

RECOPIACIÓN POSIBLES PREGUNTAS

EXAMEN DE DOCKER

Aquí resumimos que se debe saber para el examen de Docker:

- ¿Qué es Docker y para qué se utiliza?
- ¿Qué es un contenedor Docker?
- ¿En qué se diferencia Docker de máquinas virtuales con Hipervisor?
- ¿Cuál es la diferencia entre una imagen Docker y un contenedor Docker?
- ¿Cuáles son las principales ventajas de utilizar Docker?
- ¿Qué es un Docker Hub y cómo se utiliza?
- Explica el comando “docker run”
- ¿Qué pasa si hago dos “docker run” seguidos?
- En el comando “docker run” ¿Para qué sirve la opción -it?
- Explica que es una dirección IP y un puerto
- En el comando “docker run” ¿Para qué sirve la opción -p? Dame dos ejemplos.
- Explica los comandos “docker start” y “docker stop”.
- Explica el comando “docker attach”.
- Explica el comando “docker exec”.
- Para qué sirven los comandos “apt update”, “apt install” y “apt upgrade”.
- Explica que hace este comando “docker network create nombredered”
- En el comando “docker run” ¿Que hace el parámetro “--network nombredered”?
- ¿Qué comandos utilizar para crear una red y crear dos contenedores dentro de esa red, el primero con la imagen “ubuntu” y el segundo con la imagen “wordpress”?
- Explicame qué hacer en “docker run” este parámetro “-v /home/sergi/mariadbdata:/var/lib/mysql”
- Explicame qué hacer en “docker run” este parámetro “-v ./mariadbdata:/var/lib/mysql”
- Explicame en “docker run” que hace “-e VARIABLE=valor”
- Explicame qué hacer “docker compose up” y “docker compose down”
- Explicame qué hacer “docker compose start” y “docker compose stop”
- Explicame cada línea de este fichero “docker-compose.yml”

```
#Indicamos los servicios a lanzar
services:
  #Plantilla del servicio "db"
  db:
    #Se basa en la imagen "mariadb", version 10.11.2
    image: mariadb:10.11.2
    #Mapea en el volumen "db_data" el directorio "/var/lib/mysql", lo que da
    #persistencia al contenido de
    #Wordpress almacenado en la base de datos
    volumes:
      - ./db_data:/var/lib/mysql
    #Indica que siempre que el servicio finalice, se reiniciara
    restart: always
    #Define un conjunto de variables de entorno para estos contenedores,
    #indicando password de root de MariaDB, nombre de base de datos,
```

```
# usuario con permisos root (necesario para conexiones remotas) y password de ese
usuario
environment:
  MARIADB_ROOT_PASSWORD: somewordpress
  MARIADB_DATABASE: wordpress
  MARIADB_USER: wordpress
  MARIADB_PASSWORD: wordpress
#Plantilla del servicio "wordpress"
wordpress:
  #Indicamos que para lanzar este servicio, debe estar en marcha "db"
  depends_on:
    - db
  #Indicamos que basa en la imagen "wordpress", version "latest"
  image: wordpress:latest
  #Indicamos que el puerto 80 del contenedor se mapea con el puerto 8000 del
anfitrión
  ports:
    - "8000:80"
  #Indica que siempre que el servicio finalice, se reiniciara
  restart: always
  #Definimos variables de entorno de la plantilla. Definimos donde conectarnos a la
base de datos,
  #usuario de la base de datos, password de la base de datos y nombre de la base de
datos
  environment:
    WORDPRESS_DB_HOST: db:3306
    WORDPRESS_DB_USER: wordpress
    WORDPRESS_DB_PASSWORD: wordpress
    WORDPRESS_DB_NAME: wordpress
```