Docker

⇒ Tool for running applications in an Isolated Environment.
⇒ Similar to Virtual Machine.
⇒ App run in same environment.
⇒ Standard for software deployment

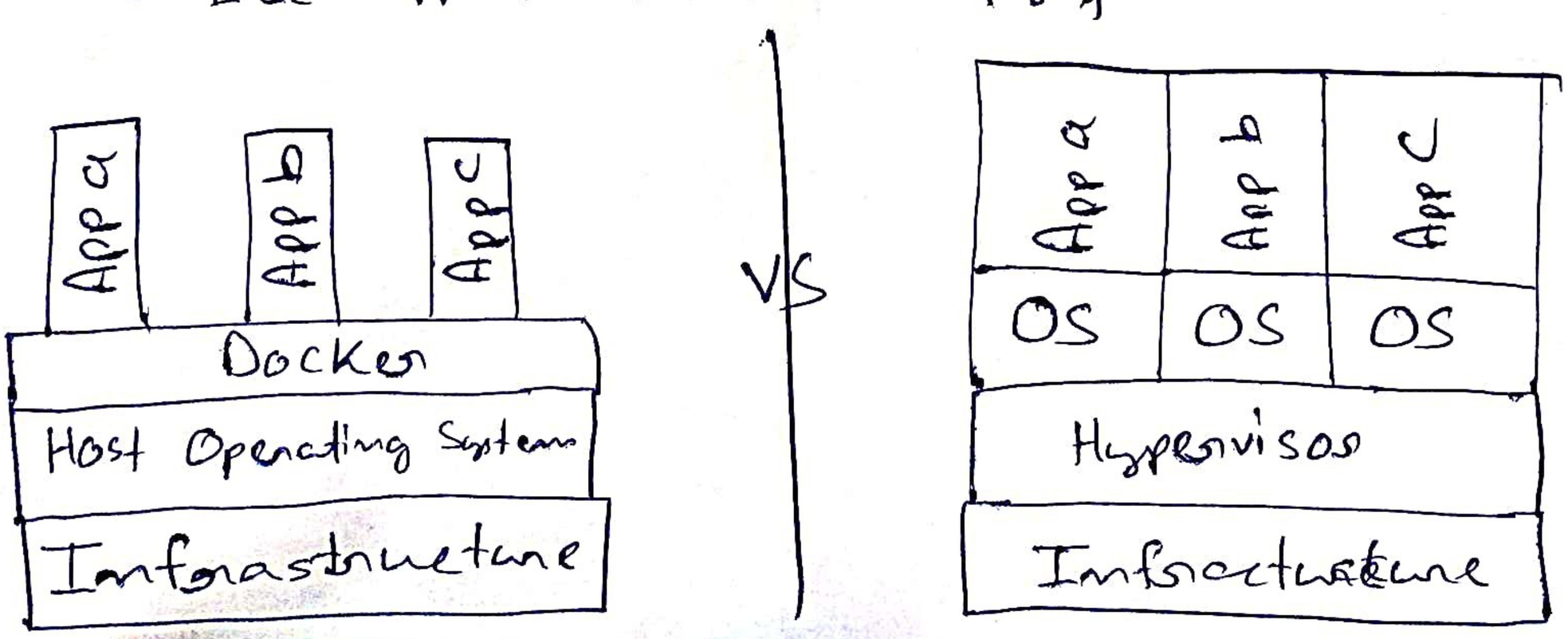
* Container Vs Virtual Machine

Container

→ Abstraction at the app layer that packages code and dependencies together.
→ Multiple Containers can run on the same machine & share the OS Kernel with other Containers.
→ Each running as isolated processes in user space.

Virtual Machine

→ Abstraction of physical hardware turning one server into many servers.
→ Hypervisor allows multiple VMs to run on a single machine.
→ Each VM include a full copy of an Operating System.



★ Docker Image

⇒ Template for creating an environment of your choice.

⇒ Has everything need to run your Apps.

↳ OS, Software, App code

★ Containers

⇒ Running instance of an image.

docker
↳ gives all the commands & options you can use with docker.

docker --version

docker ps ——→ list Containers

docker images
↳ Shows list of docker Image you have locally.

★ Running Containers

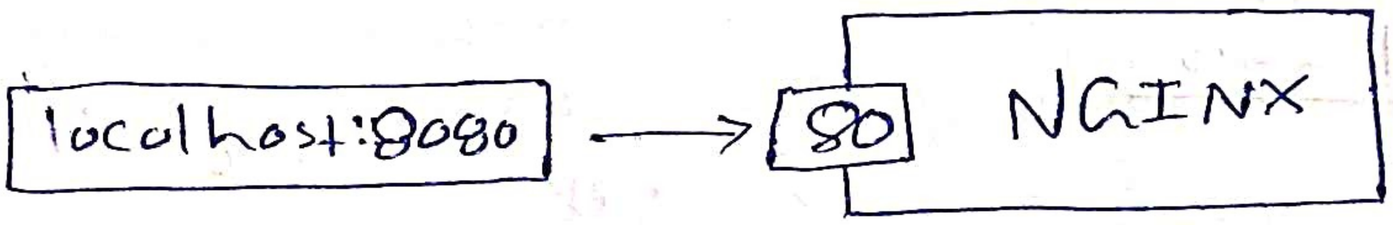
docker run <image.name>:<Tag>

Exaple: docker run nginx:latest

docker Container ls
↳ List all running Containers

docker run -d <img.name>:<Tag> {run in ditached mode}
docker stop <Contained_Id> {Stop the Container}

★ Exposing ports



docker run -d -p 8080:80 nginx:latst

★ Exposing multiple ports

#docker run -d -p 3000:80 -P 8080:80 nginx:latest

★ Managing Containers

docker stop <Container Id or name>

docker start < Cont Id or name>

↳ To start stoped container

docker ps -a

↳ Shows all the Containes, not just running.

docker rm < Cont Id or name>

↳ To remove the Container.

docker rm $(docker ps -aq)

↳ To remove all the Containers.

↳ You Cannot remove a running Container

docker rm -f $(docker ps -aq)

↳ Removes all the Containers irrespective of whether it is running or not.

★ Naming Containers

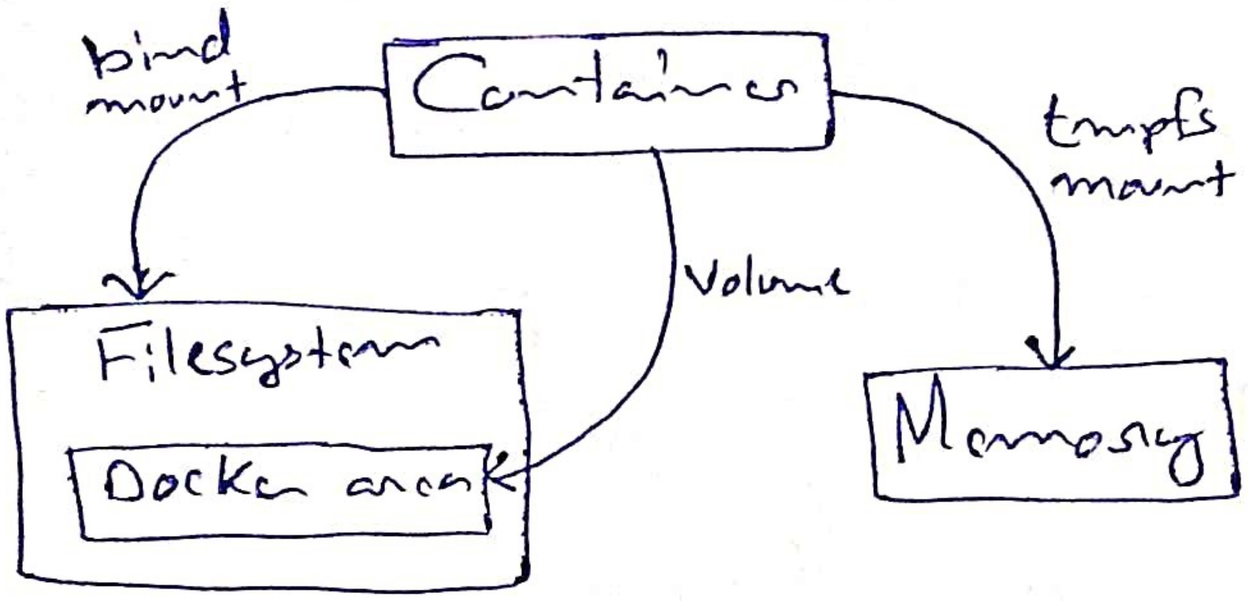
docker run --name <name> nginx:latest

★ Docker PS --Format

↳ To format the output of docker PS.

* Volumes

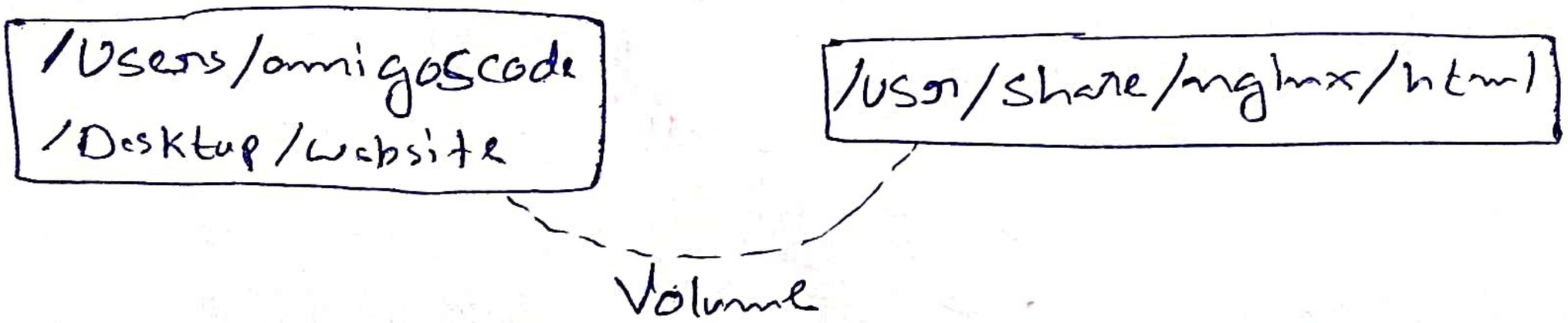
⇒ Allows sharing of data. Files & Folders.



⇒ Between host & Containers

⇒ Between Containers.

* Volumes between host & Container



Host

Container

```
# docker run --name website -v $(pwd):/usr/share/nginx/html
  :ro -d -p 8080:80 nginx:latest
```

↓

read only

```
# docker exec -it <Container name> bash
```

↓ {Interact with the Container using bash Commands}

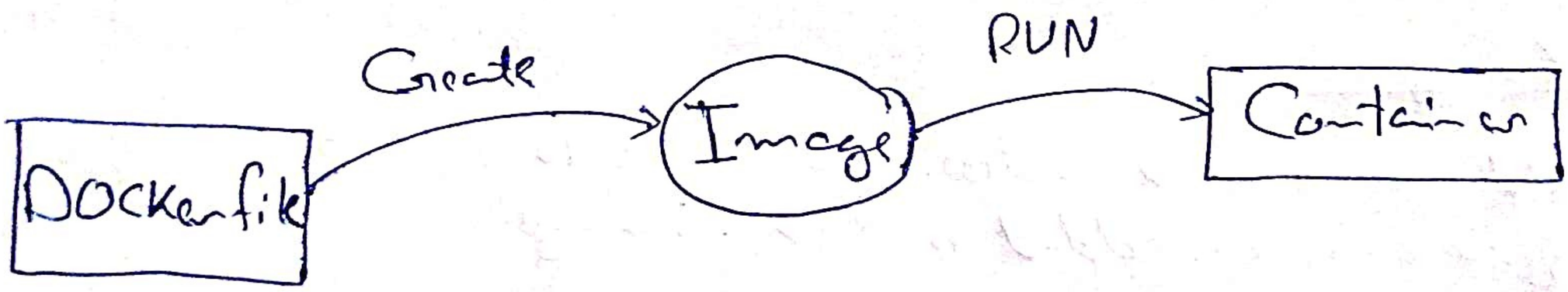
★ Volumes (Between Containers)

docker run --name website_copy --volumes-from website -d -P 8081:80 nginx:latest

→ Volumes shared between the two containers website & website_copy

★ DOCKERFILE

→ Build our own images.



★ Creating dockerfile

⇒ Docker Image has everything to run your apps
↳ OS, Software, App Code

⇒ Create a file named Dockerfile in the root of your project..

FROM <base_image>
↳ The name of the base image to use

ADD . /usr/share/nginx/html
↓ ↓
Some target

docker build --tag <name>:<tag> .

(or)
-t

Another Example

FROM node:latest → base image
WORKDIR /app → Create app directory if do not exist and enter into it
ADD . . → Add content of current directory of host to current directory of image
RUN npm install → Install dependencies
CMD node index.js → Run this command

★ .DOCKERFILEIGNORE (Docker File Ignore)

.dockerignore

→ Add the files & folders you want to ignore to be added to docker image

★ Caching and Layers

⇒ State of image is cashed at every step of docker file.

⇒ At certain steap if things changes, then it stops using caches from then on.

⇒ Each step Create a Layer that is cashed.

★ Alpine
↳ Its a small & simple linux distribution

#docker pull node:lts-alpine

#docker image rm <Image-id>
↳ To Remove a docker Image.

★ Tag, Versioning and Tagging

⇒ Allows you to control image version.

★ ~~Tagging overview~~

★ Docker registries → {Computer Server that holds all of our Images}

⇒ Highly scalable & server side application that stores & lets you distribute Docker images.

⇒ Used in your CD/CI Pipeline.

Docker Hub

docker push <username>/<imagename>:<tag>

docker login → for login

docker rmi <Image name> → To remove image

★ Docker Inspect

docker Inspect <container id or name>

★ Docker Logs

docker logs <container id or name>

→ -f for showing logs continuously

* Docker Exec

docker exec -it <container id> /bin/bash