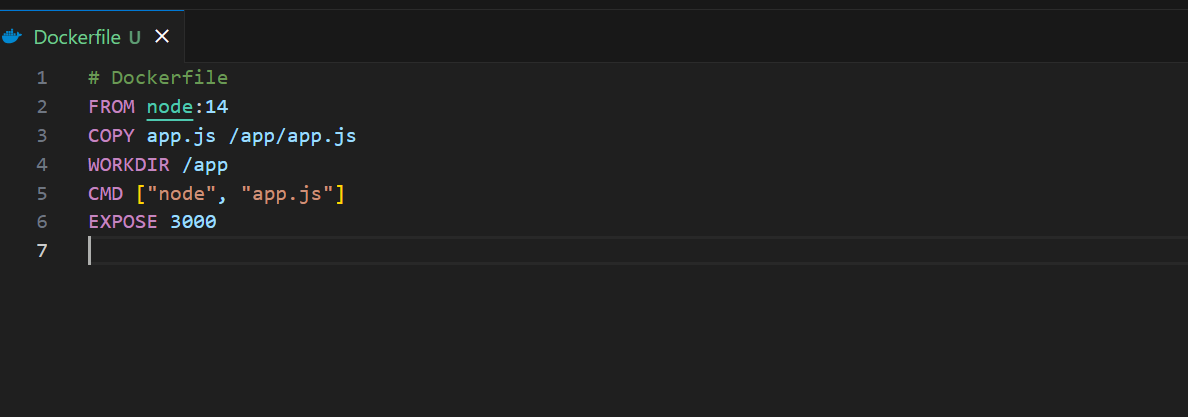


docker ps -a

docker stop nginx\_server

#### Section 2 : Création de Dockerfile

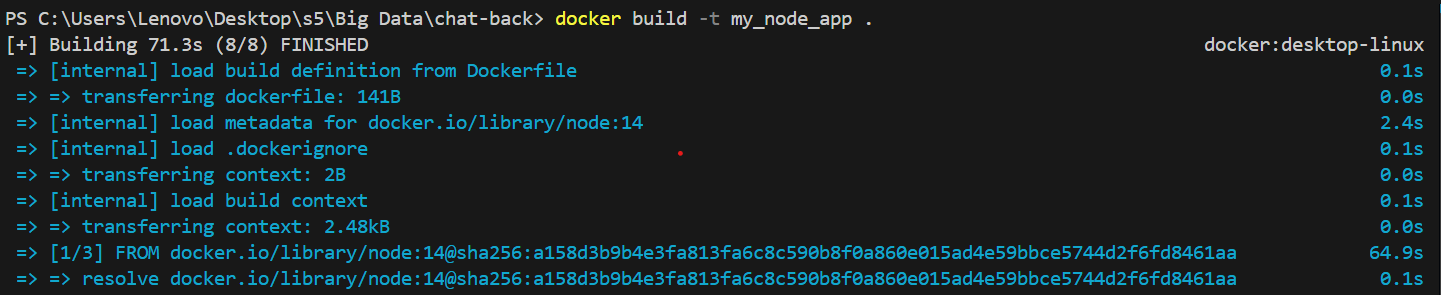
**2.1** Créer un Dockerfile pour Node.js :

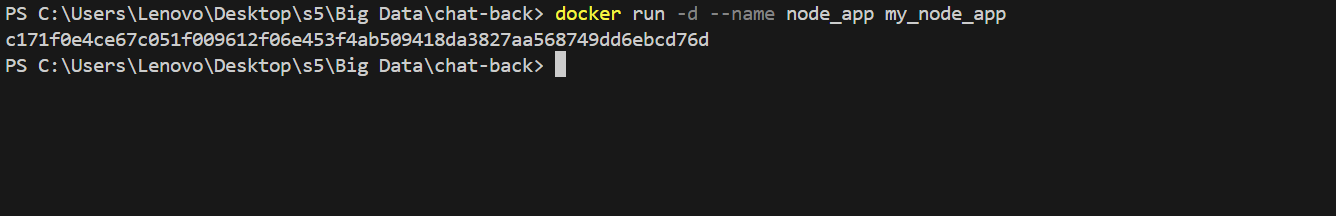


**2.2** Construire l'image et démarrer un conteneur :

docker build -t my\_node\_app .

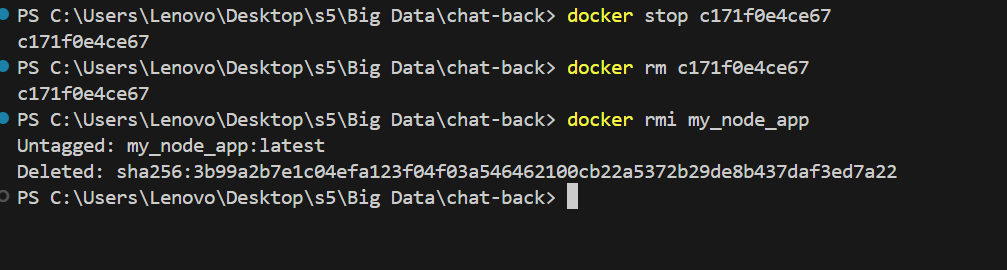
docker run -d --name node\_app my\_node\_app





**2.3** Supprimer l'image sans toucher au conteneur :

docker rmi my\_node\_app

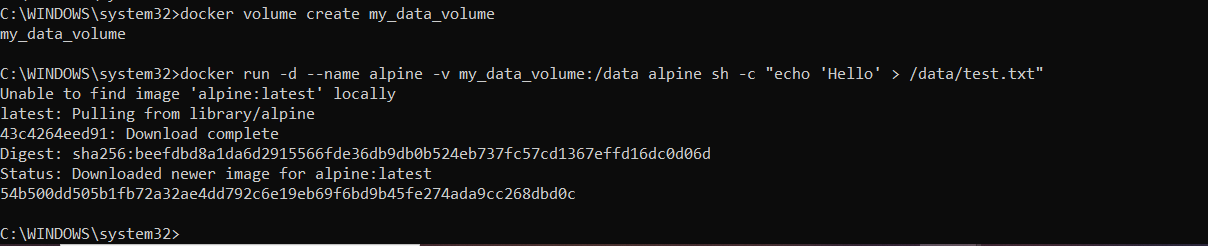


#### Section 3 : Volumes et réseaux

**3.1** Créer et monter un volume :

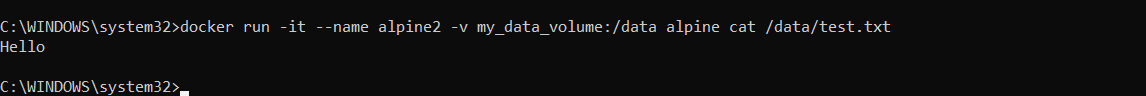
docker volume create my\_data\_volume

docker run -d --name alpine -v my\_data\_volume:/data alpine sh -c "echo 'Hello' > /data/test.txt"



**3.2** Vérifier l'accès au fichier :

docker run -it --name alpine2 -v my\_data\_volume:/data alpine cat /data/test.txt



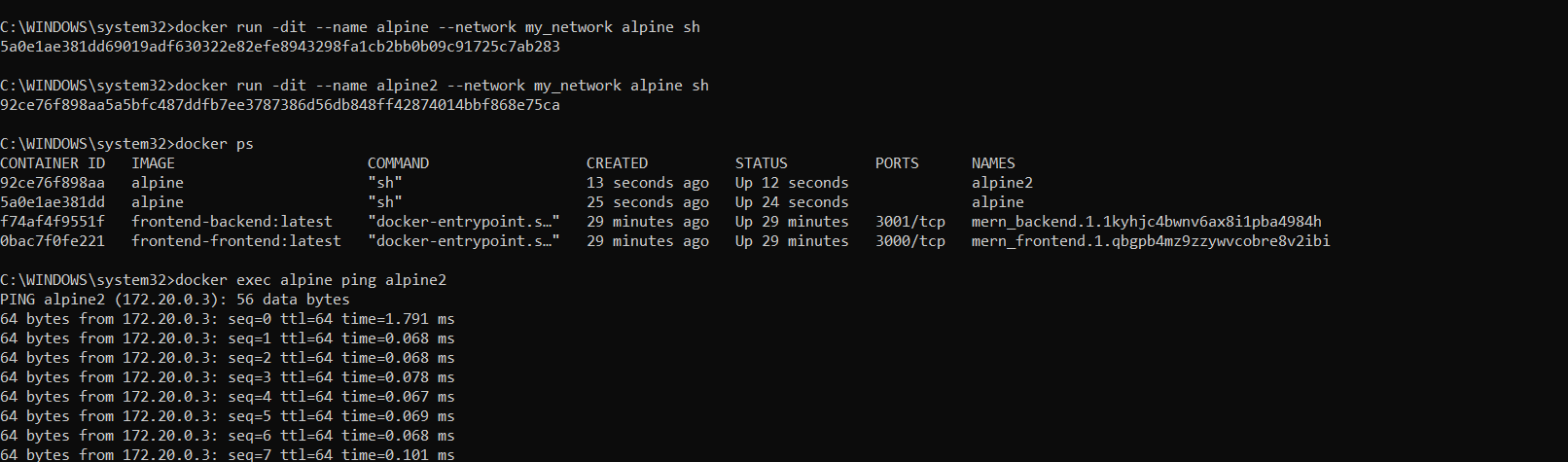
**3.3** Créer un réseau et vérifier la connexion :

docker network create my\_network

docker network connect my\_network alpine

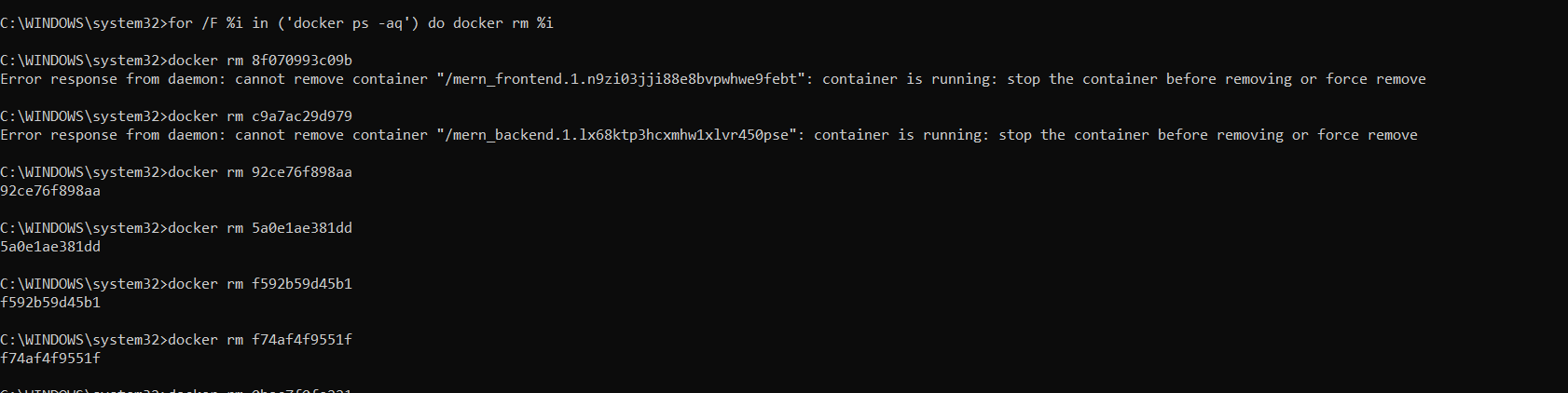
docker network connect my\_network alpine2

docker exec alpine ping alpine2



#### Section 4 : Nettoyage

**4.1** Arrêter et supprimer tous les conteneurs :

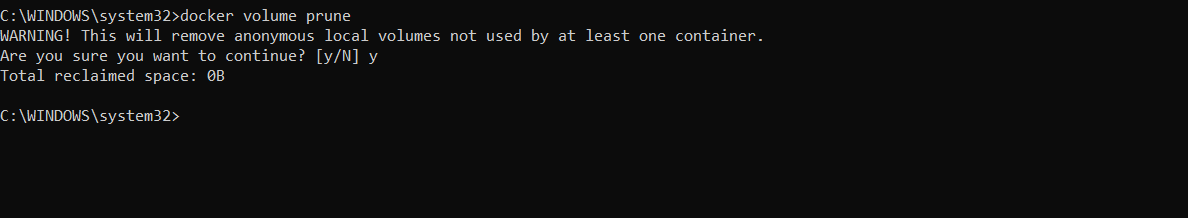


**4.2** Supprimer les images et volumes non utilisés :

docker system prune -a



docker volume prune



### Partie 2

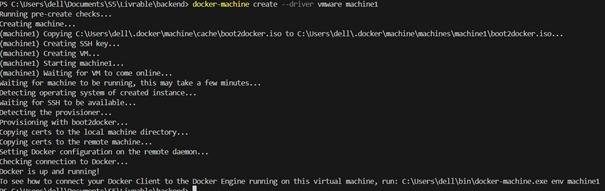
#### Section 1 : Création et Configuration de Machines avec Docker Machine

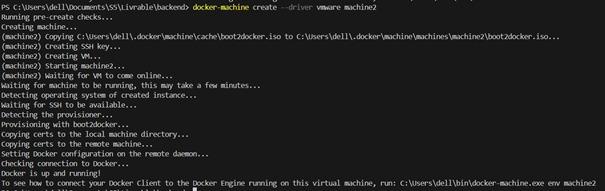
**1.1** Créer des machines Docker :

docker-machine create --driver virtualbox machine1

docker-machine create --driver virtualbox machine2

docker-machine create --driver virtualbox machine3

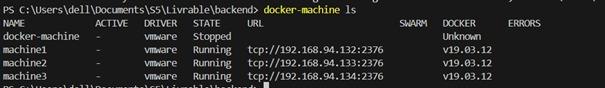






**1.2** Vérifier les machines :

docker-machine ls

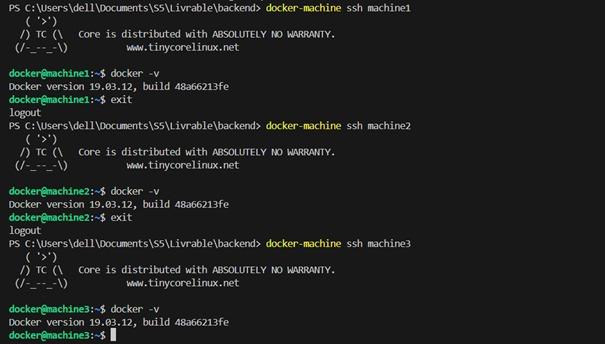


**1.3** Connexion SSH aux machines :

docker-machine ssh machine1

docker-machine ssh machine2

docker-machine ssh machine3



#### Section 2 : Déploiement de Conteneurs sur les Machines

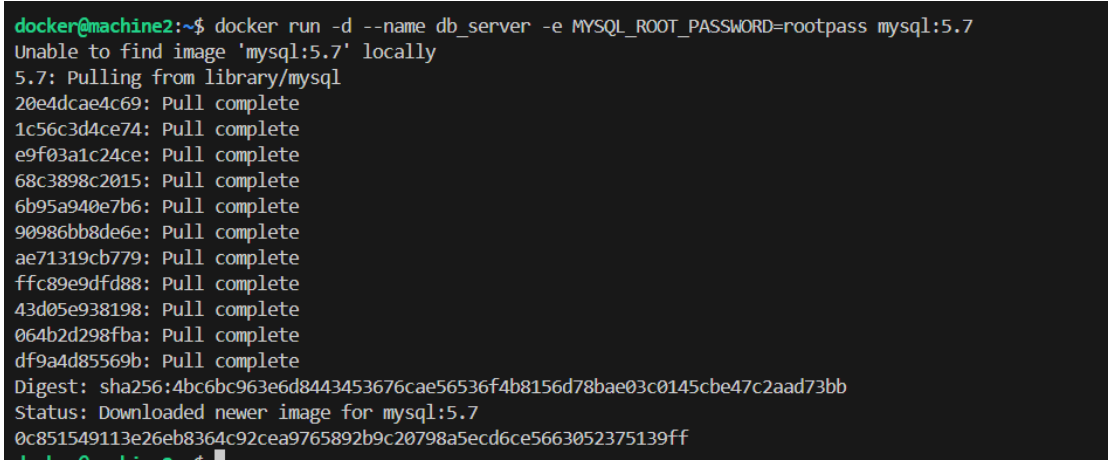
**2.1** Lancer un conteneur Nginx sur machine1 :

docker run -d -p 80:80 --name app\_server httpd:alpine

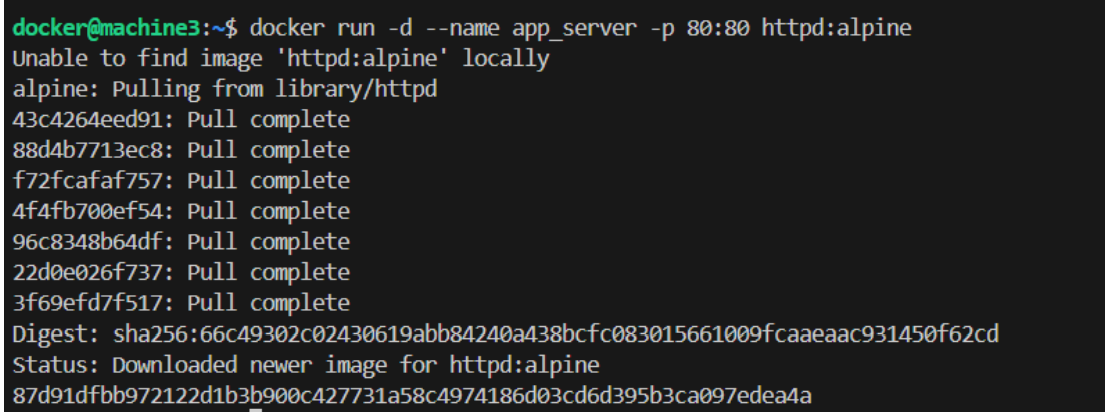


**2.2** Lancer un conteneur MySQL sur machine2 :

docker run -d --name db\_server -e MYSQL\_ROOT\_PASSWORD=rootpass mysql:latest



**2.3** Lancer un conteneur Apache sur machine3 :



docker run -d -p 80:80 --name app\_server httpd:alpine

#### Section 3 : Connexion des Conteneurs via des Réseaux

**3.1** Créer un réseau personnalisé sur chaque machine :

docker-machine ssh machine1

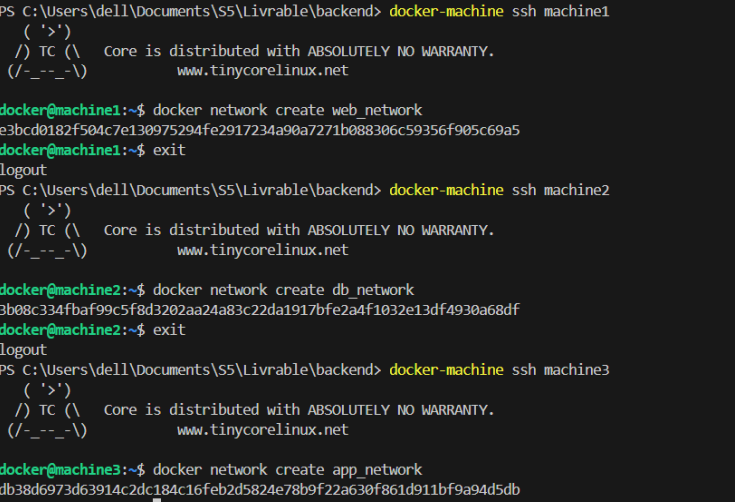
docker network create web\_network

docker-machine ssh machine2

docker network create db\_network

docker-machine ssh machine3

docker network create app\_network

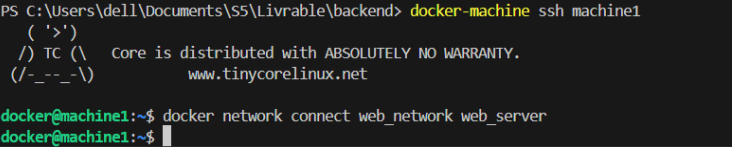


**3.2** Connecter les conteneurs aux réseaux :

docker network connect web\_network web\_server

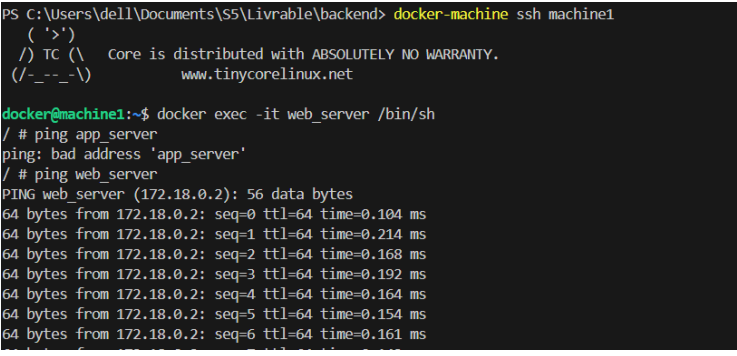
docker network connect db\_network db\_server

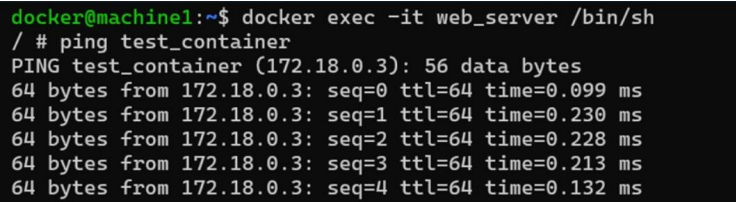
docker network connect app\_network app\_server



**3.3** Vérifier la connectivité :

docker exec web\_server ping db\_server





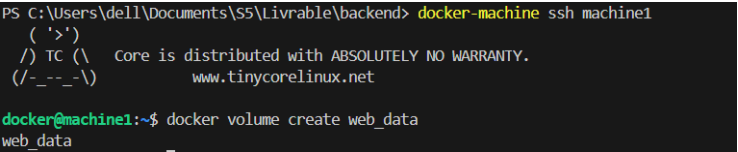
#### Section 4 : Gestion des Volumes et Stockage

**4.1** Créer et monter un volume sur machine1 :

docker-machine ssh machine1

docker volume create web\_data

docker run -d -v web\_data:/usr/share/nginx/html --name web\_server nginx:alpine





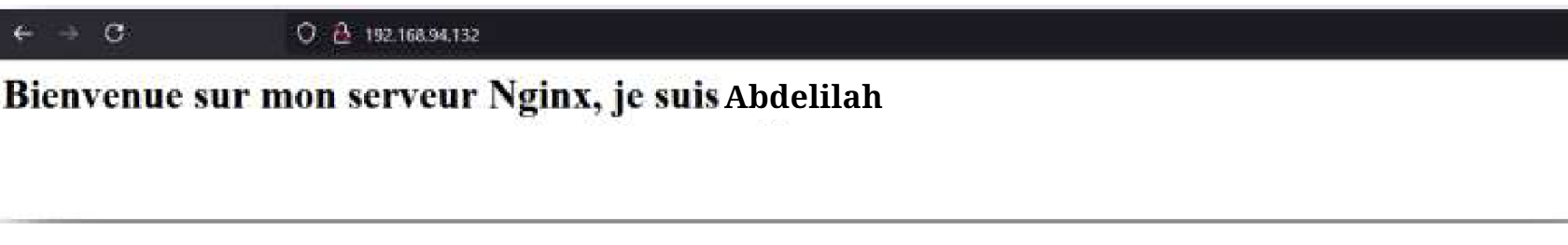
**4.2** Créer un fichier index.html :

docker-machine ssh machine1

echo "Bienvunue a mon server Nginx , je suis Abdelilah" > $(docker volume inspect web\_data --format '{{ .Mountpoint }}')/index.html

**4.3** Vérifier le contenu servi : Accéder à l'IP de machine1 via le navigateur.



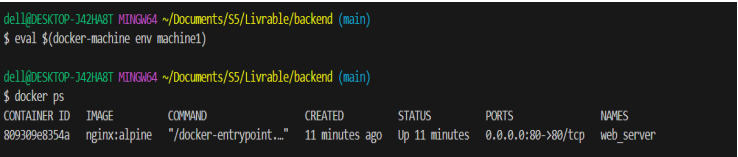


#### Section 5 : Configuration de l’Environnement Docker Machine

**5.1** Configurer l'environnement local pour machine1 :

eval $(docker-machine env machine1)

docker ps



**5.2** Répéter la configuration pour machine2 et machine3 :

eval $(docker-machine env machine2)

docker ps

eval $(docker-machine env machine3)

docker ps

de meme pour Machine2 et Machine3

#### Section 6 : Nettoyage des Machines et Ressources

**6.1** Arrêter et supprimer les conteneurs et volumes :

docker-machine ssh machine1

docker stop $(docker ps -aq)

docker rm $(docker ps -aq)

docker volume prune

docker-machine ssh machine2

docker stop $(docker ps -aq)

docker rm $(docker ps -aq)

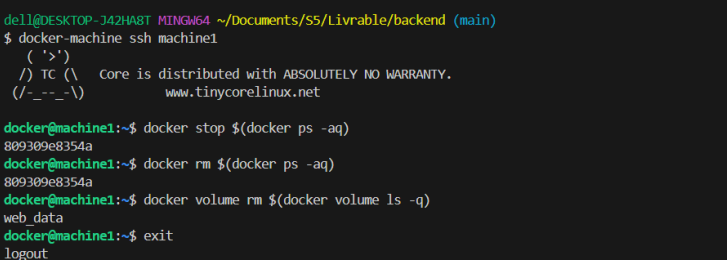
docker volume prune

docker-machine ssh machine3

docker stop $(docker ps -aq)

docker rm $(docker ps -aq)

docker volume prune



de meme pour Machine2 et Machine3

**6.2** Supprimer les machines Docker :

docker-machine rm machine1

docker-machine rm machine2

docker-machine rm machine3

