Nottingham Trent University

Department of Computer Science

Department of Computer Science

Use of docker for teaching Computer Sciences subjects in HE

Pedro Machado – Senior Lecturer in Computer Sciences @ NTU
Pedro.machado@ntu.ac.uk

Outline

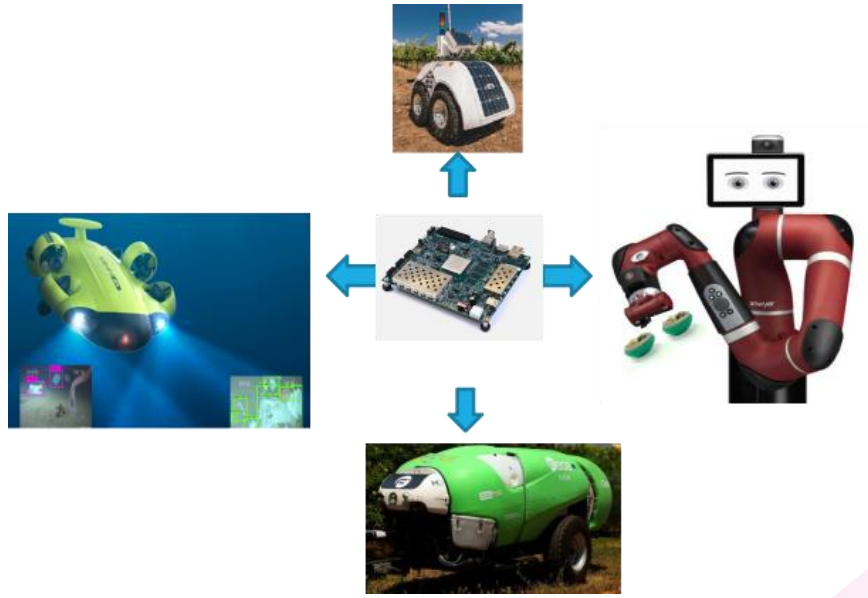
- Research interests
- Introduction to Docker
- Docker demo
- Conclusions

Research Interests

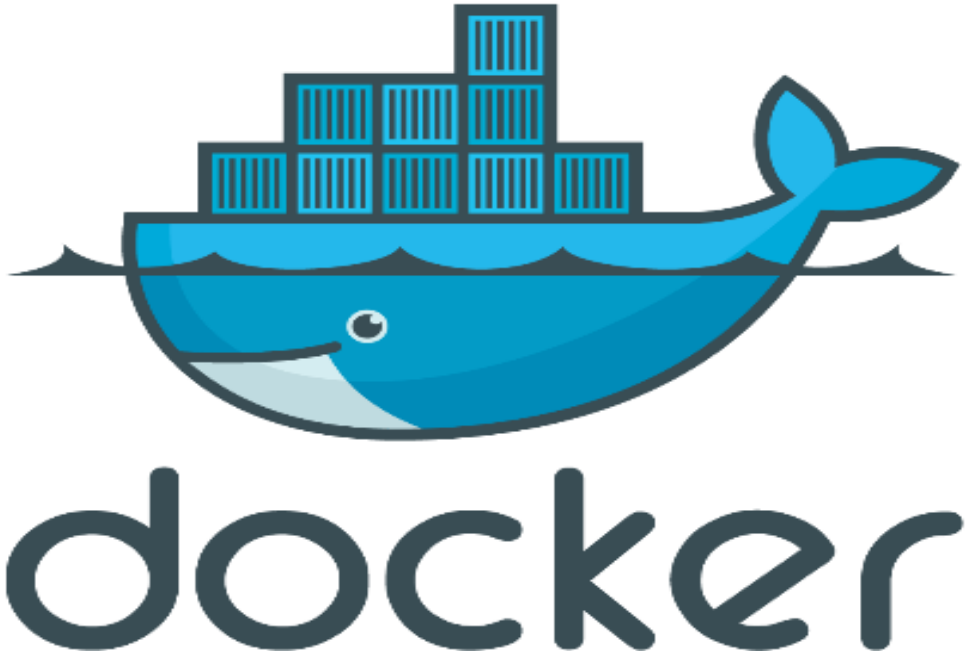
- Edge Computing
- Neuromorphic Engineering
- Robotics
- Intelligent Sensors
- Spiking Neural Networks

UPD project

IoW project



Introduction to Docker



NTU

What is Docker ?!!!

- Open platform for developers and sysadmins to build, ship and run distributed applications
- Can run on popular 64-bit Linux distributions with kernel 3.8 or later
- Supported by several cloud platforms including Amazon EC2, Google Compute Engine, Microsoft Azure and Rackspace.

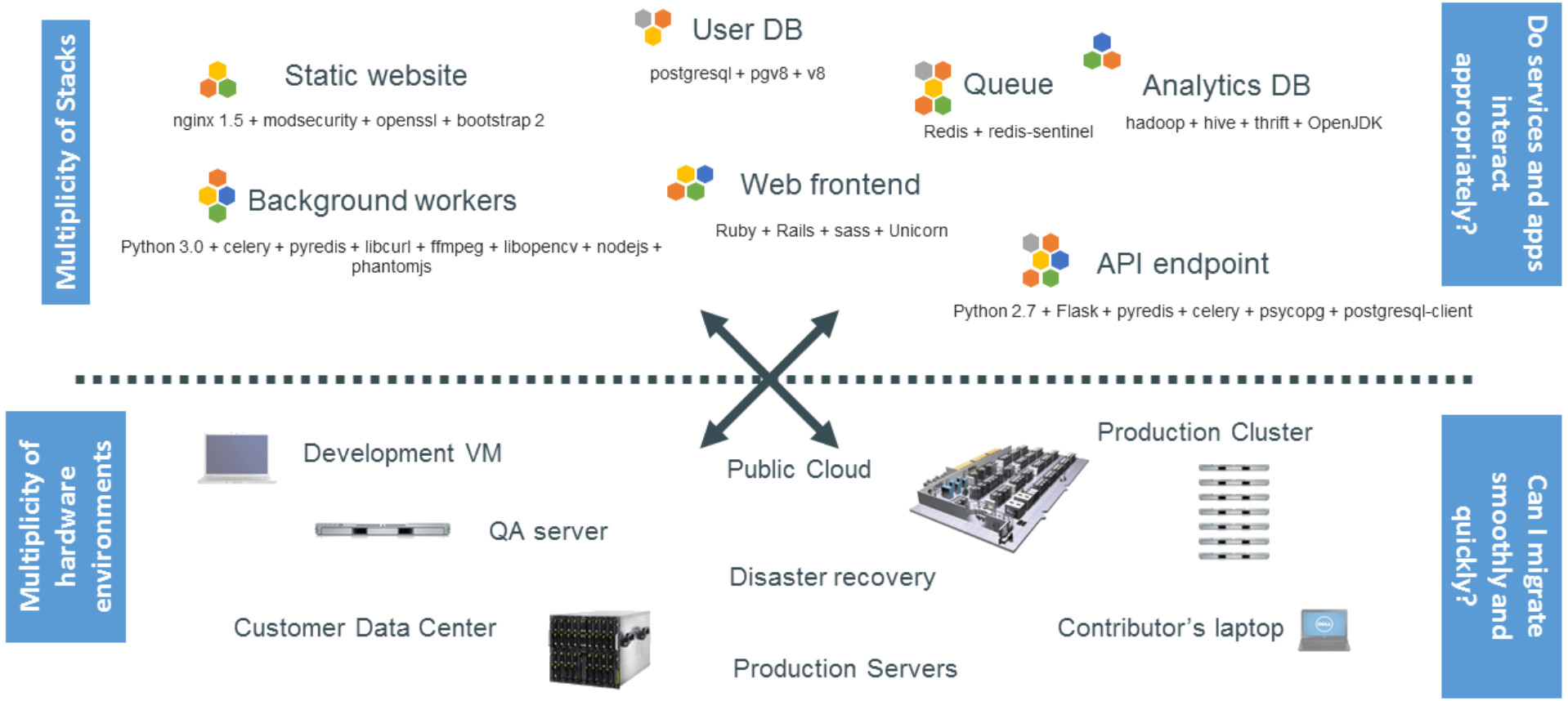
NTU

Features….

- Light-Weight
- Minimal overhead (cpu/io/network)
- Based on Linux containers
- Uses layered filesystem to save space (AUFS/LVM)
- Uses a copy-on-write filesystem to track changes
- Portable
- Can run on any Linux system, Windows or MacOS.
- Edge devices support.
- A Docker container contains everything it needs to run
- Minimal Base OS
- Libraries and frameworks
- Application code
- A docker container can run anywhere that Docker can run.

NTU

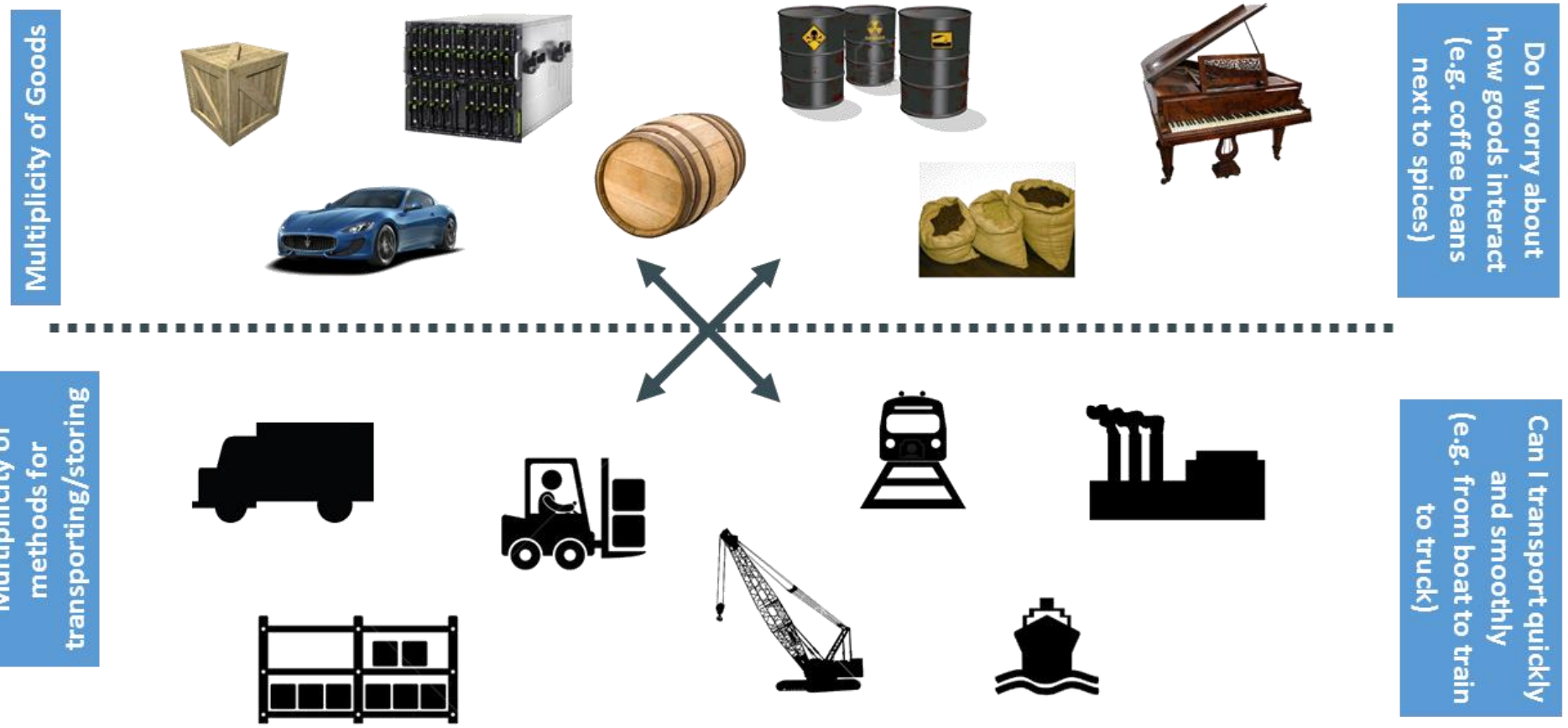
The Challenge……



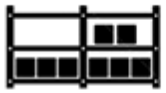
What and Where?

	Development VM	QA Server	Single Prod Server	Onsite Cluster	Public Cloud	Contributor's laptop	Customer Servers
Static website	?	?	?	?	?	?	?
Web frontend	?	?	?	?	?	?	?
Background workers	?	?	?	?	?	?	?
User DB	?	?	?	?	?	?	?
Analytics DB	?	?	?	?	?	?	?
Queue	?	?	?	?	?	?	?

Cargo Transport Pre-1960……

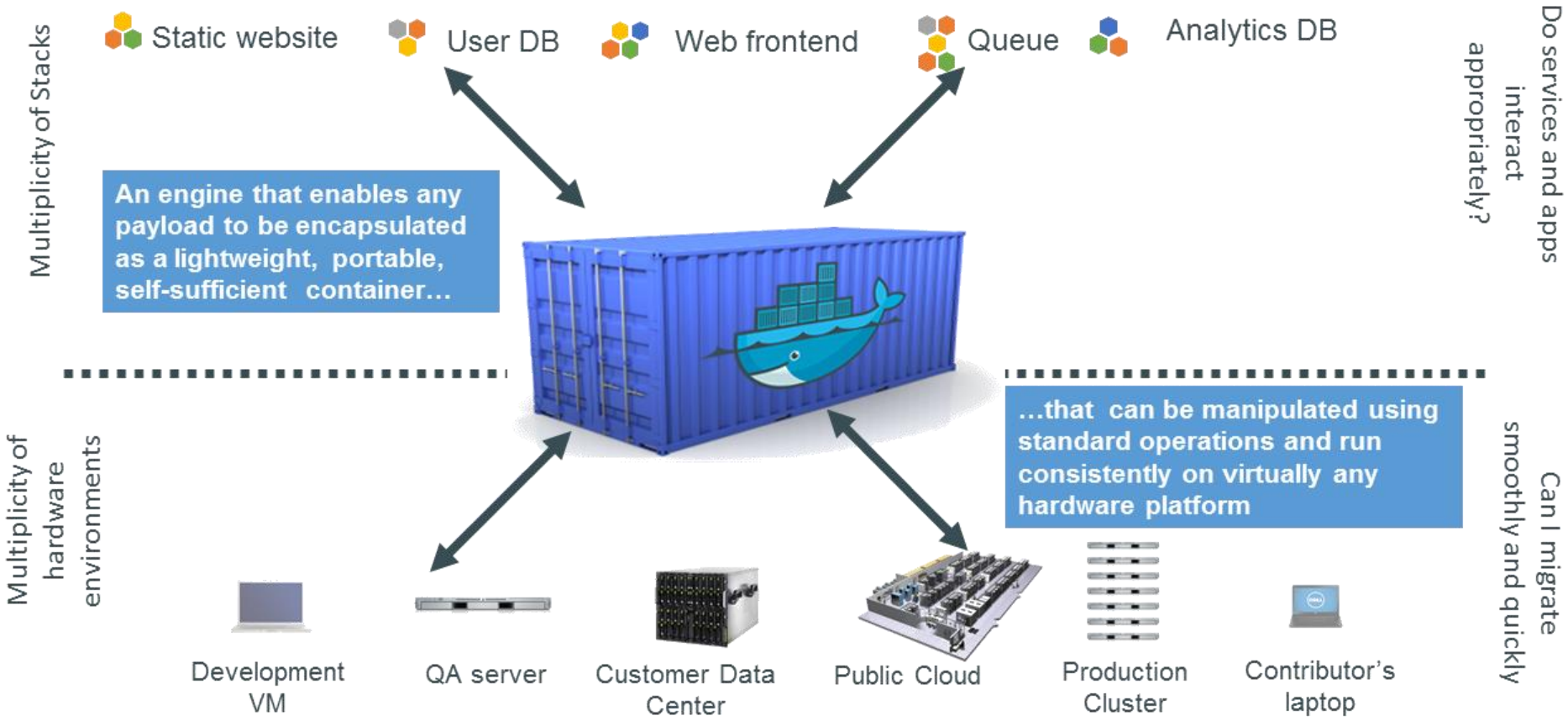


What and where?

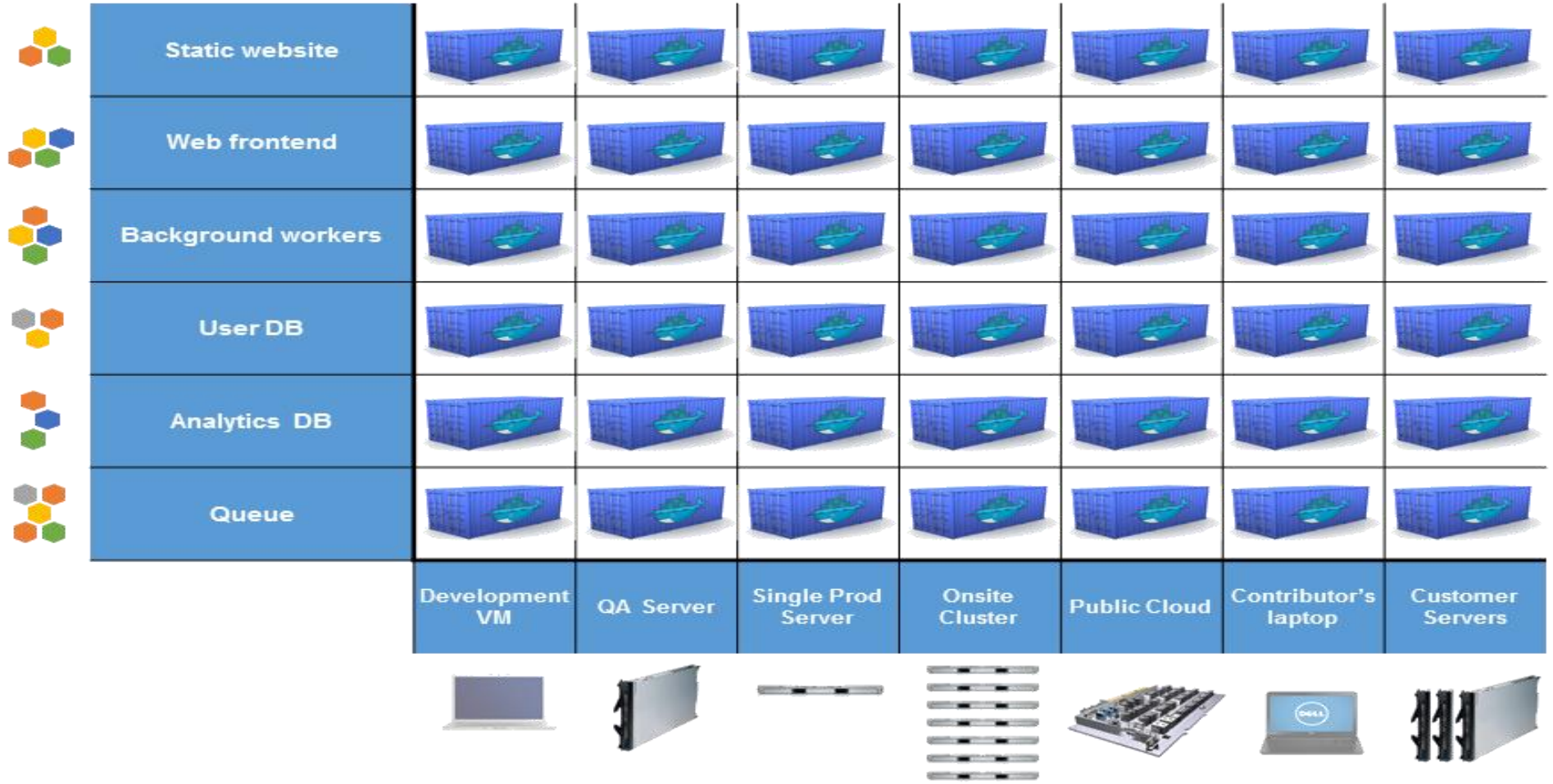
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
							

Solution: Intermodal Shipping Container

Docker is a Container System for Code……

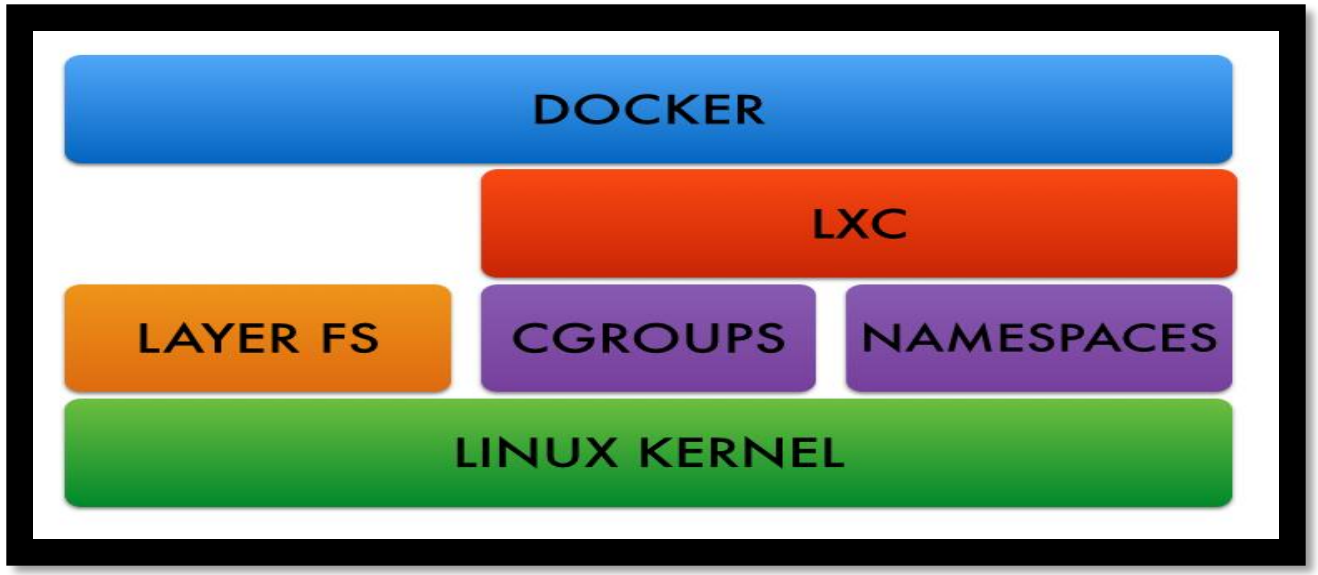


Docker provides the answer to what and where



Docker Architecture

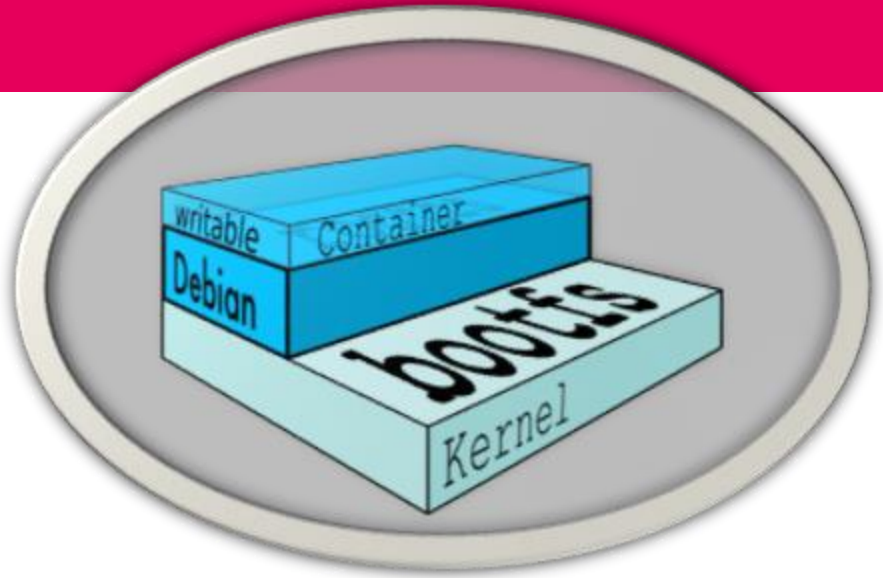
- Docker Engine
  - CLI
  - Docker Daemon
  - Docker Registry
- Docker Hub
  - Cloud service
    - Share Applications
    - Automate workflows
    - Assemble apps from components
- Docker images
- Docker containers

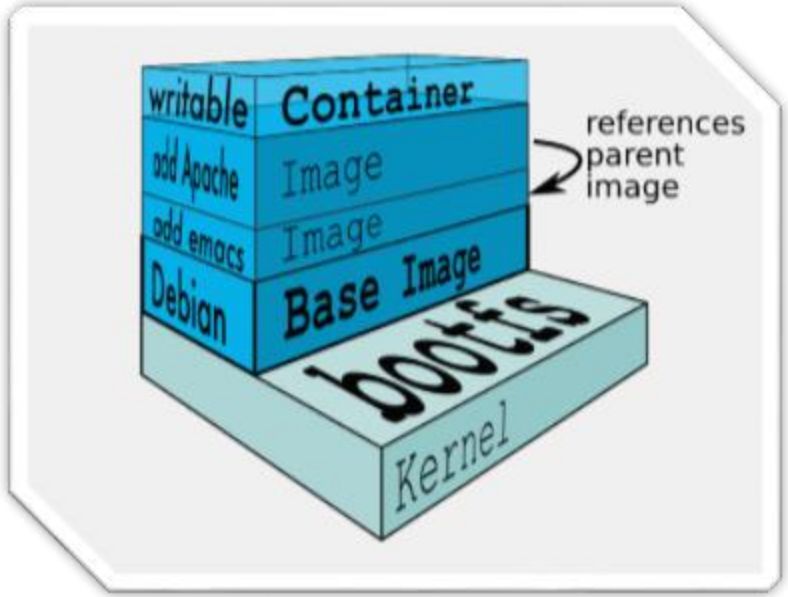


NTU

Docker images

- NOT A VHD
- NOT A FILESYSTEM
- uses a *Union File System*
- a read-only *Layer*
- do not have state
- Basically a tar file
- Has a hierarchy
    - Arbitrary depth
- Fits into the Docker Registry





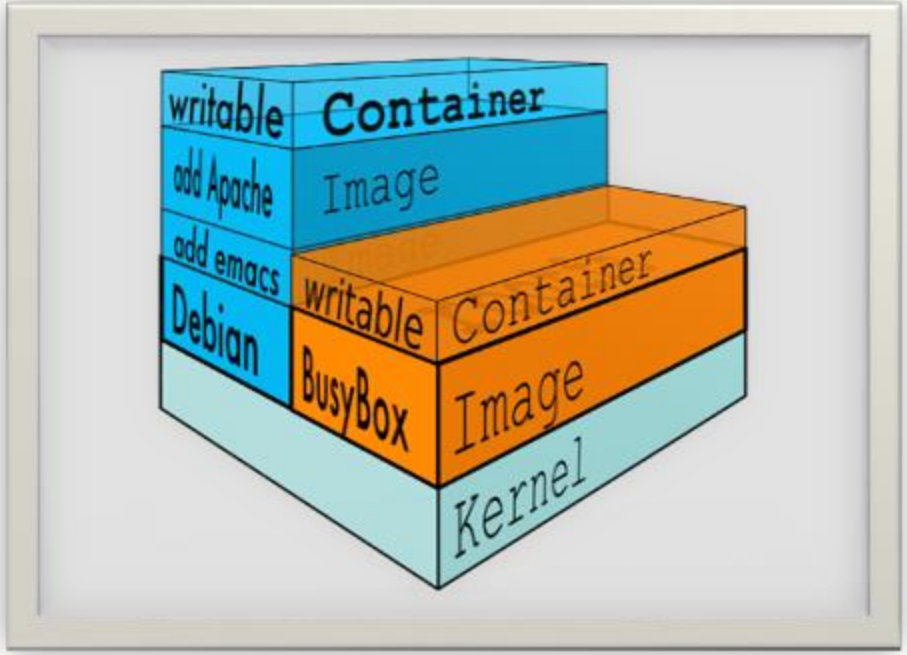
Docker Containers...

Units of software delivery (ship it!)

- run everywhere
  - regardless of kernel version
  - regardless of host distro
  - (but container and host architecture must match*)

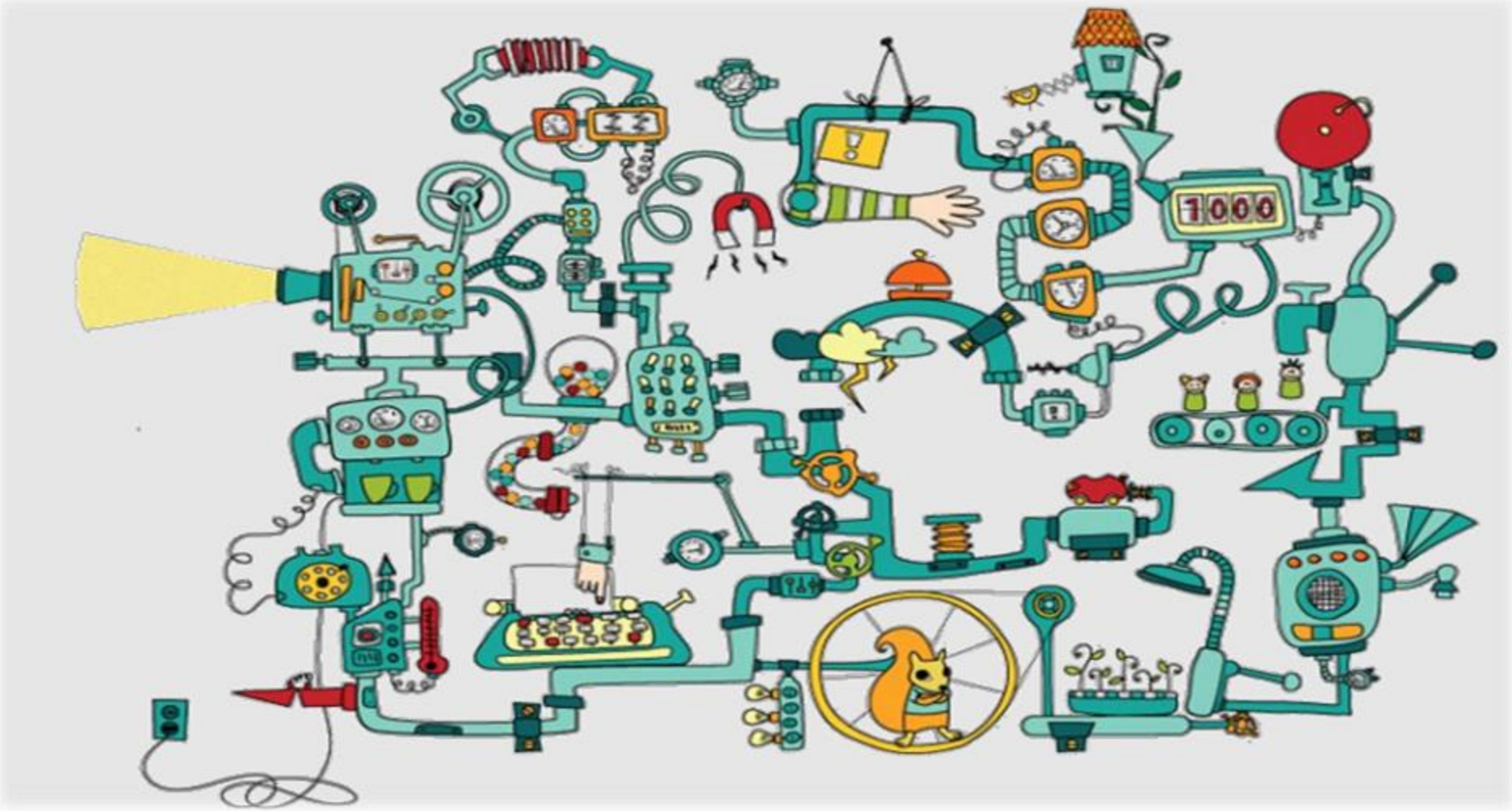
- run anything
  - if it can run on the host, it can run in the container
  - i.e., if it can run on a Linux kernel, it can run

*Unless you emulate CPU with qemu and binfmt



NTU

Containers before Docker



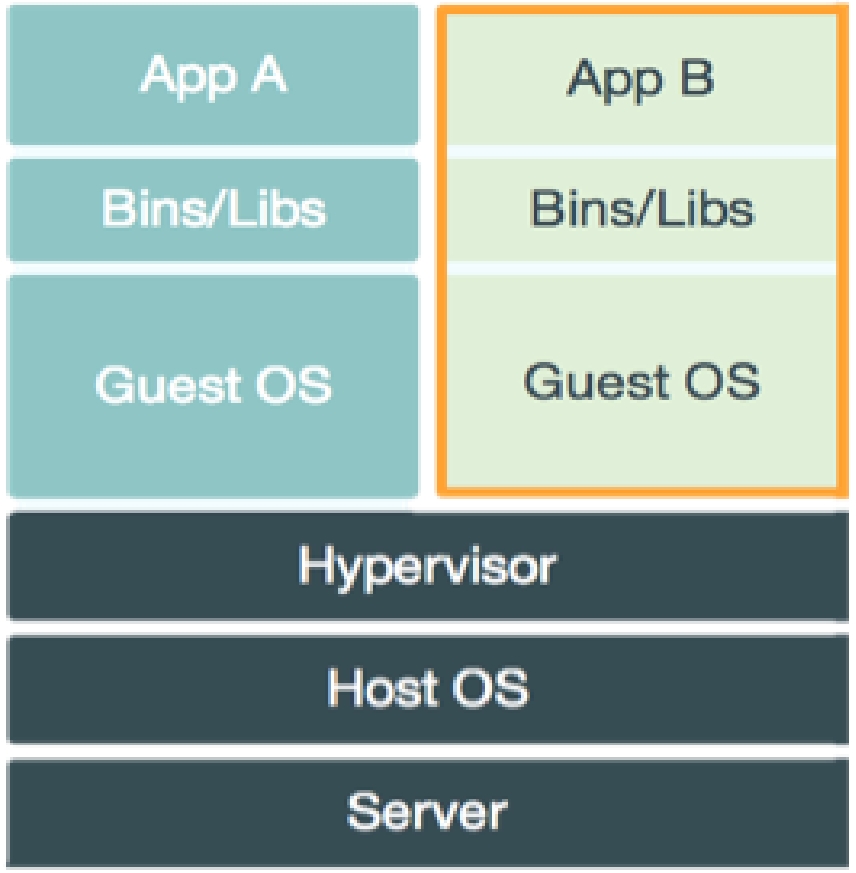

Containers after Docker

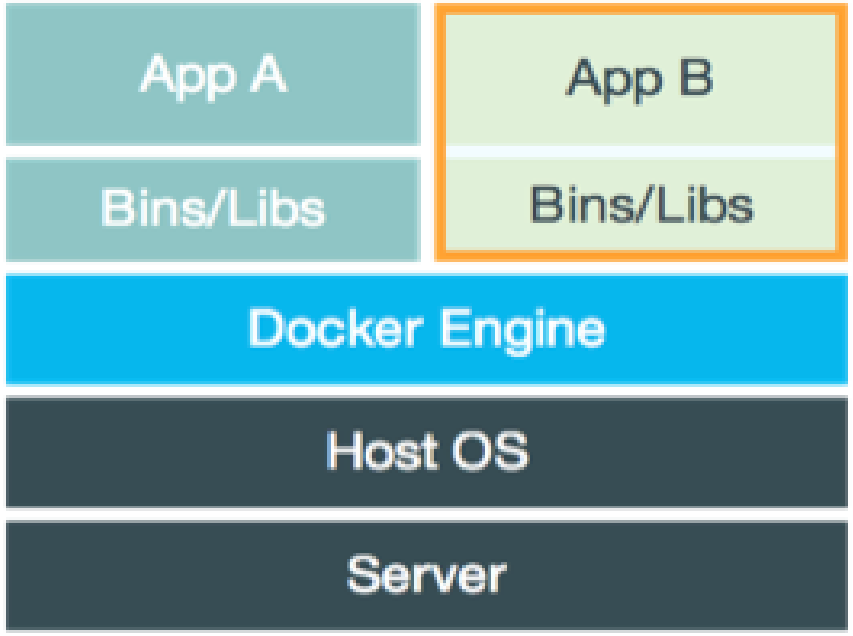
How does Docker work ?

- You can build Docker images that hold your applications
- You can create Docker containers from those Docker images to run your applications.
- You can share those Docker images via Docker Hub or your own registry

https://hub.docker.com/repository/docker/pedrombmachado/ntu_lubuntu

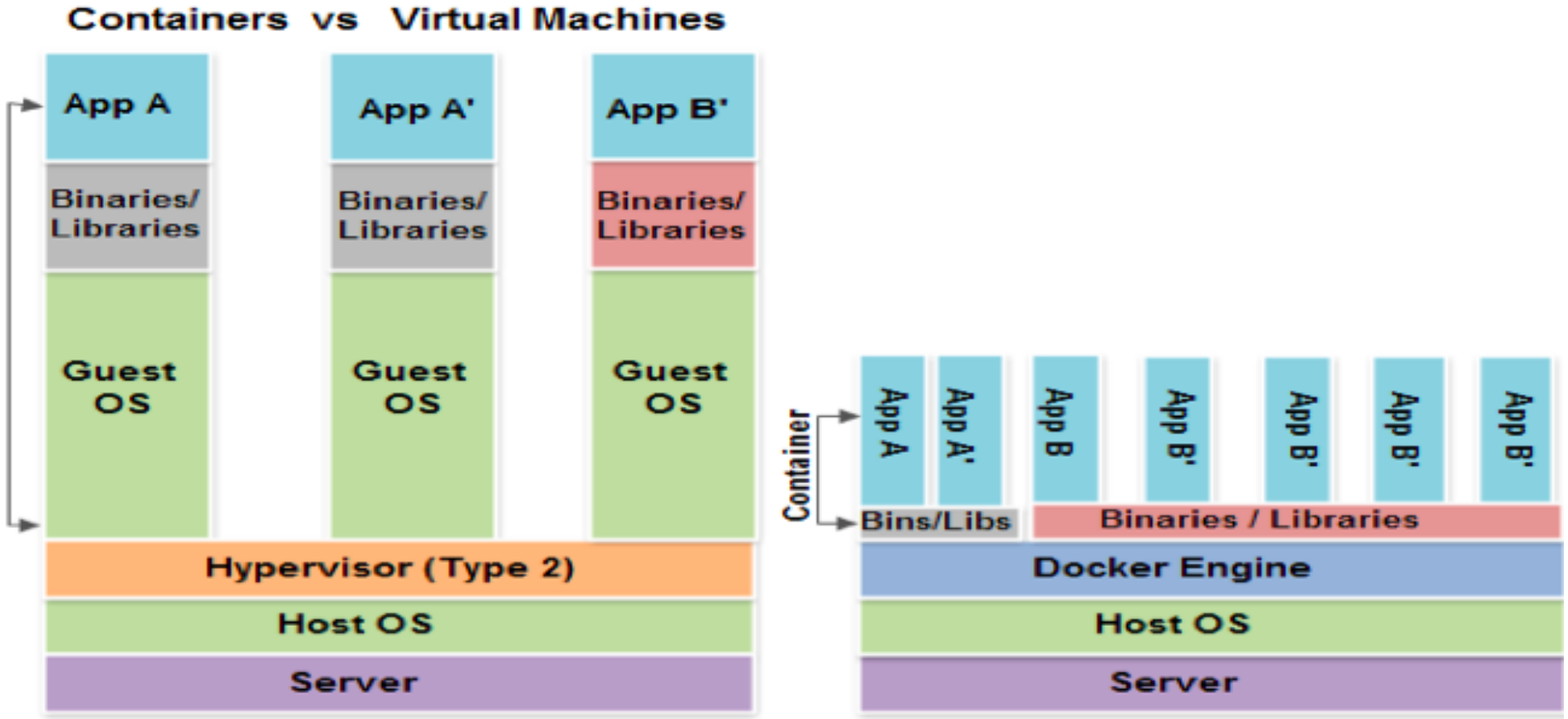
Virtual Machine Versus Container……





NTU

Virtual Machine Versus Container



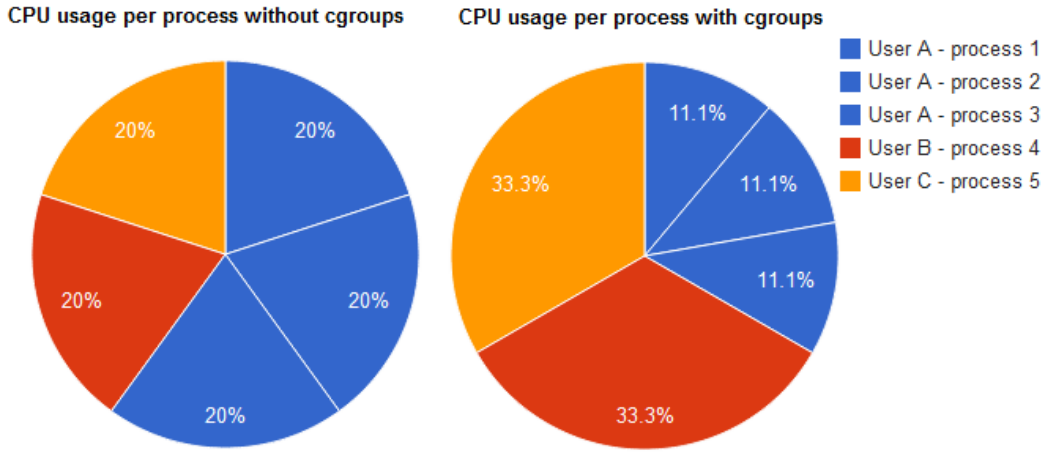
Docker Container Lifecycle

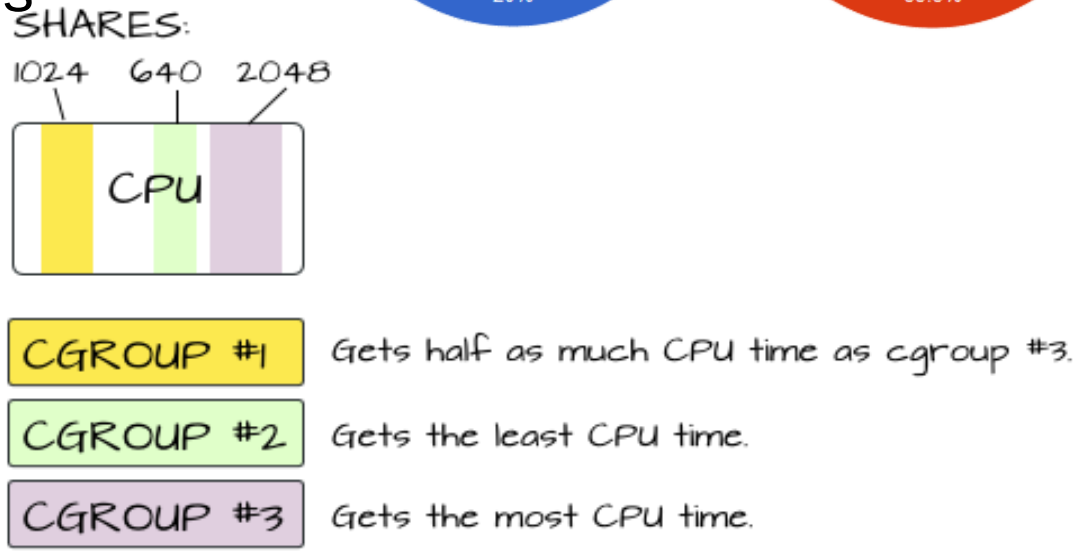
- Conception
  - **BUILD** an Image from a Dockerfile
- Birth
  - **RUN** (create+start) a container
- Reproduction
  - **COMMIT** (persist) a container to a new image
  - **RUN** a new container from an image
- Sleep
  - **KILL** a running container
- Wake
  - **START** a stopped container
- Death
  - **RM** (delete) a stopped container

- Extinction
  - **RMI** a container image (delete image)

Linux Cgroups

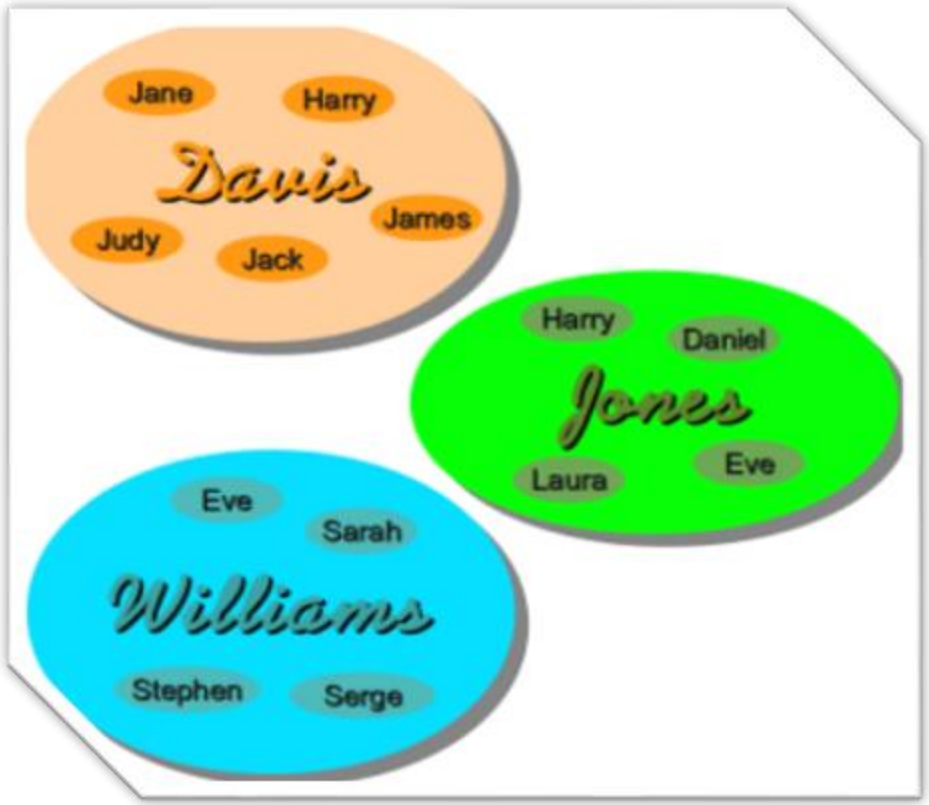
- Kernel Feature
- Groups of processes
- Control resource allocations
  - CPU
  - Memory
  - Disk
  - I/O
- May be nested





Linux Kernel Namespaces

- Kernel Feature
- Restrict your view of the system
  - Mounts (CLONE_NEWNS)
  - UTS (CLONE_NEWUTS)
    - uname() output
  - IPC (CLONE_NEWIPC)
  - PID (CLONE_NEWPID)
  - Networks (CLONE_NEWNET)
  - User (CLONE_NEWUSER)
    - Not supported in Docker yet
    - Has privileged/unprivileged modes today
- May be nested



Dockerfile

- Like a Makefile (shell script with keywords)
- Extends from a Base Image
- Results in a new Docker Image
- Imperative, not Declarative
- A Docker file lists the steps needed to build an images
- docker build is used to run a Docker file
- Can define default command for docker run, ports to expose, etc

```
# syntax=docker/dockerfile:1
FROM ubuntu:18.04
COPY . /app
RUN make /app
CMD python /app/app.py
```

NTU

```
exec        Run a command in a running container
export      Export a container's filesystem as a tar archive
history     Show the history of an image
images      List images
import      Import the contents from a tarball to create a filesystem image
info        Display system-wide information
inspect     Return low-level information on Docker objects
kill        Kill one or more running containers
load        Load an image from a tar archive or STDIN
login       Log in to a Docker registry
logout      Log out from a Docker registry
logs        Fetch the logs of a container
pause       Pause all processes within one or more containers
port        List port mappings or a specific mapping for the container
ps          List containers
pull        Pull an image or a repository from a registry
push        Push an image or a repository to a registry
rename      Rename a container
restart     Restart one or more containers
rm          Remove one or more containers
rmi         Remove one or more images
run         Run a command in a new container
save        Save one or more images to a tar archive (streamed to STDOUT by default)
search      Search the Docker Hub for images
start       Start one or more stopped containers
stats       Display a live stream of container(s) resource usage statistics
stop        Stop one or more running containers
tag         Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE
top         Display the running processes of a container
unpause     Unpause all processes within one or more containers
```

Docker CLI Commands

Docker in Higher Education

- How do we train our students?
- How do we reduce installation and configuration times?
- How do we offer the same Dev environment to all our students?

Run the docker container

Only for personal laptops: ensure that the steps described in Install Docker Desktop on Windows machines have been completed successfully.

1. Start docker desktop
2. Start PowerShell (Windows) or Terminal (Linux/Mac OS) and run the following commands and **DO NOT copy the $ sign**:
3. Load the container (**ONLY FOR LAB PCs**). **DO NOT copy the $ sign**:

```
$ docker load --input 'C:\Users\Public\Documents\Shared Virtual Machines\Docker
\comp20081.docker'

$ docker create volume docker_comp20081

$ docker run -it --rm -p "3390:3389/tcp" --name="ntu-vm-scomp20081" -v docker_comp20081:/home
/ntu-user/NetBeansProjects pedrombmachado/ntu_lubuntu:comp20081
```

Docker demo

Get docker desktop from https://docs.docker.com/get-docker/

Instructions

On AMD64/Intel64 (your laptop). DO NOT copy the $ sign:

```
$ docker volume create docker_comp20081
$ docker run -it --rm -p "3390:3389/tcp" --name="ntu-vm-comp20081" -v
docker_comp20081:/home/ntu-user/NetBeansProjects
pedrombmachado/ntu_lubuntu:comp20081
```

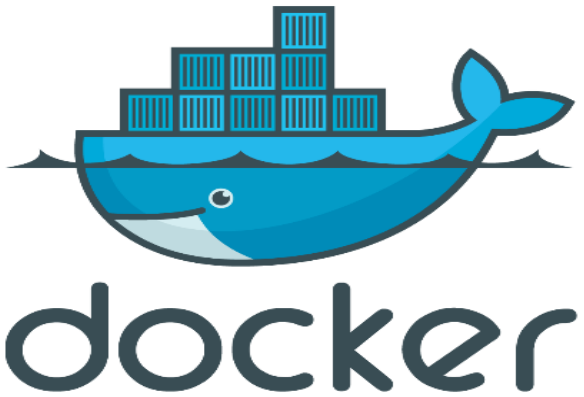
On ARM64 architecture (Mac M1/M2, Chrome book, etc.). DO NOT copy the $ sign:

```
$ docker volume create docker_soft40051
$ docker run -it --rm -p "3390:3389/tcp" --name="ntu-vm-comp20081" -v
docker_comp20081:/home/ntu-user/NetBeansProjects
pedrombmachado/ntu_lubuntu:arm64v8_comp20081
```

Docker Hub repo

Docker demo

Conclusions



- Easy to build, run & share containers
- Rapidly expanding ecosystem
- Better performance vs. VMs
- Layered file system gives us git-like control of images
- Reduces complexity of system builds
- Can be used in higher education to train students and abstract students from installing complex packages.

https://www.docker.com/

NTU

Nottingham Trent University

Department of Computer Science

Department of Computer Science

Use of docker for teaching Computer Sciences subjects in HE

Pedro Machado – Senior Lecturer in Computer Sciences @ NTU
Pedro.machado@ntu.ac.uk