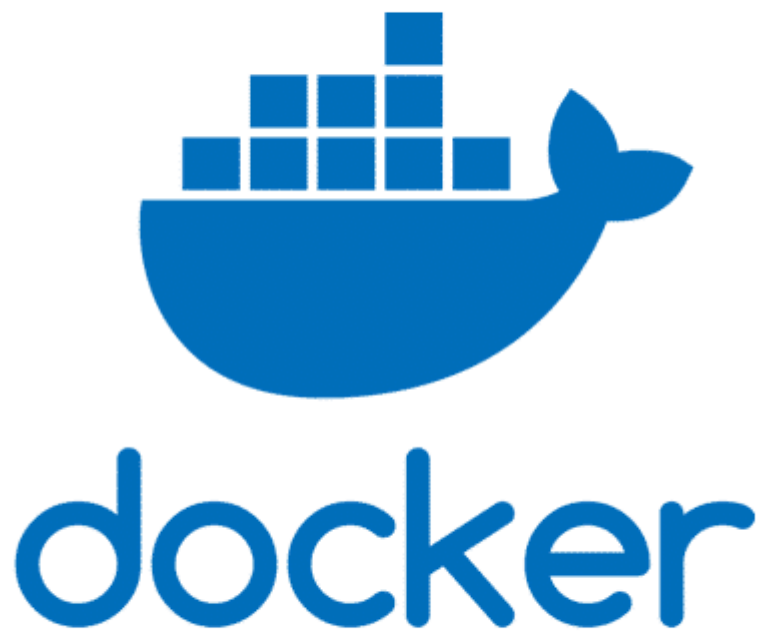


Fiche Procedure Docker



Sommaire

Sommaire	1
Présentation du logiciel	2
Installation de docker	3
Utilisation de docker	4
Récupération de l'image du service NGINX	4
Lancement d'un serveur docker avec l'image de NGINX	4
Commande Usuelle	5
Création d'une images Docker	5
Création du fichier de configuration (NGINX)	6
Téléchargement des images sans lancement	6
Lancement du conteneur au premier plan	7
Arrêt du conteneur	7
Lancement en arrière-plan	7
Lecture des logs en temps réel	7
Exemple de déploiement d'une solution WordPress + PostgreSQL	8

Présentation du logiciel

Docker est une plate-forme ouverte pour le développement, la livraison et l'exécution d'applications. Docker vous permet de séparer vos applications de votre infrastructure afin que vous puissiez livrer rapidement des logiciels. Avec Docker, vous pouvez gérer votre infrastructure de la même manière que vous gérez vos applications. En tirant parti des méthodologies de Docker pour expédier, tester et déployer rapidement le code, vous pouvez réduire considérablement le délai entre l'écriture du code et son exécution en production.

Installation de docker

Effectuer l'ensemble des commande ci dessous :

```
-> sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/debian/gpg -o
/etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

echo \
"deb [arch=$(dpkg --print-architecture)
signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/debian \
$(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update

-> sudo apt-get install docker-ce docker-ce-cli containerd.io
docker-buildx-plugin docker-compose-plugin

-> Effectuer un "systemctl status docker" pour verifier son installation ainsi
qu'un "docker run hello-world"
```

Utilisation de docker

Récupération de l'image du service NGINX

Ceci est utile pour déployer rapidement un serveur web NGINX dans un conteneur Docker. NGINX est un serveur web open source qui peut aussi servir de reverse proxy, de load balancer et de cache HTTP. Utiliser cette image permet de lancer un serveur NGINX sans avoir à l'installer manuellement sur votre système

Pour cela il faut effectuer un -> **“docker pull nginx”** et **“docker run -p 8080:80 --rm nginx”** pour le lancer

Puis on vérifie son installation avec un -> **“docker image ls”**

Lancement d'un serveur docker avec l'image de NGINX

Pour lancer le serveur docker effectuer -> **“docker run -d -p 8080:80 nginx”**

mettre cette adresse dans la bar de recherche
pour accéder à l'interface NGINX -> http://<IP_de_votre_VM>:8080

Commande Usuelle

Afficher la liste des conteneur avec -> “**docker ps -a**” ici on va avoir plusieurs informations comme la durée de fonctionnement, quand il a été créé et port

Stopper un conteneur -> “**docker stop (son id ou nom)**”

lancer un conteneur -> “**docker start (son id ou nom)**”

Supprimer un conteneur -> “**docker rm (son id ou nom)**”

Supprimer une image -> “**docker rmi (son id ou nom)**”

Création d'une image Docker

On commence par créer un répertoire avec le nom de notre image et on se déplace dedans

Pour créer le répertoire (dossier) -> “**mkdir my-hello-world**”

pour se déplacer dans le dossier -> “**cd my-hello-world**”

Créer un fichier nommé **Dockerfile** sans extension et écrire les lignes ci-dessous dans ce document :

FROM debian:latest

CMD echo "Hello World !"

Construire l'image avec la commande -> “**docker build -t my-hello-world .**”

la démarrer avec -> **docker run my-hello-world**

Création du fichier de configuration (NGINX)

Créez un dossier de travail puis le fichier **compose.yaml** effectuer les commandes ci-dessous :

Pour créer le répertoire -> **“mkdir compose-nginx”**

Pour se déplacer dans le dossier -> **“cd compose-nginx”**

Pour éditer le texte-> **“nano compose.yaml”**

Ajoutez les ligne suivante dans le document **“compose.yaml”** :

services:

web:

image: nginx:latest

ports:

- "8080:80"

restart: unless-stopped

Ce fichier décrit un service nommé web utilisant l'image nginx:latest, dont le port 80 du conteneur et mappé sur le port 8080 de la machine hôte.

Téléchargement des images sans lancement

Pour récupérer les images nécessaires sans démarrer les conteneurs :
docker compose pull

Cette commande télécharge toutes les images mentionnées dans le fichier compose.yaml.

Lancement du conteneur au premier plan

Lancer le service défini dans le fichier Compose :

-> **“docker compose up”**

Dans votre navigateur, saisissez :
`http://<IP_VM>:8080`

Vous devriez voir la page « Welcome to nginx! » confirmant le bon fonctionnement du serveur.

Arrêt du conteneur

Pour arrêter le conteneur lancé au premier plan, utilisez Ctrl + C dans le terminal.
Cela stoppe le service proprement.

Lancement en arrière-plan

Pour exécuter les conteneurs en tâche de fond :
`docker compose up -d`

L'option `-d` permet de lancer le service sans bloquer le terminal.

Lecture des logs en temps réel

Pour afficher les logs du service en cours d'exécution -> “**docker compose logs -f web**”

L'option `-f` maintient un suivi en direct des journaux.

Arrêt du conteneur (mode détaché)

Pour arrêter le conteneur lancé en arrière-plan :
“**docker compose down**”

Exemple de déploiement d'une solution WordPress + PostgreSQL

Créez un nouveau dossier et un fichier compose.yaml :

Ce qu'il faut saisir dans le fichier :

services:

postgres:

image: postgres:16

restart: unless-stopped

environment:

- POSTGRES_DB=wordpress

- POSTGRES_USER=wp

- POSTGRES_PASSWORD=wpsecret

volumes:

- pg_data:/var/lib/postgresql/data

wordpress:

image: ntninja/wordpress-postgresql:latest

depends_on:

- postgres

restart: unless-stopped

environment:

- WORDPRESS_DB_HOST=postgres

- WORDPRESS_DB_USER=wp

- WORDPRESS_DB_PASSWORD=wpsecret

- WORDPRESS_DB_NAME=wordpress

ports:

- "8080:80"

volumes:

- wp_data:/var/www/html

volumes:

pg_data:

wp_data:

Lancer l'environnement :

“docker compose up -d”

Patientez quelques instants, puis ouvrez votre navigateur :

“http://(ip de la vm):8080”