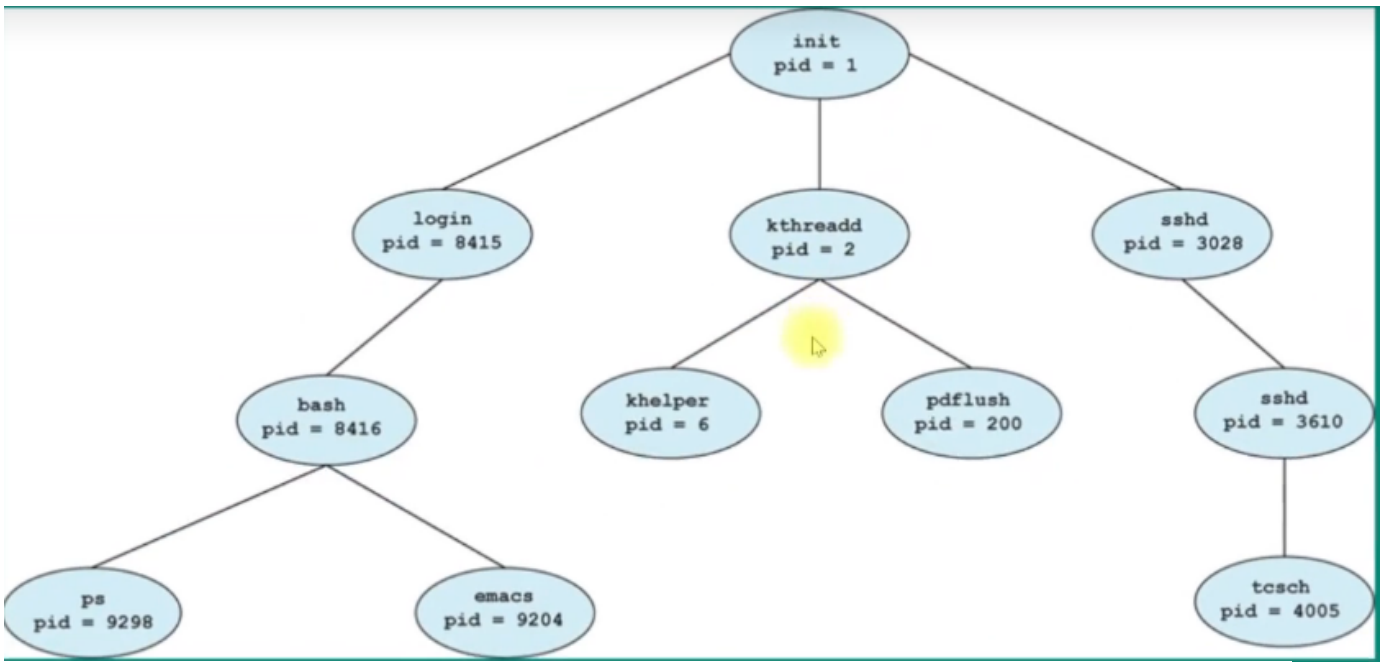
8. Introducing Containers

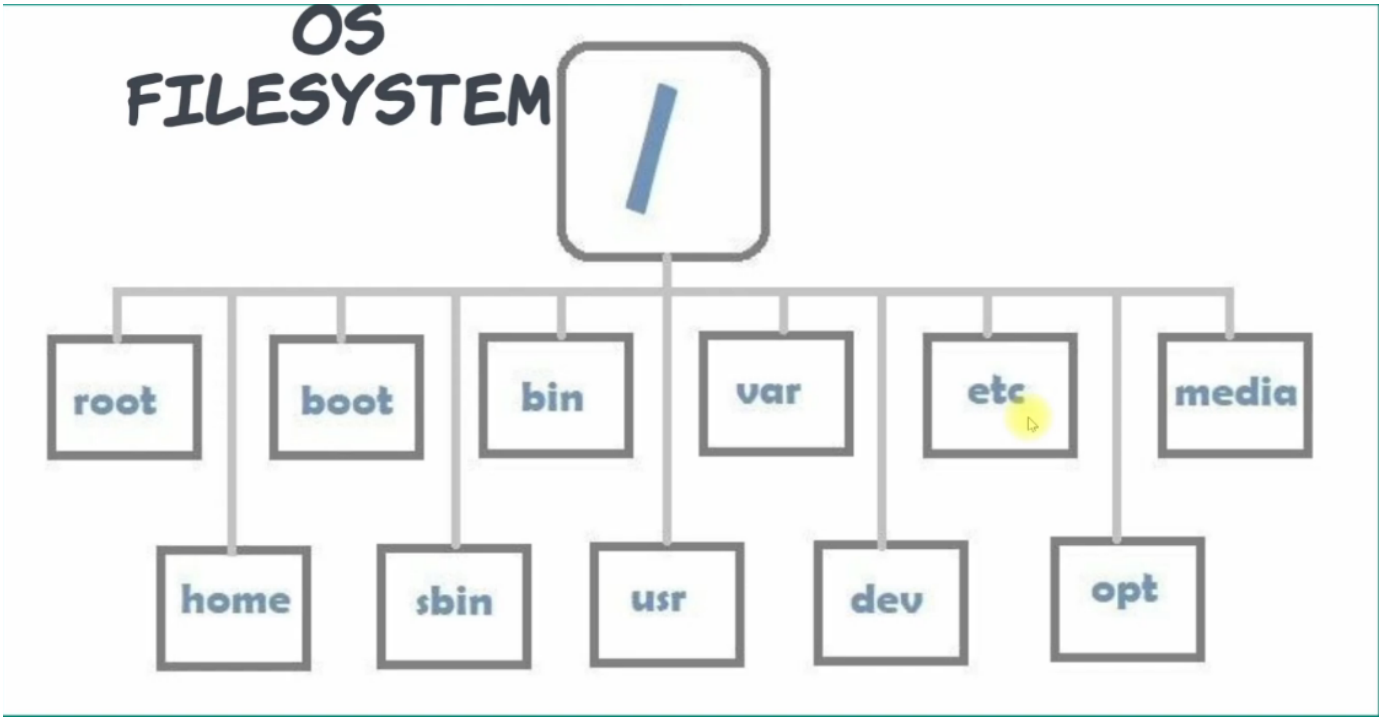
What does container do?

It Isolate your application"

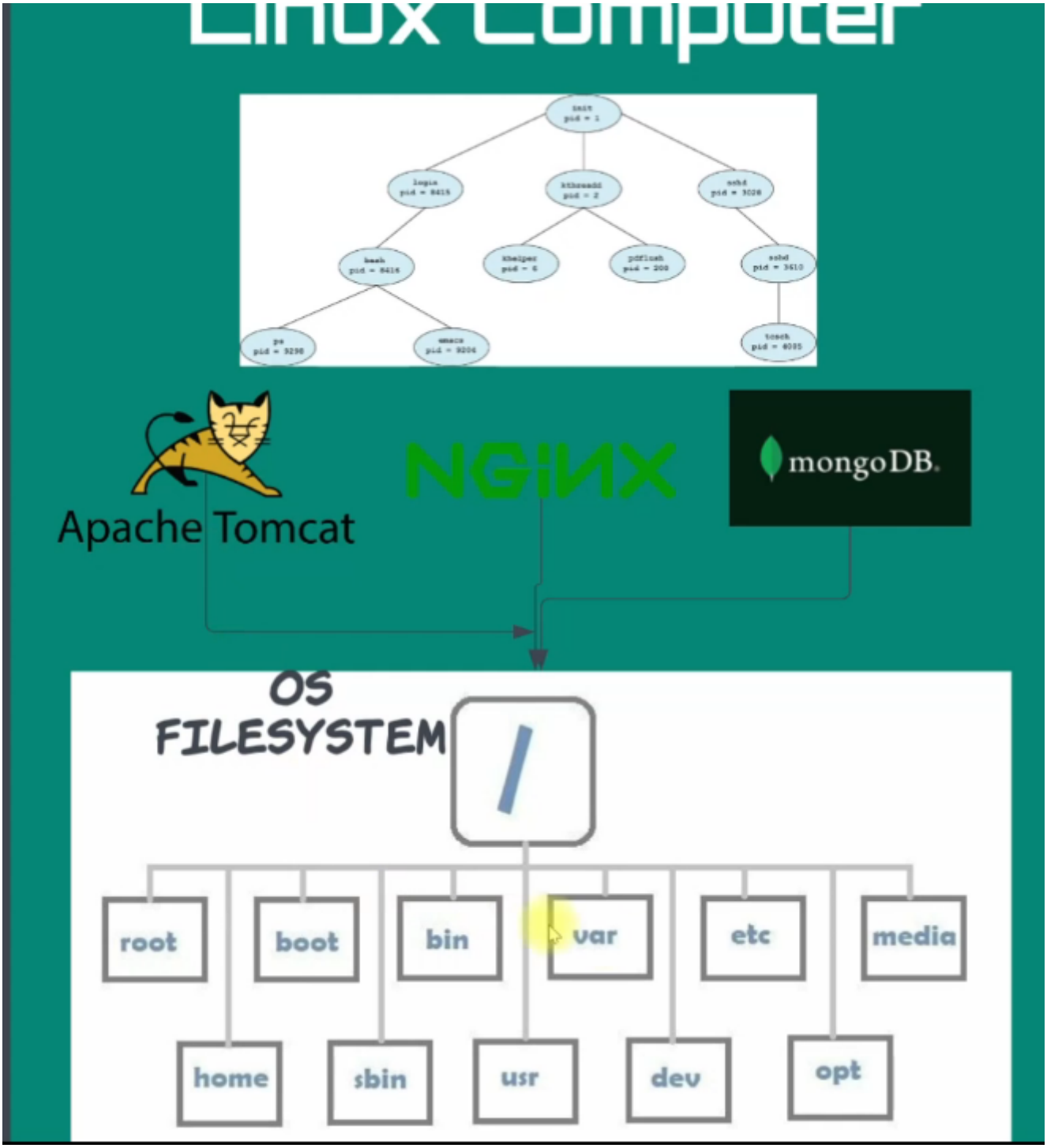


Some Important Directories

- Home Directories: `/root`,`/home/username`
- User Executable: `/bin`, `/usr/bin`, `/usr/local/bin`
- System Executables: `/sbin`, `/usr/sbin`, `/usr/local/sbin`
- Other Mountpoints: `/media`, `/mnt`
- Configuration: `/etc`
- Temporary Files: `/tmp`
- Kernels and Bootloader: `/boot`
- Server Data: `/var`, `/srv`
- System Information: `/proc`, `/sys`
- Shared Libraries: `/lib`, `/usr/lib`, `/usr/local/lib`


OS
FILESYSTEM
/
root
boot
bin
var
etc
media
home
sbin
usr
dev
opt

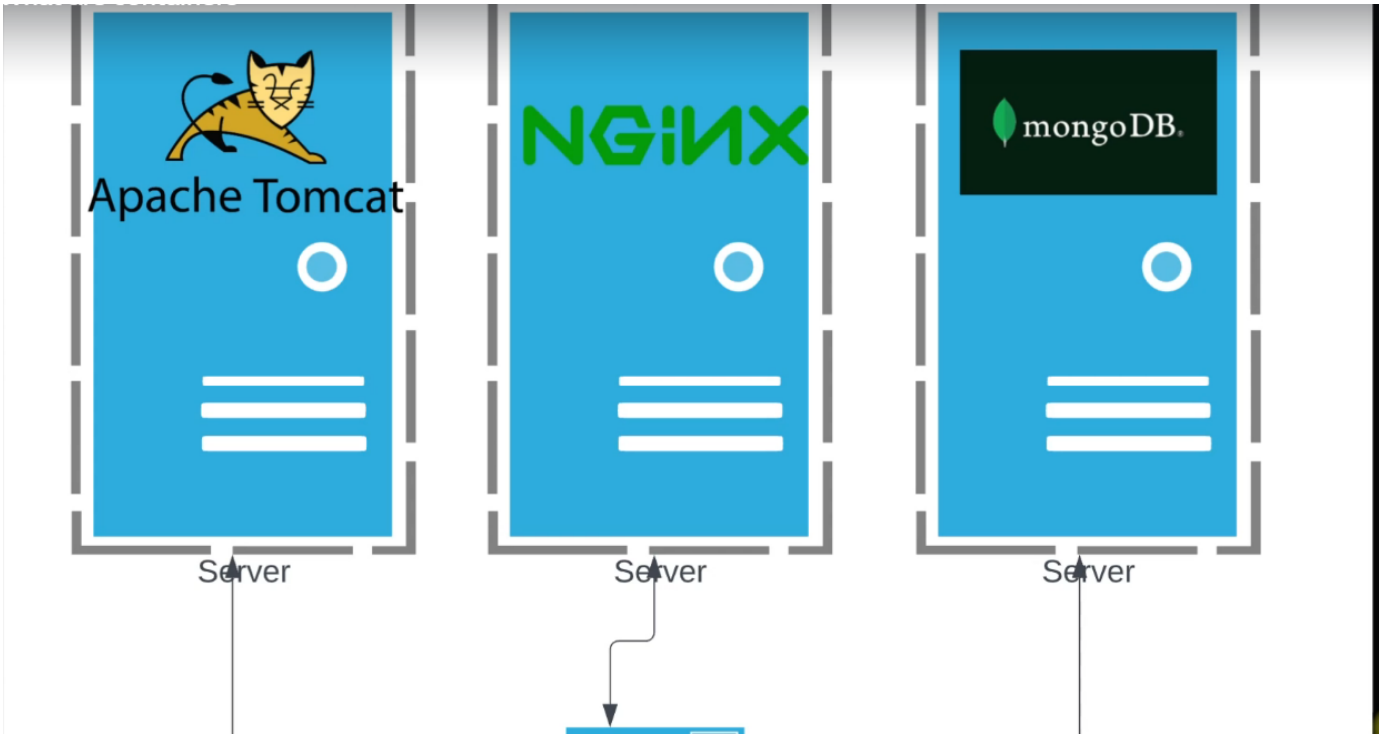
Linux computer normally running various services



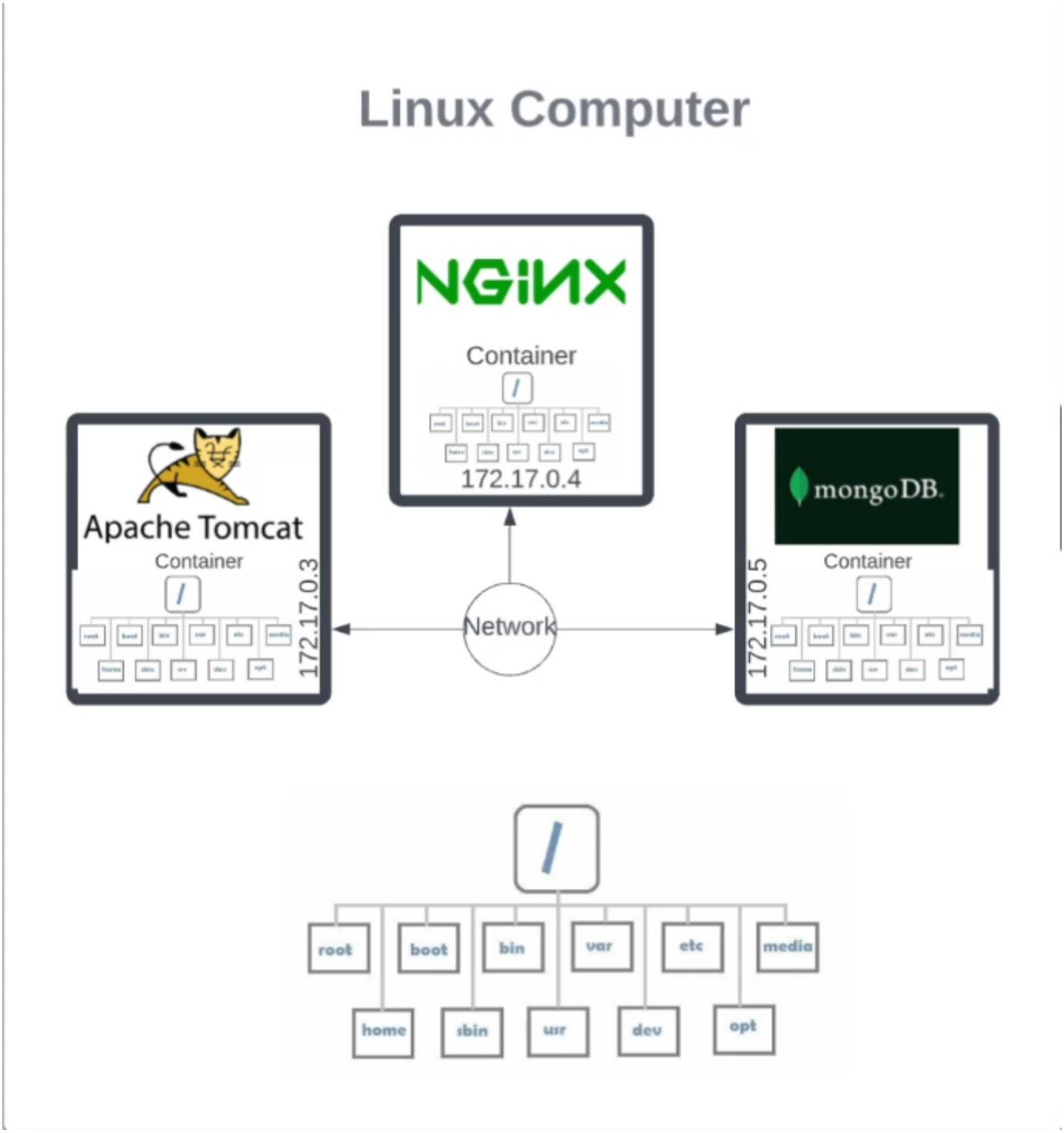
There are different services and processes running in the same computer which makes it difficult in chaning the configuration and all other files

so what we need is isolation. By isolation we mean we run different processes in different computers

which will ease out things

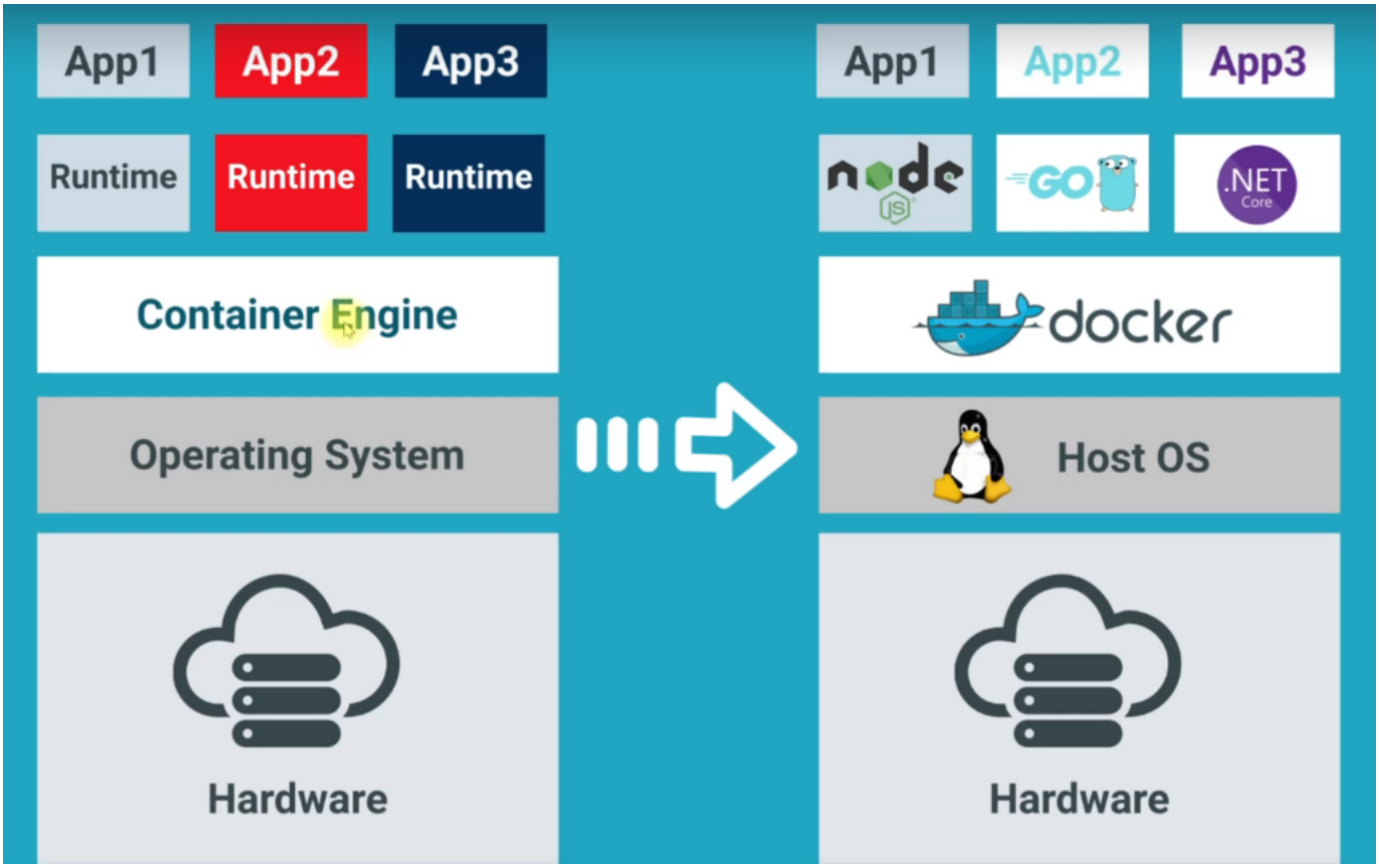


But this will obv create more problems because buying various computers for multiple processes is not possible financially. So for this problem we have **containers**.



Now with container all the problem is solved. each miniature OS or directory (container) has its own file system. This will not contain all the files that an OS has but only contain the files and configuration each specified applicaiton requires. For example In apache tomcat container PID 1 will be apache tomcat process id. Since they are very small and very light weight they can be archieved. These archieve are called as image or container image.
Its purpose is that we can ship it anywhere. We can run it in desktop and same thing we can run in production server.

But how is all this possible??? (Containerization)



What is Docker?

Docker is an open platform for developing, shipping and running applications. Where ever you have docker daemon installed you can run docker application

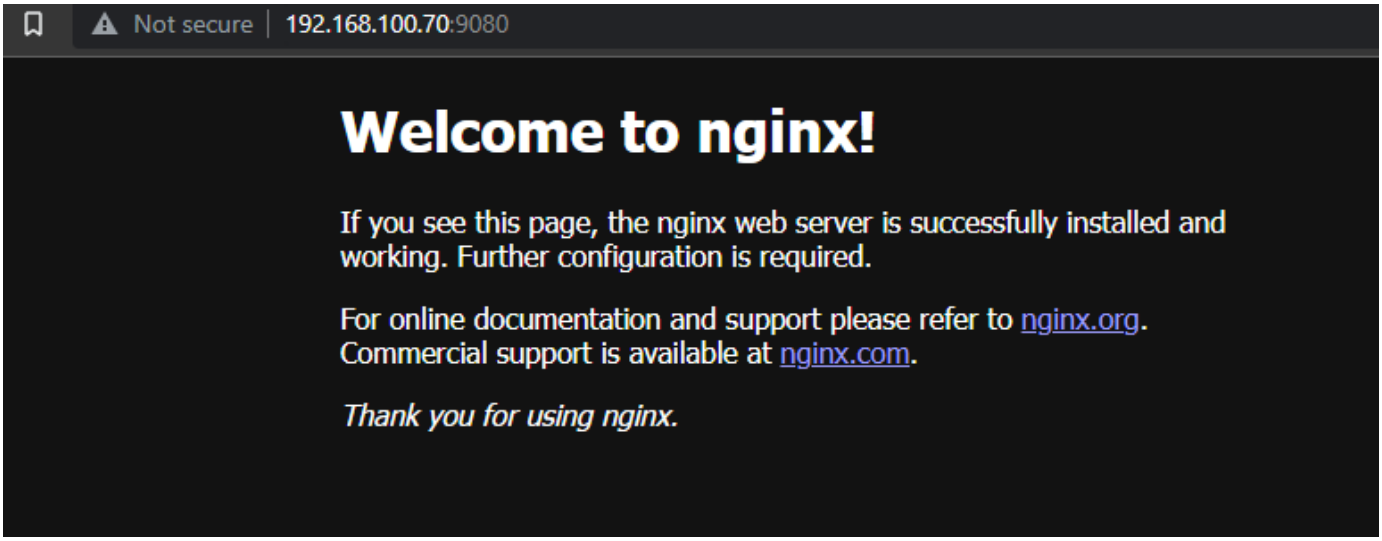
Docker provides the ability to package and run an application in a loosely isolated environment called a container.

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

`docker images` will show you images that are available
`docker ps` will show running containers
`docker ps -a` will show you all the containers

```
root@ubuntu-focal:~# docker run --name web01 -d -p 9080:80 nginx
```

-d : run it in the background
-p : port for host and container (9080 is the host port and 80 is the container port) for now host is the virtual machine and container port is running in an internal network so to access we have to do port mapping



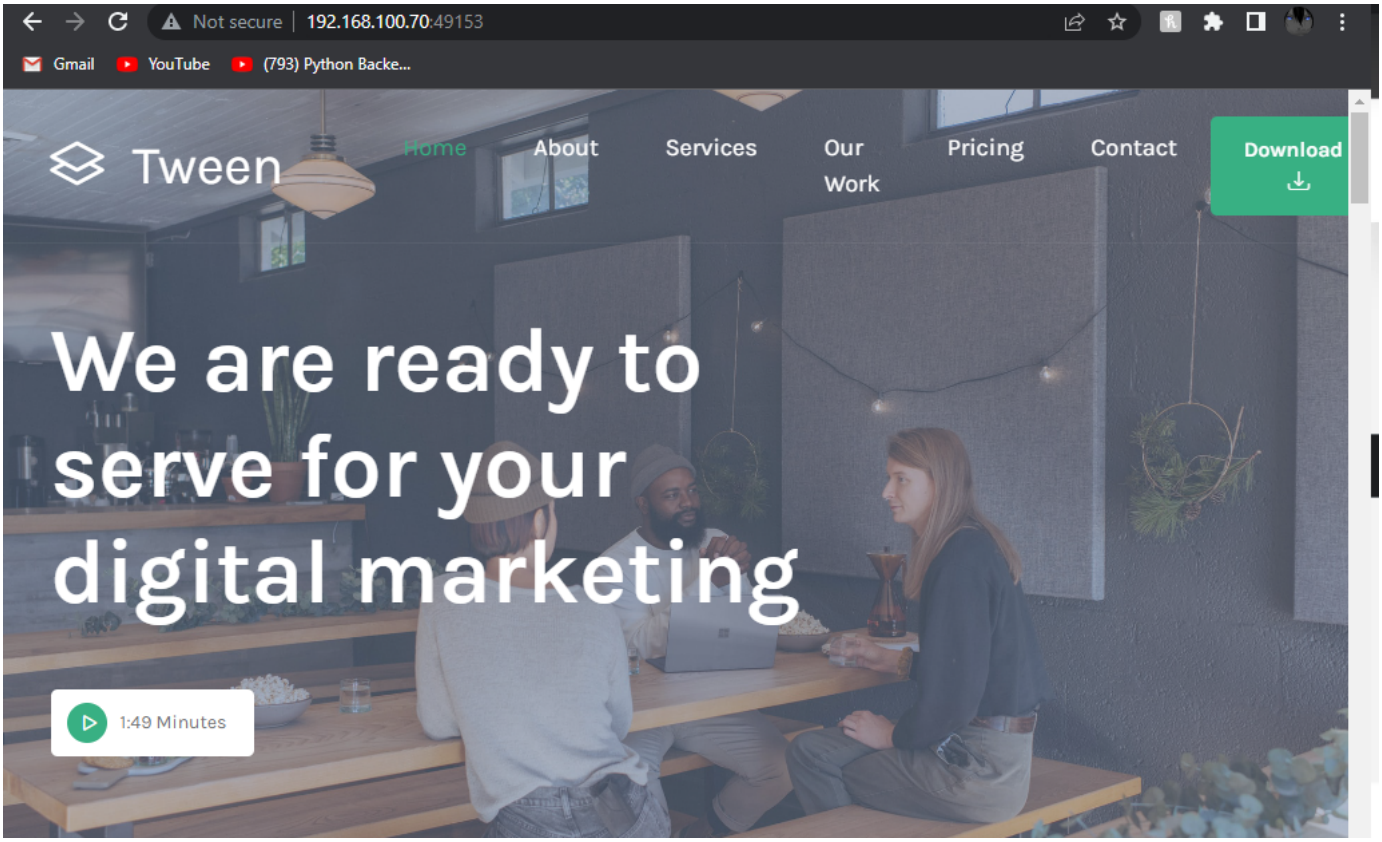
there you go this is coming straight from the container.

while creating our own image we do

```
# Building an Image
mkdir images
cd images/
vim Dockerfile

# Paste below content
FROM ubuntu:latest AS BUILD_IMAGE
RUN apt update && apt install wget unzip -y
RUN wget https://www.tooplate.com/zip-templates/2128_tween_agency.zip
RUN unzip 2128_tween_agency.zip && cd 2128_tween_agency && tar -czf tween.tgz * && mv tween.tgz /root/tween.tgz

FROM ubuntu:latest
LABEL "project"="Marketing"
ENV DEBIAN_FRONTEND=noninteractive

RUN apt update && apt install apache2 git wget -y
COPY --from=BUILD_IMAGE /root/tween.tgz /var/www/html/
RUN cd /var/www/html/ && tar xzf tween.tgz
CMD ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]
VOLUME /var/log/apache2
WORKDIR /var/www/html/
EXPOSE 80
root@ubuntu-focal:~/images# docker build -t tesimg .
```

this command will build our docker image.. where tesimg is the image name and '.' is the current working directory.

```
root@ubuntu-focal:~/images# docker images
REPOSITORY    TAG       IMAGE ID       CREATED         SIZE
tesimg        latest    d4431526e6e9   2 hours ago     250MB
<none>        <none>    42f9e5968b72   2 hours ago     127MB
nginx         latest    670dcc86b69d   12 days ago     142MB
ubuntu        latest    27941809078c   7 weeks ago     77.8MB
hello-world   latest    feb5d9fea6a5   10 months ago   13.3kB
root@ubuntu-focal:~/images# docker run -d -P tesimg
bde0083dad00b2a0ef14cf35469272e81eb8d8d8967cd8d4df90f83c937dde23
root@ubuntu-focal:~/images# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS                                       NAMES
bde0083dad00   tesimg    "/usr/sbin/apache2ct…"   7 seconds ago   Up 6 seconds   0.0.0.0:49153->80/tcp, :::49153->80/tcp     angry_nightingale
ff17d3062570   nginx     "/docker-entrypoint.…"   2 hours ago     Up 2 hours     0.0.0.0:9080->80/tcp, :::9080->80/tcp       web01
```



to stop the docker container just do

docker stop web01 angry_nightingale
these are the name of 2 docker images.

to remove the containers
docker rm

to remove the images
docker rmi

Some question

Question 3:

Command to list all the containers including exited container.

- ○ docker ps
- ○ docker all
- ◉ **docker ps -a**