

NIPUN SINGAL
500069052
R171218069

Application Containerization Lab

EXPERIMENT-2

Deploying Your First Docker Container

◀ Step 3 of 6 ▶

Step 3 - Accessing Redis

Jane is happy that *Redis* is running, but is surprised that she cannot access it. The reason is that each container is sandboxed. If a service needs to be accessible by a process not running in a container, then the port needs to be exposed via the Host.

Once exposed, it is possible to access the process as if it were running on the host OS itself.

Jane knows that by default, *Redis* runs on port 6379. She has learned that by default other applications and library expect a *Redis* instance to be listening on the port.

Task

After reading the documentation, Jane discovers that ports are bound when containers are started

Terminal



Your Interactive Bash Terminal. A safe place to learn and execute commands.

```
$ docker search redis
```

NAME	DESCRIPTION	STARS
redis	Redis is an open source key-value store that...	9029
[OK]		
bitnami/redis	Bitnami Redis Docker Image	172
[OK]		
sameersbn/redis		82
[OK]		
grokzen/redis-cluster	Redis cluster 3.0, 3.2, 4.0, 5.0, 6.0	75
rediscommander/redis-commander	Alpine image for redis-commander - Redis man...	52
[OK]		
kubeguide/redis-master	redis-master with "Hello World!"	33
redislabs/redisearch	Redis With the RedisSearch module pre-loaded...	30
redislabs/redis	Clustered in-memory database engine compatib...	27
oliver006/redis_exporter	Prometheus Exporter for Redis Metrics. Supp...	23
redislabs/rejson	RedisJSON - Enhanced JSON data type processi...	23

Deploying Your First Docker Container

◀ Step 3 of 6 ▶

Step 3 - Accessing Redis

Jane is happy that *Redis* is running, but is surprised that she cannot access it. The reason is that each container is sandboxed. If a service needs to be accessible by a process not running in a container, then the port needs to be exposed via the Host.

Once exposed, it is possible to access the process as if it were running on the host OS itself.

Jane knows that by default, *Redis* runs on port 6379. She has learned that by default other applications and library expect a *Redis* instance to be listening on the port.

Task

After reading the documentation, Jane discovers that ports are bound when containers are started

Terminal +

[OK]

```
$ docker run -d redis
b3a0e362fac75a3b1afc4f2c684810f51722396f310115f1799d7191f71261f3
$ docker ps
CONTAINER ID      IMAGE      COMMAND      CREATED      STATUS
PORTS            NAMES
b3a0e362fac7      redis      "docker-entri...  13 seconds ago  Up 11 secon
ds              hungry_proskuriakova
$ docker ps -a
CONTAINER ID      IMAGE      COMMAND      CREATED      STATUS
PORTS            NAMES
b3a0e362fac7      redis      "docker-entri...  15 minutes ago  Up 15 minut
es              hungry_proskuriakova
$ docker run -it redis
1:C 25 Jan 10:22:07.735 # o000o000o000o Redis is starting o000o000o000o
1:C 25 Jan 10:22:07.738 # Redis version=4.0.11, bits=64, commit=00000000, modified=0, pid=1, jus
t started
1:C 25 Jan 10:22:07.738 # Warning: no config file specified, using the default config. In order
to specify a config file use redis-server /path/to/redis.conf

                                Redis 4.0.11 (00000000/0) 64 bit

                                Running in standalone mode
                                Port: 6379
```