

## Experiment 09

**Aim:** To understand Docker Architecture and Container Life Cycle, install Docker and execute docker commands to manage images and interact with Containers.

**Theory:** Docker is a popular platform that enables developers to build, package, and deploy applications as lightweight, portable, and self-sufficient containers. These containers encapsulate all the necessary dependencies and libraries required for an application to run, ensuring consistency across different environments. Here is a theoretical overview of

Docker:

Containerization:

Docker utilizes containerization technology to create isolated environments for applications. Containers are lightweight, standalone, and executable packages that include everything needed to run an application, such as code, runtime, system tools, libraries, and settings. This isolation ensures that applications run consistently across different environments, from development to production.

Docker Engine:

At the core of Docker is the Docker Engine, which is responsible for building, running, and managing containers. It consists of the Docker daemon, which manages containers, images, networks, and volumes, and the Docker client, which allows users to interact with the daemon through the Docker API.

Docker Images:

Docker images are read-only templates used to create containers. They contain the application code, runtime, libraries, dependencies, and other files needed to run the application. Images are built using Dockerfiles, which are text files that define the steps needed to create the image.

Docker Containers:

Containers are instances of Docker images that are running as isolated processes on a host machine. They are lightweight, portable, and can be easily started, stopped, moved, and deleted. Containers provide a consistent environment for applications to run, regardless of the underlying infrastructure.

Benefits of Docker:

**Portability:** Docker containers can run on any platform that supports Docker, making it easy to deploy applications across different environments.

**Efficiency:** Containers share the host OS kernel, reducing overhead and improving resource utilization.

**Isolation:** Containers provide a level of isolation that helps prevent conflicts between applications and dependencies.

**Scalability:** Docker enables easy scaling of applications by quickly spinning up additional containers.

**Consistency:** Docker ensures that applications run the same way in development, testing, and production environments.

## Output:

```

C:\Users\202>docker run redis
Unable to find image 'redis:latest' locally
latest: Pulling from library/redis
8ale25ce7c4f: Pull complete
8ab039a68e51: Pull complete
2b12a49dcfb9: Pull complete
cdf9868f47ac: Pull complete
e73ea5d3136b: Pull complete
890ad32c613f: Pull complete
4f4fb700ef54: Pull complete
ba517b76f92b: Pull complete
Digest: sha256:7dd707032d90c6eaafd566f62a00f5b0116ae08fd7d6cbbb0f311b82b47171a2
Status: Downloaded newer image for redis:latest
1:C 13 Mar 2024 03:19:03.928 * oOoOoOoOoOoOo Redis is starting oOoOoOoOoOoOo
1:C 13 Mar 2024 03:19:03.928 * Redis version=7.2.4, bits=64, commit=00000000, mod
1:C 13 Mar 2024 03:19:03.928 # Warning: no config file specified, using the defau
s.conf
1:M 13 Mar 2024 03:19:03.929 * monotonic clock: POSIX clock_gettime
1:M 13 Mar 2024 03:19:03.929 * Running mode=standalone, port=6379.
1:M 13 Mar 2024 03:19:03.929 * Server initialized
1:M 13 Mar 2024 03:19:03.929 * Ready to accept connections tcp
1:signal-handler (1710300105) Received SIGINT scheduling shutdown...
1:M 13 Mar 2024 03:21:45.877 * User requested shutdown...
1:M 13 Mar 2024 03:21:45.877 * Saving the final RDB snapshot before exiting.
1:M 13 Mar 2024 03:21:45.887 * DB saved on disk
1:M 13 Mar 2024 03:21:45.887 # Redis is now ready to exit, bye bye...

```

```

C:\Users\202>docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
redis         latest    170ale90f843   2 months ago  138MB

```

```

C:\Users\202>docker pull redis
Using default tag: latest
latest: Pulling from library/redis
Digest: sha256:7dd707032d90c6eaafd566f62a00f5b0116ae08fd7d6cbbb0f311b82b47171a2
Status: Image is up to date for redis:latest
docker.io/library/redis:latest

```

```
C:\Users\202>docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED    STATUS    PORTS    NAMES
C:\Users\202>docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED    STATUS    PORTS    NAMES
052aeb0ee88    redis    "docker-entrypoint.s..." About a minute ago    Up 4 seconds    6379/tcp    container121
1c4472744083    redis    "docker-entrypoint.s..." 6 minutes ago        Up 10 seconds    6379/tcp    modest_herschel
C:\Users\202>
```

```
C:\Users\202>docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED    STATUS    PORTS    NAMES
052aeb0ee88    redis    "docker-entrypoint.s..." About a minute ago    Up 4 seconds    6379/tcp    container121
1c4472744083    redis    "docker-entrypoint.s..." 6 minutes ago        Up 10 seconds    6379/tcp    modest_herschel
C:\Users\202>
C:\Users\202>
C:\Users\202>docker stop 052aeb0ee88
052aeb0ee88
C:\Users\202>docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED    STATUS    PORTS    NAMES
1c4472744083    redis    "docker-entrypoint.s..." 9 minutes ago        Up 2 minutes     6379/tcp    modest_herschel
```

```
C:\Users\202>docker start 052aeb0ee88
052aeb0ee88
C:\Users\202>docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED    STATUS    PORTS    NAMES
052aeb0ee88    redis    "docker-entrypoint.s..." 5 minutes ago        Up 8 seconds     6379/tcp    container121
1c4472744083    redis    "docker-entrypoint.s..." 10 minutes ago       Up 3 minutes     6379/tcp    modest_herschel
```

```
C:\Users\202>docker rm 052aeb0ee88
052aeb0ee88
```

```
C:\Users\202>docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
redis         latest    170a1e90f843  2 months ago  138MB
```

```
C:\Users\202>docker exec -d 1c4472744083 touch /tmp/execWorks
C:\Users\202>docker exec -it 1c4472744083 bash
root@1c4472744083:/data# |
```

```
C:\Users\202>docker restart 1c4472744083
1c4472744083

C:\Users\202>docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES
1c4472744083   redis    "docker-entrypoint.s..." 13 minutes ago Up 3 seconds  6379/tcp     modest_herschel
```

```
C:\Users\202>docker inspect 1c4472744083
[
  {
    "Id": "1c44727440831475b093dcbf93163064b819bdd9ad8378bb3a4fa847dc411d80",
    "Created": "2024-03-13T03:19:03.418741433Z",
    "Path": "docker-entrypoint.sh",
    "Args": [
      "redis-server"
    ],
    "State": {
      "Status": "running",
      "Running": true,
      "Paused": false,
      "Restarting": false,
      "OOMKilled": false,
      "Dead": false,
      "Pid": 2112,
      "ExitCode": 0,
      "Error": "",
      "StartedAt": "2024-03-13T03:32:13.750463204Z",
      "FinishedAt": "2024-03-13T03:32:13.145321277Z"
    },
    "Image": "sha256:170a1e90f8436daa6778aeea3926e716928826c215ca23a8dfd8055f663f9428",
    "ResolvConfPath": "/var/lib/docker/containers/1c44727440831475b093dcbf93163064b819bdd9a"
```

```
C:\Users\202>docker commit 1c4472744083 new_image_name:redis2
sha256:33e4284a7e92a4a1331555d01f6e078fc496e3a3ed8eb7f84f2678261ad07e83
```

```
C:\Users\202>docker images
REPOSITORY      TAG         IMAGE ID      CREATED        SIZE
new_image_name  redis2      33e4284a7e92  4 seconds ago  138MB
new_image_name  tag         61ab016507fa  36 seconds ago 138MB
redis           latest      170a1e90f843  2 months ago  138MB
```

**Conclusion:** Docker revolutionizes the software development and deployment process by providing a powerful platform for containerization. By encapsulating applications and their dependencies into lightweight, portable containers.