Assignment

NOTE: Do not forget to see manual page using "--help" option in command when searching for options/commands for a particular task.

1. Install Docker, either on your native OS or on a VM. Make sure it runs. type "docker -v" to check if it's installed.

try below commands for help

docker --help ---> This command shows all available options and commands to work with images and containers

docker images --help ---> This command shows all the avaialble options and commands to work with docker images

docker ps --help ---> This command shows all the avaialble options and commands to work with docker containers

--->

C:\Users\Neha Kala>docker -v

Docker version 20.10.6, build 370c289

C:\Users\Neha Kala>docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

netdata/netdata latest adac7fe94f33 9 days ago 328MB

nehakala/myapp latest a6e5f8b09c22 10 days ago 72.7MB

gcr.io/k8s-minikube/kicbase v0.0.22 bcd131522525 11 days ago 1.09GB

docker101tutorial latest 27089658d4c0 11 days ago 28MB

jenkins/jenkins latest 2acc5493bcb8 12 days ago 580MB

busybox latest c55b0f125dc6 13 days ago 1.24MB

httpd latest 45561c1ef88a 13 days ago 138MB

redis latest ccee4cdf984f 13 days ago 105MB

ubuntu latest 7e0aa2d69a15 3 weeks ago 72.7MB

alpine/git latest c99c7d810bc1 3 weeks ago 25.1MB

nginx latest 62d49f9bab67 4 weeks ago 133MB

docker/getting-started latest 3ba8f2ff0727 8 weeks ago 27.9MB

hello-world latest d1165f221234 2 months ago 13.3kB

anapsix/nyancat latest 5bd1209bb5e2 5 years ago 2.77MB

C:\Users\Neha Kala>docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

--------------------------------------------------------------------

C:\Users\Neha Kala>docker ps --help

Usage: docker ps [OPTIONS]

List containers

Options:

-a, --all Show all containers (default shows just running)

-f, --filter filter Filter output based on conditions provided

--format string Pretty-print containers using a Go template

-n, --last int Show n last created containers (includes all

states) (default -1)

-l, --latest Show the latest created container (includes all

states)

--no-trunc Don't truncate output

-q, --quiet Only display container IDs

-s, --size Display total file sizes

-----------------------------------------------------------------------

C:\Users\Neha Kala>docker images --help

Usage: docker images [OPTIONS] [REPOSITORY[:TAG]]

List images

Options:

-a, --all Show all images (default hides intermediate images)

--digests Show digests

-f, --filter filter Filter output based on conditions provided

--format string Pretty-print images using a Go template

--no-trunc Don't truncate output

-q, --quiet Only show image IDs

-----------------------------------------------------------------------

C:\Users\Neha Kala>docker --help

Usage: docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Options:

--config string Location of client config files (default

"C:\\Users\\Neha Kala\\.docker")

-c, --context string Name of the context to use to connect to the

daemon (overrides DOCKER\_HOST env var and

default context set with "docker context use")

-D, --debug Enable debug mode

-H, --host list Daemon socket(s) to connect to

-l, --log-level string Set the logging level

("debug"|"info"|"warn"|"error"|"fatal")

(default "info")

--tls Use TLS; implied by --tlsverify

--tlscacert string Trust certs signed only by this CA (default

"C:\\Users\\Neha Kala\\.docker\\ca.pem")

--tlscert string Path to TLS certificate file (default

"C:\\Users\\Neha Kala\\.docker\\cert.pem")

--tlskey string Path to TLS key file (default "C:\\Users\\Neha

Kala\\.docker\\key.pem")

--tlsverify Use TLS and verify the remote

-v, --version Print version information and quit

Management Commands:

app\* Docker App (Docker Inc., v0.9.1-beta3)

builder Manage builds

buildx\* Build with BuildKit (Docker Inc., v0.5.1-docker)

compose\* Docker Compose (Docker Inc., 2.0.0-beta.1)

config Manage Docker configs

container Manage containers

context Manage contexts

image Manage images

manifest Manage Docker image manifests and manifest lists

network Manage networks

node Manage Swarm nodes

plugin Manage plugins

scan\* Docker Scan (Docker Inc., v0.8.0)

secret Manage Docker secrets

service Manage services

stack Manage Docker stacks

swarm Manage Swarm

system Manage Docker

trust Manage trust on Docker images

volume Manage volumes

Commands:

attach Attach local standard input, output, and error streams to a running container

build Build an image from a Dockerfile

commit Create a new image from a container's changes

cp Copy files/folders between a container and the local filesystem

create Create a new container

diff Inspect changes to files or directories on a container's filesystem

events Get real time events from the server

exec Run a command in a running container

export Export a container's filesystem as a tar archive

history Show the history of an image

images List images

import Import the contents from a tarball to create a filesystem image

info Display system-wide information

inspect Return low-level information on Docker objects

kill Kill one or more running containers

load Load an image from a tar archive or STDIN

login Log in to a Docker registry

logout Log out from a Docker registry

logs Fetch the logs of a container

pause Pause all processes within one or more containers

port List port mappings or a specific mapping for the container

ps List containers

pull Pull an image or a repository from a registry

push Push an image or a repository to a registry

rename Rename a container

restart Restart one or more containers

rm Remove one or more containers

rmi Remove one or more images

run Run a command in a new container

save Save one or more images to a tar archive (streamed to STDOUT by default)

search Search the Docker Hub for images

start Start one or more stopped containers

stats Display a live stream of container(s) resource usage statistics

stop Stop one or more running containers

tag Create a tag TARGET\_IMAGE that refers to SOURCE\_IMAGE

top Display the running processes of a container

unpause Unpause all processes within one or more containers

update Update configuration of one or more containers

version Show the Docker version information

wait Block until one or more containers stop, then print their exit codes

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. Find a image from dockerhub of your choice(recommeded: nginx), don't use browser, pull the official image from dockerhub

--->

C:\Users\Neha Kala>docker pull nginx

Using default tag: latest

latest: Pulling from library/nginx

69692152171a: Pull complete

49f7d34d62c1: Pull complete

5f97dc5d71ab: Pull complete

cfcd0711b93a: Pull complete

be6172d7651b: Pull complete

de9813870342: Pull complete

Digest: sha256:df13abe416e37eb3db4722840dd479b00ba193ac6606e7902331dcea50f4f1f2

Status: Downloaded newer image for nginx:latest

docker.io/library/nginx:latest

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. List all the available images in your machine/vm, make sure you see recently pulled image in the list.

and

4. Find out the "Full" ImageId of the image that you pulled and write it below.

--->

C:\Users\Neha Kala>docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

nginx latest f0b8a9a54136 4 days ago 133MB

netdata/netdata latest adac7fe94f33 9 days ago 328MB

nehakala/myapp latest a6e5f8b09c22 10 days ago 72.7MB

gcr.io/k8s-minikube/kicbase v0.0.22 bcd131522525 11 days ago 1.09GB

docker101tutorial latest 27089658d4c0 11 days ago 28MB

jenkins/jenkins latest 2acc5493bcb8 12 days ago 580MB

busybox latest c55b0f125dc6 13 days ago 1.24MB

httpd latest 45561c1ef88a 13 days ago 138MB

redis latest ccee4cdf984f 13 days ago 105MB

ubuntu latest 7e0aa2d69a15 3 weeks ago 72.7MB

alpine/git latest c99c7d810bc1 3 weeks ago 25.1MB

docker/getting-started latest 3ba8f2ff0727 8 weeks ago 27.9MB

hello-world latest d1165f221234 2 months ago 13.3kB

anapsix/nyancat latest 5bd1209bb5e2 5 years ago 2.77MB

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5. Create a container of your image

--->

C:\Users\Neha Kala>docker container create nginx

884f715d9ef450b7b569e86284075d8ac4c746c0ca49c0a51c68c0b85dcbbc60

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

6. List all the running containers

--->docker ps

PS C:\WINDOWS\system32> docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

8f2a4ccd20eb nginx "/docker-entrypoint.…" 16 minutes ago Up 12 minutes 80/tcp optimistic\_mcclintock

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

7. List all the running and stopped containers

--->docker ps -a

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

8. Find out the "Full" containerId of the container and write it below.

--->

PS C:\WINDOWS\system32> docker inspect --format="{{.Id}}" nginx

sha256:f0b8a9a541369db503ff3b9d4fa6de561b300f7363920c2bff4577c6c24c5cf6

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

9. Find out how many image layers are used to build this image.

--->

PS C:\WINDOWS\system32> docker image inspect nginx -f '{{.RootFS.Layers}}'

[sha256:02c055ef67f5904019f43a41ea5f099996d8e7633749b6e606c400526b2c4b33

sha256:431f409d4c5a8f79640000705665407ff22d73e043472cb1521faa6d83afc5e8

sha256:4b8db2d7f35aa38ac283036f2c7a453ebfdcc8d7e83a2bf3b55bf8847f8fafaf

sha256:c9732df61184e9e8d08f96c6966190c59f507d8f57ea057a4610f145c59e9bc4

sha256:eeb14ff930d4c2c04ece429112c16a536985f0cba6b13fdb52b00853107ab9c4

sha256:f0f30197ccf95e395bbf4efd65ec94b9219516ae5cafe989df4cf220eb1d6dfa]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

10. Get the Apache Tomcat 7 server image from the docker hub.

--->docker pull tomcat:7.0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

11. Run the Apache Tomcat 7, I mean create a container of Apache Tomcat.

--->docker container start 9e09cb521853

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

12. Find out what is the IP Address of the Apache Tomcat Container that it is running on

and

13. Which Port it is using?

--->

PS C:\WINDOWS\system32> docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

2dc584e1c0e6 tomcat:8.0 "catalina.sh run" 12 minutes ago Up 11 minutes 0.0.0.0:8082->8080/tcp, :::8082->8080/tcp my-tomcat-container

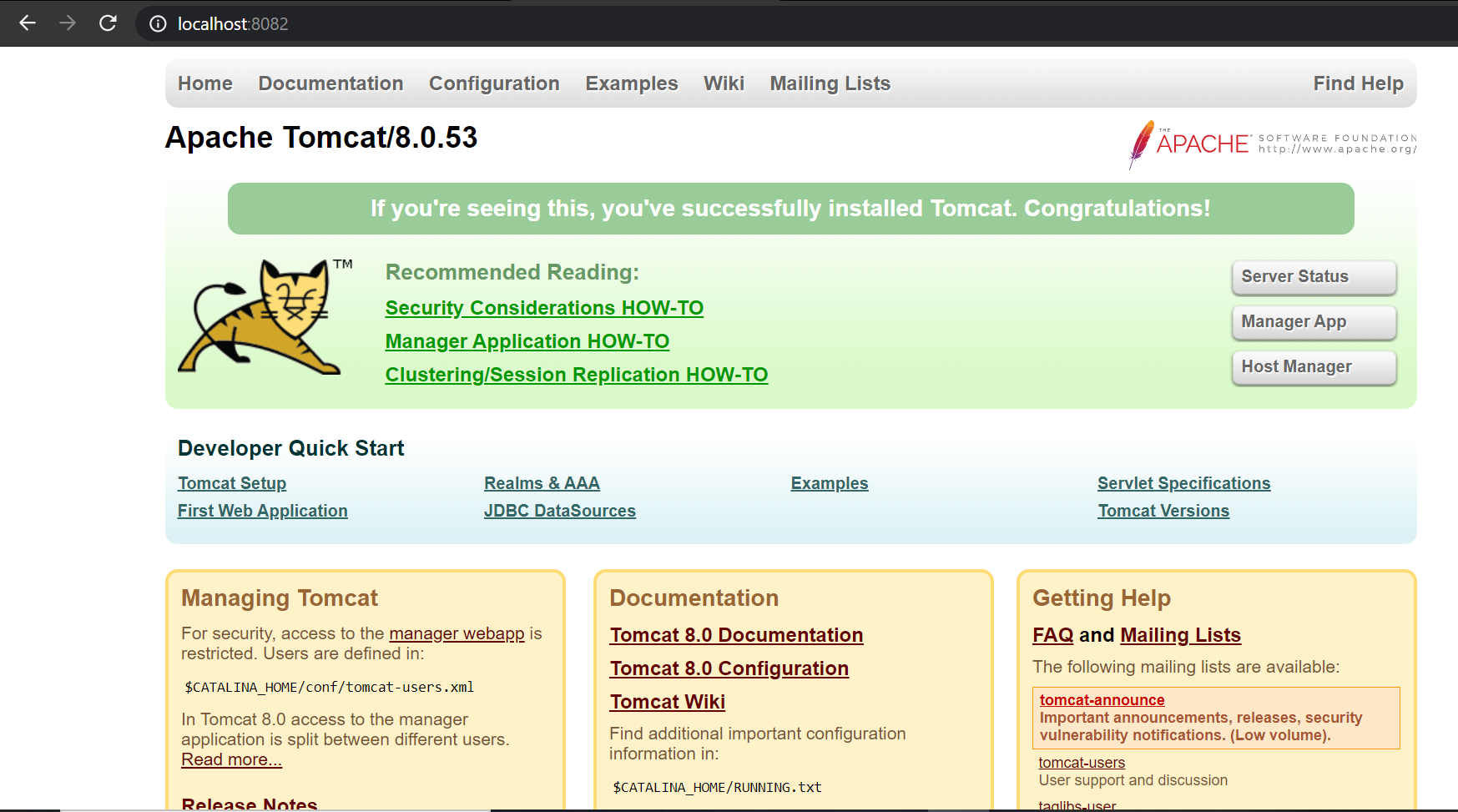
PS C:\WINDOWS\system32> docker inspect --format '{{ .NetworkSettings.IPAddress }}' 2dc584e1c0e6

172.17.0.2

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

14. Try to access the Tomcat's home page from your machine/vm.

--->



PS C:\WINDOWS\system32> docker image pull tomcat:8.0

8.0: Pulling from library/tomcat

f189db1b88b3: Pull complete

3d06cf2f1b5e: Pull complete

edd0da9e3091: Pull complete

eb7768aae14e: Pull complete

e2780f585e0f: Pull complete

e5ed720afeba: Pull complete

d9e134700cfc: Pull complete

e4804b33d02a: Pull complete

b9df0c24315e: Pull complete

49fdae8eaa20: Pull complete

1aea3d9a32e6: Pull complete

Digest: sha256:8ecb10948deb32c34aeadf7bf95d12a93fbd3527911fa629c1a3e7823b89ce6f

Status: Downloaded newer image for tomcat:8.0

docker.io/library/tomcat:8.0

PS C:\WINDOWS\system32> docker image ls

REPOSITORY TAG IMAGE ID CREATED SIZE

nginx latest f0b8a9a54136 5 days ago 133MB

myimage 1.0 a6e5f8b09c22 10 days ago 72.7MB

nehakala/myapp latest a6e5f8b09c22 10 days ago 72.7MB

gcr.io/k8s-minikube/kicbase v0.0.22 bcd131522525 11 days ago 1.09GB

tomcat 8.0 ef6a7c98d192 2 years ago 356MB

PS C:\WINDOWS\system32> docker container create --publish 8082:8080 --name my-tomcat-container tomcat:8.0

2dc584e1c0e604e1a428df2d0ebec589e3ef20b0c2326dce0cd13ef2504dbe57

PS C:\WINDOWS\system32> docker container start my-tomcat-container

my-tomcat-container

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

15. What is the disk size of Apache Tomcat image?

--->docker images

356MB

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

16. Find out list of all environment variables that is configured for tomcat image, can you see JAVA\_HOME and CATALINA\_HOME? What did you notice about it?

--->

17. Find out which port is exposed for tomcat?

--->8082

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

18. Run multiple conntainers of tomcat on different port and access it's home page.

--->

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

19. Pull ubuntu os from dockerhub, try to pull 2 images of ubuntu, Except the latest one.

--->

C:\Users\Neha Kala>docker pull ubuntu:xenial

xenial: Pulling from library/ubuntu

92473f7ef455: Pull complete

fb52bde70123: Pull complete

64788f86be3f: Pull complete

33f6d5f2e001: Pull complete

Digest: sha256:eed7e1076bbc1f342c4474c718e5438af4784f59a4e88ad687dbb98483b59ee4

Status: Downloaded newer image for ubuntu:xenial

docker.io/library/ubuntu:xenial

C:\Users\Neha Kala>docker pull ubuntu:rolling

rolling: Pulling from library/ubuntu

c830499a6a92: Pull complete

b38f134463e2: Pull complete

2fd6a415fd8e: Pull complete

Digest: sha256:be154cc2b1211a9f98f4d708f4266650c9129784d0485d4507d9b0fa05d928b6

Status: Downloaded newer image for ubuntu:rolling

docker.io/library/ubuntu:rolling

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

20. Run the container of ubuntu in attached mode.

--->

PS C:\WINDOWS\system32> docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

nginx latest f0b8a9a54136 5 days ago 133MB

myimage 1.0 a6e5f8b09c22 11 days ago 72.7MB

nehakala/myapp latest a6e5f8b09c22 11 days ago 72.7MB

gcr.io/k8s-minikube/kicbase v0.0.22 bcd131522525 11 days ago 1.09GB

ubuntu xenial aefd7f02ae24 3 weeks ago 134MB

ubuntu rolling 01244e8938d1 3 weeks ago 74.1MB

tomcat 8.0 ef6a7c98d192 2 years ago 356MB

PS C:\WINDOWS\system32> docker run -it aefd7f02ae24

root@87ec73e6e331:/#

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

21. Run the container of another ubuntu in detached mode.

--->

PS C:\WINDOWS\system32> docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

nginx latest f0b8a9a54136 5 days ago 133MB

myimage 1.0 a6e5f8b09c22 11 days ago 72.7MB

nehakala/myapp latest a6e5f8b09c22 11 days ago 72.7MB

gcr.io/k8s-minikube/kicbase v0.0.22 bcd131522525 11 days ago 1.09GB

ubuntu xenial aefd7f02ae24 3 weeks ago 134MB

ubuntu rolling 01244e8938d1 3 weeks ago 74.1MB

tomcat 8.0 ef6a7c98d192 2 years ago 356MB

PS C:\WINDOWS\system32> docker run -it -d 01244e8938d1

761be7034997dc7d9ed36eb13cc3d73bd4e51052eb1eee415ea7ad48a4ea55e0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

22. Check how many ubuntu containers are running and stopped

-->

PS C:\WINDOWS\system32> docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

7e8879be9421 aefd7f02ae24 "/bin/bash" 53 seconds ago Exited (0) 43 seconds ago vibrant\_noyce

761be7034997 01244e8938d1 "/bin/bash" 2 minutes ago Up 2 minutes gifted\_booth

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

23. Is the tomcat container running? If no, start one.

24. Check the logs, generated by tomcat container(don't forget to make request to tomcat's home page to see the log).

-->docker logs 2dc584e1c0e6

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

25. Check if ubuntu conatiner is running? If no, start one in attached mode to the terminal.

26. Login as root user in ubuntu container

--->

PS C:\WINDOWS\system32> docker exec -it 7e8879be9421 bash

root@7e8879be9421:/#

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

27. Create a file with any name in root directory

28. Install software of your choice in ubuntu container using "apt-get install"

29. Now exit the ubuntu shell, are you back to your host machine, if not, come back to the host machine.

-->

PS C:\WINDOWS\system32> docker run -it aefd7f02ae24 bash

root@416c380b54dc:/# apt install nginx

Reading package lists... Done

Building dependency tree

Reading state information... Done

E: Unable to locate package nginx

root@416c380b54dc:/# exit

exit

PS C:\WINDOWS\system32> docker ps -l

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

416c380b54dc aefd7f02ae24 "bash" 47 seconds ago Exited (100) 6 seconds ago admiring\_volhard

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

30. Check if the ubuntu container is running.

-->

PS C:\WINDOWS\system32> docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

7e8879be9421 aefd7f02ae24 "/bin/bash" About an hour ago Up 55 minutes vibrant\_noyce

761be7034997 01244e8938d1 "/bin/bash" About an hour ago Up About an hour gifted\_booth

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

31. Create a new ubuntu container out of the same image as that previous container in attached mode

32. Login as a root user

33. Check if you can see the file created in previous container, you will not see the file as well as software that you installed in the previous container.

Now kill this Container.

--->

PS C:\WINDOWS\system32> docker run -it aefd7f02ae24

root@d6fa4b8463bc:/# ls

bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var

root@d6fa4b8463bc:/# exit

exit

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

34. Do you have the previous ubuntu container where you created the file and installed the software? If no reapeat step 25 to 29.

35. Create an Image out of the existing container.

--->

C:\Users\Neha Kala>docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

7e8879be9421 aefd7f02ae24 "/bin/bash" 2 hours ago Up 2 hours vibrant\_noyce

761be7034997 01244e8938d1 "/bin/bash" 2 hours ago Up 2 hours gifted\_booth

2dc584e1c0e6 tomcat:8.0 "catalina.sh run" 5 hours ago Up 5 hours 0.0.0.0:8082->8080/tcp, :::8082->8080/tcp my-tomcat-container

C:\Users\Neha Kala>docker commit vibrant\_noyce neha

sha256:5799755747dc7a3e1f4ab4748d7c6a350aa891a7f324c41187c929d6a0dd8ec5

C:\Users\Neha Kala>docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

neha latest 5799755747dc 10 seconds ago 134MB

nginx latest f0b8a9a54136 5 days ago 133MB

nehakala/myapp latest a6e5f8b09c22 11 days ago 72.7MB

myimage 1.0 a6e5f8b09c22 11 days ago 72.7MB

gcr.io/k8s-minikube/kicbase v0.0.22 bcd131522525 11 days ago 1.09GB

ubuntu xenial aefd7f02ae24 3 weeks ago 134MB

ubuntu rolling 01244e8938d1 3 weeks ago 74.1MB

tomcat 8.0 ef6a7c98d192 2 years ago 356MB

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

36. Now Create a Container out of this image and login into it to see if you can see the file and software installed by you in the previous container.

--->

C:\Users\Neha Kala>docker run -it neha

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

37. Do you have running tomcat container? If yes, Stop it and kill all tomcat container.

--->

C:\Users\Neha Kala>docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

7e8879be9421 aefd7f02ae24 "/bin/bash" 2 hours ago Up 2 hours vibrant\_noyce

761be7034997 01244e8938d1 "/bin/bash" 2 hours ago Up 2 hours gifted\_booth

2dc584e1c0e6 tomcat:8.0 "catalina.sh run" 5 hours ago Up 5 hours 0.0.0.0:8082->8080/tcp, :::8082->8080/tcp my-tomcat-container

C:\Users\Neha Kala>docker stop 2dc584e1c0e6

2dc584e1c0e6

C:\Users\Neha Kala>docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

7e8879be9421 aefd7f02ae24 "/bin/bash" 2 hours ago Up 2 hours vibrant\_noyce

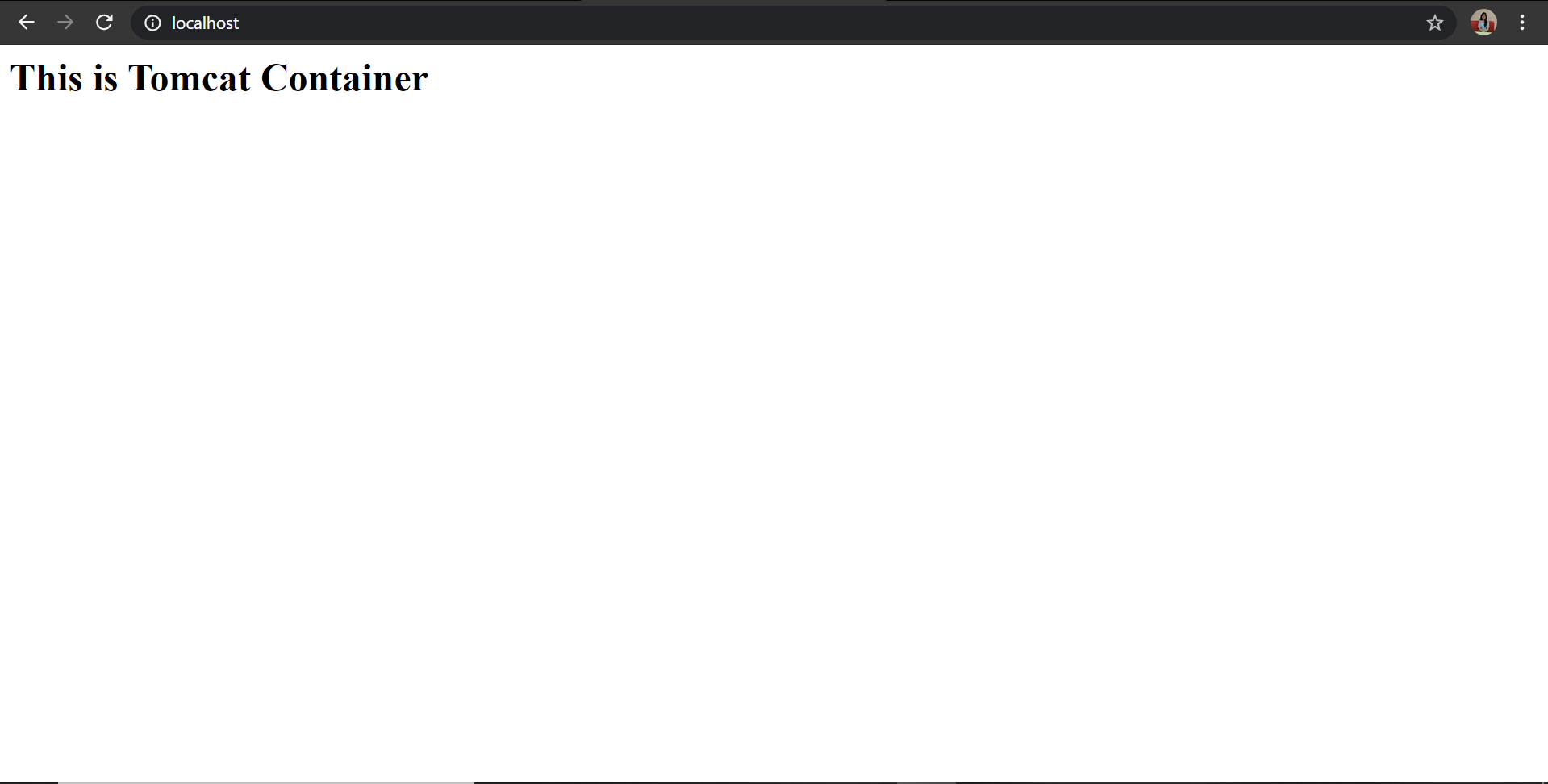
761be7034997 01244e8938d1 "/bin/bash" 2 hours ago Up 2 hours gifted\_booth

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

38. Create an index.html file with following code in it:-

<h1>This is Tomcat Container</h1>

Now, Start a tomcat container in such a way that on hitting its URL for home page it should show the above html page.



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

39. type below command:-

docker images --help

Now, try to run command that proves the concept of following three options:-

1. -a

2. -f

3. -q

write atleast 1 command using each option above and prove their concepts as described in the --help.

--->

-a, --all Show all images (default hides intermediate images)

-f, --filter filter Filter output based on conditions provided

-q, --quiet Only show image IDs

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

40. type below command:-

docker ps --help

Now, try to run command that proves the concept of following six options:-

1. -a

2. -f

3. -q

4. -n

5. -l

6. -s

write atleast 1 command using each option above and prove their concepts as described in the --help.

--->

1) -a, --all Show all containers (default shows just running)

>docker ps -a

>docker ps --all

---------------------------------------------

2) -f, --filter filter Filter output based on conditions provided

--format string Pretty-print containers using a Go template

>docker ps --all --format "table {{.ID}}\t{{.Image}}"

PS C:\WINDOWS\system32> docker ps --all --format "table {{.ID}}\t{{.Image}}"

>>

CONTAINER ID IMAGE

1b093a0ccaca ubuntu:rolling

e8c5aea4636b ubuntu:xenial

9216739c05ac 01244e8938d1

9e09cb521853 tomcat:7.0

8f2a4ccd20eb nginx

3e70ca503c72 gcr.io/k8s-minikube/kicbase:v0.0.22

6ef06301892c nehakala/myapp

ac624da1d1ef 2acc5493bcb8

1e5606b05c81 docker/getting-started

--------------------------------------------

3) -q, --quiet Only display container IDs

> docker ps --quiet

> docker ps --all --quiet

PS C:\WINDOWS\system32> docker ps --quiet

PS C:\WINDOWS\system32> docker ps --all --quiet

1b093a0ccaca

e8c5aea4636b

9216739c05ac

9e09cb521853

8f2a4ccd20eb

3e70ca503c72

6ef06301892c

ac624da1d1ef

1e5606b05c81

------------------------------------------------

4)-n, --last int Show n last created containers (includes all

states) (default -1)

>docker ps --last 3

PS C:\WINDOWS\system32> docker ps --last 3

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

1b093a0ccaca ubuntu:rolling "/bin/bash" 39 minutes ago Exited (0) 39 minutes ago boring\_cannon

e8c5aea4636b ubuntu:xenial "/bin/bash" 49 minutes ago Exited (0) 49 minutes ago zen\_ellis

9216739c05ac 01244e8938d1 "/bin/bash" 2 hours ago Exited (0) 2 hours ago hopeful\_jackson

-------------------------------------------------------

5)-l, --latest Show the latest created container (includes all

states)

--no-trunc Don't truncate output

>docker ps --all --latest

>docker ps --all --no-trunc

PS C:\WINDOWS\system32> docker ps --all --latest

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

1b093a0ccaca ubuntu:rolling "/bin/bash" 45 minutes ago Exited (0) 45 minutes ago boring\_cannon

------------------------------------------------------------

6)-s, --size Display total file sizes

>docker ps --all --size

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

41. Type below command:-

docker --help

you will various sections of commands apart from options like "Managemnet Commands" and "Commands".

Write some texts below describing the use of "Management Commands".

Use each command mentioned below and prove its concepts as described in the --help desription.

write what you have understood from the output of the command after its successful execution.

1. cp

2. create

3. export

4. history

5. info

6. login

7. logout

8. rename

9. save

10. stats

11. top

---->

cp Copy files/folders between a container and the local filesystem

create Create a new container

export Export a container's filesystem as a tar archive

history Show the history of an image

info Display system-wide information

login Log in to a Docker registry

logout Log out from a Docker registry

rename Rename a container

save Save one or more images to a tar archive (streamed to STDOUT by default)

stats Display a live stream of container(s) resource usage statistics

top Display the running processes of a container

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

42. Kill all running container in one liner command.

--->docker ps -q

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

43. Delete all images in one liner command.

--->docker rmi -f

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

44. Create a simple Dockerfile, build it and run it.

--->

Dockerfile-

From ubuntu

RUN dir

CMD ["echo", "Hello! , Your Dockerfile has been executed Successfully"]

--->

C:\Users\Neha Kala\Desktop\Dockerfiles\DF>docker build -t myimage:1.0 .

[+] Building 5.7s (7/7) FINISHED

=> [internal] load build definition from Dockerfile 0.8s

=> => transferring dockerfile: 31B 0.0s

=> [internal] load .dockerignore 1.1s

=> => transferring context: 2B 0.0s

=> [internal] load metadata for docker.io/library/ubuntu:latest 3.4s

=> [auth] library/ubuntu:pull token for registry-1.docker.io 0.0s

=> [1/2] FROM docker.io/library/ubuntu@sha256:cf31af331f38d1d7158470e095b132acd126a7180a54f263d386da88eb681d93 0.3s

=> => resolve docker.io/library/ubuntu@sha256:cf31af331f38d1d7158470e095b132acd126a7180a54f263d386da88eb681d93 0.3s

=> CACHED [2/2] RUN dir 0.0s

=> exporting to image 0.4s

=> => exporting layers 0.0s

=> => writing image sha256:a6e5f8b09c22df3faba85eb04d271db1ee48ce0b3511a70be98ba9d424e1fa51 0.1s

=> => naming to docker.io/library/myimage:1.0

C:\Users\Neha Kala\Desktop\Dockerfiles\DF>docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

myimage 1.0 a6e5f8b09c22 10 days ago 72.7MB

C:\Users\Neha Kala\Desktop\Dockerfiles\DF>docker run a6e5f8b09c22

Hello! , Your Dockerfile has been executed Successfully

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

45. Create one or more Dockerfile that demostrate the following commands in Dockerfile (Write a PoC for each in one or more Dockerfile)

1. USER

2. RUN

3. ENV

4. CMD

5. ENTRYPOINT

6. EXPOSE

7. VOLUME

--->

FROM ubuntu

RUN apt-get update

RUN apt-get -y install apache2

ADD . /var/www/html

ENTRYPOINT apachectl -D FOREGROUND

ENV name DevOps Neha

------------------------------------------------------

FROM is used to define base image, on which we will be building.

ADD - ADD<source><destination in container> -used to add files to the container being built.

RUN is used to add layers to the base image, by installing components. Each RUN statement adds new layer to the docker image.

CMD is used to run commands on the start of the container. These commands run only when there is no arguement specified while running the container.

ENTRYPOINT is used strictly run commands the moment the container initializes.

ENV is used to define environment variables in the container run-time.

46. Dockerhub:-

Find a application you care about on docker hub.

Launch the container.

Install another application in it.

Save (commit) the image.

Upload that to docker hub in your account and share it with a colleague, ask them to use your image and run the container out of it in their machine/vm.

--->

>docker pull Ubuntu

>docker run -it ubuntu bash

>apt-get update

>apt-get install nginx

>docker ps –a

>docker commit ab74f948f82c nehakala/nginx

>docker login

>docker push nehakala/nginx

