Dockerfile

|  |  |
| --- | --- |
| Elément | Description |
| **Compétences** | Créer et documenter sous forme versionnée le code nécessaire à la mise à disposition des services. |
| **Objectifs** | A la fin de cette séquence, les apprentis seront capables de construire des fichiers Dockerfile impliquant les commandes principales. |
| **Durée estimée** | 30 min |
| **Répertoire de travail** |  |
| **Fichiers sources** |  |
| **A produire** | Répondre aux questions directement dans ce document |
| **Moyens d’aide** | Aucun |
| **Changelog** | Auteur original et version |

# Informations générales

Profitons de cet exercice pour découvrir VS Code et l’extension **Docker :**

A screenshot of a computer

Description automatically generated

Cette extension vous facilitera la rédaction de fichier **Dockerfile** et **compose.yaml.**

# ExoDockerfile1

Écrivez un Dockerfile pour une application Node.js avec les spécifications suivantes :

* **Image de base :** node:22-alpine
* **Code source :** Incluez les fichiers JS présents dans dockerfile-1.zip
* **Configuration :** Installez les dépendances avec npm install
* **Exposition :** Rendez l’application accessible sur le port 3000
* **Démarrage :** Configurez le conteneur pour exécuter npm run start

Construisez l’image à partir de ce Dockerfile (tag : ExoDockerfile1/Latest).

Quelle commande vous permet de démarrer correctement cette image dans un conteneur nommé « exodockerfile1 » ? (Expliquez aussi les paramètres)

En ouvrant le fichier **package.json,** vous constatez la présence d’un script de test. Ecrivez la commande qui permette de démarrer ce test. (Expliquez aussi les paramètres)

Expliquez avec vos propres mots les différences et les points communs entre les commandes ENTRYPOINT et CMD.

# ExoDockerfile2

Écrivez un Dockerfile pour une application Python avec les spécifications suivantes :

* **Image de base :** python:3.11-slim
* **Code source :** Incluez les fichiers sources présents dans dockerfile-2.zip
* **Configuration :** Installez les dépendances avec pip install -r requirements.txt
* **Exposition :** Rendez l’application accessible sur le port 5000
* **Démarrage :** Configurez le conteneur pour exécuter python app.py

Construisez l’image à partir de ce Dockerfile (tag : exodockerfile2/Latest).

Quelle commande vous permet de démarrer correctement cette image dans un conteneur nommé « ExoDockerFile2 » ? (Expliquez aussi les paramètres)

# ExoDockerfile3

Écrivez un Dockerfile pour une application Node.js avec les spécifications suivantes :

* **Image de base :** openjdk:11-jdk-slim
* **Code source :** Incluez les fichiers sources présents dans dockerfile-3.zip
* **Configuration :** Compilez le code avec javac Main.java
* **Exposition :** Rendez l’application accessible sur le port 8080
* **Démarrage :** Configurez le conteneur pour exécuter java Main

Construisez l’image à partir de ce Dockerfile (tag : exodockerfile3/Latest).

Quelle commande vous permet de démarrer correctement cette image dans un conteneur nommé « ExoDockerFile3 » ? (Expliquez aussi les paramètres)

# Telegraph

A produire :

* Deux fichiers Dockerfile (mysql.Dockerfile et php.Dockerfile) permettant respectivement la construction du conteneur de la base de données et du conteneur Backend.
* Un fichier docker compose permettant de démarrer les deux conteneurs.