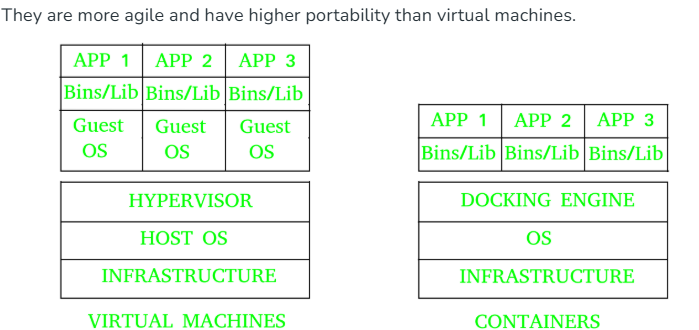
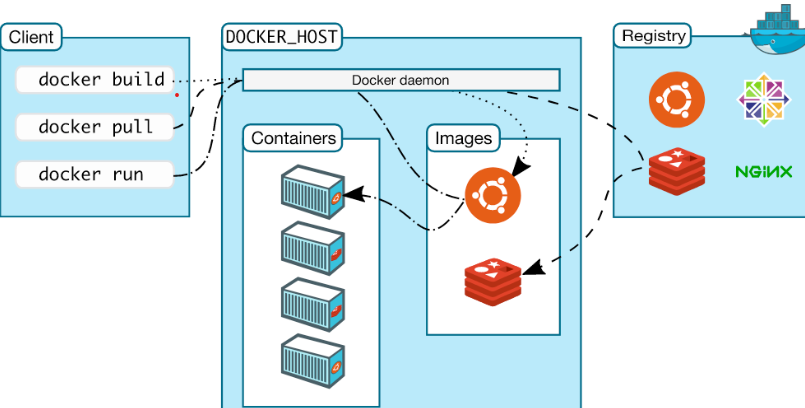
**Virtual Machine vs Container**

****

**How containers are logically isolated?**

1. Each container has its own filesystem & processes so one can’t see other’s files & processes.
2. Each container gets own private IP and hostname. Public IP is common i.e. host IP for external communication.
3. Each container runs its processes in its own ControlGroup which control how much resources a container can use so that other containers are not affected. (ControlGroup is a feature of linux kernel which controls limit of resources that processes can use)

**Architecture**



**Commands**

Docker build: Uses docker file to create an image. Image has: code, libraries, dependencies, runtime

Docker push: sends the built image to container registry (docker hub or acr)

Docker run: creates a container with my app and everything inside it

**Terminologies**

Docker daemon: dockerd listens to dockerAPI requests and manages all docker objects like images, containers, volumes, networks. It is heart of docker.

Docker client: It is docker CLI. Communicates with dockerd what to do using dockerAPI

Docker registry: version control for docker images

**Process:** Install docker. Add Ubuntu user to docker, since docker runs as admin. So we need to add our user to docker group: sudo usermod –aG docker Ubuntu

**Docker file for python project**

FROM python:3.10  #get python image from docker hub. Here python image already includes minimal Linux OS

WORKDIR /app #set current working directory inside image to /app. All commands copy, run executed inside it

COPY requirements.txt #copy this file from host machine (where docker runs) to /app inside container

RUN pip install requirements.txt #install python packages listed in requirements.txt

COPY . . #copy everything from project folder on host to /app inside image. Keep unwanted files .dockerignore

CMD ["python","app.py"]  #Entrypoint. Here CMD to run when container starts.here it runs python.py inside container

docker build –t user\_name/repo\_name:latest .

docker push user\_name/repo\_name:latest