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

root@DESKTOP-PDOJQQA:ubuntu-nginx# **docker container run -itd --name NO --restart no httpd**

69af6a7780797548e3658ef88869fd20ac60732f01846d09a4bb1e7335ef4cfa

root@DESKTOP-PDOJQQA:ubuntu-nginx# **docker container ls -a**

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

69af6a778079 httpd "httpd-foreground" 10 seconds ago Up 10 seconds 80/tcp NO

root@DESKTOP-PDOJQQA:ubuntu-nginx# **docker inspect 69af6a778079**

"NetworkMode": "default",

"PortBindings": {},

"RestartPolicy": {

**"Name": "no",**

**"MaximumRetryCount": 0**

},

root@DESKTOP-PDOJQQA:ubuntu-nginx**# docker container run -itd --name ON\_FAILURE --restart on-failure:5 httpd**

32bf6943413ae8686fe052bdfa7a87538cd4d357542218077f5565b0e1580059

root@DESKTOP-PDOJQQA:ubuntu-nginx# **docker container ls -a**

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

**32bf6943413a** httpd "httpd-foreground" 6 seconds ago Up 6 seconds 80/tcp ON\_FAILURE

69af6a778079 httpd "httpd-foreground" 10 minutes ago Up 10 minutes 80/tcp NO

root@DESKTOP-PDOJQQA:ubuntu-nginx# **docker inspect 32bf6943413a**

"NetworkMode": "default",

"PortBindings": {},

"RestartPolicy": {

**"Name": "on-failure",**

**"MaximumRetryCount": 5**

root@DESKTOP-PDOJQQA:ubuntu-nginx# **docker container run -itd --name always-policy --restart always httpd sleep 10**

b2717980a4f7610c0715c2e0ac445d9824fb03eec860eb9d509fd52f99cd907e

root@DESKTOP-PDOJQQA:ubuntu-nginx# **docker container ls -a**

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

**b2717980a4f7**  httpd "sleep 10" 6 seconds ago Up 5 seconds 80/tcp always-policy

32bf6943413a httpd "httpd-foreground" 10 minutes ago Up 10 minutes 80/tcp ON\_FAILURE

69af6a778079 httpd "httpd-foreground" 21 minutes ago Up 21 minutes 80/tcp NO

root@DESKTOP-PDOJQQA:ubuntu-nginx# **docker inspect b2717980a4f7**

"NetworkMode": "default",

"PortBindings": {},

"RestartPolicy": {

**"Name": "always",**

**"MaximumRetryCount": 0**

root@DESKTOP-PDOJQQA:ubuntu-nginx# **docker container ls -a**

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

b2717980a4f7 httpd "sleep 10" 5 minutes ago **Up 2 seconds**  80/tcp always-policy

32bf6943413a httpd "httpd-foreground" 15 minutes ago Up 15 minutes 80/tcp ON\_FAILURE

69af6a778079 httpd "httpd-foreground" 26 minutes ago Up 26 minutes 80/tcp NO

root@DESKTOP-PDOJQQA:ubuntu-nginx# docker container ls -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

b2717980a4f7 httpd "sleep 10" 5 minutes ago **Up 4 seconds** 80/tcp always-policy

32bf6943413a httpd "httpd-foreground" 15 minutes ago Up 15 minutes 80/tcp ON\_FAILURE

69af6a778079 httpd "httpd-foreground" 26 minutes ago Up 26 minutes 80/tcp NO

root@DESKTOP-PDOJQQA:ubuntu-nginx# docker container ls -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

b2717980a4f7 httpd "sleep 10" 5 minutes ago **Up 5 seconds** 80/tcp always-policy

32bf6943413a httpd "httpd-foreground" 15 minutes ago Up 15 minutes 80/tcp ON\_FAILURE

69af6a778079 httpd "httpd-foreground" 26 minutes ago Up 26 minutes 80/tcp NO

root@DESKTOP-PDOJQQA:ubuntu-nginx# **docker inspect always-policy | grep -i restartcount**

"RestartCount": 47,

root@DESKTOP-PDOJQQA:ubuntu-nginx# **docker container run -itd --name unless-stopped --restart unless-stopped httpd sleep 10**

0c44dc3d8a54ec997fd6695fba65a704610b41ed43137c0e77fe00189e764b29

root@DESKTOP-PDOJQQA:ubuntu-nginx# **docker container ls -a**

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

**0c44dc3d8a54** httpd "sleep 10" 5 seconds ago Up 4 seconds 80/tcp unless-stopped

b2717980a4f7 httpd "sleep 10" 9 minutes ago Up 7 seconds 80/tcp always-policy

32bf6943413a httpd "httpd-foreground" 20 minutes ago Up 20 minutes 80/tcp ON\_FAILURE

69af6a778079 httpd "httpd-foreground" 31 minutes ago Up 31 minutes 80/tcp NO

root@DESKTOP-PDOJQQA:ubuntu-nginx# **docker inspect 0c44dc3d8a54**

"RestartPolicy": {

**"Name": "unless-stopped",**

**"MaximumRetryCount": 0**

root@DESKTOP-PDOJQQA:ubuntu-nginx# **docker container ls -a**

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

0c44dc3d8a54 httpd "sleep 10" About a minute ago Up 7 seconds 80/tcp unless-stopped

0c44dc3d8a54 httpd "sleep 10" 2 minutes ago Up 10 seconds 80/tcp unless-stopped

0c44dc3d8a54 httpd "sleep 10" 2 minutes ago Up Less than a second 80/tcp unless-stopped

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

root@DESKTOP-PDOJQQA:ubuntu-nginx# **docker container ls -a**

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

0c44dc3d8a54 httpd "sleep 10" 3 hours ago Up 4 seconds 80/tcp unless-stopped

b2717980a4f7 httpd "sleep 10" 3 hours ago Up 5 seconds 80/tcp always-policy

32bf6943413a httpd "httpd-foreground" 3 hours ago Up 3 hours 80/tcp ON\_FAILURE

69af6a778079 httpd "httpd-foreground" 3 hours ago Up 3 hours 80/tcp NO

root@DESKTOP-PDOJQQA:ubuntu-nginx# **docker update --restart=always NO b2717980a4f7**

**NO**

b2717980a4f7

root@DESKTOP-PDOJQQA:ubuntu-nginx# docker inspect NO | grep "RestartPolicy | NO"

root@DESKTOP-PDOJQQA:ubuntu-nginx# docker inspect NO | grep "RestartPolicy|NO"

root@DESKTOP-PDOJQQA:ubuntu-nginx# docker inspect NO | grep "RestartPolicy|no"

root@DESKTOP-PDOJQQA:ubuntu-nginx# docker inspect NO | egrep "RestartPolicy|no"

**"RestartPolicy": {**

**"NanoCpus": 0,**

root@DESKTOP-PDOJQQA:ubuntu-nginx# docker update --restart=always NO

NO

root@DESKTOP-PDOJQQA:ubuntu-nginx# **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.

# root@DESKTOP-PDOJQQA:ubuntu-nginx# docker container run -itd -p 8080:80 --name NEW nginx

# Unable to find image 'nginx:latest' locally

# latest: Pulling from library/nginx

# af107e978371: Already exists

# 336ba1f05c3e: Pull complete

# 8c37d2ff6efa: Pull complete

# 51d6357098de: Pull complete

# 782f1ecce57d: Pull complete

# 5e99d351b073: Pull complete

# 7b73345df136: Pull complete

# Digest: sha256:2bdc49f2f8ae8d8dc50ed00f2ee56d00385c6f8bc8a8b320d0a294d9e3b49026

# Status: Downloaded newer image for nginx:latest

# 2e4dc5eee1e22460effb4bac6d5904a9cdef0e22d87fd7e10d8a0aabc33e8d7c

# root@DESKTOP-PDOJQQA:ubuntu-nginx# docker container ls -a

# CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

# 2e4dc5eee1e2 nginx "/docker-entrypoint.…" 3 minutes ago Up 3 minutes 0.0.0.0:8080->80/tcp, :::8080->80/tcp NEW

# root@DESKTOP-PDOJQQA:ubuntu-nginx# vi index.html

# root@DESKTOP-PDOJQQA:ubuntu-nginx# docker container cp index.html NEW:/usr/share/nginx/html/

# Successfully copied 2.05kB to NEW:/usr/share/nginx/html/

# root@DESKTOP-PDOJQQA:ubuntu-nginx# docker container exec -it NEW /bin/bash

# root@2e4dc5eee1e2:/# cd usr/share/nginx/html/

# root@2e4dc5eee1e2:/usr/share/nginx/html# ls

# 50x.html index.html

# root@2e4dc5eee1e2:/usr/share/nginx/html# cat index.html

# HI

# Devops

# root@DESKTOP-PDOJQQA:ubuntu-nginx# curl 172.17.0.6

# HI

# Devops

# 4. Run a docker container with CPU and Memory limit.  docker container run --help      ref links :- <https://phoenixnap.com/kb/docker-memory-and-cpu-limit>                      <https://www.baeldung.com/ops/docker-memory-limit>

# root@DESKTOP-PDOJQQA:ubuntu-nginx# docker container run -itd -m 100m --memory-reservation=100m --cpus=2 --cpu-shares=20 --name LOAD nginx

# 085c16a007caeeadc883d0bf8c838718f9f301507df955da238c59eb3c50ef5a

# root@DESKTOP-PDOJQQA:ubuntu-nginx# docker container ls -a

# CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

# 085c16a007ca nginx "/docker-entrypoint.…" 7 seconds ago Up 6 seconds 80/tcp LOAD

# root@DESKTOP-PDOJQQA:ubuntu-nginx# docker stats LOAD

# CONTAINER ID NAME CPU % MEM USAGE / LIMIT MEM % NET I/O BLOCK I/O PIDS

# 085c16a007ca LOAD 0.00% 4.414MiB / 100MiB 4.41% 656B / 0B 0B / 0B 5

# CONTAINER ID NAME CPU % MEM USAGE / LIMIT MEM % NET I/O BLOCK I/O PIDS

# 085c16a007ca LOAD 0.00% 4.414MiB / 100MiB 4.41% 726B / 0B 0B / 0B 5

# root@DESKTOP-PDOJQQA:ubuntu-nginx# docker container exec -it 085c16a007ca lscpu

# Architecture: x86\_64

# CPU op-mode(s): 32-bit, 64-bit

# Address sizes: 39 bits physical, 48 bits virtual

# Byte Order: Little Endian

# CPU(s): 4

# On-line CPU(s) list: 0-3

# Vendor ID: GenuineIntel

# Model name: Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz

# CPU family: 6

# Model: 78

# Thread(s) per core: 2

# Core(s) per socket: 2

# Socket(s): 1

# Stepping: 3

# BogoMIPS: 4800.00

# Flags: fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ss ht sysc

# all nx pdpe1gb rdtscp lm constant\_tsc arch\_perfmon rep\_good nopl xtopology cpuid pni pclmulqdq ssse3 fma cx1

# 5.  Update CUP and Memory of docker container using docker update.  ref links :- <https://docs.docker.com/engine/reference/commandline/update/>

# root@DESKTOP-PDOJQQA:ubuntu-nginx# docker container run -itd -m 100m --cpus=2 --name LIMITED nginx

# 37a12830b76ab73852a9c818b849ab01dd367d788ce0e0252a5c37db5b5e59c0

# root@DESKTOP-PDOJQQA:ubuntu-nginx# docker container ls -a

# CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

# 37a12830b76a nginx "/docker-entrypoint.…" 6 seconds ago Up 5 seconds 80/tcp LIMITED

# 085c16a007ca nginx "/docker-entrypoint.…" 14 minutes ago Up 14 minutes 80/tcp LOAD

# root@DESKTOP-PDOJQQA:ubuntu-nginx# docker stats LIMITED

# CONTAINER ID NAME CPU % MEM USAGE / LIMIT MEM % NET I/O BLOCK I/O PIDS

# 37a12830b76a LIMITED 0.00% 4.414MiB / 100MiB 4.41% 656B / 0B 0B / 0B 5

# CONTAINER ID NAME CPU % MEM USAGE / LIMIT MEM % NET I/O BLOCK I/O PIDS

# 37a12830b76a LIMITED 0.00% 4.414MiB / 100MiB 4.41% 656B / 0B 0B / 0B 5

# root@DESKTOP-PDOJQQA:ubuntu-nginx# docker update -m 100m --cpus=2 LIMITED

# LIMITED

# root@DESKTOP-PDOJQQA:ubuntu-nginx# docker stats LIMITED

# CONTAINER ID NAME CPU % MEM USAGE / LIMIT MEM % NET I/O BLOCK I/O PIDS

# 37a12830b76a LIMITED 0.00% 4.414MiB / 100MiB 4.41% 796B / 0B 0B / 0B 5

# CONTAINER ID NAME CPU % MEM USAGE / LIMIT MEM % NET I/O BLOCK I/O PIDS

# 37a12830b76a LIMITED 0.00% 4.414MiB / 100MiB 4.41% 796B / 0B 0B / 0B 5

# root@DESKTOP-PDOJQQA:ubuntu-nginx# docker container exec -it 085c16a007ca lscpu

# Architecture: x86\_64

# CPU op-mode(s): 32-bit, 64-bit

# Address sizes: 39 bits physical, 48 bits virtual

# Byte Order: Little Endian

# CPU(s): 4

# On-line CPU(s) list: 0-3

# Vendor ID: GenuineIntel

# Model name: Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz

# CPU family: 6

# Model: 78

# Thread(s) per core: 2

# Core(s) per socket: 2

# Socket(s): 1

# Stepping: 3

# BogoMIPS: 4800.00