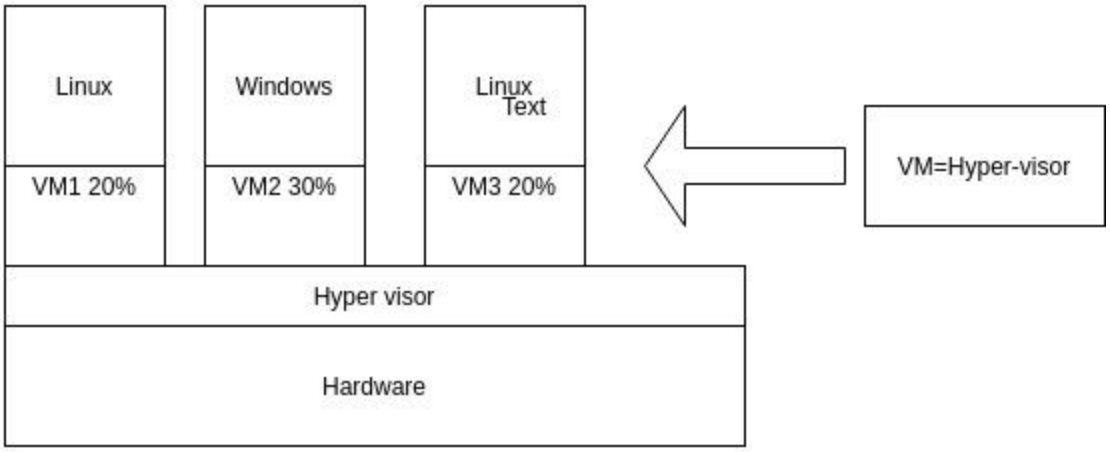
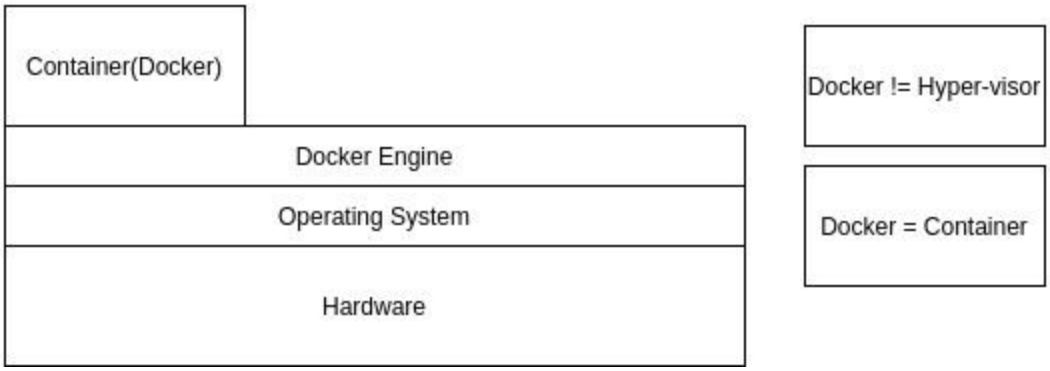
1.Docker

a) Physical Servers

b) Virtual Servers



c) Docker



1. Docker Engine is not in the docker project it is use as registry,ocestration,service and security
2. Registry - Place store your docker images (docker hope-largest repository),customize existing images and push it back

3. Ocestration - is the process all instances get together and go for a common goal . ex: If we have to deploy some services(auth,login) in different host what ocestraion does get responsibility move those services to host and controlling steps - Kubernetes
4. Docker is a kind of persistence or like a VM
5. Legacy applications doesn't support docker architecture - half true , better migrate
6. Docker fit with micro level service architecture
7. OCI - open container initiative (Docker + RKT)
8. Use principal call allocate and share(this much of time from container) - can do process allocation for container(allocating processing power)/limit the cpu CFS(completely fair scheduler) quota / limit cpu real time period in microseconds(out 100000ms need much 50000ms) / if allocated time exceed it won't affect other containers/ex :Unix -Nice - you may have 32 cores it is not you have to run 32 processes.
9. No of containers can exceed no of cores(CPU)/same as memory
10. Docker is a container which is another process in operating system
11. Docker toolbox - include linux image to process docker command(windows ,mac)
12. Dockerfile -
    FROM ubuntu:latest - command docker repository go and download : is version
    CMD ["echo","hello docker"] - command for print
    RUN - command for run
    WORKDIR - command for change working directory
    EXPOSE - expose localhost current port
    ENTRYPOINT - when start docker this is the thing you have to do
    ADD - command copy file given location add it into container given location
13. Docker commands
    docker build . - need to build location