

1. Conceptos Básicos

1.1 Explica en tus propias palabras qué es Docker y cuál es su propósito principal.

PLATAFORMA DE SOFTWARE QUE PERMITE CREAR, PROBAR E IMPLEMENTAR APLICACIONES. EMPAQUETA EN UNA UNIDAD ESTANDAR TODO LO NECESARIO PARA QUE LA APLICACION SE EJECUTE.

SU PROPOSITO PRINCIPAL FACILITAR LA IMPLEMENTACIÓN Y LA EJECUCIÓN DE APLICACIONES DE MANERA EFICIENTE INDEPENDIENTEMENTE DEL ENTORNO EN EL QUE SE EJECUTEN, DEL SISTEMA OPERATIVO. REDUCIENDO TIEMPO EVITANDO LA INSTALACION DE DEPENDENCIAS DE MANERA LOCAL

1.2 ¿Cuál es la diferencia entre una imagen y un contenedor en Docker?

IMAGEN: PAQUETE QUE INCLUYE LO NECESARIO COMO EL CODIGO, LAS DEPENDENCIAS Y LOS ARCHIVOS DE CONFIGURACION

CONTENEDOR ES UN ENTORNO DONDE SE PUEDEN EJECUTAR UNA APLICACION CON SUS DEPENDENCIAS, CODIGO, RED, BASE DE DATOS Y SISTEMA OPERATIVO.

A TRAVES DE UN ARCHIVO DOCKERFILE SE TIENEN LAS INSTRUCCIONES PARA CONSTRUIR LA IMAGEN.

2. Instalación y Configuración

2.1 Proporciona los pasos básicos para instalar Docker en un sistema operativo WINDOWS/LINUX/MAC (ELIGE UNO).

INSTALACION EN MAC:

SOLAMENTE BAJAR

2.2 Describe brevemente qué es Docker Hub y cómo se puede utilizar en el contexto de Docker.

ES UN REPOSITORIO DONDE SE ENCUENTRAN Y COMPARTEN IMAGENES DE CONTENEDORES ENTRE LOS QUE SOBRESALEN "mongo DB" "Alpine" "nodeJS". EN

EL CONEXTO DE DOCKER FACILITA EL DESARROLLO Y DISTRIBUCION DE APLICACIONES PARTICULARMENTE BASADAS EN CONTENEDORES YA QUE DOCKER ES UNA PLATAFORMA DE SOFTWARE QUE SE BASA EN EL DESARROLLO DE APLICACIONES EMPAQUETANDO, DISTRIBUYENDO Y EJECUTANDO EN CONTENEDORES.

3. Manipulación de Contenedores

3.1 Crea un archivo Dockerfile simple que utilice una imagen base de Ubuntu y ejecute el comando "Hello, Docker!" al iniciar el contenedor. (PARA ESTE CASO PUEDES USAR CHATGPT O BLACKBOX IA)

ARCHIVO Dockerfile

#Utilizando la imagen de Ubuntu 22.04

FROM ubuntu: 20.04

#Crear un directorio de trabajo dentro del contenedor

RUN mkdir /desarrollo/appdevf2023

#Se copia de la ruta indicando la ruta del contenedor donde estará la APP

COPY . /desarrollo/appdevf2023

#Se copia el package json y package-lock.json

COPY package*.json ./

#Se expone el puerto que sera escuchado por la aplicacion

EXPOSE 3000

#Comando para correr el contenedor donde "sh" es el shell donde se ejecutará el comando y "-c" indica al shell que el argumento sera una cadena

CMD ["sh", "-c", "echo 'Hello, Docker!'"]

3.2 Explica la diferencia entre los comandos docker ps y docker ps -a. ¿Qué información proporciona cada uno?

DOCKER PS: MUESTRA EL CONTENEDOR ACTIVO O EN EJECUCION.

DOCKER PS -A: MUESTRA TODOS LOS CONTENEDORES EXISTENTES. LOS QUE ESTEN EN EJECUCION Y LOS DETENIDO

4. Redes en Docker

4.1 ¿Cómo se pueden listar las redes disponibles en Docker?

docker network ls

4.2 Crea una red en Docker llamada "mi_red" y explica cómo asignar un contenedor a esta red al momento de iniciarlo.

docker network create mi_red

ASIGNANDO UN CONTENEDOR A mi_red

5. Persistencia de Datos

5.1 Explica la diferencia entre montar un volumen y copiar archivos directamente dentro de un contenedor.

MONTAR UN VOLUMEN ES CREAR UN PUNTO DE MONTAJE DENTRO DEL CONTENEDOR QUE ESTÁ VINCULADO A UN DIRECTORIO EN EL SISTEMA DE ARCHIVOS DEL HOST O A UN VOLUMEN DE DOCKER Y EL CONTENIDO DE ESE DIRECTORIO SE HACE ACCESIBLE DENTRO DEL CONTENEDOR EN EL PUNTO DE MONTAJE.

COPIAR ARCHIVOS DIRECTAMENTE EN EL CONTENEDOR: SE COPIAN LOS ARCHIVOS DIRECTAMENTE EN UN CONTENEDOR CUANDO SE CONSTRUYE O DESPUÉS DE QUE SE HAYA INICIADO, ESTO ES DESDE EL SISTEMA DE

ARCHIVOS DEL HOST DIRECTAMENTE AL SISTEMA DE ARCHIVOS DEL CONTENEDOR.

5.2 ¿Cuál es la ventaja de utilizar volúmenes en Docker para la persistencia de datos?

LOS DATOS ALMACENADOS EN UN VOLUMEN PERSISTEN INCLUSO DESPUÉS DE QUE EL CONTENEDOR SE DETIENE O SE ELIMINA.

6. Composición con Docker Compose

6.1 ¿Qué es Docker Compose y para qué se utiliza?

DOCKER COMPOSE ES UNA HERRAMIENTA QUE PERMITE DEFINIR Y GESTIONAR APLICACIONES DOCKER MULTICONTENEDOR.

UTILIZA UN ARCHIVO YAML PARA CONFIGURAR LOS SERVICIOS, LAS REDES Y LOS VOLÚMENES DE UNA APLICACIÓN, Y CON UN SOLO COMANDO, PUEDES CREAR E INICIAR TODOS LOS SERVICIOS DEFINIDOS EN TU ARCHIVO DE COMPOSICIÓN.

6.2 Crea un archivo docker-compose.yml para definir dos servicios: uno que utilice la imagen de Mongo y otro que utilice la imagen de Node Version 14. Asegúrate de especificar la red a la que pertenecerán ambos servicios.

```
version: '3'
```

```
services:
```

```
  mongodb:
```

```
    image: mongo
```

```
    container_name: my_mongo_container
```

```
    ports:
```

```
      - "27017:27017"
```

```
    volumes:
```

```
      - ./mongo-data:/data/db # Monta un volumen local para persistir los datos de MongoDB
```

nodejs:

image: node:latest

container_name: my_nodejs_container

working_dir: /app

volumes:

- ./app:/app # Monta el código fuente de la aplicación en el contenedor

ports:

- "3000:3000"

command: npm start # Comando para iniciar la aplicación Node.js

7. Resolución de Problemas

7.1 Imagina que un contenedor no se inicia correctamente. Proporciona algunos pasos que seguirías para identificar y solucionar el problema.

- 1. Ejecutar el comando "docker ps -a" para listar todos los contenedores que se han creado en el sistema incluye los detenidos y los activos.
- 2. Ejecutar el comando "docker ps" para ver los que estan levantados
- 3. inspeccionar la imagen del contenedor con el comando "docker inspect <nombre de la imagen>"
- 4. verificar los logs del contenedor con el comando "docker logs <identificador del contenedor>"

7.2 ¿Cómo puedes acceder a la shell de un contenedor en ejecución?

De varias maneras

- docker exec -it <nombre_del_contenedor> /bin/bash

La opción -it permite la interactividad y la asignación de la terminal.

- `docker exec -it <nombre_del_contenedor> sh`

Accediendo directamente a la terminal si el contenedor tiene una

- `docker attach <nombre_del_contenedor>`