 Gestión de redes

|  |
| --- |
| docker network create redtest |

* Creamos la red “redtest”

|  |
| --- |
| docker network ls |

* Nos permite ver el listado de redes existentes.

|  |
| --- |
| docker network rm redtest |

* Borramos la red “redtest”.

|  |
| --- |
| docker run -it --network redtest ubuntu /bin/bash |

* Conectamos el contenedor que creamos a la red “redtest”.

|  |
| --- |
| docker network connect IDRED IDCONTENEDOR |

* Conectamos un contenedor a una red.

|  |
| --- |
| docker network disconnect IDRED IDCONTENEDOR |

* Desconectamos un contenedor de una red

 Volúmenes

|  |
| --- |
| docker run -d -it --name appcontainer -v /home/sergi/target:/app nginx:latest |

* Creamos un contenedor y asignamos un volumen con “binding mount”.

|  |
| --- |
| docker run -d -it --name appcontainer -v micontenedor:/app nginx:latest |

* Creamos un contenedor y asignamos un volumen Docker llamado “micontenedor”.

|  |
| --- |
| docker volume create/ls/rm mivolumen |

* Permite crear, listar o eliminar volúmenes Docker.

|  |
| --- |
| docker run -d -it --tmpfs /app nginx |

* Permite crear un contenedor y asociar un volumen “tmpfs”.

|  |
| --- |
| docker run --rm --volumes-from contenedor1 -v /home/sergi/backup:/backup ubuntu bash -c "cd /datos && tar cvf /backup/copiaseguridad.tar ." |

* Permite realizar una copia de seguridad de un volumen asociado a “contenedor1” y que se monta en “/datos”. Dicha copia finalmente acabará en “/home/sergi/backup” de la máquina anfitrión.