(A) Docker Installation:

Installing docker on windows:

Docker can be installed only on Windows 10 Pro 64 bit version or Windows 2016 Server Edition

(i) Open https://docs.docker.com/docker-for-windows/install/
(ii) Download and install docker
(iii) Docker commands will work on windows only through power shell

Note: docker when installed on windows activates an application called Hyper-v. This application will not allow any other virtualization software's to run.

Installing docker on linux:

(i) Open get.docker.com
(ii) Copy the first two commands and paste in terminal

Curl -fsSL get.docker.com -o get-docker.sh
sh get-docker.sh

Installing docker in Mac:

(i) Open https://docs.docker.com/docker-for-mac/install/
(ii) Download and install docker

To verify the installation of docker, open terminal

```
Yukteshs-MacBook-Pro:~ ychinmka@in.ibm.com$ docker version
Client: Docker Engine - Community
```

To verify the installation of only docker host

```
Yukteshs-MacBook-Pro:~ ychinmka@in.ibm.com$ docker -v
Docker version 19.03.8, build afacb8b
Yukteshs-MacBook-Pro:~ ychinmka@in.ibm.com$
```

or

```
Yukteshs-MacBook-Pro:~ ychinmka@in.ibm.com$ docker --version
Docker version 19.03.8, build afacb8b
```

To see the detailed information of dockers info in our docker host

```
Yukteshs-MacBook-Pro:~ ychinmka@in.ibm.com$ docker info
Client:
 Debug Mode: false

Server:
 Containers: 2
```

To see command help

```
Yukteshs-MacBook-Pro:~ ychinmka@in.ibm.com$ docker --help

Usage:  docker [OPTIONS] COMMAND
```

(B) Important Docker commands

Docker Image Commands:

1. To download an image

   docker pull image_name

2. To see list of all images in the docker host (our machine where docker is installed)

   docker images
   Or
   docker images ls

3. To search for an image on hub.docker.com

   docker search image_name

4. To delete an image from our docker host

   docker rmi image_name
   docker image prune -a (helps in deleting all docker images)

5. To upload a docker image

   docker push imagename

6. To create an image from a container

   docker commit container_id/container_name new_image_name

7. To create an image from a docker file

   docker build -t new_image_name path_of_dockerfile

Docker Container Commands:

8. To see the list of running containers

   docker container ls

9. To see the list of all containers (running and stopped)

   docker ps -a

10. To find detailed info about a container

    docker inspect container_name/container_id

11. To see the logs generated by a container

    docker logs container_name/container_id

12. To start a stopped container

    docker start container_name/container_id

13. To stop a running container

    docker stop container_name/container_id

14. To restart a container

docker restart container_name/container_id

15. To restart after 10 seconds

    docker restart -t 10 container_name/container_id

16. To remove a stopped container

    docker rm container_name/container_id

17. To stop all running containers

    docker stop $(docker ps -aq)

18. To remove all stopped containers

    docker rm $(docker ps -aq)

19. To remove all containers (running and stopped)

    docker rm -f $(docker ps -aq)

20. To create a container from an image

    docker run image_name

(C) Working with Docker commands:

In this section will learn how to use docker commands in creating images and containers

Let me run a hello-world container

```
Yukteshs-MacBook-Pro:~ ychinmka@in.ibm.com$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
0e03bdcc26d7: Pull complete
Digest: sha256:8e3114318a995a1ee497790535e7b88365222a21771ae7e53687ad76563e8e76
Status: Downloaded newer image for hello-world:latest

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/
```

Now, hello-world image got downloaded from the Docker hub into Docker host (our machine), Now check for images

```
Yukteshs-MacBook-Pro:~ ychinmka@in.ibm.com$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
hello-world         latest              bf756fb1ae65        4 months ago        13.3kB
```

Recent created ***hello-world*** image it is showing

Let me pull the ubuntu image from the docker hub

```
Yukteshs-MacBook-Pro:~ ychinmka@in.ibm.com$ docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
d51af753c3d3: Pull complete
fc878cd0a91c: Pull complete
6154df8ff988: Pull complete
fee5db0ff82f: Pull complete
Digest: sha256:747d2dbbaaee995098c9792d99bd333c6783ce56150d1b11e333bbceed5c54d7
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
Yukteshs-MacBook-Pro:~ ychinmka@in.ibm.com$
```

Let us run this ubuntu image, when we run any image it will create docker container

```
Yukteshs-MacBook-Pro:~ ychinmka@in.ibm.com$ docker run -it -d ubuntu
bf9058d5f9c88c0013c432cdb3f93e19ead565b90ed769b8a58382a44c7f8d50
```

Let us see how list of containers in the docker host

```
Yukteshs-MacBook-Pro:~ ychinmka@in.ibm.com$ docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED              STATUS                      PORTS
 NAMES
bf9058d5f9c8        ubuntu              "/bin/bash"         About a minute ago   Up About a minute
 silly_robinson
e846b917a4ec        hello-world         "/hello"            29 minutes ago       Exited (0) 29 minutes ago
 sharp_leavitt
Yukteshs-MacBook-Pro:~ ychinmka@in.ibm.com$
```

Let us access the running container

```
Yukteshs-MacBook-Pro:~ ychinmka@in.ibm.com$ docker exec -it bf9058d5f9c8 bash
root@bf9058d5f9c8:/# echo hello
hello
root@bf9058d5f9c8:/# exit
exit
Yukteshs-MacBook-Pro:~ ychinmka@in.ibm.com$
```

We have accessed ubuntu terminal printed hello and given exit command to come out from the container environment

Now, let us stop the ubuntu container

```
Yukteshs-MacBook-Pro:~ ychinmka@in.ibm.com$ docker stop bf9058d5f9c8
bf9058d5f9c8
```

Creating an image from the local image

```
Yukteshs-MacBook-Pro:~ ychinmka@in.ibm.com$ docker commit bf9058d5f9c8 silly_robinson/ubuntu
sha256:ae93070aa61c9241fb4c0b2a4295361ed8c484177dba8bed528cfb48c0b14678
```

Let us check total available images

```
Yukteshs-MacBook-Pro:~ ychinmka@in.ibm.com$ docker images
REPOSITORY                TAG                 IMAGE ID            CREATED             SIZE
silly_robinson/ubuntu     latest              ae93070aa61c        2 minutes ago       73.9MB
ubuntu                    latest              1d622ef86b13        2 weeks ago         73.9MB
hello-world               latest              bf756fb1ae65        4 months ago        13.3kB
```

Let us see all the existing containers now

```
Yukteshs-MacBook-Pro:~ ychinmka@in.ibm.com$ docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS                      PORTS               NAMES
b9f41439826e        hello-world         "/hello"            18 minutes ago      Exited (0) 18 minutes ago                       objective_blackwell
bf9058d5f9c8        ubuntu              "/bin/bash"         34 minutes ago      Exited (0) 23 minutes ago                       silly_robinson
e846b917a4ec        hello-world         "/hello"            About an hour ago   Exited (0) About an hour ago                    sharp_leavitt
```

Let us delete a container

```
Yukteshs-MacBook-Pro:~ ychinmka@in.ibm.com$ docker rm bf9058d5f9c8
bf9058d5f9c8
```

Let us see all the existing containers again now

```
Yukteshs-MacBook-Pro:~ ychinmka@in.ibm.com$ docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS                      PORTS               NAMES
b9f41439826e        hello-world         "/hello"            20 minutes ago      Exited (0) 20 minutes ago                       objective_blackwell
e846b917a4ec        hello-world         "/hello"            About an hour ago   Exited (0) About an hour ago                    sharp_leavitt
```

This will show all the containers which are in stop & start state

Following command helps in showing running containers

```
Yukteshs-MacBook-Pro:~ ychinmka@in.ibm.com$  docker container ls
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
Yukteshs-MacBook-Pro:~ ychinmka@in.ibm.com$
```

Basically, the understanding is there is no running container for now

Similarly, we can also remove docker images

```
Yukteshs-MacBook-Pro:~ ychinmka@in.ibm.com$ docker images
REPOSITORY              TAG                 IMAGE ID            CREATED             SIZE
silly_robinson/ubuntu   latest              ae93070aa61c        9 hours ago         73.9MB
ubuntu                  latest              1d622ef86b13        2 weeks ago         73.9MB
hello-world             latest              bf756fb1ae65        4 months ago        13.3kB
Yukteshs-MacBook-Pro:~ ychinmka@in.ibm.com$
```

Currently we are having 3 docker images, Now let us try to delete 1 image

```
Yukteshs-MacBook-Pro:~ ychinmka@in.ibm.com$ docker rmi ae93070aa61c
Untagged: silly_robinson/ubuntu:latest
Deleted: sha256:ae93070aa61c9241fb4c0b2a4295361ed8c484177dba8bed528cfb48c0b14678
Deleted: sha256:925c742afe92a45cd24b0f27f84631181ac2d7b9c08e5254b362445fbf5b24a6
Yukteshs-MacBook-Pro:~ ychinmka@in.ibm.com$
```

Now let us total docker images

```
Yukteshs-MacBook-Pro:~ ychinmka@in.ibm.com$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
ubuntu              latest              1d622ef86b13        2 weeks ago         73.9MB
hello-world         latest              bf756fb1ae65        4 months ago        13.3kB
Yukteshs-MacBook-Pro:~ ychinmka@in.ibm.com$
```

(D) Building a Docker File:

As discussed, Docker File is simple text file which is used for creating dockerimages . It uses a syntax called GO which has certain predefined keywords

Steps:

1) Create a file named Dockerfile
2) Add instructions into dockerfile
3) Build dockerfile to create image
4) Run image to create a container

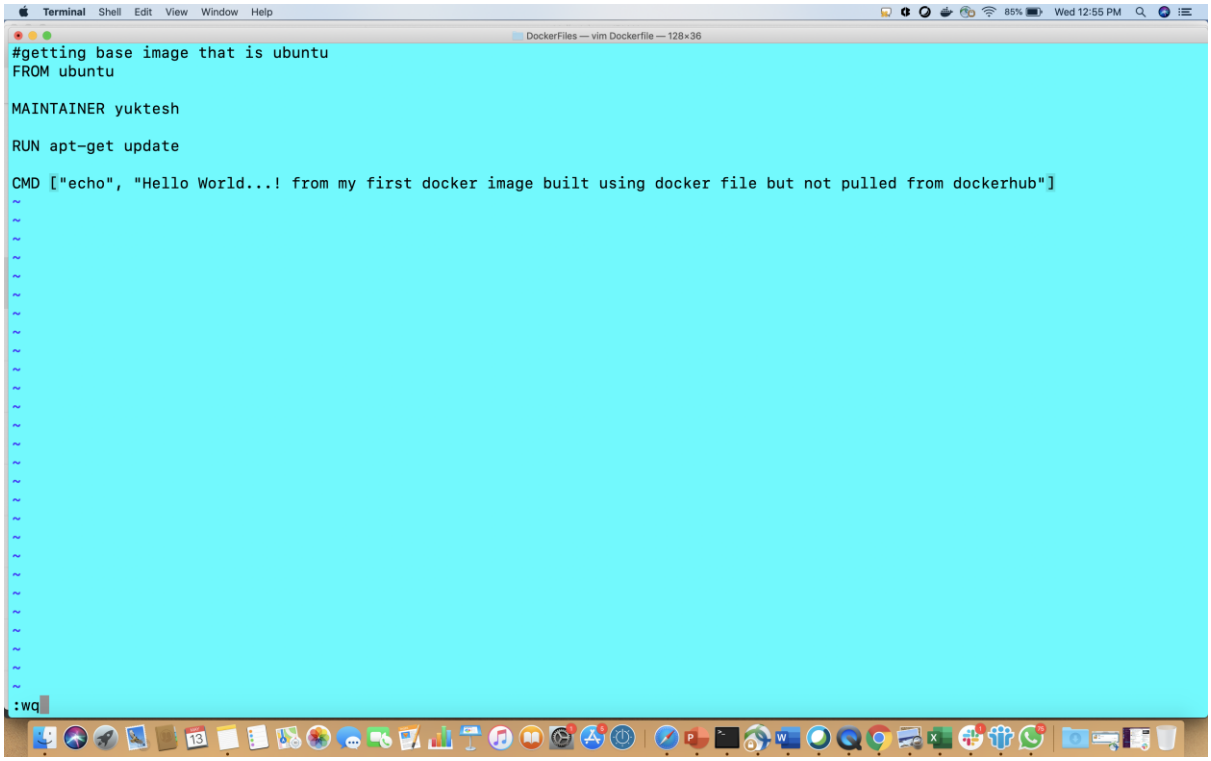
(i)Creating a file names Dockerfile

```
Yukteshs-MacBook-Pro:Documents ychinmka@in.ibm.com$ cd DockerFiles/
Yukteshs-MacBook-Pro:DockerFiles ychinmka@in.ibm.com$ touch Dockerfile
```

(ii)Add instructions into dockerfile

```
Yukteshs-MacBook-Pro:DockerFiles ychinmka@in.ibm.com$ vim Dockerfile
```

It will move to vim directory -> press ‘i’ to insert code, start typing below show 5 lines -> press ‘escape’ ->:wq and enter to save and exit



To view my dockerfile content

```
Yukteshs-MacBook-Pro:DockerFiles ychinmka@in.ibm.com$ cat Dockerfile
#getting base image that is ubuntu
FROM ubuntu

MAINTAINER yuktesh

RUN apt-get update

CMD ["echo", "Hello World...! from my first docker image built using docker file but not pulled from dockerhub"]
Yukteshs-MacBook-Pro:DockerFiles ychinmka@in.ibm.com$
```

(iii) Now we can build dockerfile to create the docker image

```
Yukteshs-MacBook-Pro:DockerFiles ychinmka@in.ibm.com$ docker build -t myimage1:1.0 .
Sending build context to Docker daemon  2.048kB
Step 1/4 : FROM ubuntu
 ---> 1d622ef86b13
```

Let us see now docker images

```
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
myimage1            1.0                 a70764a20ab5        5 minutes ago       95.3MB
ubuntu              latest              1d622ef86b13        2 weeks ago         73.9MB
hello-world         latest              bf756fb1ae65        4 months ago        13.3kB
Yukteshs-MacBook-Pro:DockerFiles ychinmka@in.ibm.com$
```

We can see our myimage1 docker image got created with TAG as 1.0

(iv) Now let us run this docker image to create container

```
Yukteshs-MacBook-Pro:DockerFiles ychinmka@in.ibm.com$ docker run a70764a20ab5
Hello World...! from my first docker image built using docker file but not pulled from dockerhub
Yukteshs-MacBook-Pro:DockerFiles ychinmka@in.ibm.com$
```

(e) Docker Compose:

Following are points we should understand:

(i) What and why docker compose?
(ii) How to install
(iii) How to create a docker compose file
(iv) How to use docker compose file to create services
(v) Basic commands

(i) What and why docker compose:

- Tool for defining and running multi container docker applications
- Use yaml files to configure application services
- Can start all services with single command
  docker compose up
- Can stop all services with single command
  docker compose down
- ***Can scale up selected services when required***

(ii) How to install

Already installed on windows and mac with docker

```
Yukteshs-MacBook-Pro:~ ychinmka@in.ibm.com$ docker-compose -v
docker-compose version 1.25.4, build 8d51620a
Yukteshs-MacBook-Pro:~ ychinmka@in.ibm.com$
```

Or

```
Yukteshs-MacBook-Pro:~ ychinmka@in.ibm.com$ docker-compose --version
docker-compose version 1.25.4, build 8d51620a
Yukteshs-MacBook-Pro:~ ychinmka@in.ibm.com$
```

For linux there are 2 ways to install

(a) we should get onto below link and install

https://docs.docker.com/compose/install/

(b) Using pip application, for this you must have python installed

pip install -U docker-compose

(iii) <u>How to create docker compose file</u>

- Create docker compose file at nay location on your system
- Name must be ***docker-compose.yml***

```
Yukteshs-MacBook-Pro:~ ychinmka@in.ibm.com$ mkdir DockerComposeFile
Yukteshs-MacBook-Pro:~ ychinmka@in.ibm.com$ cd DockerComposeFile/
Yukteshs-MacBook-Pro:DockerComposeFile ychinmka@in.ibm.com$ touch docker-compose.yml
Yukteshs-MacBook-Pro:DockerComposeFile ychinmka@in.ibm.com$ ls
docker-compose.yml
Yukteshs-MacBook-Pro:DockerComposeFile ychinmka@in.ibm.com$ vim docker-compose.yml
```

Write below code in vi editor by click i -> type code -> esc -> :wq! enter

```
services:

  web:
    image: nginx

  database:
    image: redis
~
~
~
"docker-compose.yml" 7L, 65C
```

You can check the code

```
Yukteshs-MacBook-Pro:DockerComposeFile ychinmka@in.ibm.com$ cat docker-compose.yml
services:

  web:
    image: nginx

  database:
    image: redis
Yukteshs-MacBook-Pro:DockerComposeFile ychinmka@in.ibm.com$
```

-Now check the validity of the file by ***docker-compose config***

```
Yukteshs-MacBook-Pro:DockerComposeFile ychinmka@in.ibm.com$ docker-compose config
ERROR: The Compose file './docker-compose.yml' is invalid because:
Unsupported config option for services: 'web'
Yukteshs-MacBook-Pro:DockerComposeFile ychinmka@in.ibm.com$ vim docker-compose.yml
Yukteshs-MacBook-Pro:DockerComposeFile ychinmka@in.ibm.com$ cat docker-compose.yml
version: '1'

services:

  web:
    image: nginx

  database:
    image: redis
Yukteshs-MacBook-Pro:DockerComposeFile ychinmka@in.ibm.com$ docker-compose config
ERROR: Version in "./docker-compose.yml" is invalid. You might be seeing this error because you're using the wrong Compose file
version. Either specify a supported version (e.g "2.2" or "3.3") and place your service definitions under the `services` key, or
 omit the `version` key and place your service definitions at the root of the file to use version 1.
For more on the Compose file format versions, see https://docs.docker.com/compose/compose-file/
Yukteshs-MacBook-Pro:DockerComposeFile ychinmka@in.ibm.com$ vim docker-compose.yml
Yukteshs-MacBook-Pro:DockerComposeFile ychinmka@in.ibm.com$ docker-compose config
services:
  database:
    image: redis
  web:
    image: nginx
version: '3.0'

Yukteshs-MacBook-Pro:DockerComposeFile ychinmka@in.ibm.com$
```

(vi) <u>How to use docker compose file to create services</u>

->Run docker-compose file using command ***docker-compose up -d***
(detach mode – which helps in adding more to this compose file)

```
Yukteshs-MacBook-Pro:DockerComposeFile ychinmka@in.ibm.com$ docker-compose up -d
Creating network "dockercomposefile_default" with the default driver
Pulling web (nginx:)...
latest: Pulling from library/nginx
54fec2fa59d0: Pull complete
4ede6f09aefe: Pull complete
f9dc69acb465: Pull complete
Digest: sha256:86ae264c3f4acb99b2dee4d0098c40cb8c46dcf9e1148f05d3a51c4df6758c12
Status: Downloaded newer image for nginx:latest
Pulling database (redis:)...
latest: Pulling from library/redis
54fec2fa59d0: Already exists
9c94e11103d9: Pull complete
04ab1bfc453f: Pull complete
a22fde870392: Pull complete
def16cac9f02: Pull complete
1604f5999542: Pull complete
Digest: sha256:f7ee67d8d9050357a6ea362e2a7e8b65a6823d9b612bc430d057416788ef6df9
Status: Downloaded newer image for redis:latest
Creating dockercomposefile_database_1 ... done
Creating dockercomposefile_web_1      ... done
Yukteshs-MacBook-Pro:DockerComposeFile ychinmka@in.ibm.com$
```

Let us see running containers

```
Yukteshs-MacBook-Pro:DockerComposeFile ychinmka@in.ibm.com$ docker container ls
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS               NAM
ES
7cb9fb926628        redis               "docker-entrypoint.s…"   25 minutes ago      Up 25 minutes       6379/tcp            doc
kercomposefile_database_1
be15386ab8da        nginx               "nginx -g 'daemon of…"   25 minutes ago      Up 25 minutes       80/tcp              doc
kercomposefile_web_1
Yukteshs-MacBook-Pro:DockerComposeFile ychinmka@in.ibm.com$
```

Or

```
Yukteshs-MacBook-Pro:DockerComposeFile ychinmka@in.ibm.com$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS               NAM
ES
7cb9fb926628        redis               "docker-entrypoint.s…"   25 minutes ago      Up 25 minutes       6379/tcp            doc
kercomposefile_database_1
be15386ab8da        nginx               "nginx -g 'daemon of…"   25 minutes ago      Up 25 minutes       80/tcp              doc
kercomposefile_web_1
```

-Now let us stop all these services by running ***docker-compose down***

```
Yukteshs-MacBook-Pro:DockerComposeFile ychinmka@in.ibm.com$ docker-compose down
Stopping dockercomposefile_database_1 ... done
Stopping dockercomposefile_web_1      ... done
Removing dockercomposefile_database_1 ... done
Removing dockercomposefile_web_1      ... done
Removing network dockercomposefile_default
Yukteshs-MacBook-Pro:DockerComposeFile ychinmka@in.ibm.com$ docker container ls
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
Yukteshs-MacBook-Pro:DockerComposeFile ychinmka@in.ibm.com$
```

Now no containers are running ie., we managed to stop 2 containers using 1 command