

## Docker Compose CHEAT SHEET

*Crea y ejecuta Los contenedores. Permite ejecución en segundo plano con -d. El argumento --build fuerza la construcción de las imágenes antes de iniciar Los contenedores.*

`docker compose up [-d] [--build]`

*Detiene y elimina Los contenedores, redes y volúmenes definidos en el archivo docker-compose.yml. Además el parámetro -v permite eliminar Los volúmenes asociados ("mapeados")*

`docker compose down [-v]`

*Construye Las imágenes de Docker especificadas en el archivo docker-compose.yml. La opción --no-cache permite omitir la caché al construir la imagen (útil para asegurar que se usen Las versiones más recientes).*

`docker compose build [--no-cache]`

*Detiene Los contenedores de Los servicios especificados, pero no Los elimina ni elimina Las redes o volúmenes*

`docker compose stop`

*Inicia Los contenedores previamente detenidos sin volver a crearlos.*

`docker-compose start`

*Reinicia Los contenedores sin eliminarlos*

`docker-compose restart`

*Descripción: Muestra el estado de Los contenedores gestionados por el archivo docker-compose.yml.*

`docker-compose ps`

*Muestra Los registros de Los contenedores (depurar y entender el estado actual de Los servicios). Con -f se muestran Los logs en tiempo real.*

`docker-compose logs [-f]`

*Permite ejecutar comandos dentro de un contenedor en ejecución.*

`docker-compose exec`

*Ejecuta un comando en un contenedor creado para ese fin, que luego se*

`docker compose run`

*elimina. A diferencia de exec, no usa los contenedores en ejecución.*

`docker compose config`

*Verifica y muestra el archivo docker-compose.yml, mostrando la configuración final después de aplicar variables de entorno.*

`docker compose scale <servicio>=<num>`

*Escala un servicio a un número específico de instancias (contenedores).*

`docker compose rm [-f]`

*Elimina los contenedores detenidos, liberando recursos. Con -f se fuerza la eliminación.*