|  |  |  |
| --- | --- | --- |
| **Num** | **Command** | **Description** |
| 1 | docker run -ti <docker\_image> bash | To Run and open that image |
| 2 | docker run --memory maximum-allowed-memory image-name command |  |
| 3 | docker run -ti ubuntu:latest bash |  |
| 4 | docker ps -l  docker ps -a | List container |
| 5 | docker commit <container\_ID>  docker tag <commit\_ID> <new\_image\_name> |  |
| 6 | docker commit <container\_name> <new\_image\_name> |  |
| 7 | docker run --rm -ti ubuntu sleep 5 | Delete the container when exits |
| 8 | docker run -ti ubuntu bash -c “sleep 3; echo all done” |  |
| 9 | docker run -d -ti ubuntu bash |  |
| 10 | docker attach <container\_name> |  |
| 11 | ctrl+p ctrl+q | detach container, leaves container |
| 12 | docker exec -ti <container\_name> bash |  |
| 13 | docker network create <network\_name> | create network |
| 14 | docker run -ti --net <network\_name> --name <container\_name> ubuntu:latest bash | create container with particular network |
| 15 | docker run -ti -e <ENV\_VARIABLE>=<message\_or\_content> --name <container\_name> ubuntu:latest bash | create env variable inside docker |
| 16 | docker run -ti –link <container\_name> <ENV\_VARIABLE>=<message\_or\_content> --name <container\_name> ubuntu:latest bash | link container to mentioned container |
| 17 | docker rmi <Image\_ID>  docker rmi <Image\_Name>:<tag> | remove image |
| 18 | docker run -ti -v <path\_of\_data\_folder>:<path\_of\_data\_inside\_the\_container> ubuntu:latest bash | create Ephemeral data |
| 19 | docker run -ti --volumes-from <container\_name> ubuntu:latest bash | use data from another container. |

|  |  |  |
| --- | --- | --- |
| Sr | Topic |  |
| 1 | Volume | Virtual Discs to store and share data.  Not part of images.   1. Persistent: available when container goes also. 2. Ephemeral: Data gets evaporate when no container is using it |
| 2 | Caching in Dockerfile | Docker skips lines that have not changed since the last build.   * The parts that change the most belong at the end of the Dockerfile. |
| 3 | Docker Tips | * Build everything from scratch * Tag your builds with the git hash of the code that built it * Use small Base images, Such as Alpine * Build images you share publicly from Dockerfiles, Always * Don’t ever leave passwords in dockerfiles. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |