DOCEKR Hands On Lab 3

---

1. Run four HTTPD Docker containers with distinct, meaningful names, and apply restart policies (NO, On-Failure, Always, and Unless-Stopped) to each of the four containers, respectively. Demonstrate that the restart policies function as expected.
Ans:

----NO----

```
root@DESKTOP-Q1VPUEC:~# docker container run -itd --name NO --restart no httpd
95d0a37dbea5cc348fb0fae464d0b208265689653462c3d366755ab01814d191
root@DESKTOP-Q1VPUEC:~# docker container ls
CONTAINER ID   IMAGE     COMMAND              CREATED          STATUS          PORTS     NAMES
95d0a37dbea5   httpd     "httpd-foreground"   15 seconds ago   Up 11 seconds   80/tcp    NO
root@DESKTOP-Q1VPUEC:~# docker container stop NO
NO
root@DESKTOP-Q1VPUEC:~# docker container ls -a
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS                      PORTS     NAMES
95d0a37dbea5   httpd          "httpd-foreground"       41 seconds ago   Exited (0) 6 seconds ago              NO
```

----ON_FAILURE----

```
root@DESKTOP-Q1VPUEC:~# docker container run -itd --name ON_FAILURE --restart on-failure:5 httpd
3711799d5a89924cf2f71e76caed55c75ac7343810a8ce8fd488ab6601fcad29
root@DESKTOP-Q1VPUEC:~# docker container ls
CONTAINER ID   IMAGE     COMMAND              CREATED          STATUS          PORTS     NAMES
3711799d5a89   httpd     "httpd-foreground"   8 seconds ago    Up 5 seconds    80/tcp    ON_FAILURE
root@DESKTOP-Q1VPUEC:~# docker container inspect ON_FAILURE | egrep "RestartPolicy|on-failure|RetryCount"
        "RestartPolicy": {
            "Name": "on-failure",
            "MaximumRetryCount": 5
root@DESKTOP-Q1VPUEC:~# docker container ls
CONTAINER ID   IMAGE     COMMAND              CREATED          STATUS              PORTS     NAMES
3711799d5a89   httpd     "httpd-foreground"   7 minutes ago    Up About a minute   80/tcp    ON_FAILURE
root@DESKTOP-Q1VPUEC:~# ps -ef | grep container
root       331     1  0 11:41 ?        00:00:01 /usr/bin/containerd
root       477     1  0 11:41 ?        00:00:03 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
root      1308     1  0 12:14 ?        00:00:00 /usr/bin/containerd-shim-runc-v2 -namespace moby -id
3711799d5a89924cf2f71e76caed55c75ac7343810a8ce8fd488ab6601fcad29 -address
/run/containerd/containerd.sock
root      1472   732  0 12:18 pts/2    00:00:00 grep --color=auto container
root@DESKTOP-Q1VPUEC:~# kill -9 1308
root@DESKTOP-Q1VPUEC:~# docker container ls
CONTAINER ID   IMAGE     COMMAND              CREATED          STATUS          PORTS     NAMES
3711799d5a89   httpd     "httpd-foreground"   11 minutes ago   Up 8 seconds    80/tcp    ON_FAILURE
```

----ALWAYS----

```
root@DESKTOP-Q1VPUEC:~# docker container run -itd --name ALWAYS --restart always httpd sleep 10
0de80e609920765e2d674db9664a726000b6aca61dd9696163711e802d31bc27
root@DESKTOP-Q1VPUEC:~# docker container ls
CONTAINER ID   IMAGE     COMMAND              CREATED          STATUS          PORTS     NAMES
0de80e609920   httpd     "sleep 10"           7 seconds ago    Up 5 seconds    80/tcp    ALWAYS
```

```
root@DESKTOP-Q1VPUEC:~# docker inspect ALWAYS | egrep "RestartPolicy|always"
        "RestartPolicy": {
            "Name": "always",
root@DESKTOP-Q1VPUEC:~# docker inspect ALWAYS | grep -i restartcount
                                "RestartCount": 10,
root@DESKTOP-Q1VPUEC:~# docker container ls
CONTAINER ID   IMAGE     COMMAND            CREATED         STATUS         PORTS     NAMES
0de80e609920   httpd     "sleep 10"         13 minutes ago   Up 4 seconds    80/tcp    ALWAYS
root@DESKTOP-Q1VPUEC:~# docker container ls
root@DESKTOP-Q1VPUEC:~# docker container stop ALWAYS
ALWAYS
root@DESKTOP-Q1VPUEC:~# docker container ls -a
CONTAINER ID   IMAGE     COMMAND            CREATED         STATUS                  PORTS     NAMES
0de80e609920   httpd     "sleep 10"         11 minutes ago   Exited (0) 5 seconds ago           ALWAYS
3711799d5a89   httpd     "httpd-foreground"   40 minutes ago   Up 29 minutes           80/tcp    ON_FAILURE
```

----UNLESS_STOPPED----

```
root@DESKTOP-Q1VPUEC:~# docker container run -itd --name UNLESS-STOPPED --restart unless-stopped httpd sleep
10
e6d646cd779e43a4b778f77b721d0b23bbe93d6e560c614645f5cf2f8cd5f7e7
root@DESKTOP-Q1VPUEC:~# docker container ls
CONTAINER ID   IMAGE     COMMAND            CREATED         STATUS                         PORTS     NAMES
e6d646cd779e   httpd     "sleep 10"         9 seconds ago    Up 6 seconds                    80/tcp    UNLESS-STOPPED
root@DESKTOP-Q1VPUEC:~# docker inspect UNLESS-STOPPED | grep -i restartcount
      "RestartCount": 4,
root@DESKTOP-Q1VPUEC:~# docker container stop UNLESS-STOPPED
UNLESS-STOPPED
root@DESKTOP-Q1VPUEC:~# docker container ls -a
CONTAINER ID   IMAGE     COMMAND            CREATED           STATUS                  PORTS     NAMES
e6d646cd779e   httpd     "sleep 10"         About a minute ago   Exited (0) 4 seconds ago           UNLESS-STOPPED
root@DESKTOP-Q1VPUEC:~# docker container restart UNLESS-STOPPED
UNLESS-STOPPED
root@DESKTOP-Q1VPUEC:~# docker container ls -a
CONTAINER ID   IMAGE     COMMAND            CREATED         STATUS         PORTS     NAMES
e6d646cd779e   httpd     "sleep 10"         2 minutes ago    Up 2 seconds    80/tcp    UNLESS-STOPPED
root@DESKTOP-Q1VPUEC:~# docker inspect UNLESS-STOPPED | egrep "RestartPolicy|unless-stopped"
        "RestartPolicy": {
            "Name": "unless-stopped",
root@DESKTOP-Q1VPUEC:~# docker inspect UNLESS-STOPPED | grep -i "restartcount"
      "RestartCount": 19,
```

---

2. Change the restart policy of a above running container from the default to a custom policy using the docker update command.
e.g. docker update --help
Ans:

```
root@DESKTOP-Q1VPUEC:~# docker inspect NO | egrep "RestartPolicy|no"
        "RestartPolicy": {
            "Name": "no",
root@DESKTOP-Q1VPUEC:~# docker update --restart=always NO
NO
```

```
root@DESKTOP-Q1VPUEC:~# docker inspect NO | egrep "RestartPolicy|always"
        "RestartPolicy": {
            "Name": "always",
```

---

3. Launch an NGINX container with a meaningful name and expose it on the host's port 80. Create an "index.html" file containing the text "Hello there, Let's be the Team CloudEthiX," and copy the file to the container's "/usr/share/nginx/html/" location. Access the container in a browser to verify that the webpage displays correctly.
Ans:

```
root@DESKTOP-Q1VPUEC:~# docker container run -itd -p 8080:80 --name NEW nginx
247559b5459a7bfa77911032bbe25c47e726f0c8217c21b1b485010d5c2fbf24
root@DESKTOP-Q1VPUEC:~# docker container ls -a
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS                        PORTS
NAMES
247559b5459a   nginx     "/docker-entrypoint.…"   11 seconds ago   Up 9 seconds                  0.0.0.0:8080-
>80/tcp, :::8080->80/tcp   NEW
root@DESKTOP-Q1VPUEC:~# vim index.html
root@DESKTOP-Q1VPUEC:~# docker container cp index.html NEW:/usr/share/nginx/html/
Successfully copied 2.05kB to NEW:/usr/share/nginx/html/
root@DESKTOP-Q1VPUEC:~# docker container exec -it NEW /bin/bash
root@247559b5459a:/# cd /usr/share/nginx/html/
root@247559b5459a:/usr/share/nginx/html# ls
50x.html  index.html
root@247559b5459a:/usr/share/nginx/html# cat index.html
Hello there, Let's be the Team CloudEthiX.
root@247559b5459a:/usr/share/nginx/html# read escape sequence
root@DESKTOP-Q1VPUEC:~# docker inspect NEW
                "Gateway": "172.17.0.1",
                "IPAddress": "172.17.0.7",

root@DESKTOP-Q1VPUEC:~# curl 172.17.0.7
Hello there, Let's be the Team CloudEthiX.
```

---

4. Run a docker container with CPU and Memory limit. docker container run --help
Ans:

```
root@DESKTOP-Q1VPUEC:~# docker container run -itd -m 200m --memory-reservation=100m --cpus=2 --cpu-
shares=20 --name LOAD nginx
d1f2ba7dbb4430f6a674ab613c601e93f5071c57f33c98ad6b82a29dffa5ca10
root@DESKTOP-Q1VPUEC:~# docker container ls
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS                NAMES
d1f2ba7dbb44   nginx     "/docker-entrypoint.…"   10 seconds ago   Up 8 seconds    80/tcp
LOAD
root@DESKTOP-Q1VPUEC:~# docker stats LOAD
CONTAINER ID   NAME      CPU %     MEM USAGE / LIMIT   MEM %     NET I/O     BLOCK I/O   PIDS
d1f2ba7dbb44   LOAD      0.00%     3.027MiB / 200MiB   1.51%     656B / 0B   0B / 0B     3
root@DESKTOP-Q1VPUEC:~# docker container exec -it LIMITED lscpu
Architecture:            x86_64
  CPU op-mode(s):        32-bit, 64-bit
```

```
  Address sizes:        48 bits physical, 48 bits virtual
  Byte Order:           Little Endian
CPU(s):                 2
```

---

5. Update CUP and Memory of docker container using docker update.
Ans:

```
root@DESKTOP-Q1VPUEC:~# docker container run -itd -m 200m --cpus=1 --name LIMITED nginx
190ade9bd13ab3ce6ed20023a59b89b7a01bee569c15d759e1d78adf9693a040
root@DESKTOP-Q1VPUEC:~# docker container ls -a
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS     NAMES
190ade9bd13a   nginx     "/docker-entrypoint.…"   5 minutes ago   Up 5 minutes   80/tcp    LIMITED

root@DESKTOP-Q1VPUEC:~# docker stats LIMITED
CONTAINER ID   NAME      CPU %     MEM USAGE / LIMIT   MEM %     NET I/O       BLOCK I/O   PIDS
190ade9bd13a   LIMITED   0.00%     3.008MiB / 200MiB   1.50%     1.02kB / 0B   0B / 0B     3

root@DESKTOP-Q1VPUEC:~# docker update -m 300m --cpus=2 LIMITED
LIMITED
root@DESKTOP-Q1VPUEC:~# docker stats LIMITED
CONTAINER ID   NAME      CPU %     MEM USAGE / LIMIT   MEM %     NET I/O       BLOCK I/O   PIDS
190ade9bd13a   LIMITED   0.00%     3.008MiB / 300MiB   1.00%     1.02kB / 0B   0B / 0B     3
root@DESKTOP-Q1VPUEC:~# docker container exec -it LIMITED lscpu
Architecture:          x86_64
  CPU op-mode(s):      32-bit, 64-bit
  Address sizes:       48 bits physical, 48 bits virtual
  Byte Order:          Little Endian
CPU(s):                2
```
