

EXPERIMENT 2

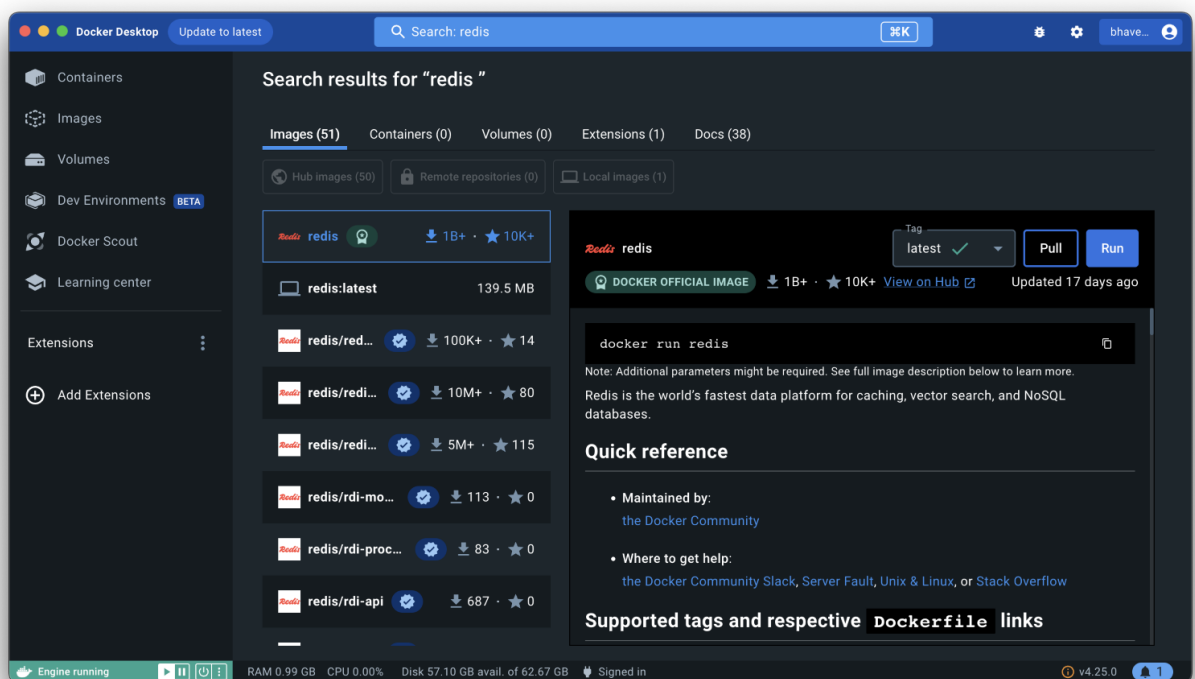
Name : Bhavesh Sanjiv Kapur

SapID: 500105635

Roll no: R2142220057

Aim : Working with Docker – Basic Commands

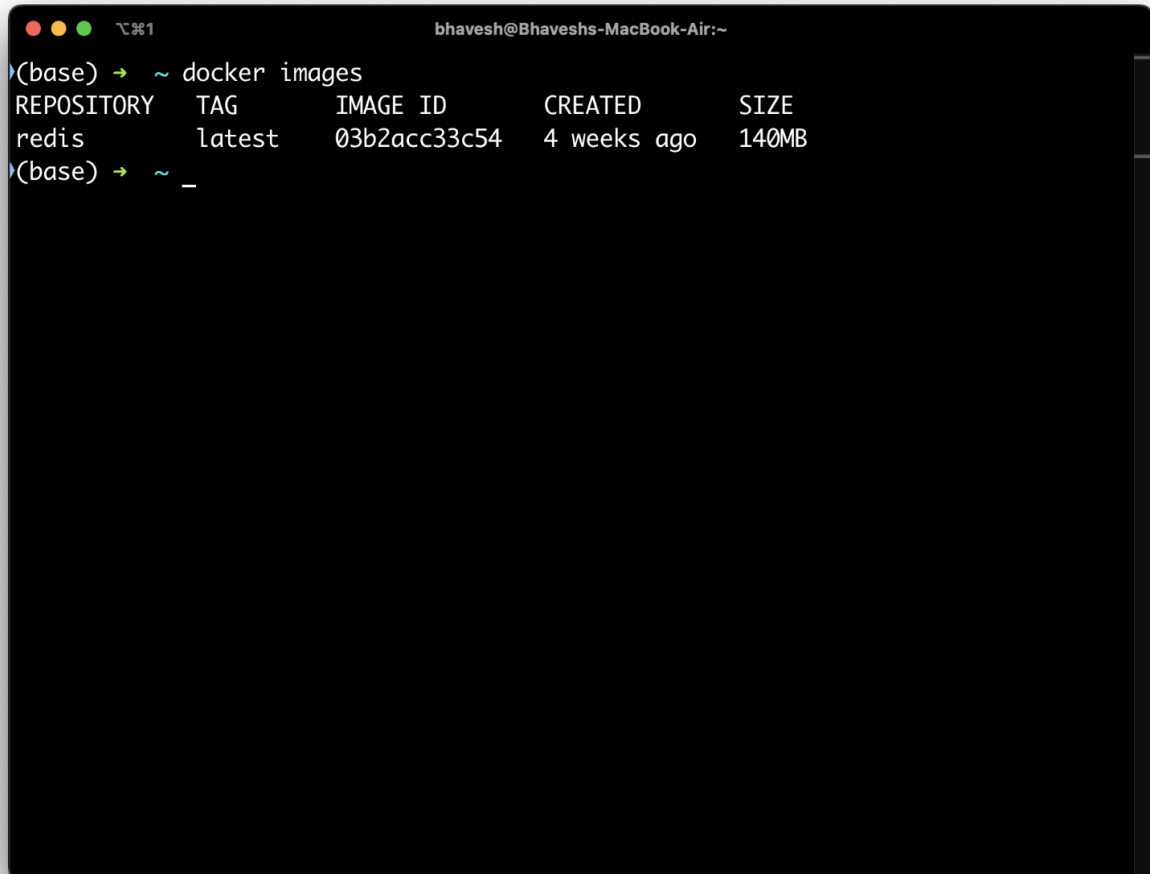
1. Finding a Redis Image on Docker Hub:



Pulling redis image on local machine to use it offline
~ docker pull redis
this command will pull the redis image from docker hub.

```
bhaves@Bhaves-MacBook-Air:~  
(base) → ~ docker pull redis  
Using default tag: latest  
latest: Pulling from library/redis  
aa6fbc30c84e: Pull complete  
323d31fa11ec: Pull complete  
67d6949a391e: Pull complete  
223ac6cd20bc: Pull complete  
50f092169b10: Pull complete  
1ef69efe6db3: Pull complete  
4f4fb700ef54: Pull complete  
39ad5645d4ce: Pull complete  
Digest: sha256:878983f8f5045b28384fc300268cec62bca3b14d5e1a448bec21f28cfcc7bf78  
Status: Downloaded newer image for redis:latest  
docker.io/library/redis:latest  
  
What's Next?  
View a summary of image vulnerabilities and recommendations → docker scout quickview redis  
(base) → ~ _
```

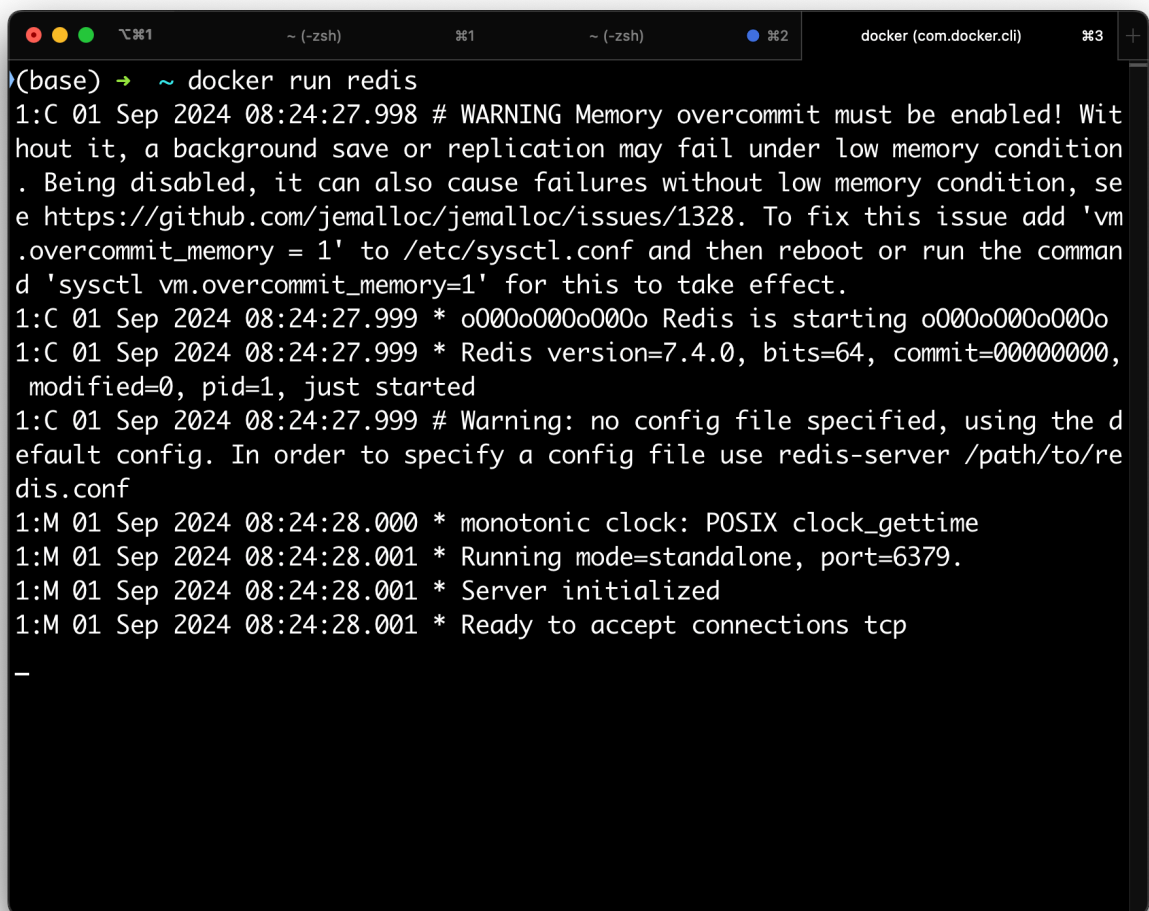
We can check what all images are pull from docker hub ~
docker images
this command helps to see all the images downloaded on to
the local machine.

A terminal window on a Mac with a black background and white text. The window title is 'bhavesh@Bhaveshs-MacBook-Air:~'. The prompt is '(base) → ~'. The command 'docker images' has been entered. The output is a table with five columns: REPOSITORY, TAG, IMAGE ID, CREATED, and SIZE. There is one row of data for the 'redis' image.

```
(base) → ~ docker images
REPOSITORY  TAG      IMAGE ID    CREATED     SIZE
redis        latest   03b2acc33c54 4 weeks ago 140MB
(base) → ~ _
```

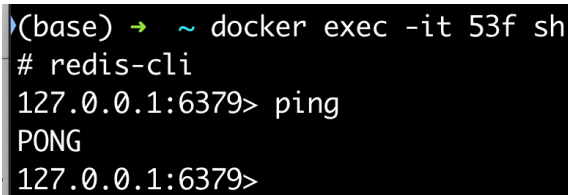
As redis image has been pulled from the docker hub
we can use it.

2. Running the docker container having the redis image:



```
(base) → ~ docker run redis
1:C 01 Sep 2024 08:24:27.998 # WARNING Memory overcommit must be enabled! Without it, a background save or replication may fail under low memory condition. Being disabled, it can also cause failures without low memory condition, see https://github.com/jemalloc/jemalloc/issues/1328. To fix this issue add 'vm.overcommit_memory = 1' to /etc/sysctl.conf and then reboot or run the command 'sysctl vm.overcommit_memory=1' for this to take effect.
1:C 01 Sep 2024 08:24:27.999 * o000o000o000o Redis is starting o000o000o000o
1:C 01 Sep 2024 08:24:27.999 * Redis version=7.4.0, bits=64, commit=00000000, modified=0, pid=1, just started
1:C 01 Sep 2024 08:24:27.999 # Warning: no config file specified, using the default config. In order to specify a config file use redis-server /path/to/redis.conf
1:M 01 Sep 2024 08:24:28.000 * monotonic clock: POSIX clock_gettime
1:M 01 Sep 2024 08:24:28.001 * Running mode=standalone, port=6379.
1:M 01 Sep 2024 08:24:28.001 * Server initialized
1:M 01 Sep 2024 08:24:28.001 * Ready to accept connections tcp
-
```

To run redis we will use the command
~docker run redis
this will create a container.



```
(base) → ~ docker exec -it 53f sh
# redis-cli
127.0.0.1:6379> ping
PONG
127.0.0.1:6379> _
```

This shows that the redis container is working fine.

3. Docker ps and Docker ps -a

These commands show the list of docker containers on our local machine

On running `~docker ps` and `~docker ps -a` the output is :

```
(base) → ~ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES
53fbb46cbf9e   redis    "docker-entrypoint.s..." 5 minutes ago  Up 5 minutes  6379/tcp     modest_matsumoto
(base) → ~ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES
53fbb46cbf9e   redis    "docker-entrypoint.s..." 5 minutes ago  Up 5 minutes  6379/tcp     modest_matsumoto
(base) → ~ _
```

`~docker ps` just shows the current running containers on the local machine

whereas

`~docker ps -a` shows the list of all running and stopped containers on the local machine

4. Creating Docker volume

`~docker volume create volumeBhavesh`

this creates a new volume

```
(base) → ~ docker volume create volumeBhavesh
volumeBhavesh
(base) → ~ docker volume ls
DRIVER      VOLUME NAME
local       0ae4ad30cd2b1dd56f7e698bc93cdbe4b368540ca1f0c7159816205c32e56cc0
local       0efcd0c1093f345927a6ad6c7710ba9a09aed8289101555dedbfa34e97279e9a
local       4b931dacbce5bdc3e997ae5eae2a681273a6633bf95482e47b01774ee8069bc6
local       d036f39a36132dd8281dec905fe75522b4eb53c8e46ab6b67612b4f6688b0809
local       volumeBhavesh
(base) → ~ _
```

Running the redis container with the new volume:

`~docker run -d --name redisBhavesh -v`

`volumeBhavesh:/data redis:latest`

```
(base) → ~ docker run -d --name redisBhavesh2 -v volumeBhavesh:/data redis:latest
9b734252bd23b0a1630144d5baaea45db34b92950895d2e62f3a9465a2f271cb
(base) → ~ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES
9b734252bd23   redis:latest "docker-entrypoint.s..." 7 seconds ago  Up 7 seconds  6379/tcp     redisBhavesh2
53fbb46cbf9e   redis    "docker-entrypoint.s..." About an hour ago  Exited (1) 7 minutes ago  redisBhavesh
(base) → ~ _
```

this creates another container

