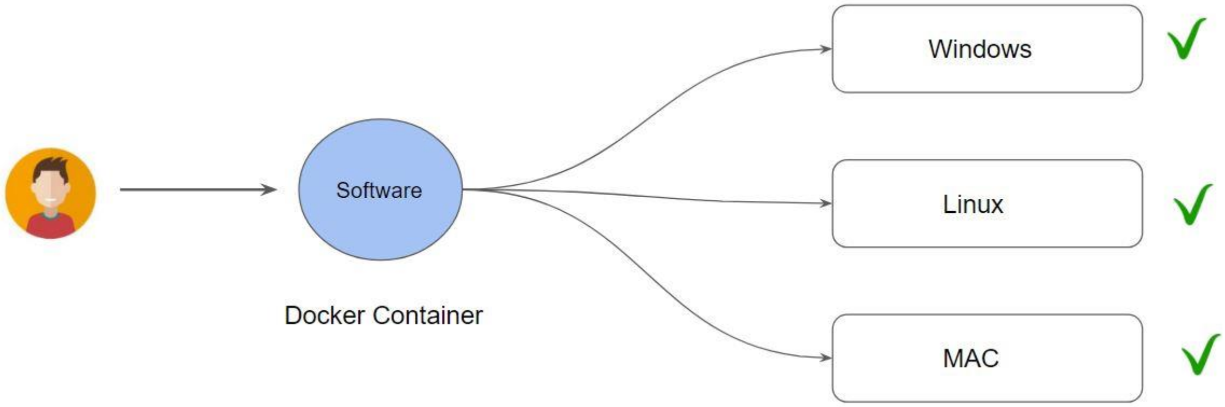
Module 1: Introduction to Docker

Docker is a technology designed to make it easier to create, deploy, and run applications by using containers.

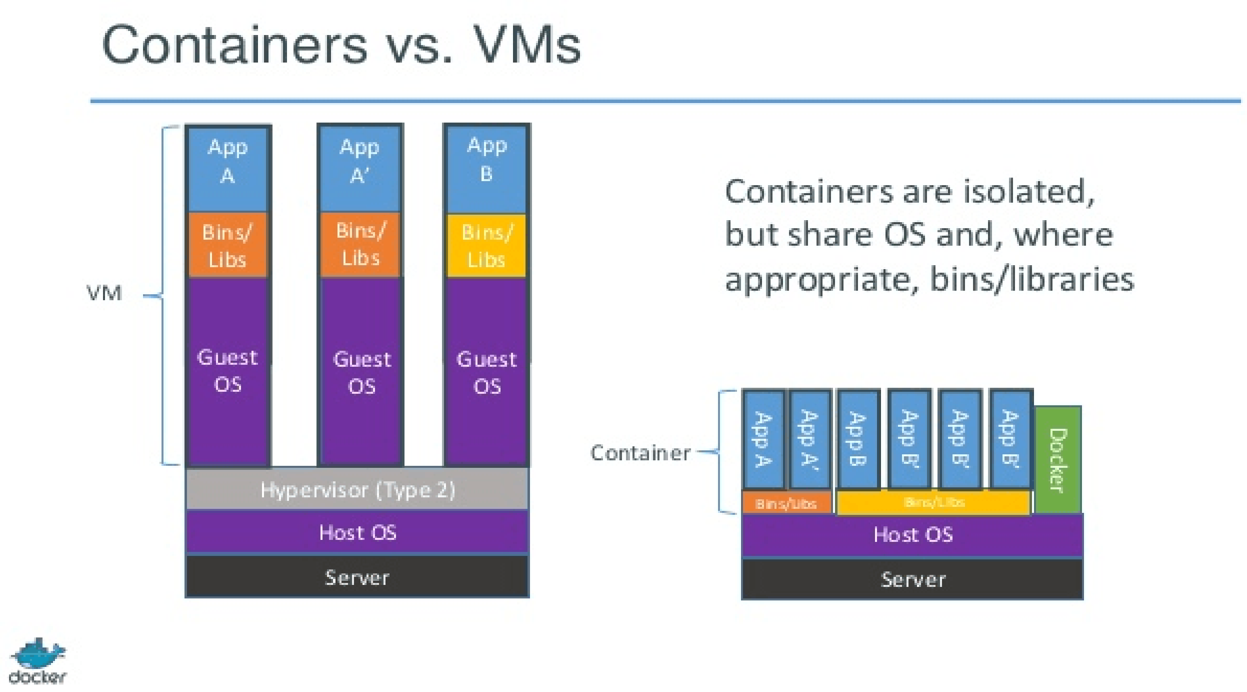
Docker is an open platform, once we build a docker image, we can run it anywhere, say it windows, Linux, mac whether on a laptop, data center, or in the cloud.

It follows the build once, run anywhere approach.



1.2 Docker Containers vs Virtual Machines

- Virtual Machine contains the entire Operating System.
- The container uses the resource of the host operating system



Containers vs. VMs

Containers are isolated, but share OS and, where appropriate, bins/libraries

Module 2: Installing Docker

Docker works on a wide variety of operating systems, this includes:

- Windows
- Linux
- MAC

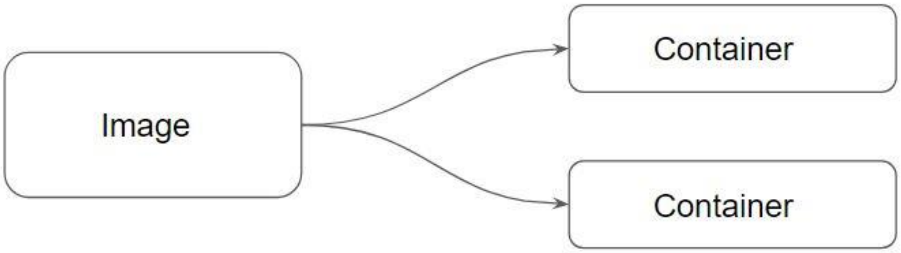
The installation of Docker is pretty straight forward in each one of them.

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

Module 3: Docker [Image vs Containers]

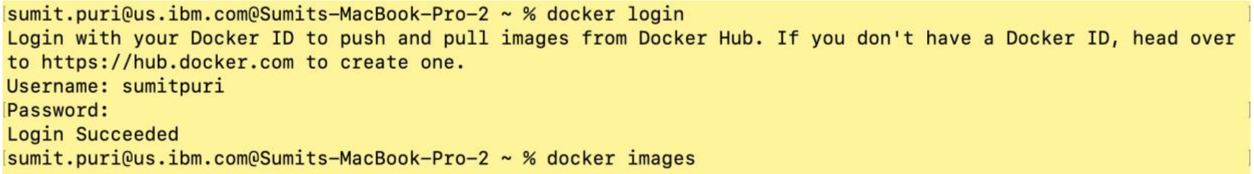
Docker Image is a file that contains all the necessary dependency and configurations which are required to run an application.

Docker Containers is basically a running instance of an image.



Module 4: Images

Use docker login command to download images from Docker Hub registry.



```
sumit.puri@us.ibm.com@Sumits-MacBook-Pro-2 ~ % docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over
to https://hub.docker.com to create one.
Username: sumitpuri
Password:
Login Succeeded
sumit.puri@us.ibm.com@Sumits-MacBook-Pro-2 ~ % docker images
```

Docker Registry is a storage and distribution system for docker images.

IBM’s Docker Registry is called IBM Container Registry (ICR). Install ICR and view images in ICR:

```
ibmcloud plugin install container-registry
ibmcloud login --sso
ibmcloud cr login
ibmcloud cr images –include-ibm
```

Download (Pull) Docker Images using:

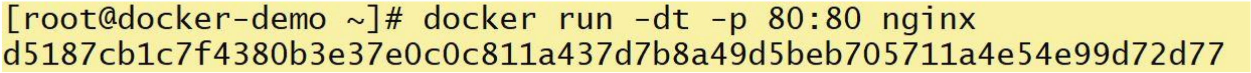
docker pull <image-name>

docker pull busybox
docker pull alpine

Module 5: Container Identification

When you create a Docker container, it is assigned a universally unique identifier (UUID).

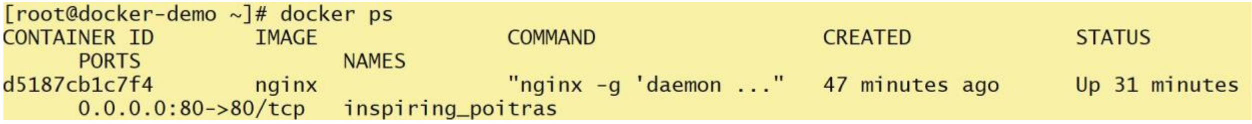
These can help identify the docker container among others.



```
[root@docker-demo ~]# docker run -dt -p 80:80 nginx
d5187cb1c7f4380b3e37e0c0c811a437d7b8a49d5beb705711a4e54e99d72d77
```

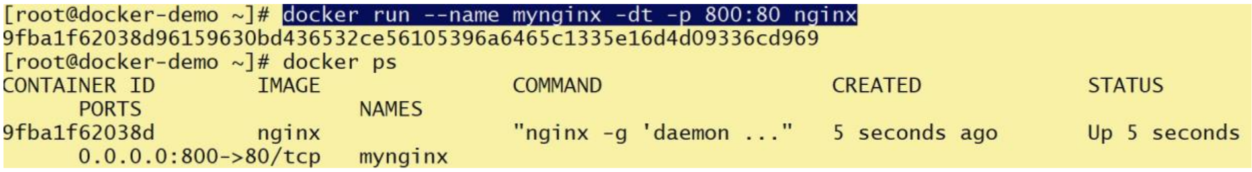
To help humans, Docker also allows us to supply container names.

By default, if we do not specify the name, docker supplies a randomly-generated name from two words, joined by an underscore



```
[root@docker-demo ~]# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS
      PORTS               NAMES
d5187cb1c7f4        nginx               "nginx -g 'daemon ..."   47 minutes ago      Up 31 minutes
      0.0.0.0:80->80/tcp   inspiring_poitras
```

By adding **--name=meaningful_name** argument during the docker run command, we can specify our own name to the containers.

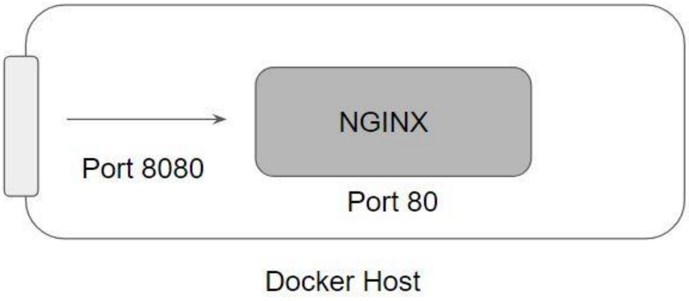


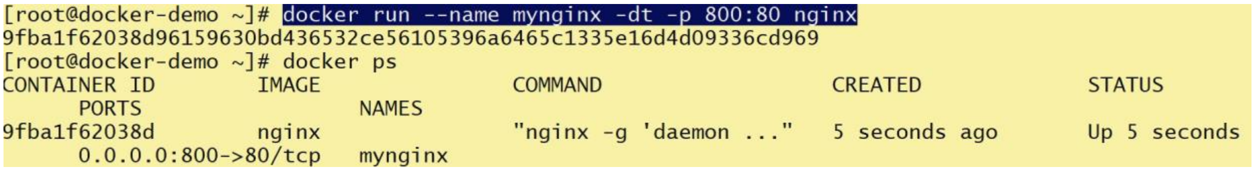
```
[root@docker-demo ~]# docker run --name mynginx -dt -p 800:80 nginx
9fba1f62038d96159630bd436532ce56105396a6465c1335e16d4d09336cd969
[root@docker-demo ~]# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS
      PORTS                NAMES
9fba1f62038d        nginx               "nginx -g 'daemon ..."   5 seconds ago       Up 5 seconds
      0.0.0.0:800->80/tcp   mynginx
```

Module 6: Port Binding

By default Docker containers can make connections to the outside world, but the outside world cannot connect to containers.

If we want containers to accept incoming connections from the world, you will have to bind it to a host port.





```
[root@docker-demo ~]# docker run --name mynginx -dt -p 800:80 nginx
9fba1f62038d96159630bd436532ce56105396a6465c1335e16d4d09336cd969
[root@docker-demo ~]# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS
      PORTS                  NAMES
9fba1f62038d        nginx               "nginx -g 'daemon ..."   5 seconds ago       Up 5 seconds
      0.0.0.0:800->80/tcp    mynginx
```

Module 7: Attached and Detached Mode

When we start a docker container, we need to decide if we want to run in a default foreground mode or the detached mode.

You may want to use this if you want a container to run but do not want to view and follow all its output.

Module 8: Removing Docker Containers

Docker containers can be removed with the help of **docker container rm** command.

Description	Command
Remove single container	docker container rm CONTAINER
Stop all the containers	docker container stop $(docker container ls -aq)
Remove all the containers	docker container rm $(docker container ls -aq)

Module 9: New Docker CLI Commands

Prior to docker 1.13, the docker run command was only available.

The cli commands were then refactored to have the form docker COMMAND SUBCOMMAND, wherein this case the COMMAND is container and the SUBCOMMAND is run

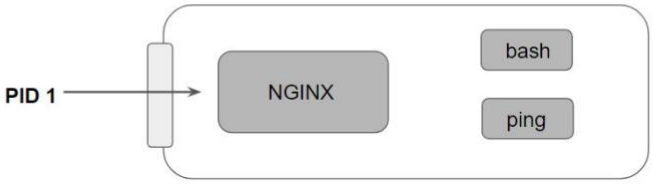
Older Approach: docker run
Newer Approach: docker container run

Both of these approaches will work perfectly.

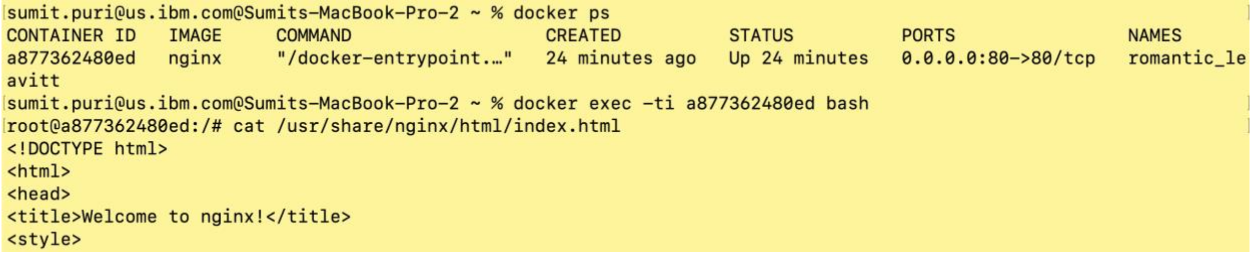
Module 10: docker container exec

The docker container exec command runs a new command in a running container.

The command started using docker exec only runs while the container's primary process (PID 1) is running, and it is not restarted if the container is restarted.



Nginx Docker Container

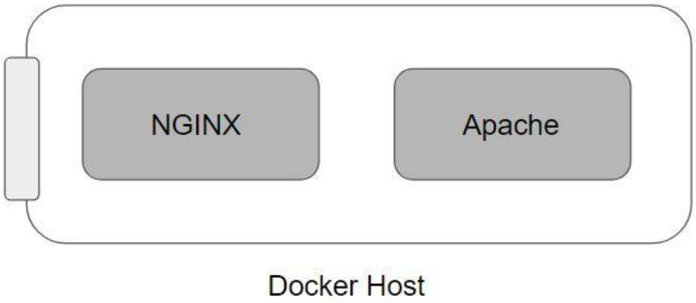


```
sumit.puri@us.ibm.com@Sumits-MacBook-Pro-2 ~ % docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS                NAMES
a877362480ed   nginx     "/docker-entrypoint.…"   24 minutes ago   Up 24 minutes   0.0.0.0:80->80/tcp   romantic_le
avitt
sumit.puri@us.ibm.com@Sumits-MacBook-Pro-2 ~ % docker exec -ti a877362480ed bash
root@a877362480ed:/# cat /usr/share/nginx/html/index.html
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
```

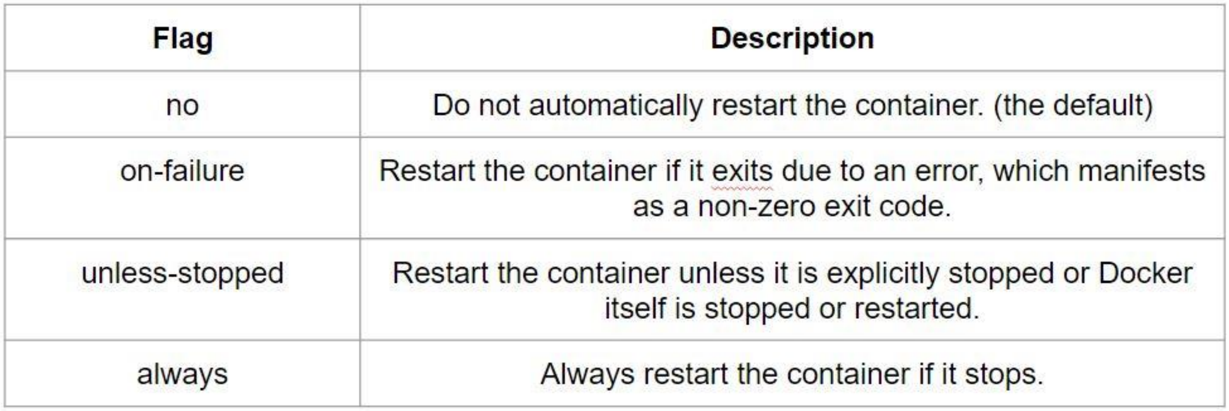
Module 11: Restart Policies

By default, Docker containers will not start when they exit or when docker daemon is restarted.

Docker provides restart policies to control whether your containers start automatically when they exit, or when Docker restarts.



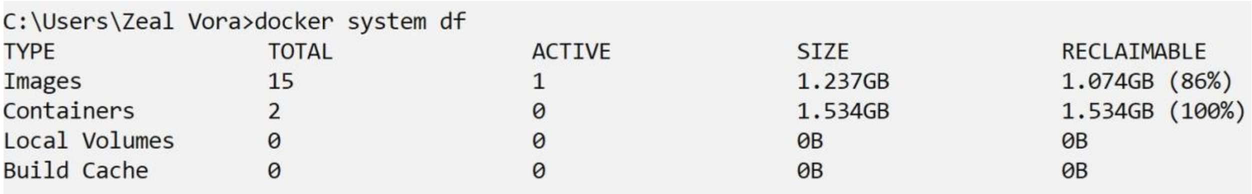
We can specify the restart policy by using the --restart flag with docker run command.



Flag	Description
no	Do not automatically restart the container. (the default)
on-failure	Restart the container if it exits due to an error, which manifests as a non-zero exit code.
unless-stopped	Restart the container unless it is explicitly stopped or Docker itself is stopped or restarted.
always	Always restart the container if it stops.

Module 12: Disk Usage Metrics

The docker system df command displays information regarding the amount of disk space used by the docker daemon.



```
C:\Users\Zeal Vora>docker system df
TYPE                TOTAL               ACTIVE              SIZE                RECLAIMABLE
Images              15                  1                   1.237GB             1.074GB (86%)
Containers          2                   0                   1.534GB             1.534GB (100%)
Local Volumes       0                   0                   0B                  0B
Build Cache         0                   0                   0B                  0B
```

Module 13: Automatically Delete Container On Exit

By default, containers that are exited are not removed by Docker.

With the **--rm** flag, the user can specify that whenever a container exits, it should automatically be removed.