1. Base Images.
2. By default, the topmost directory (which is root directory) inside a container is the working directory.
3. By Default Image each name that we generate, a name is given.
4. docker container prune 🡸 To remove all stopped containers.
5. docker image
   1. prune : Removed only Dangling images (having no tag)
      1. -a: To remove dangling images too.
6. **docker rm** <container\_name> [ container\_name container\_name…]
7. docker rmi
8. docker inspect <image-id>
9. docker cp
   1. Host to Container :
      1. docker cp <host-file-or-folder-address> <container-id>:/test (Where \test is the location inside container: If not then will be created).
   2. Container to Host:
      1. docker cp <container-id>:/test <host-file-or-folder-address>
10. docker log : to print the log of the running container till now.
    1. -f : means follow.
11. Attached and Detached
    1. Attached: run Command
    2. Detached: start Command.
    3. Interactive mode: Requires attached mode.
12. Share Images:
    1. Docker Hub:
       1. Public Registry,
    2. Private Registry
13. Docker Hub:
    1. docker push username/<image\_name>
       1. To push on private 🡺 docker push <URL:Name> <image-name>
    2. docker pull <image\_name>
    3. docker login
    4. docker logout
14. d
15. AtGraphical user interface, application

    Description automatically generated

Docker Image

1. Text

   Description automatically generated
2. Command to build image 🡺 docker build . (dot).
   1. Tagging/naming an image 🡺 name : tag = Image Tag = Unique ID.  
      -t <name:tag> where tag is optional -t = --tag  
      Where <name> is repository/group and tag is a specific image in that group.  
      Like 🡺 node:4.9 where node is a group of images and 4.9 is specific version in that group.