////// TASK : 1 /////

PS C:\Users\Tanishka2020> docker --version

Docker version 28.1.1, build 4eba377

PS C:\Users\Tanishka2020> docker pull

docker: 'docker pull' requires 1 argument

Usage: docker pull [OPTIONS] NAME[:TAG|@DIGEST]

Run 'docker pull --help' for more information

PS C:\Users\Tanishka2020> docker pull --help

Usage: docker pull [OPTIONS] NAME[:TAG|@DIGEST]

Download an image from a registry

Aliases:

docker image pull, docker pull

Options:

-a, --all-tags Download all tagged images in the repository

--disable-content-trust Skip image verification (default true)

--platform string Set platform if server is multi-platform

capable

-q, --quiet Suppress verbose output

PS C:\Users\Tanishka2020> docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

hello-world latest 940c619fbd41 4 months ago 20.4kB

PS C:\Users\Tanishka2020> docker run

docker: 'docker run' requires at least 1 argument

Usage: docker run [OPTIONS] IMAGE [COMMAND] [ARG...]

See 'docker run --help' for more information

PS C:\Users\Tanishka2020> docker run --help

Usage: docker run [OPTIONS] IMAGE [COMMAND] [ARG...]

Create and run a new container from an image

Aliases:

docker container run, docker run

Options:

--add-host list Add a custom host-to-IP mapping

(host:ip)

--annotation map Add an annotation to the

container (passed through to the

OCI runtime) (default map[])

-a, --attach list Attach to STDIN, STDOUT or STDERR

--blkio-weight uint16 Block IO (relative weight),

between 10 and 1000, or 0 to

disable (default 0)

--blkio-weight-device list Block IO weight (relative device

weight) (default [])

--cap-add list Add Linux capabilities

--cap-drop list Drop Linux capabilities

--cgroup-parent string Optional parent cgroup for the

container

--cgroupns string Cgroup namespace to use

(host|private)

'host': Run the container in

the Docker host's cgroup

namespace

'private': Run the container in

its own private cgroup namespace

'': Use the cgroup

namespace as configured by the

default-cgroupns-mode

option on the daemon (default)

--cidfile string Write the container ID to the file

--cpu-period int Limit CPU CFS (Completely Fair

Scheduler) period

--cpu-quota int Limit CPU CFS (Completely Fair

Scheduler) quota

--cpu-rt-period int Limit CPU real-time period in

microseconds

--cpu-rt-runtime int Limit CPU real-time runtime in

microseconds

-c, --cpu-shares int CPU shares (relative weight)

--cpus decimal Number of CPUs

--cpuset-cpus string CPUs in which to allow execution

(0-3, 0,1)

--cpuset-mems string MEMs in which to allow execution

(0-3, 0,1)

-d, --detach Run container in background and

print container ID

--detach-keys string Override the key sequence for

detaching a container

--device list Add a host device to the container

--device-cgroup-rule list Add a rule to the cgroup allowed

devices list

--device-read-bps list Limit read rate (bytes per

second) from a device (default [])

--device-read-iops list Limit read rate (IO per second)

from a device (default [])

--device-write-bps list Limit write rate (bytes per

second) to a device (default [])

--device-write-iops list Limit write rate (IO per second)

to a device (default [])

--disable-content-trust Skip image verification (default

true)

--dns list Set custom DNS servers

--dns-option list Set DNS options

--dns-search list Set custom DNS search domains

--domainname string Container NIS domain name

--entrypoint string Overwrite the default ENTRYPOINT

of the image

-e, --env list Set environment variables

--env-file list Read in a file of environment

variables

--expose list Expose a port or a range of ports

--gpus gpu-request GPU devices to add to the

container ('all' to pass all GPUs)

--group-add list Add additional groups to join

--health-cmd string Command to run to check health

--health-interval duration Time between running the check

(ms|s|m|h) (default 0s)

--health-retries int Consecutive failures needed to

report unhealthy

--health-start-interval duration Time between running the check

during the start period

(ms|s|m|h) (default 0s)

--health-start-period duration Start period for the container

to initialize before starting

health-retries countdown

(ms|s|m|h) (default 0s)

--health-timeout duration Maximum time to allow one check

to run (ms|s|m|h) (default 0s)

--help Print usage

-h, --hostname string Container host name

--init Run an init inside the container

that forwards signals and reaps

processes

-i, --interactive Keep STDIN open even if not attached

--ip string IPv4 address (e.g., 172.30.100.104)

--ip6 string IPv6 address (e.g., 2001:db8::33)

--ipc string IPC mode to use

--isolation string Container isolation technology

--kernel-memory bytes Kernel memory limit

-l, --label list Set meta data on a container

--label-file list Read in a line delimited file of

labels

--link list Add link to another container

--link-local-ip list Container IPv4/IPv6 link-local

addresses

--log-driver string Logging driver for the container

--log-opt list Log driver options

--mac-address string Container MAC address (e.g.,

92:d0:c6:0a:29:33)

-m, --memory bytes Memory limit

--memory-reservation bytes Memory soft limit

--memory-swap bytes Swap limit equal to memory plus

swap: '-1' to enable unlimited swap

--memory-swappiness int Tune container memory swappiness

(0 to 100) (default -1)

--mount mount Attach a filesystem mount to the

container

--name string Assign a name to the container

--network network Connect a container to a network

--network-alias list Add network-scoped alias for the

container

--no-healthcheck Disable any container-specified

HEALTHCHECK

--oom-kill-disable Disable OOM Killer

--oom-score-adj int Tune host's OOM preferences

(-1000 to 1000)

--pid string PID namespace to use

--pids-limit int Tune container pids limit (set

-1 for unlimited)

--platform string Set platform if server is

multi-platform capable

--privileged Give extended privileges to this

container

-p, --publish list Publish a container's port(s) to

the host

-P, --publish-all Publish all exposed ports to

random ports

--pull string Pull image before running

("always", "missing", "never")

(default "missing")

-q, --quiet Suppress the pull output

--read-only Mount the container's root

filesystem as read only

--restart string Restart policy to apply when a

container exits (default "no")

--rm Automatically remove the

container and its associated

anonymous volumes when it exits

--runtime string Runtime to use for this container

--security-opt list Security Options

--shm-size bytes Size of /dev/shm

--sig-proxy Proxy received signals to the

process (default true)

--stop-signal string Signal to stop the container

--stop-timeout int Timeout (in seconds) to stop a

container

--storage-opt list Storage driver options for the

container

--sysctl map Sysctl options (default map[])

--tmpfs list Mount a tmpfs directory

-t, --tty Allocate a pseudo-TTY

--ulimit ulimit Ulimit options (default [])

--use-api-socket Bind mount Docker API socket and

required auth

-u, --user string Username or UID (format:

<name|uid>[:<group|gid>])

--userns string User namespace to use

--uts string UTS namespace to use

-v, --volume list Bind mount a volume

--volume-driver string Optional volume driver for the

container

--volumes-from list Mount volumes from the specified

container(s)

-w, --workdir string Working directory inside the

container

PS C:\Users\Tanishka2020> docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

PS C:\Users\Tanishka2020> docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

157e089da671 hello-world "/hello" 15 minutes ago Exited (0) 10 minutes ago practical\_brattain

PS C:\Users\Tanishka2020> docker stop 157e089da671

157e089da671

PS C:\Users\Tanishka2020> docker start 157e089da671

157e089da671

PS C:\Users\Tanishka2020> docker rm 157e089da671

157e089da671

PS C:\Users\Tanishka2020> docker rmi 157e089da671

Error response from daemon: No such image: 157e089da671:latest

PS C:\Users\Tanishka2020> docker start 157e089da671

Error response from daemon: No such container: 157e089da671

Error: failed to start containers: 157e089da671

PS C:\Users\Tanishka2020> docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

PS C:\Users\Tanishka2020> docker run hello-world

Hello from Docker!

This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:

1. The Docker client contacted the Docker daemon.

2. The Docker daemon pulled the "hello-world" image from the Docker Hub.

(amd64)

3. The Docker daemon created a new container from that image which runs the

executable that produces the output you are currently reading.

4. The Docker daemon streamed that output to the Docker client, which sent it

to your terminal.

To try something more ambitious, you can run an Ubuntu container with:

$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:

https://hub.docker.com/

For more examples and ideas, visit:

https://docs.docker.com/get-started/

PS C:\Users\Tanishka2020> docker exec -it

docker: 'docker exec' requires at least 2 arguments

Usage: docker exec [OPTIONS] CONTAINER COMMAND [ARG...]

See 'docker exec --help' for more information

PS C:\Users\Tanishka2020> docker login

Authenticating with existing credentials... [Username: bhagyada09]

i Info → To login with a different account, run 'docker logout' followed by 'docker login'

Login Succeeded

PS C:\Users\Tanishka2020> docker push

docker: 'docker push' requires 1 argument

Usage: docker push [OPTIONS] NAME[:TAG]

Run 'docker push --help' for more information

PS C:\Users\Tanishka2020> docker pull ubuntu

Using default tag: latest

latest: Pulling from library/ubuntu

d9d352c11bbd: Pull complete

Digest: sha256:b59d21599a2b151e23eea5f6602f4af4d7d31c4e236d22bf0b62b86d2e386b8f

Status: Downloaded newer image for ubuntu:latest

docker.io/library/ubuntu:latest

PS C:\Users\Tanishka2020> docker run -it ubuntu /bin/bash

root@22b7ae1e6947:/# apt update

Get:1 http://archive.ubuntu.com/ubuntu noble InRelease [256 kB]

Get:2 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]

Get:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]

Get:4 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1093 kB]

Get:5 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]

Ign:6 http://archive.ubuntu.com/ubuntu noble/restricted amd64 Packages

Get:7 http://archive.ubuntu.com/ubuntu noble/universe amd64 Packages [19.3 MB]

Get:8 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [22.1 kB]

Get:9 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [1100 kB]

Get:10 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [1442 kB]

Get:11 http://archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [331 kB]

Get:12 http://archive.ubuntu.com/ubuntu noble/main amd64 Packages [1808 kB]

Get:13 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [26.7 kB]

Get:14 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1403 kB]

Get:15 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1418 kB]

Get:16 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [1495 kB]

Get:17 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [31.8 kB]

Get:18 http://archive.ubuntu.com/ubuntu noble-backports/main amd64 Packages [48.0 kB]

Get:6 http://archive.ubuntu.com/ubuntu noble/restricted amd64 Packages [117 kB]

Fetched 30.3 MB in 9min 23s (53.8 kB/s)

Reading package lists... Done

Building dependency tree... Done

Reading state information... Done

2 packages can be upgraded. Run 'apt list --upgradable' to see them.

root@22b7ae1e6947:/#