

EXPERIMENT 2

AIM: Working with Docker (Basic Commands)

Steps to Complete:

1. Search a Redis image on DockerHub

```
om-vats@om-vats-VirtualBox:~$ sudo docker pull redis
[sudo] password for om-vats:
Using default tag: latest
latest: Pulling from library/redis
e4fff0779e6d: Pull complete
d1dde3db2ec5: Pull complete
1d321a003dde: Pull complete
d65aedb2f012: Pull complete
4018f93716a2: Pull complete
b0967b02e8cf: Pull complete
4f4fb700ef54: Pull complete
d288b86f5d06: Pull complete
Digest: sha256:878983f8f5045b28384fc300268cec62bca3b14d5e1a448bec21f28cfcc7bf78
Status: Downloaded newer image for redis:latest
docker.io/library/redis:latest
om-vats@om-vats-VirtualBox:~$ sudo docker search redis
NAME                                DESCRIPTION
STARS      OFFICIAL  AUTOMATED
redis                                Redis is the world's fastest data platform f...
12987      [OK]
redis/redisinsight                  Redis Insight - our best official GUI for Re...
14
redis/redis-stack-server            redis-stack-server installs a Redis server w...
80
```

```

0
redis/rdi-operator
0
redis/rdi-collector-initializer    Init container for RDI Collector
0
circleci/redis                    CircleCI images for Redis
17                                [OK]
bitnami/redis                     Bitnami container image for Redis
300                               [OK]
chainguard/redis                  Minimalist Wolfi-based Redis image.
1
elestio/redis                     Redis, verified and packaged by Elestio
0
redislabs/redis                   Clustered in-memory database engine compatib...
41
rapidfort/redis                   RapidFort optimized, hardened image for Redi...
21
ubuntu/redis                      Redis, an open source key-value store. Long-...
22
drud/redis                        redis
0                                [OK]
metabrainz/redis                  MetaBrainz redis cluster
0                                [OK]
cing/redis

```

2. Run Docker Container of Redis Image in background

```

1
redislabs/ng-redis-raft           Redis with redis raft module
0
google/apigee-redis               apigee-redis
1
redislabs/ru101-lab               Redis University 101
0
om-vats@om-vats-VirtualBox:~$ docker run --name my-redis -d redis
docker: permission denied while trying to connect to the Docker daemon socket at
unix:///var/run/docker.sock: Post "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/containers/create?name=my-redis": dial unix /var/run/docker.sock: connect: permission denied.
See 'docker run --help'.
om-vats@om-vats-VirtualBox:~$ sudo docker run --name my-redis -d redis
[sudo] password for om-vats:
Sorry, try again.
[sudo] password for om-vats:
e562a30accd45810ca55be525c90b56527f910602c6db94f9eec245cab256104
om-vats@om-vats-VirtualBox:~$ sudo docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED          STATUS
PORTS         NAMES
e562a30accd4   redis     "docker-entrypoint.s..." About a minute ago Up About
a minute      6379/tcp  my-redis
om-vats@om-vats-VirtualBox:~$

```

3. Run Docker PS and Docker PS -a

```

1
redislabs/ng-redis-raft          Redis with redis raft module
0
google/apigee-redis              apigee-redis
1
redislabs/ru101-lab              Redis University 101
0
om-vats@om-vats-VirtualBox:~$ docker run --name my-redis -d redis
docker: permission denied while trying to connect to the Docker daemon socket at
unix:///var/run/docker.sock: Post "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/containers/create?name=my-redis": dial unix /var/run/docker.sock: connect: permission denied.
See 'docker run --help'.
om-vats@om-vats-VirtualBox:~$ sudo docker run --name my-redis -d redis
[sudo] password for om-vats:
Sorry, try again.
[sudo] password for om-vats:
e562a30accd45810ca55be525c90b56527f910602c6db94f9eec245cab256104
om-vats@om-vats-VirtualBox:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS          NAMES
e562a30accd4   redis    "docker-entrypoint.s..." About a minute ago Up About a minute 6379/tcp       my-redis
om-vats@om-vats-VirtualBox:~$

```

4. Run Docker Container and take its console

```

om-vats@om-vats-VirtualBox:~$ sudo docker run --name my-redis -d redis
[sudo] password for om-vats:
Sorry, try again.
[sudo] password for om-vats:
e562a30accd45810ca55be525c90b56527f910602c6db94f9eec245cab256104
om-vats@om-vats-VirtualBox:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS          NAMES
e562a30accd4   redis    "docker-entrypoint.s..." About a minute ago Up About a minute 6379/tcp       my-redis
om-vats@om-vats-VirtualBox:~$ sudo docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS          NAMES
e562a30accd4   redis    "docker-entrypoint.s..." 11 minutes ago Up 11 minutes 6379/tcp       my-redis
om-vats@om-vats-VirtualBox:~$ sudo docker run -it --name my-redis-container redis
1:C 01 Sep 2024 14:27:33.638 # WARNING Memory overcommit must be enabled! Without it, a background save or replication may
under low memory condition. Being disabled, it can also cause failures without low memory condition, see https://github.com
lloc/jemalloc/issues/1328. To fix this issue add 'vm.overcommit_memory = 1' to /etc/sysctl.conf and then reboot or run the
nd 'sysctl vm.overcommit_memory=1' for this to take effect.
1:C 01 Sep 2024 14:27:33.642 * o000o000o000o Redis is starting o000o000o000o
1:C 01 Sep 2024 14:27:33.642 * Redis version=7.4.0, bits=64, commit=00000000, modified=0, pid=1, just started
1:C 01 Sep 2024 14:27:33.642 # Warning: no config file specified, using the default config. In order to specify a config fi
e redis-server /path/to/redis.conf
1:M 01 Sep 2024 14:27:33.647 * Increased maximum number of open files to 10032 (it was originally set to 1024).
1:M 01 Sep 2024 14:27:33.658 * monotonic clock: POSIX clock_gettime

```



```
"LogPath": "/var/snap/docker/containers/323761/c7806b401916a641a09c1fd28a4d9c0321fd28a4d9c032fdd9b8a517eb2f9442",
"Name": "/my-redis-container",
"RestartCount": 0,
"Driver": "overlay2",
"Platform": "linux",
"MountLabel": "",
"ProcessLabel": "",
"AppArmorProfile": "docker-default",
"ExecIDs": null,
"HostConfig": {
  "Binds": null,
  "ContainerIDFile": "",
  "LogConfig": {
    "Type": "json-file",
    "Config": {}
  },
  "NetworkMode": "default",
  "PortBindings": {},
  "RestartPolicy": {
    "Name": "no",
    "MaximumRetryCount": 0
  },
  "AutoRemove": false,
  "VolumeDriver": "",
  "VolumesFrom": null,
  "ConsoleSize": [
    24,
    80
  ],
  "CapAdd": null,
```