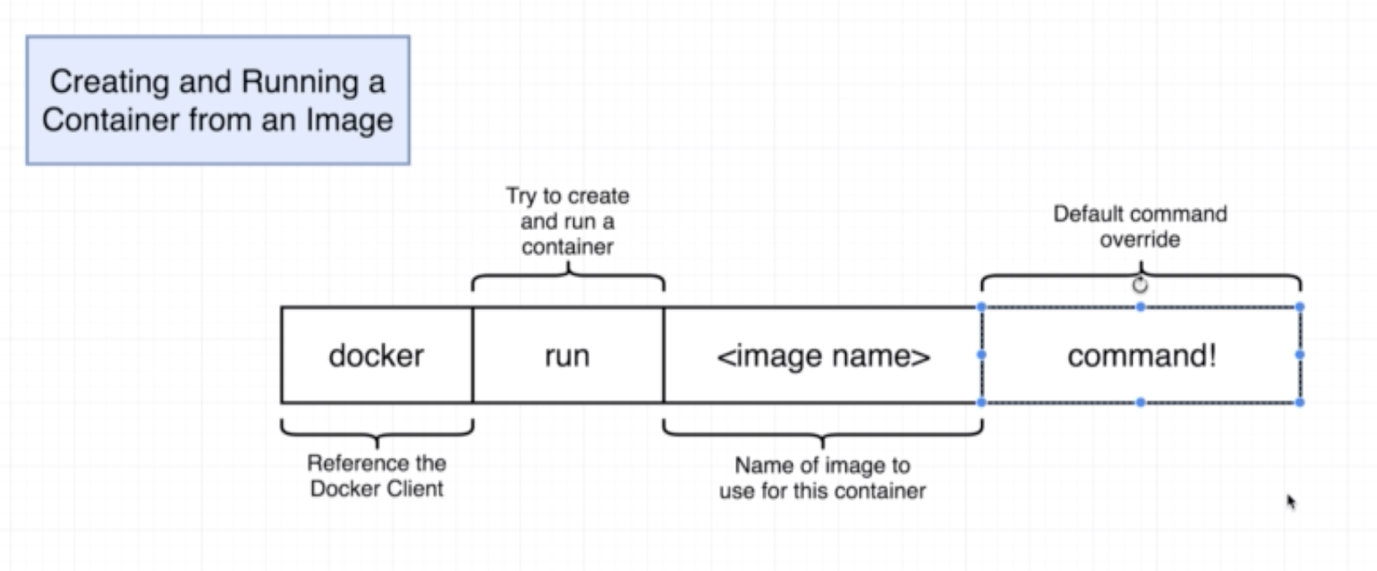
**Docker Client**

* **Introduction:**

**Diagram

Description automatically generated**

* **Overriding Default Commands:**

****

docker run busybox echo hi there

In above command “echo hi there” is the override command.

* **Listing running containers:**

docker ps

docker ps --all

* **Container Lifecycle:**

**A close up of a sign

Description automatically generated**

**Diagram

Description automatically generated**

Docker create:

Diagram

Description automatically generated

Docker Startr:

Diagram

Description automatically generated

docker create hello-world

The above command creates the container with the id

53162f953b54cd8ffd838176b73052b82b69b9f73a9f7697f09f1f5915a70acf

docker start -a 53162f953b54cd8ffd838176b73052b82b69b9f73a9f7697f09f1f5915a70acf

Note: docker start -> execute the primary start up command in the image.

docker start -a id

a -> stands for attach, start this container and watch for whatever the container sends as an output and print in the terminal.

* **Restarting Stopped Containers:**

Execute below command to restart the stopped container.

docker ps –all

Select the container\_id you want to restart and execute

docker start -a id.

* **Removing stopped containers:**

docker system prune

Note: this command deletes all the images and image cache also.

* **Retrieving Log Outputs:**

**Diagram

Description automatically generated**

docker create busybox

docker start id\_of\_busybox

Normal way to get the output is by passing the “-a” option, but we can do it another way by “docker logs” command.

Note: docker logs command can be used even after the container is stopped and it will not bring up the container. Just it will read the record from logs.

* **Stopping containers:**

**Diagram

Description automatically generated**

**Diagram

Description automatically generated**

**Diagram

Description automatically generated**

* **Multi command containers:**

docker create redis-server

docker start redis-server

Go to new terminal and run try to run “redis-cli”, you will get “couldn’t connect to redis”

This is because redis-server is running inside the container and we are trying to access redis-cli from outside the container.

**Diagram

Description automatically generated**

**Diagram

Description automatically generated**

* **Executing commands in running container:**

**Diagram

Description automatically generated**

docker exec container\_id -it redis-cli

Now you will get the redis-cli instance.

* **Purpose of IT flag:**

**A picture containing diagram

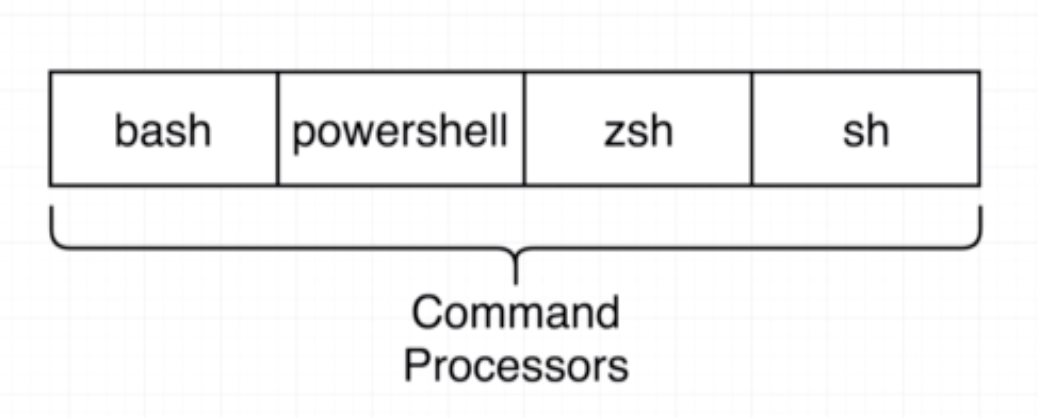
Description automatically generated**

**Diagram

Description automatically generated**

* **Getting a command prompt in a container:**

docker exec -it container\_id sh



We can use the docker run command to get the shell as below

docker run it image\_name sh