

Lesson 02 Demo 01

Demonstrating Lifecycle of Containers

Objective: To demonstrate the lifecycle of containers for efficient management and optimization of Docker container orchestration

Tools required: Ubuntu

Prerequisites: None

Steps to be followed:

1. Demonstrate Docker container lifecycle management

Step 1: Demonstrate Docker container lifecycle management

- 1.1 Pull a Docker image from Docker hub using the following command:

sudo docker pull nginx

```
labsuser@ip-172-31-1-240:~$ sudo docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
8a1e25ce7c4f: Pull complete
e78b137be355: Pull complete
39fc875bd2b2: Pull complete
035788421403: Pull complete
87c3fb37cbf2: Pull complete
c5cdd1ce752d: Pull complete
33952c599532: Pull complete
Digest: sha256:6db391d1c0cfb30588ba0bf72ea999404f2764febf0f1f196acd5867ac7efa7e
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
labsuser@ip-172-31-1-240:~$
```

I

- 1.2 Execute the following command to run a container from the Nginx image:
sudo docker run -d --name my-nginx -p 8080:80 nginx

```
labsuser@ip-172-31-1-240:~$ sudo docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
8a1e25ce7c4f: Pull complete
e78b137be355: Pull complete
39fc875bd2b2: Pull complete
035788421403: Pull complete
87c3fb37cbf2: Pull complete
c5cdd1ce752d: Pull complete
33952c599532: Pull complete
Digest: sha256:6db391d1c0cfb30588ba0bf72ea999404f2764feb0f1f196acd5867ac7efa7e
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
labsuser@ip-172-31-1-240:~$ sudo docker run -d --name my-nginx -p 8080:80 nginx
3f08374c93a13a17dfcc1a09b921da5011e7d26de6fc439455de3781eb8d838a
labsuser@ip-172-31-1-240:~$
```

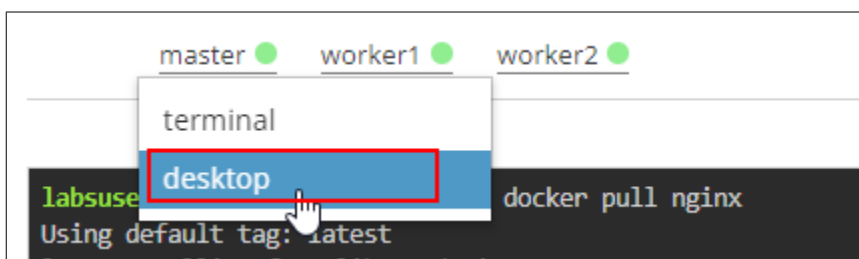
- 1.3 Run the following command to list the running containers:
sudo docker ps

```
labsuser@ip-172-31-1-240:~$ sudo docker run -d --name my-nginx -p 8080:80 nginx
3f08374c93a13a17dfcc1a09b921da5011e7d26de6fc439455de3781eb8d838a
labsuser@ip-172-31-1-240:~$ sudo docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
3f08374c93a1	nginx	"/docker-entrypoint..."	About a minute ago	Up About a minute	0.0.0.0:8080->80/tcp, :::8080->80/tcp	my-nginx

```
labsuser@ip-172-31-1-240:~$
```

- 1.4 Click on the **master** dropdown and select the **desktop** option as shown in the screenshot below:



- 1.5 Open a web browser and navigate to <http://localhost:8080> to view the default Nginx welcome page



- 1.6 Execute the following command to inspect the details of the running container, including its configuration and networking information as shown in the screenshots below:
sudo docker inspect my-nginx

```
labsuser@ip-172-31-1-240:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
3f08374c93a1   nginx    "/docker-entrypoint..." About a minute ago Up About a minute 0.0.0.0:8080->80/tcp, :::8080->80/tcp my-nginx

labsuser@ip-172-31-1-240:~$ sudo docker inspect my-nginx
[
  {
    "Id": "3f08374c93a13a17dfcc1a09b921da5011e7d26de6fc439455de3781eb8d838a",
    "Created": "2024-03-21T01:59:33.800952176Z",
    "Path": "/docker-entrypoint.sh",
    "Args": [
      "nginx",
      "-g",
      "daemon off;"
    ],
    "State": {
      "Status": "running",
      "Running": true,
      "Paused": false,
      "Restarting": false,
      "OOMKilled": false,
```

```

"ExecIDs": null,
"HostConfig": {
  "Binds": null,
  "ContainerIDFile": "",
  "LogConfig": {
    "Type": "json-file",
    "Config": {}
  },
  "NetworkMode": "default",
  "PortBindings": {
    "80/tcp": [
      {
        "HostIp": "",
        "HostPort": "8080"
      }
    ]
  },
  "RestartPolicy": {
    "Name": "no",
    "MaximumRetryCount": 0
  }
}

```

1.7 Run the following command to stop the running container:

sudo docker stop my-nginx

```

"Links": null,
"Aliases": null,
"NetworkID": "793a42f465bad73df46661742d9d83041c785ba9ba71aa45e6fa3e7937949efe",
"EndpointID": "03ded4a2f0466e4059fdccba661a65dc7fea5cb7b5fcc1d527d88aa7f228d68b",
"Gateway": "172.17.0.1",
"IPAddress": "172.17.0.2",
"IPPrefixLen": 16,
"IPv6Gateway": "",
"GlobalIPv6Address": "",
"GlobalIPv6PrefixLen": 0,
"MacAddress": "02:42:ac:11:00:02",
"DriverOpts": null
}
}
}
}
1
labsuser@ip-172-31-1-240:~$ sudo docker stop my-nginx
my-nginx
labsuser@ip-172-31-1-240:~$

```

1.8 Run the following command to list the running containers:

sudo docker ps

```
labsuser@ip-172-31-1-240:~$ sudo docker stop my-nginx
my-nginx
labsuser@ip-172-31-1-240:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED    STATUS    PORTS    NAMES
labsuser@ip-172-31-1-240:~$
```

1.9 Use the following command to start the container as shown in the screenshot below:

sudo docker start my-nginx

```
labsuser@ip-172-31-1-240:~$ sudo docker stop my-nginx
my-nginx
labsuser@ip-172-31-1-240:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED    STATUS    PORTS    NAMES
labsuser@ip-172-31-1-240:~$ sudo docker start my-nginx
my-nginx
labsuser@ip-172-31-1-240:~$
```

1.10 Run the following command to list the running containers:

sudo docker ps

```
labsuser@ip-172-31-1-240:~$ sudo docker stop my-nginx
my-nginx
labsuser@ip-172-31-1-240:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED    STATUS    PORTS    NAMES
labsuser@ip-172-31-1-240:~$ sudo docker start my-nginx
my-nginx
labsuser@ip-172-31-1-240:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED    STATUS    PORTS    NAMES
3f08374c93a1  nginx    "/docker-entrypoint..." 11 minutes ago Up About a minute 0.0.0.0:8080->80/tcp, :::8080->80/tcp my-nginx
labsuser@ip-172-31-1-240:~$
```

1.11 Run the following command to remove the running container:

sudo docker rm my-nginx

```
labsuser@ip-172-31-1-240:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED    STATUS    PORTS    NAMES
3f08374c93a1  nginx    "/docker-entrypoint..." 11 minutes ago Up About a minute 0.0.0.0:8080->80/tcp, :::8080->80/tcp my-nginx
labsuser@ip-172-31-1-240:~$ sudo docker rm my-nginx
Error response from daemon: You cannot remove a running container 3f08374c93a13a17dfcc1a09b921da5011e7d26de6fc439455de3781eb8d838a. Stop the container
before attempting removal or force remove
labsuser@ip-172-31-1-240:~$
```

Note: You need to first stop the running container and then remove it as shown in the next steps.

1.12 Run the following command to stop the running container:

sudo docker stop my-nginx

```
labsuser@ip-172-31-1-240:~$ sudo docker rm my-nginx
Error response from daemon: You cannot remove a running container 3f08374c93a13a17dfcc1a09b921da5011e7d26de6fc439455de3781eb8d838a. Stop the container
before attempting removal or force remove
labsuser@ip-172-31-1-240:~$ sudo docker stop my-nginx
my-nginx
labsuser@ip-172-31-1-240:~$
```

1.13 Use the following command to remove the running container:

sudo docker rm my-nginx

```
labsuser@ip-172-31-1-240:~$ sudo docker stop my-nginx
my-nginx
labsuser@ip-172-31-1-240:~$ sudo docker rm my-nginx
my-nginx
labsuser@ip-172-31-1-240:~$
```

1.14 Check the list of running and stopped containers to ensure that the running container has been removed using the following command:

sudo docker ps -a

```
labsuser@ip-172-31-1-240:~$ sudo docker stop my-nginx
my-nginx
labsuser@ip-172-31-1-240:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED    STATUS    PORTS    NAMES
3f08374c93a1   nginx    "/docker-entrypoint..." 11 minutes ago Up About a minute 0.0.0.0:8080->80/tcp, :::8080->80/tcp my-nginx
labsuser@ip-172-31-1-240:~$ sudo docker start my-nginx
my-nginx
labsuser@ip-172-31-1-240:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED    STATUS    PORTS    NAMES
3f08374c93a1   nginx    "/docker-entrypoint..." 11 minutes ago Up About a minute 0.0.0.0:8080->80/tcp, :::8080->80/tcp my-nginx
labsuser@ip-172-31-1-240:~$ sudo docker rm my-nginx
Error response from daemon: You cannot remove a running container 3f08374c93a13a17dfcc1a09b921da5011e7d26de6fc439455de3781eb8d838a. Stop the container
before attempting removal or force remove
labsuser@ip-172-31-1-240:~$ sudo docker stop my-nginx
my-nginx
labsuser@ip-172-31-1-240:~$ sudo docker rm my-nginx
my-nginx
labsuser@ip-172-31-1-240:~$ sudo docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED    STATUS    PORTS    NAMES
labsuser@ip-172-31-1-240:~$
```

By following these steps, you have successfully demonstrated the lifecycle of containers for efficient management and optimization of Docker container orchestration.