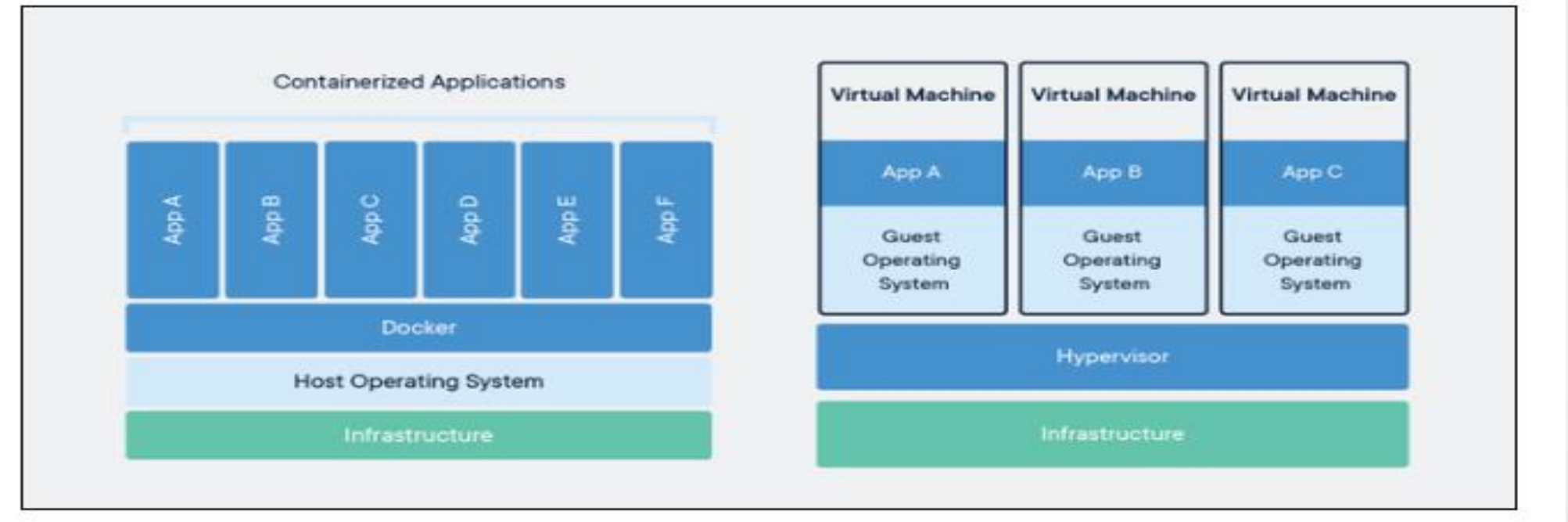
Docker

- Docker is a tool for running your applications inside containers.
- Containers package all the dependencies and code your app needs to run into a single file, which will run the same way on any machine.

- **What Is Docker?**
- Docker is similar in concept to Virtual Machines, except it's much more lightweight.
- Instead of running an entire separate operating system (which is a massive overhead), Docker runs containers, which use the same host operating system, and only virtualize at a software level.

What can I use Docker for?

- **Fast, consistent delivery of your applications**
- Docker streamlines the development lifecycle by allowing developers to work in standardized environments using local containers which provide your applications and services. Containers are great for continuous integration and continuous delivery (CI/CD) workflows.
- **Consider the following example scenario:**
- Your developers write code locally and share their work with their colleagues using Docker containers.
- They use Docker to push their applications into a test environment and execute automated and manual tests.
- When developers find bugs, they can fix them in the development environment and redeploy them to the test environment for testing and validation.
- When testing is complete, getting the fix to the customer is as simple as pushing the updated image to the production environment.



- Docker Engine runs on Linux, Windows, and macOS, and supports Linux and Windows for Docker containers.
- The exact flavor of Linux doesn't actually matter; most versions of Linux will run the same kernel, and only differ in the user software.
- Docker can install this user software to the container, allowing you to run a CentOS container on Ubuntu.
  You couldn't though, for example, run FreeBSD on Ubuntu, since the kernels are different.

Most Used Docker Commands

```
docker --version
docker --help
docker pull
docker run
docker build
docker login
docker push
```

```
docker ps
docker images
docker stop
docker kill
docker rm
docker rmi
docker exec
```

```
docker commit
docker import
docker export
docker container
docker compose
docker swarm
docker service
```

docker --version

This command returns the version of Docker which is installed

```
vaagdevi@ubuntu:~$ docker --version
Docker version 20.10.21, build 20.10.21-0ubuntu1~20.04.2
```

```
docker --help
```

This command returns a list of commands available in Docker along with the possible flags (options)

docker pull

```
$ docker pull ubuntu
```

This command pulls a new Docker image from the Docker Hub

docker images

$ docker images

This command lists down all the images in your local repo

```
vaagdevi@ubuntu:~$ sudo docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
hello-world   latest    9c7a54a9a43c   3 weeks ago    13.3kB
ubuntu        latest    3b418d7b466a   4 weeks ago    77.8MB
```

DOCKER PULL UBUNTU

```
vaagdevi@ubuntu:~$ docker pull ubuntu
Using default tag: latest
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post "http://%2Fvar
buntu&tag=latest": dial unix /var/run/docker.sock: connect: permission denied
vaagdevi@ubuntu:~$ sudo docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
dbf6a9befcde: Extracting [================>                                  ]   9.83MB/29.53MB
dbf6a9befcde: Pull complete
```

docker run

```
$ docker run ubuntu
```

This command executes a Docker image on your local repo & creates a running Container out of it

DOCKER RUN HELLO WORLD

```
vaagdevi@ubuntu:~$ sudo 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/
```

Basic Docker Commands

docker build

```
$ docker build -t MyUbuntuImage .
```

This command is used to compile the Dockerfile, for building custom Docker images based on the

```
vaagdevi@ubuntu:~/Downloads/demo$ sudo docker build -t "mycustom" .
Sending build context to Docker daemon  2.048kB
Step 1/2 : FROM ubuntu
 ---> 3b418d7b466a
Step 2/2 : RUN echo Hi,this is shazia from India!
 ---> Running in a176b1d806df
Hi,this is shazia from India!
Removing intermediate container a176b1d806df
 ---> 920eb85ea2bc
Successfully built 920eb85ea2bc
Successfully tagged mycustom:latest
```

Dockerfile(no extension to be given)

Open

```
1 FROM ubuntu
2 RUN echo Hi,this is shazia from India!
```

PWD and creating a directory

```
vaagdevi@ubuntu:~$ pwd
/home/vaagdevi
```

```
vaagdevi@ubuntu:~$ cd Downloads
vaagdevi@ubuntu:~/Downloads$ mkdir demo
vaagdevi@ubuntu:~/Downloads$ cd demo
vaagdevi@ubuntu:~/Downloads/demo$ cat DockerFile
cat: DockerFile: No such file or directory
vaagdevi@ubuntu:~/Downloads/demo$ cat >DockerFile
hi
hello
welcome
vaagdevi@ubuntu:~/Downloads/demo$ cat DockerFile
hi
hello
welcome
vaagdevi@ubuntu:~/Downloads/demo$
```

Basic Docker Commands

docker container

This command is used to perform various operations on the container. Refer to www.docs.docker.com *for more info.*

```
$ docker container logs
$ docker container kill
$ docker container rm
$ docker container run
$ docker container start
```

Docker container

```
docker container logs 4b8b34222783
```

To remove the container

```
docker container rm ecc253de9001
```

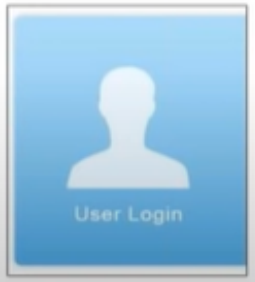
Don’t try the above command

Basic Docker Commands

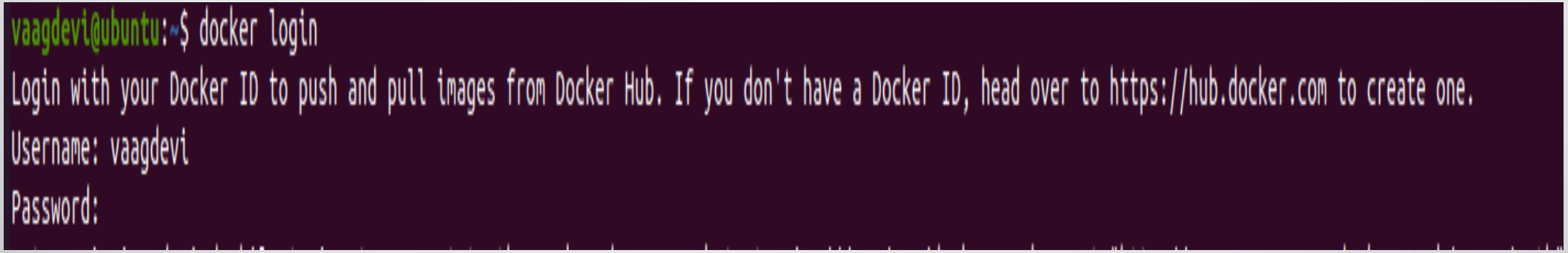
docker login

```
$ docker login
```

This command is used to Login to Docker Hub repo from the CLI



Username:vaagdevi
Password:vaag123456



docker ps

```
$ docker ps
```

This command lists all the running containers in the host
If '–a' flag is specified, shutdown containers are also displayed

```
$ docker ps -a
```

Ps commands

```
vaagdevi@ubuntu:~/Downloads/demo$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```

```
vaagdevi@ubuntu:~/Downloads/demo$ sudo docker ps -a
CONTAINER ID   IMAGE         COMMAND    CREATED          STATUS                      PORTS     NAMES
4b8b34222783   hello-world   "/hello"   11 minutes ago   Exited (0) 11 minutes ago             elated_hamilton
3edd15becc0d   hello-world   "/hello"   21 minutes ago   Exited (0) 21 minutes ago             wizardly_villani
```

docker stop

```
$ docker stop fe6e370a1c9c
```

This command shuts down the container whose Container ID is specified in arguments. Container is shut down gracefully by waiting for other dependencies to shut

docker kill

```
$ docker kill fe6e370a1c9c
```

This command kills the container by stopping its execution immediately. Its similar to force kill

docker rm

```
$ docker rm fe6e370a1c9c
```

This command removes the container whose Container ID is specified in arguments

docker rmi

```
$ docker rmi MyUbuntuImage
```

This command removes the image whose name has been specified in arguments