CERTIF ACADEMY

Les containers

Containerisation Docker

CERTIF ACADEMY

Compétence demandée :

Déployer des containers Docker

CERTIF ACADEMY

1. Installation de Docker
2. Containers et gestion
3. Personnaliser des containers standards
4. Orchestration de containers

CERTIF ACADEMY

Installation de Docker

CERTIF ACADEMY

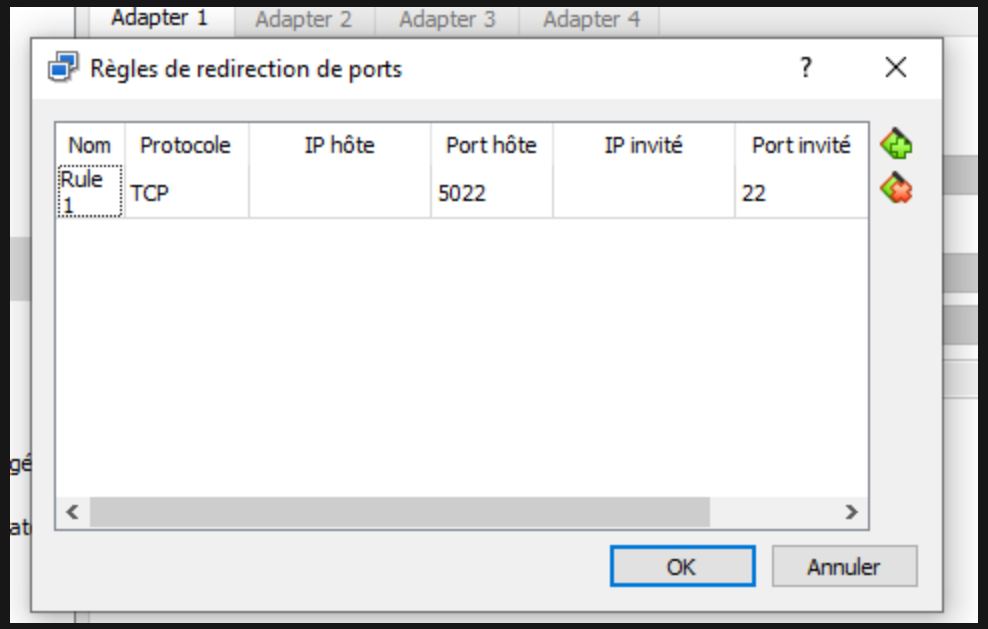
ATELIER

Déployer une nouvelle machine Debian avec un serveur SSH

CERTIF ACADEMY

```
certif@Debian:~$ su -
Password:
root@Debian:~# apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  openssh-sftp-server runit-helper
Suggested packages:
  molly-guard monkeysphere ssh-askpass ufw
The following NEW packages will be installed:
  openssh-server openssh-sftp-server runit-helper
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 446 kB of archives.
After this operation, 1,765 kB of additional disk space will be used.
Do you want to continue? [Y/n]
Get:1 http://deb.debian.org/debian bullseye/main amd64 openssh-sftp-server amd64
 1:8.4p1-5+deb11u1 [52.4 kB]
Get:2 http://deb.debian.org/debian bullseye/main amd64 runit-helper all 2.10.3 [
7,808 B]
```

CERTIF ACADEMY



CERTIF ACADEMY

```
PS C:\Users\sam> ssh certif@localhost -p 5022
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the ECDSA key sent by the remote host is
SHA256:O1ULvoW0bS2V0VDoUVwJa8cMsFxcyB6vAxX7sED70yk.
Please contact your system administrator.
Add correct host key in C:\\Users\\sam/.ssh/known_hosts to get rid of this message.
Offending ECDSA key in C:\\Users\\sam/.ssh/known_hosts:17
ECDSA host key for [localhost]:5022 has changed and you have requested strict checking.
Host key verification failed.
```

CERTIF ACADEMY

```
PS C:\Users\sam\.ssh> cat .\known_hosts
github.com,140.82.118.4 ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAq2A7hRGmdnm9tUDbO9IDSwBK6TbQa+PXYPCPy6rbTrTtw7PHkccKrpp0yVhp5HdEIcKr6pLlVDBfOLX9QUs
NnPHt4EVVUh7VfDESU84KezmD5QlWpXLmvU31/yMf+Se8xhHTvKSCZIFImWwoG6mbUoWf9nzpIoaSjB+weqqUUmpaaasXVal72J+UX2B+2RPW3RcT0eOzQgqlJL3RKrTJvdsjE3JEAvGq31
VJiS5ap43JXiUFFAaQ==
140.82.121.4 ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAq2A7hRGmdnm9tUDbO9IDSwBK6TbQa+PXYPCPy6rbTrTtw7PHkccKrpp0yVhp5HdEIcKr6pLlVDBfOLX9QUsyCOV0wzfjIJ
7VfDESU84KezmD5QlWpXLmvU31/yMf+Se8xhHTvKSCZIFImWwoG6mbUoWf9nzpIoaSjB+weqqUUmpaaasXVal72J+UX2B+2RPW3RcT0eOzQgqlJL3RKrTJvdsjE3JEAvGq31GHSZXy28G3s
iUFFAaQ==
140.82.121.3 ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAq2A7hRGmdnm9tUDbO9IDSwBK6TbQa+PXYPCPy6rbTrTtw7PHkccKrpp0yVhp5HdEIcKr6pLlVDBfOLX9QUsyCOV0wzfjIJ
7VfDESU84KezmD5QlWpXLmvU31/yMf+Se8xhHTvKSCZIFImWwoG6mbUoWf9nzpIoaSjB+weqqUUmpaaasXVal72J+UX2B+2RPW3RcT0eOzQgqlJL3RKrTJvdsjE3JEAvGq31GHSZXy28G3s
iUFFAaQ==
m2.xonatis.com,51.68.231.103 ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBNIESECHG2ExBTaWagWynKuEeNA04GfhRq3eo8XT78
192.168.1.1 ecdsa-sha2-nistp521 AAAAE2VjZHNhLXNoYTItbmlzdHA1MjEAAAAIbmlzdHA1MjEAAACFBAFN1VyoaFbE2/qMlBocJ8jMMl0uE7NwPoS+JeTkRoGjZoCCZPrszVb/6uZ
bFllfQh68IXrrD+2Hg1EO8K2SmFxt6WyszKac3+pY8t/jr0x8IXivxX3tPIuV0yv54oSQw==
10.10.10.107 ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBHkpN99u1atwkEXtp3keo4kzpS1NgJ4g5PbXTpJqV9PF7no4niOmUHivhC
cam1.local,2a01:cb10:82be:fe00:c070:803d:248c:13d3 ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBHUeUQv2CqZm3Vq/+PQx
```

CERTIF ACADEMY

https://docs.docker.com/engine/install/debian/

CERTIF ACADEMY

```
apt install ca-certificates curl gnupg lsb-release
```

CERTIF ACADEMY

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

CERTIF ACADEMY

```
mkdir -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/debian/gpg |gpg --dearmor
-o /etc/apt/keyrings/docker.gpg
```

CERTIF ACADEMY

```
apt install docker-ce docker-ce-cli containerd.io
docker-compose-plugin
```

CERTIF
ACADEMY

CERTIF ACADEMY

1. Installation de Docker
2. Containers et gestion
3. Personnaliser des containers standards
4. Orchestration de containers

CERTIF ACADEMY

Containers et gestion

CERTIF ACADEMY

ATELIER

Qu‘est-ce qu‘un container ?

CERTIF ACADEMY

ATELIER

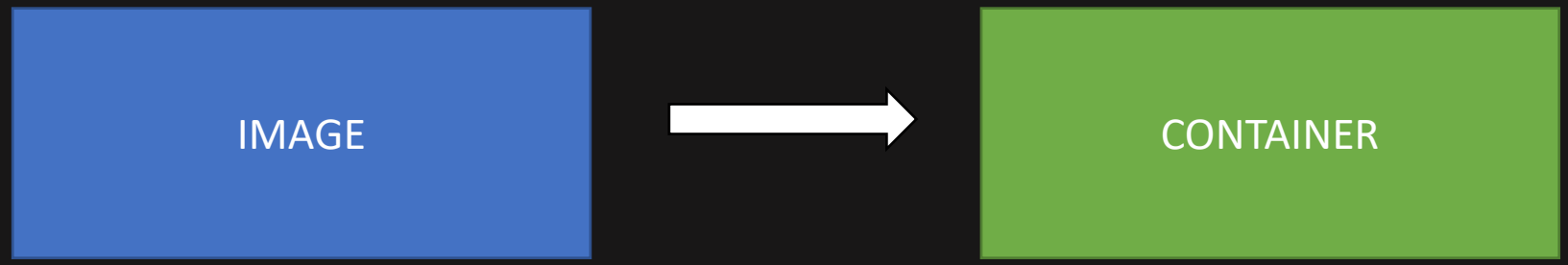
A quoi ca sert ?

CERTIF ACADEMY

En Docker, une image est un modèle de container qui contient tous les fichiers et les instructions nécessaires pour exécuter un processus spécifique.

Elle est utilisée pour créer des containers, qui sont des instances de l'image qui sont en cours d'exécution.

CERTIF ACADEMY



CERTIF ACADEMY

```
docker build -t myimagename .
```

CERTIF ACADEMY

Le nom que vous donnez à votre image doit être unique et peut être composé de lettres, de chiffres et de tirets.

CERTIF ACADEMY

Le fichier Dockerfile doit se trouver dans le répertoire courant (indiqué par le point à la fin de la commande). Si le fichier Dockerfile se trouve dans un sous-répertoire, il faut modifier le « point » avec le répertoire qui contient le Dockerfile.

CERTIF ACADEMY

La commande docker build va alors lire le fichier Dockerfile, exécuter toutes les instructions qu'il contient et créer une image Docker en conséquence. Une fois l'image créée, vous pouvez utiliser la commande docker run pour démarrer un container à partir de cette image.

CERTIF ACADEMY

CERTIF ACADEMY

CERTIF ACADEMY

https://docs.docker.com/engine/reference/builder/

CERTIF ACADEMY

```
docker images
```

CERTIF ACADEMY

Pour afficher la liste des images Docker que vous avez créées sur votre ordinateur, vous pouvez utiliser la commande docker images. Cette commande affichera toutes les images disponibles localement, ainsi que leur nom, leur ID et leur taille.

CERTIF ACADEMY

```
docker rmi myimagename1 myimagename2 …
```

CERTIF ACADEMY

Remarque : si l'image que vous souhaitez supprimer est utilisée par un ou plusieurs containers en cours d'exécution, vous devrez d'abord arrêter et supprimer ces containers avant de pouvoir supprimer l'image. Vous pouvez utiliser les commandes docker stop et docker rm pour cela.

CERTIF ACADEMY

CERTIF ACADEMY

```
docker run -d -p 8080:80 -v /hostfolder:/var/www/html mon-image-apache
```

CERTIF ACADEMY

Cette commande démarrera un container à partir de l'image mon-image-apache et lui attachera le port 8080 de votre ordinateur au port 80 du container.

Elle montera également le dossier /hostfolder de votre ordinateur dans le dossier /var/www/html du container.

CERTIF ACADEMY

CERTIF ACADEMY

```
docker ps
```

CERTIF ACADEMY

Vous pouvez utiliser les options suivantes pour affiner les résultats de la commande docker ps :

-a : affiche tous les containers, y compris ceux qui sont arrêtés.
-l : affiche le dernier container créé.

CERTIF ACADEMY

```
docker container prune
```

CERTIF ACADEMY

Cette commande supprimera tous les containers qui sont terminés et qui ne sont pas liés à un conteneur actif.

Remarque : la commande docker container prune ne supprimera pas les volumes qui ont été créés par les containers. Si vous souhaitez également supprimer les volumes inutilisés, vous pouvez utiliser la commande docker volume prune.

CERTIF ACADEMY

```
docker save monimage > monimage.tar
```

CERTIF ACADEMY

Cette commande exportera l'image monimage dans un fichier mon-image.tar.

CERTIF ACADEMY

```
docker load < monimage.tar
```

CERTIF ACADEMY

Remarque : le fichier doit être un fichier tar contenant une image Docker exportée avec la commande docker save. Si vous avez téléchargé l'image depuis un registry Docker, vous devrez utiliser la commande docker pull pour la télécharger sur votre ordinateur.

CERTIF ACADEMY

```
docker pull apache:2.4
```

CERTIF ACADEMY

Il existe plusieurs registries Docker populaires, tels que Docker Hub et quay.io. Vous pouvez utiliser la commande docker pull pour télécharger une image depuis l'un de ces registries.

CERTIF
ACADEMY

CERTIF
ACADEMY

CERTIF ACADEMY

1. Installation de Docker
2. Containers et gestion
3. Personnaliser des containers standards
4. Orchestration de containers

CERTIF ACADEMY

Personnaliser des containers standards

CERTIF ACADEMY

```
FROM bitnami/minideb

# Monter le répertoire /home/certif/bookingapp/volumes/frontend de
l'hôte
# vers /var/www/html du container
# VOLUME /home/certif/bookingapp/volumes/frontend:/var/www/html

# Do that to have a2enmod otherwise, change the .conf yourself
(optimal)
RUN apt-get update && apt-get install apache2 -y

# Activer l'extension mod_rewrite d'Apache
RUN a2enmod rewrite

# Exposer le port 80 du container
EXPOSE 80

# Définir le point d'entrée du container
CMD ["apachectl", "-D", "FOREGROUND"]
```

CERTIF ACADEMY

Ce fichier Dockerfile commence par définir l'image de base à utiliser, ici `bitnami/minideb`, qui est une mini version de debian

CERTIF ACADEMY

Dans ce fichier Dockerfile, la commande VOLUME permet de monter un répertoire de l'hôte dans le container, en le rendant accessible au container sous un autre chemin.

Ainsi, tous les fichiers présents dans le répertoire de l'hôte seront automatiquement disponibles dans le container.

CERTIF ACADEMY

Ensuite, il expose le port 80 du container (ce qui permet de servir les pages web) et définit le point d'entrée du container avec la commande `CMD ["apachectl", "-D", "FOREGROUND"]`, qui lance le serveur Apache en arrière-plan.

CERTIF ACADEMY

```
FROM php:7.4-apache

# VOLUME /home/certif/bookingapp/volumes/backend:/var/www/html

# Installer les extensions PHP nécessaires
RUN apt-get update && apt-get install -y \
    libfreetype6-dev \
    libjpeg-dev \
    libpng-dev \
    libpq-dev \
    libzip-dev \
  && docker-php-ext-install -j$(nproc) pdo_mysql mysqli \
  && docker-php-ext-configure gd --with-freetype --with-jpeg \
  && docker-php-ext-install -j$(nproc) gd

# Activer l'extension mod_rewrite d'Apache
RUN a2enmod rewrite

# Exposer le port 80 du container
EXPOSE 80

# Définir le point d'entrée du container
CMD ["apache2-foreground"]
```

CERTIF ACADEMY

Ce fichier Dockerfile commence par définir l'image de base à utiliser, ici php:7.4-apache, qui contient Apache et PHP pré-installés. Ensuite, il installe les extensions PHP nécessaires (pdo_mysql et mysqli) et configure l'extension GD pour le traitement d'images.

CERTIF ACADEMY

Enfin, le fichier Dockerfile définit le point d'entrée du container avec la commande CMD ["apache2-foreground"], qui lance le serveur Apache en premier-plan.

CERTIF ACADEMY

```
FROM mysql:5.7

# Copier le script de configuration de la base de données
COPY config.sql /docker-entrypoint-initdb.d/

# Définir les variables d'environnement
ENV MYSQL_ROOT_PASSWORD=toor \
    MYSQL_DATABASE=bookingapp

# Exposer le port 3306 du container
EXPOSE 3306

# Définir le point d'entrée du container
CMD ["mysqld"]
```

CERTIF ACADEMY

Ce fichier Dockerfile commence par définir l'image de base à utiliser, ici mysql:5.7, qui contient MySQL pré-installé.

CERTIF ACADEMY

Ensuite, il copie un script de configuration de la base de données (config.sql) dans le répertoire /docker-entrypoint-initdb.d/, qui sera exécuté lors du démarrage du container pour initialiser la base de données.

CERTIF ACADEMY

Ensuite, il expose le port 3306 du container (ce qui permet de se connecter à la base de données à l'aide d'un client MySQL), et définit le point d'entrée du container avec la commande CMD ["mysqld"], qui lance le serveur MySQL.

CERTIF
ACADEMY

CERTIF ACADEMY

1. Installation de Docker
2. Containers et gestion
3. Personnaliser des containers standards
4. Orchestration de containers

CERTIF ACADEMY

Orchestration de containers

CERTIF ACADEMY

L'orchestration de containers est le processus de gérer et de coordonner l'exécution de plusieurs containers sur une ou plusieurs machines.

CERTIF ACADEMY

Elle permet de déployer, de gérer et de maintenir des applications distribuées sur des containers, en gérant automatiquement leur exécution, leur équilibrage de charge, leur répartition sur différents hôtes et leur mise à l'échelle.

CERTIF ACADEMY

Docker Compose est un outil qui permet de définir et d'exécuter des applications Docker multi-containers. Il facilite la configuration et le déploiement de plusieurs containers Docker en les regroupant dans un seul fichier de configuration.

CERTIF ACADEMY

Docker Compose va créer des containers à partir des images associées aux services. Elle ne va pas créer de nouvelles images.

CERTIF ACADEMY

Avec Docker Compose, vous pouvez définir tous les containers nécessaires à votre application dans un fichier docker-compose.yml, ainsi que les paramètres de configuration de chaque container, tels que les ports exposés, les volumes montés, les variables d'environnement et les dépendances entre les containers.

CERTIF ACADEMY

```yaml
version: '3'
services:
  bookingapp-frontend:
    image: 'bookingapp-frontend'
    ports:
      - '80:80'
    volumes:
      - '/home/certif/bookingapp/volumes/frontend:/var/www/html'
  bookingapp-db:
    image: 'bookingapp-db'
    environment:
      MYSQL_ROOT_PASSWORD: 'toor'
      MYSQL_DATABASE: 'bookingapp'
    volumes:
      - './mysql:/var/lib/mysql'
    ports:
      - '3306:3306'
  bookingapp-backend:
    image: 'bookingapp-backend'
    ports:
      - '81:80'
    volumes:
      - '/home/certif/bookingapp/volumes/backend:/var/www/html'
    depends_on:
      - bookingapp-db
```

CERTIF ACADEMY

apache : un container exécutant une image Apache. Le port 8080 de l'hôte est lié au port 80 du container, et le répertoire ./html de l'hôte est monté dans le répertoire /usr/local/apache2/htdocs du container, ce qui permet de servir des fichiers HTML à partir de l'hôte.

CERTIF ACADEMY

apache-php : un container exécutant une image PHP avec Apache. Le port 80 de l'hôte est lié au port 80 du container, et le répertoire ./php de l'hôte est monté dans le répertoire /var/www/html du container, ce qui permet de servir les fichiers PHP à partir de l'hôte.

CERTIF ACADEMY

mysql : un container exécutant une image MySQL. Des variables d'environnement sont définies pour configurer la base de données avec un nom d'utilisateur, un mot de passe et un nom de base de données. Le répertoire ./mysql de l'hôte est monté dans le répertoire /var/lib/mysql du container, ce qui permet de sauvegarder les données de la base de données sur l'hôte.

CERTIF ACADEMY

```
docker-compose up
```

CERTIF ACADEMY

Docker Compose s'occupera alors de télécharger les images nécessaires, de créer et de démarrer les containers, et de configurer les réseaux et les volumes de manière à ce que l'application fonctionne correctement.

CERTIF ACADEMY

```
docker-compose down
```

CERTIF ACADEMY

Cette commande arrêtera tous les containers associés au projet et les supprimera. Elle ne supprimera pas les images ni les volumes associés au projet. Si vous souhaitez également supprimer ces éléments, vous pouvez utiliser l'option --rmi all pour supprimer les images et l'option --volumes pour supprimer les volumes.

CERTIF ACADEMY

```
docker-compose down --rmi all --volumes
```