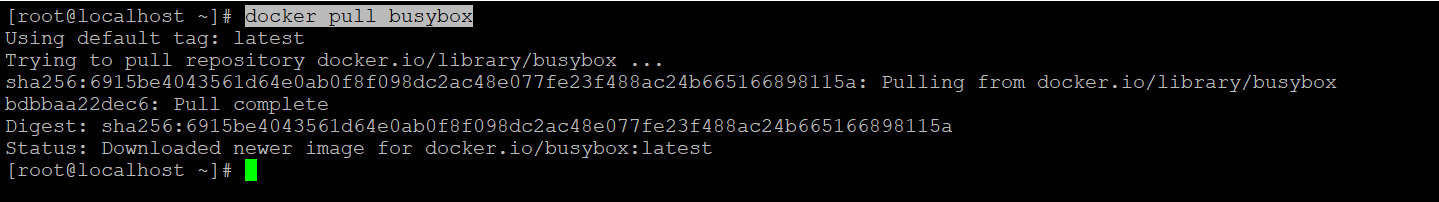
# 1. Docker - Installation

* sudo yum install -y yum-utils device-mapper-persistent-data lvm2
* sudo yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
* sudo yum install –y https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
* sudo yum-config-manager --enable rhui-REGION-rhel-server-extras
* sudo yum install -y docker-ce

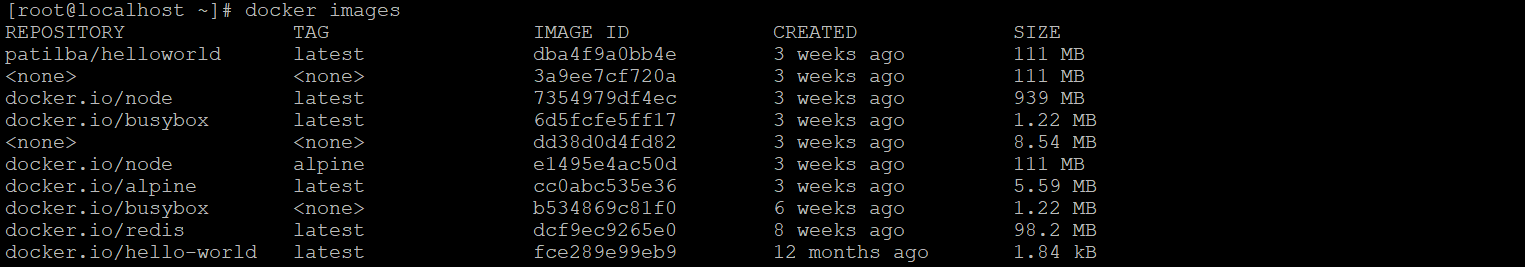
## 1.1 Pull Images from Docker Hub

* **Pull an image or a repository from a registry.**



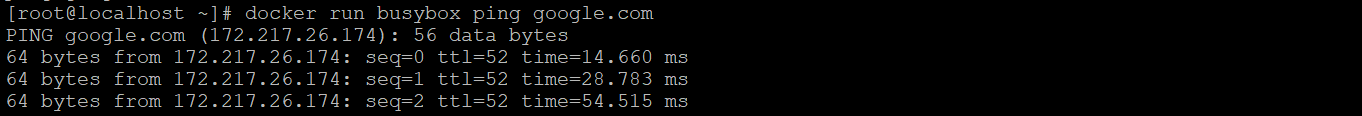
## 1.2 List all Pulled Images

* **Lists all Images**



## 1.3 Run an Image

* **Run an image by specifying image name.**



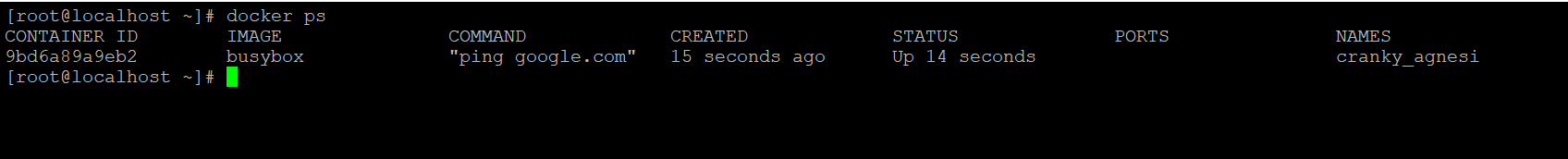
* **Run an image by specifying image ID.**



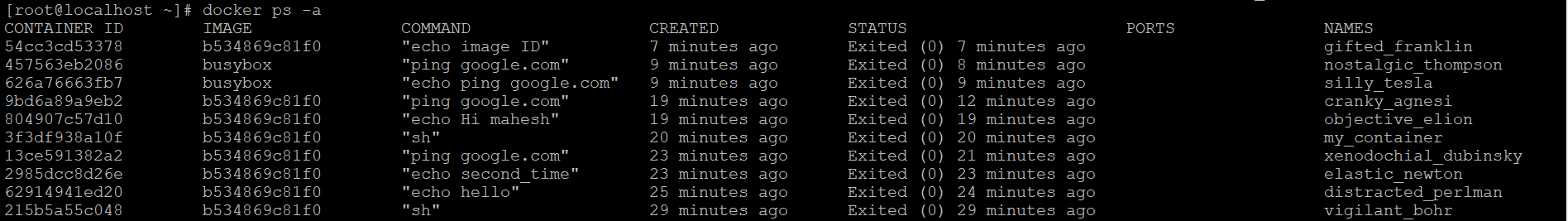
# 2. Containers

### 1.4.1 List Running Containers

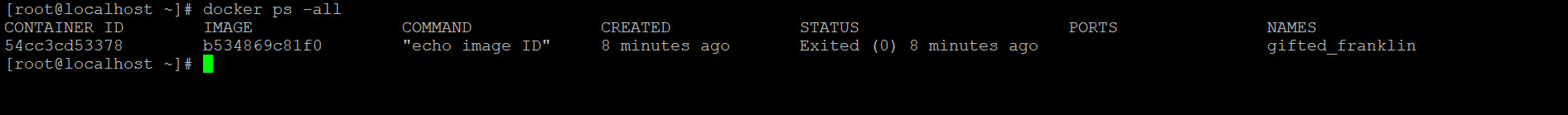
* **Lists all currently running containers.**



* **Lists all currently running and ran earlier containers.**



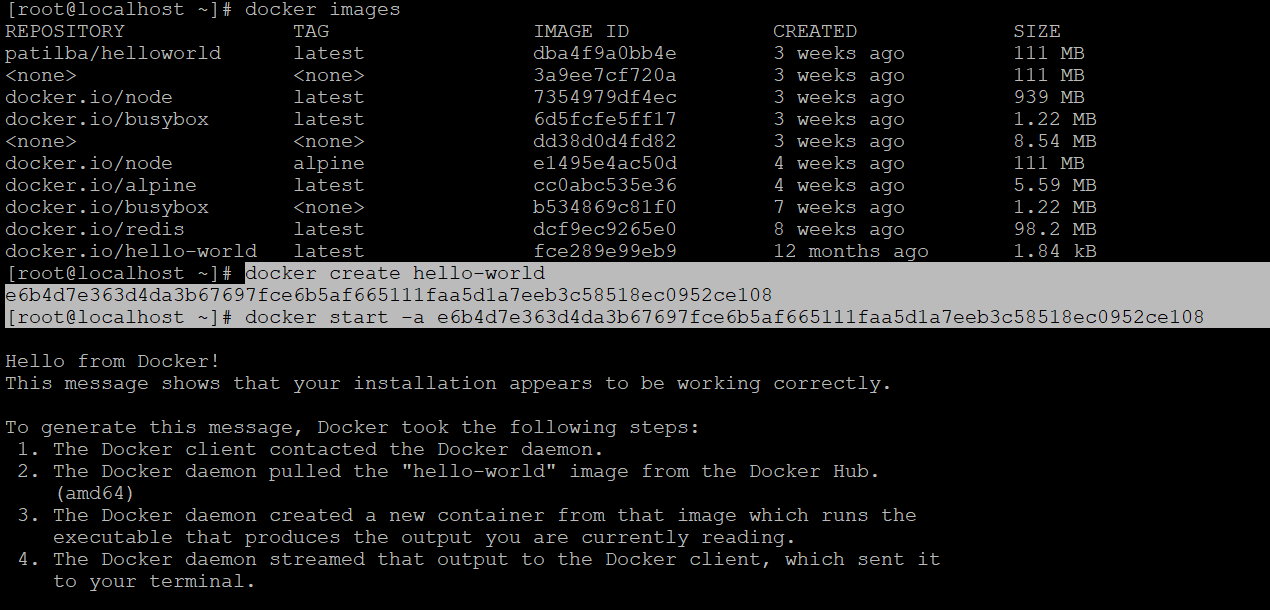
* **Lists latest ran containers.**



### 1.4.2 Create, Start and Stop Containers

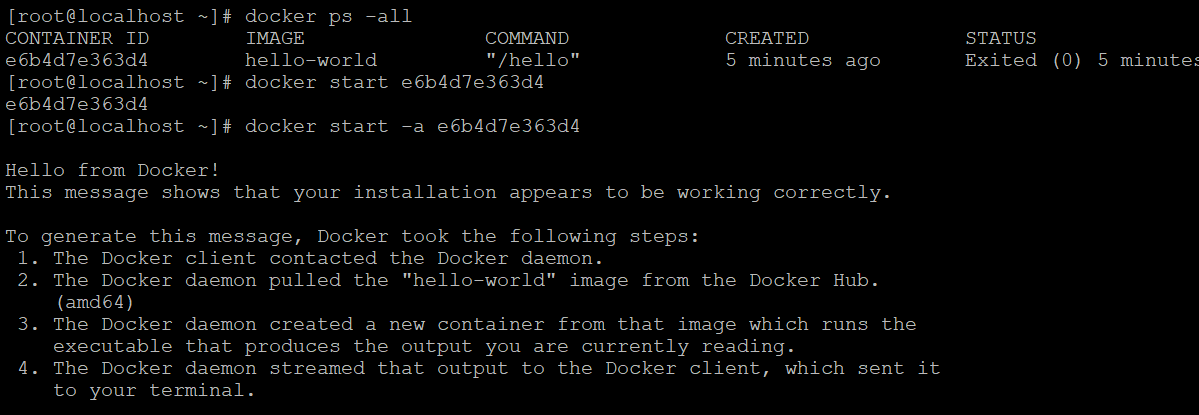
1. docker create [image]🡪prints container id
2. Docker start -a [containerid]
3. Docker stop [containerid]
4. Docker Kill [containerid]

Docker creates a container to run an image mentioned.

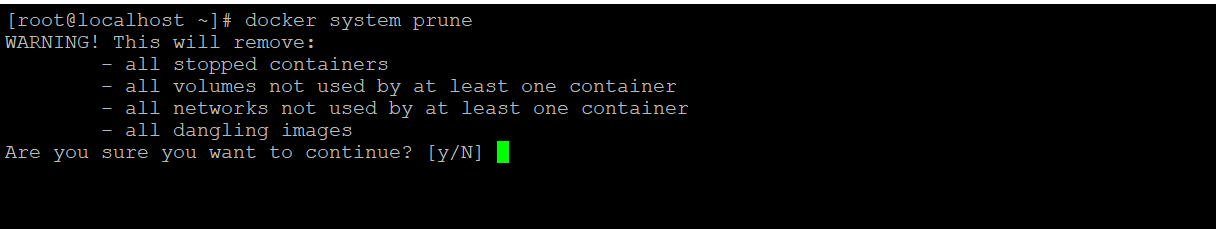


### 1.4.3 Restarting Containers

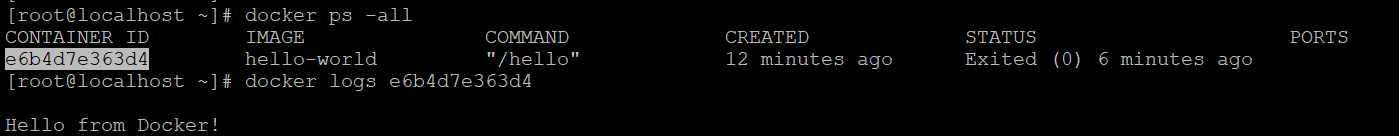
* Docker start -a [containerid]



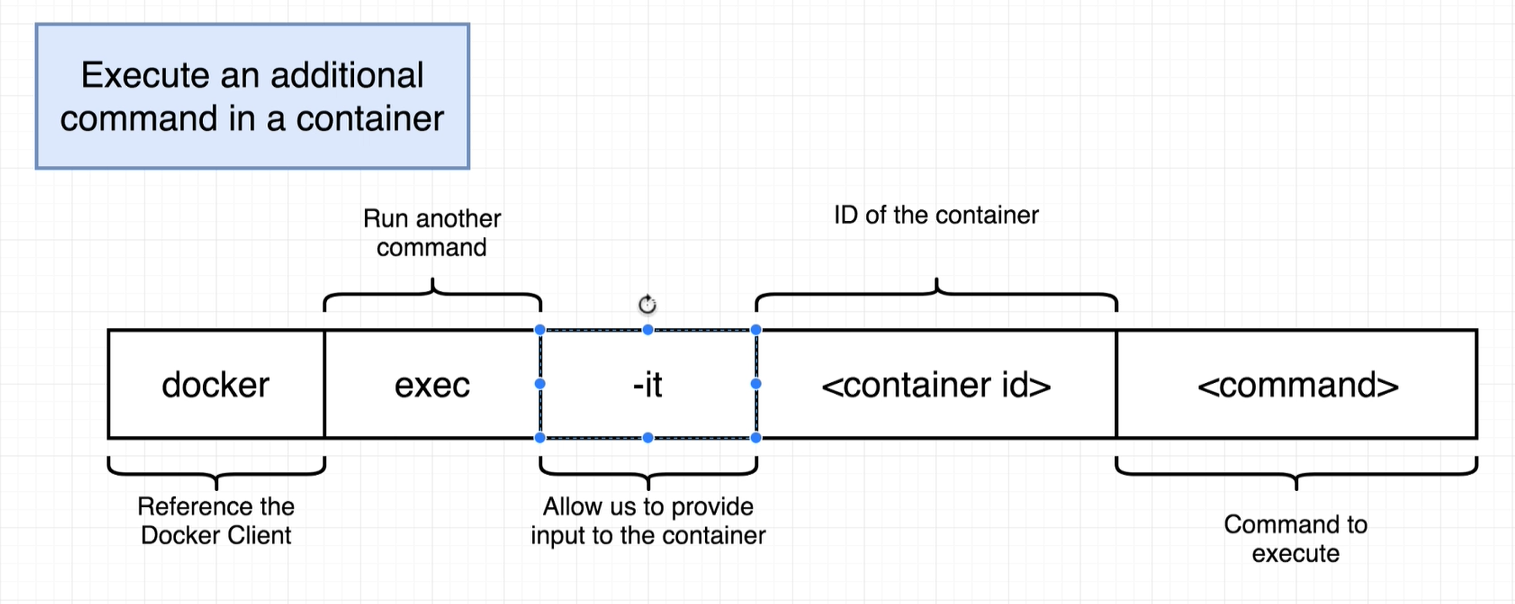
### 1.4.4 Removing Stopped Containers

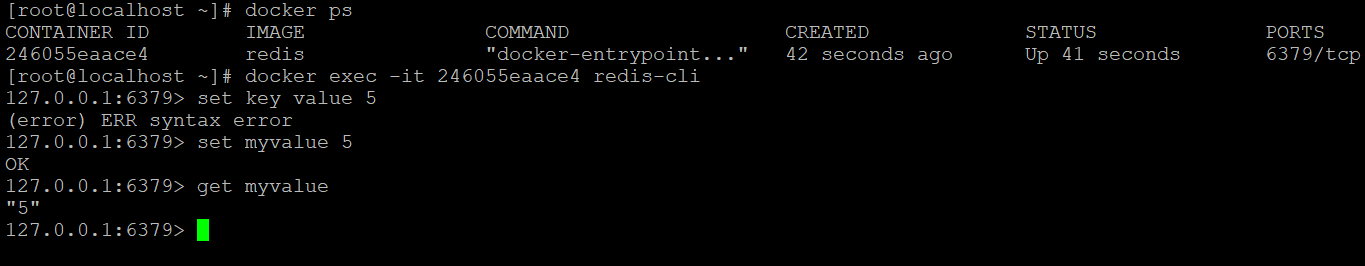


### 1.4.5 Logs

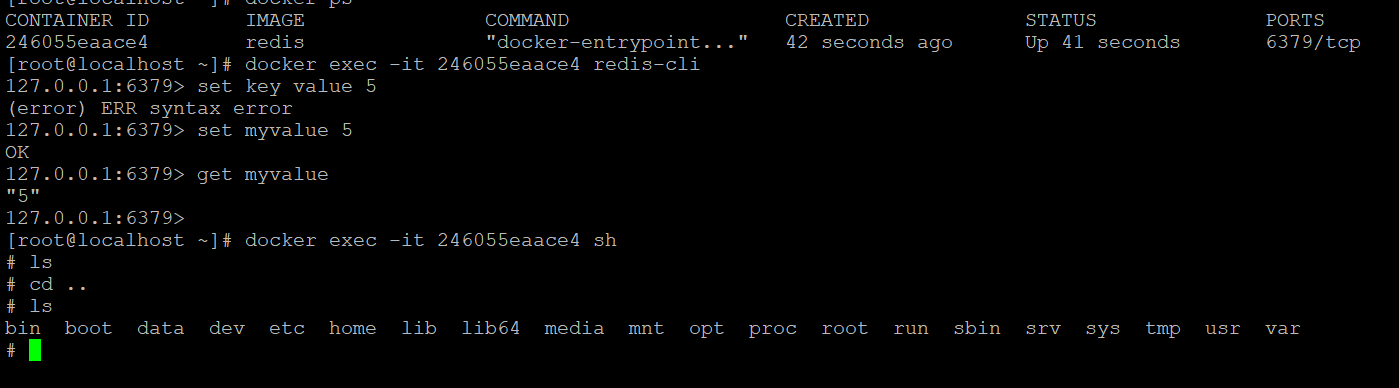


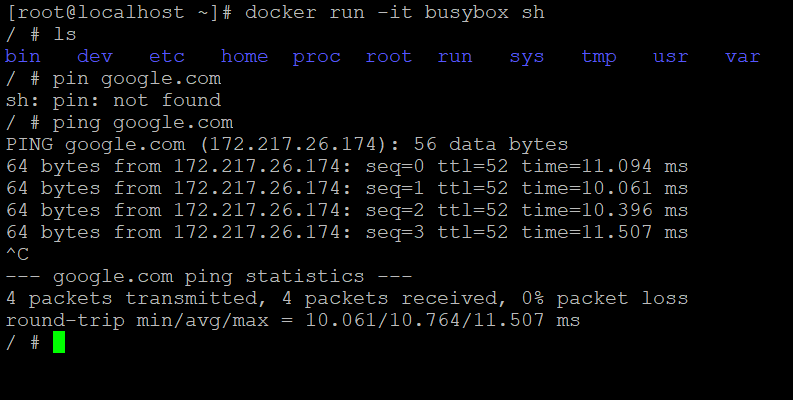
### 1.4.6 Execute a command in running container





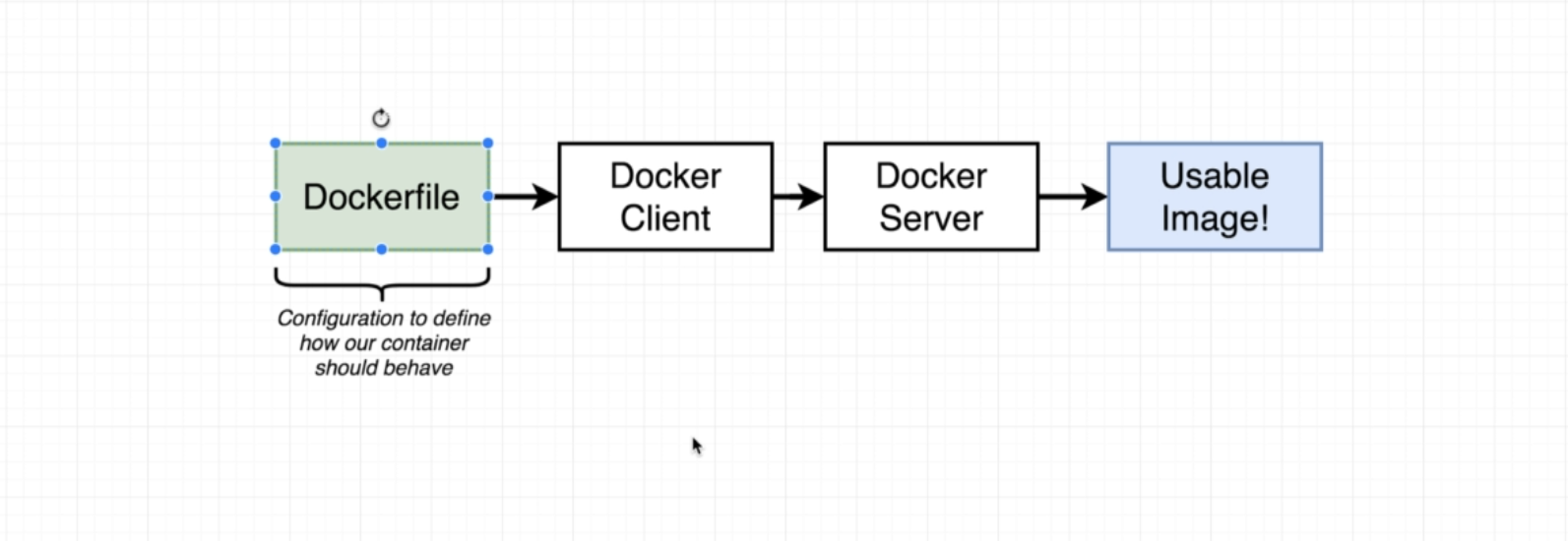
### 1.4.7 Get Command Prompt in Container (bash/sh)



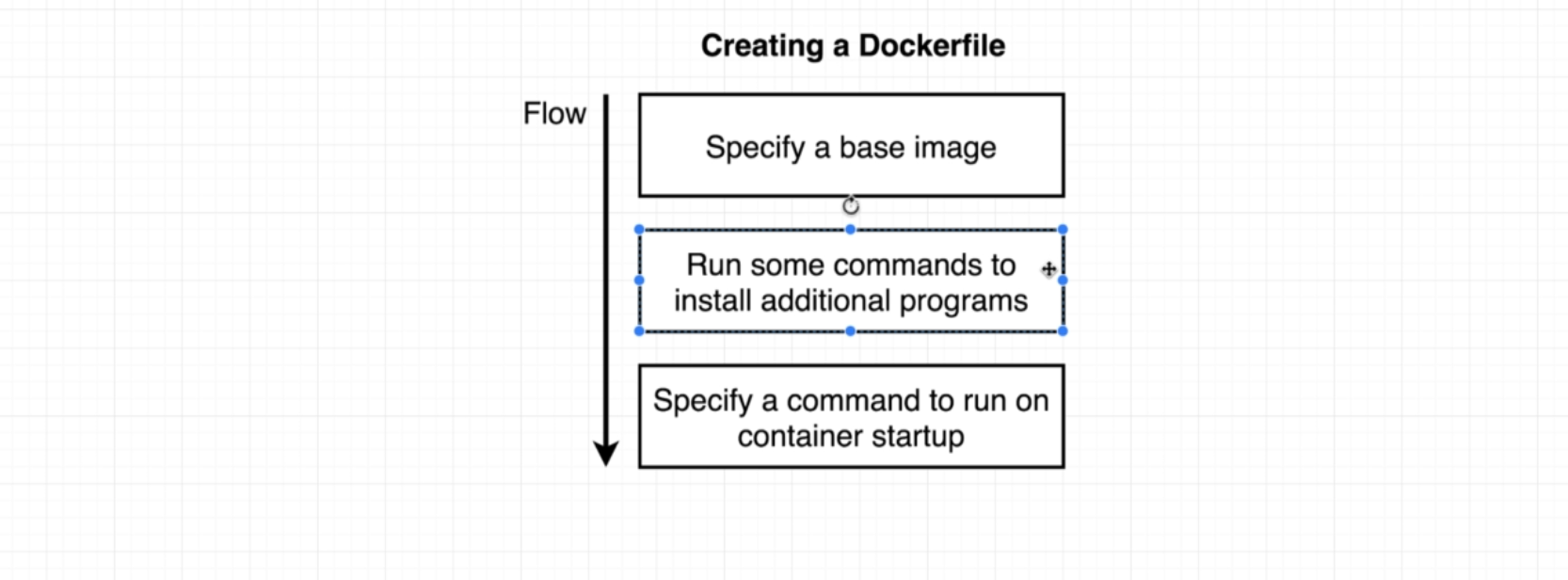


# 3. Images

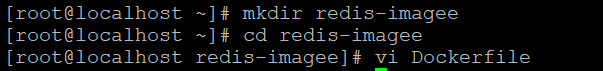
## 2.1 Flow



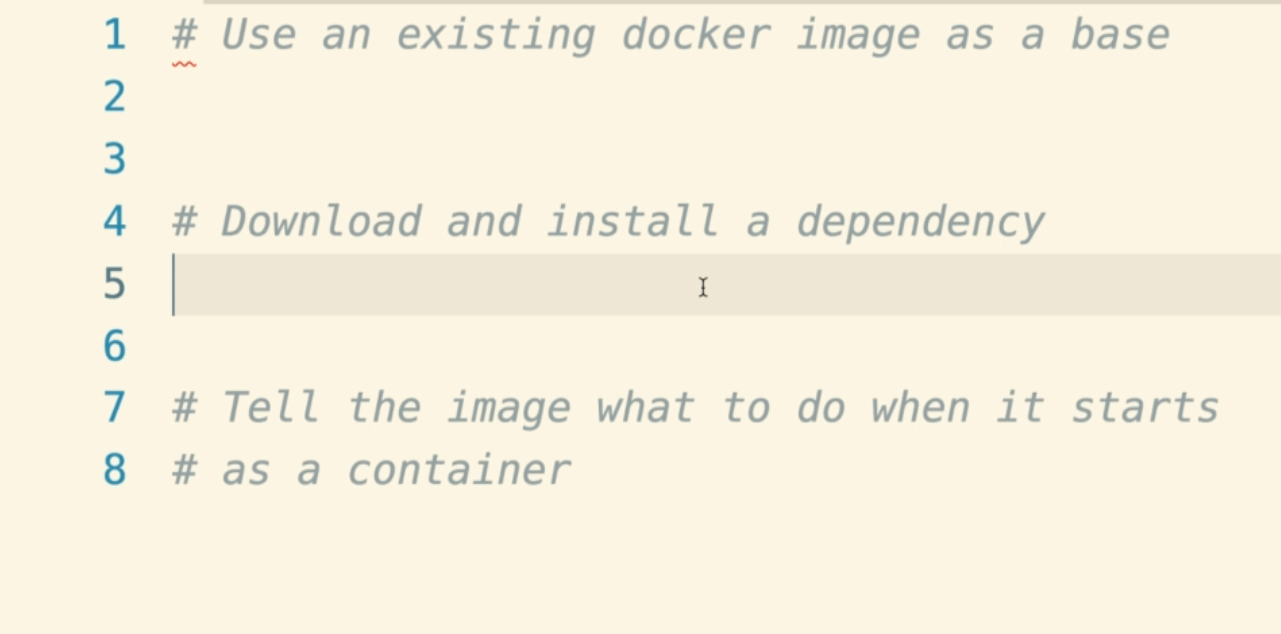
## 2.2 Steps



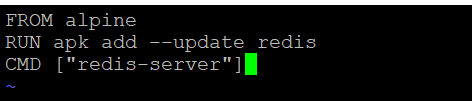
### 2.2.1 Make a Directory



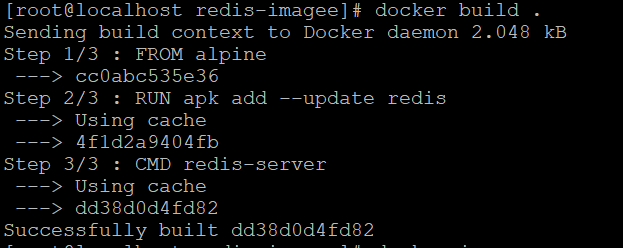
### 2.2.2 create “Dockerfile” (w/o extension).



### 2.2.3 Edit Dockerfile

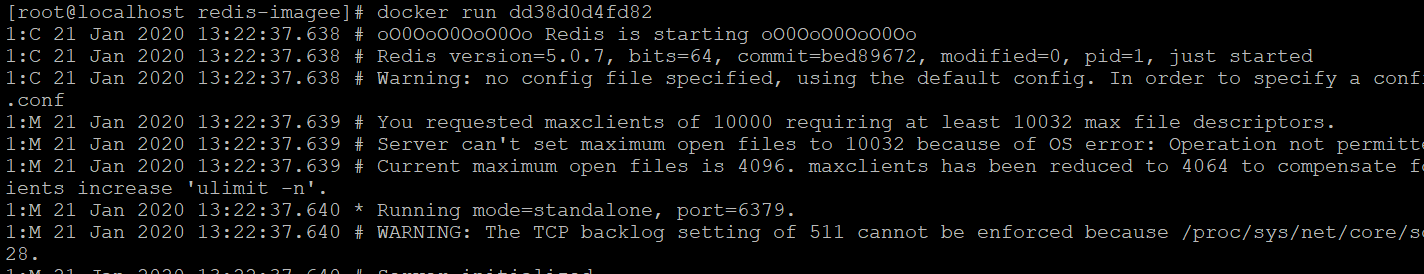


### 2.2.4 Build a redis-image



Copy image id shown like : Successfully built dd38dod4fd82

### 2.2.5 Run Image



# 4. Publish

######################################################################

PUBLISH CONTAINERS

######################################################################

Publish a Code through Nginx!

1. Install wget and unzip cmd

2. wget https://www.free-css.com/assets/files/free-css-templates/download/page249/magazee.zip

3. ls

4. unzip magazee.zip -d ./var/lib/docker/devicemapper/mnt/21f298042f86ca2f550df42affbecdc44556c7fc2d5ed369cf58266ffe62d7a8/rootfs/usr/share/nginx/html

directory is created so move all contents to nginx

goto inside of unzipped directory and type "mv \* .."

5. publish nginx image

6. Commit

docker commit [container] [image{new name if u want}]

7. push to docker hub

1. login :docker login

2. push :docker push

# find /home -iname tecmint.txt

# 5. Tar File

To create a tar ball of images

# docker export [image] -o [tar\_ball\_name].tar

To check tar file --> saved in pwd.

To import tar ball from anysource

# docker import tar\_ball\_name.tar [new name for that]

----------------------------------------------------------------------

To tag a docker image

# docker tag [image] [new\_name]

it will create copy of image with new name.

# 6. Network

# docker netork ls

----------------------------------------------------------------------

To see container details

# docker inspect [container] --> see network settings and IP address.

----------------------------------------------------------------------

To create new custom connection

# Dcoker network create [name] -d bridge

we can use during container creation

----------------------------------------------------------------------

To use new connection with container provide as argument

# Docker run -itd --network my\_con -p 8080:80 [imagename]

----------------------------------------------------------------------

To connect two network for same container

# Docker run -itd --network my\_con -p 8080:80 [container]

# docker network connect [container] [conn]

----------------------------------------------------------------------

To disconnect

# docker network disconnect [conn] [container]

To connect with Host Network

# docker network connect host [container]

# 7. Volumes

# mkdir volume123

Mounting volume123 to container

# Docker run -itd -v [/root/volume123:/data [image]]

-v = volume

here data folder is created by container and this is optional.

1. go to inside container

2. create a file

3. come out of the container

4. files are available in host machine also.

----------------------------------------------------------------------

To list volumes

# docker volume ls