



Containers

Data-Centers (Centres de Processament de Dades)

Josep Lluís Berral-García
berral@ac.upc.edu



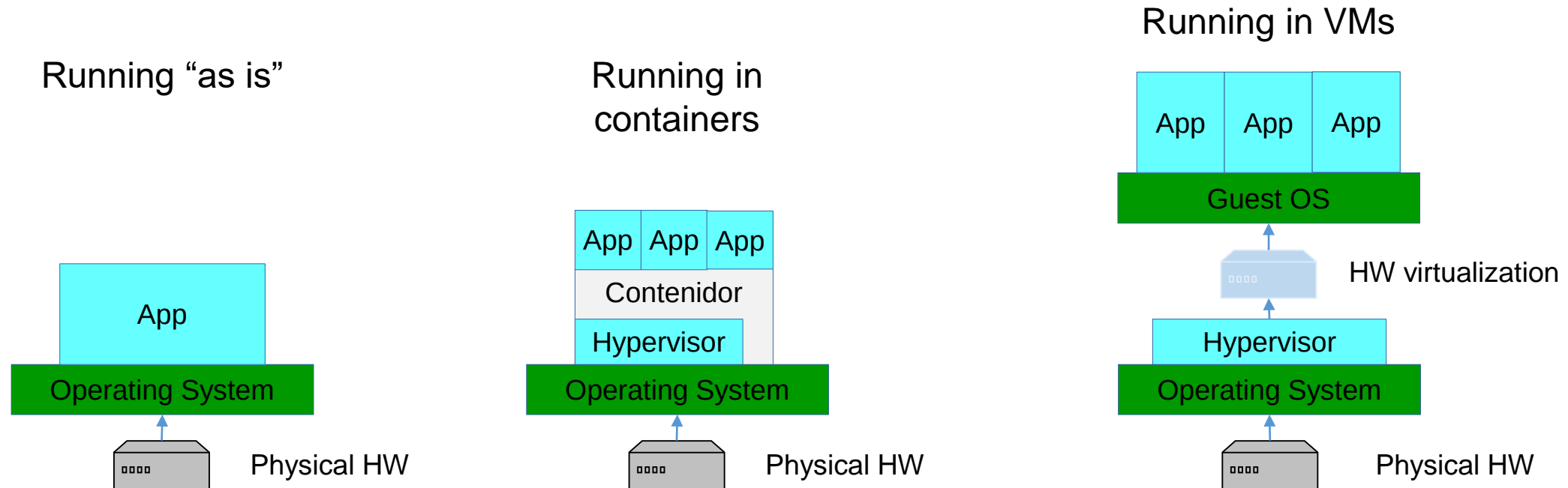
Introduction

“Independent **services** better be **encapsulated**

... so we can replicate and scale better, also deploy faster”

Isolation and Environments

- Isolation of Services
 - In between running “as is” and full system virtualization





Containers Life-Cycle

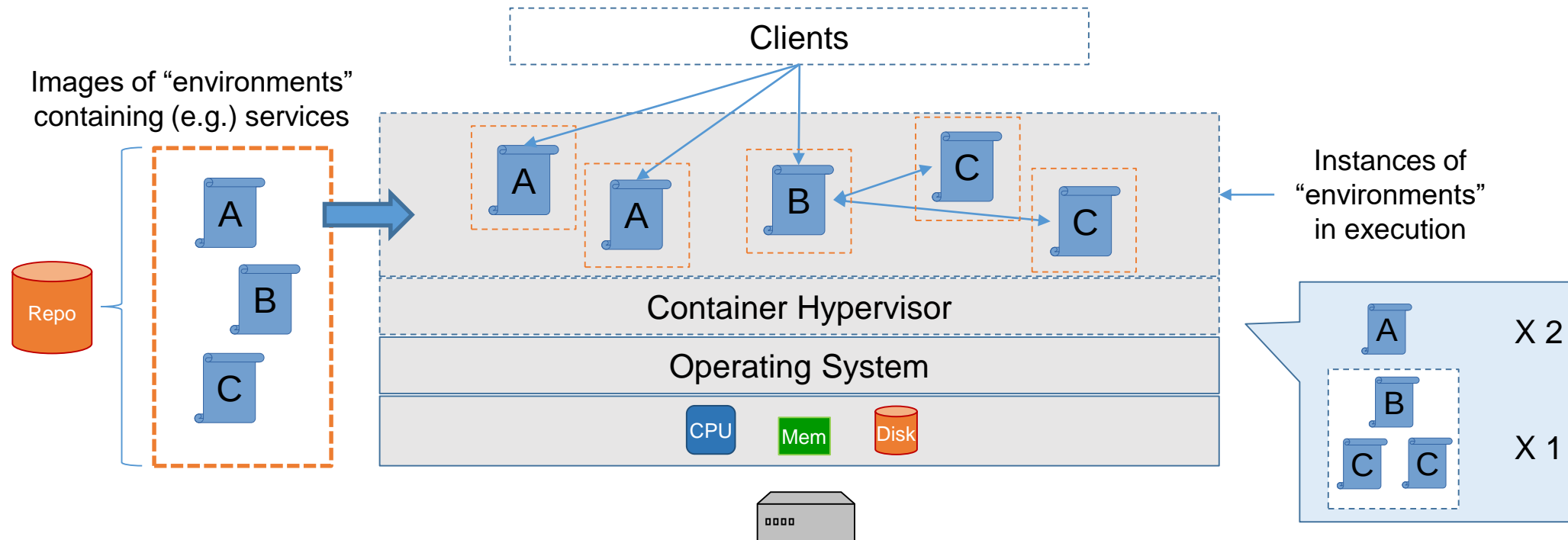
Containers ~ isolation of environments as a process, oriented to
“create-use-destroy” scenarios

Containers Life-Cycle

- VM (usually)
 - An image and disk volume exist per VM
 - Modifications are persistent: changes remain
- Containers
 - An “image” is to be deployed as many times as wanted
 - Each deployment (“instantiation”) → a “container”
 - When a container stops, it is not re-used

Container Architecture

- Encapsulation of Services



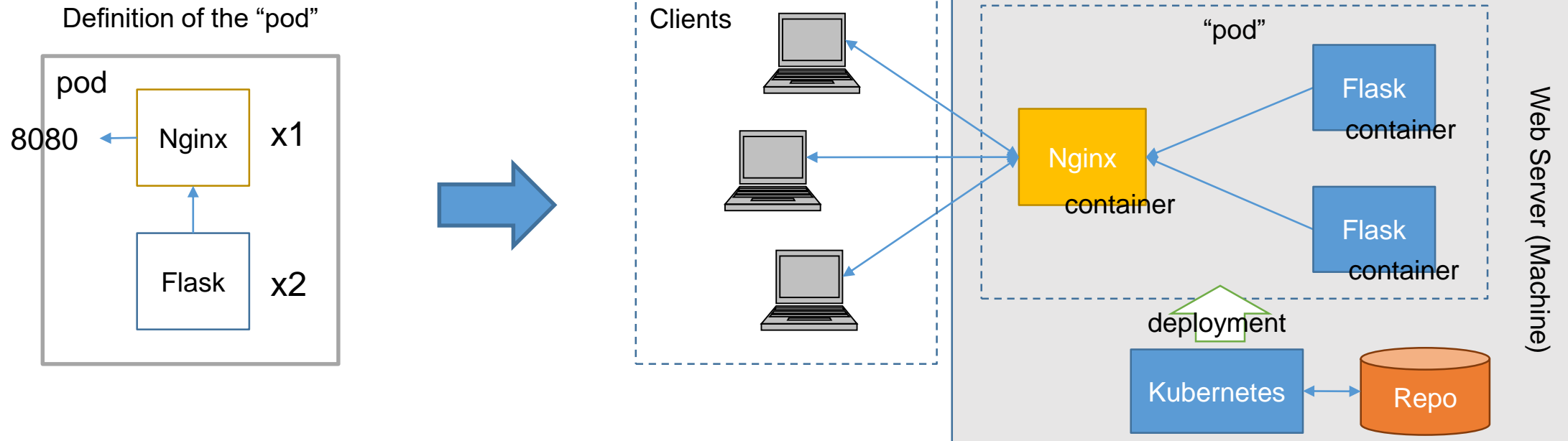
- Allow to deploy “instances” of services, quickly, orchestrated and in isolation

Custom Containers

- The “Receipe”:
 - Instructions to “build” an image
 - Contains the instructions to download an “image base”
 - ... plus the instructions to install additional programs
 - ... pus copy files from our system inside the container
 - ... plus do any other modification
 - The result is an image ready to be instantiated
 - (deployed as a/many container/s)
 - E.g., in Docker → the “dockerfile”

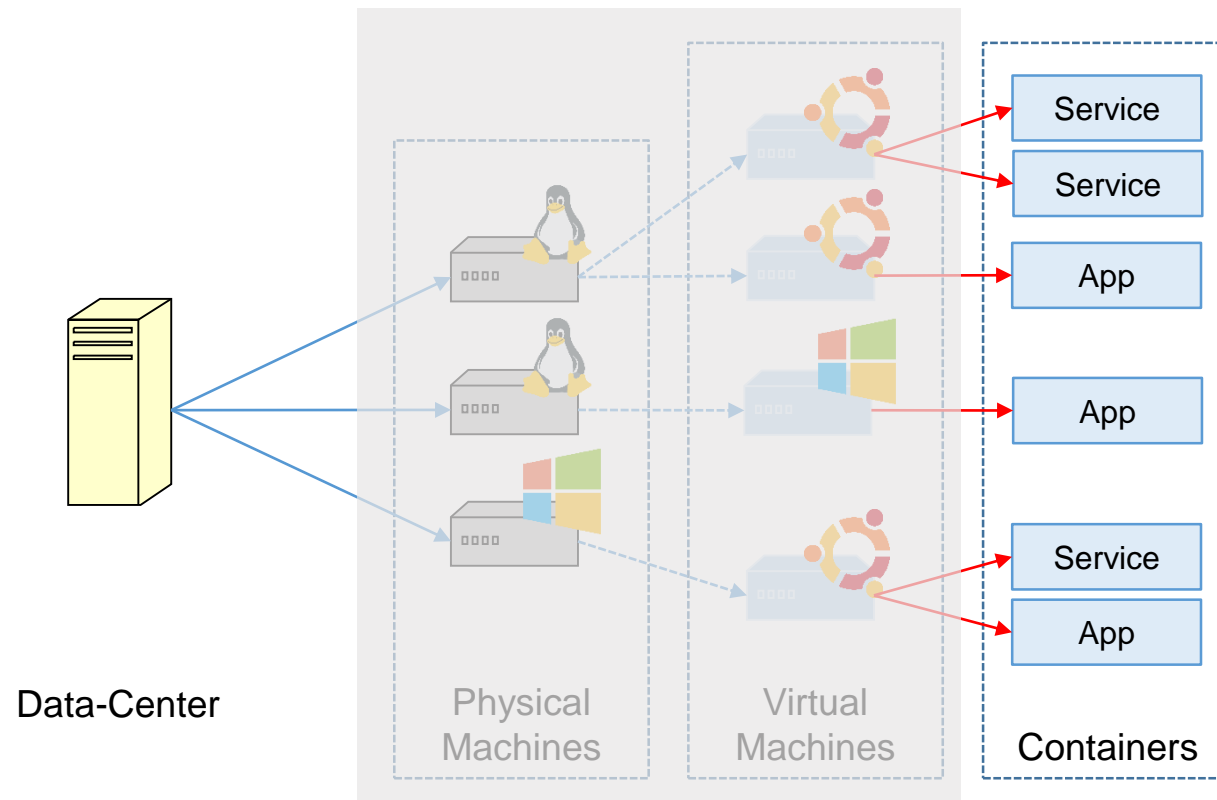
“Pods” and reunion of Services

- Grouping services in “Pods”
 - Kubernetes
 - Define architectures of containerized services (“pods”)
 - Indicate how they connect and how they are exposed
 - Deploy ALL in one shot
 - Allocate & manage available resources for the pods/containers



Virtualization in Resource Providers

- Resource providers now offer containers as a service
 - (inside VMs, to still keep control of users and enforce security)



Sessió Pràctica

- Entorn:
 - Docker → Hypervisor (gestiona i executa els contenidors)
 - Repo de Docker → Repositori públic de contenidors ja configurats
- Sistema Operatiu:
 - VM amb Ubuntu 18.04
 - Ho executarem tot dins una VM, per tenir tots el mateix entorn
 - A la pràctica tindrem 2 nivells d'imbricació (màquina → VM → contenidor → App)
- Qüestionari:
 - Durant la pràctica cal resoldre preguntes respecte el que estem executant i observant

Sessió Pràctica

- Part zero:
 - Preparar una VM amb VBox per tenir un entorn base llest
- Primera part:
 - Instal·lar Docker per poder executar contenidors
 - Provar a descarregar una imatge genèrica i fer-la funcionar
- Segona part:
 - Preparar una imatge “custom” que contingui el servidor d’aplicacions web Flask
 - Instanciar la imatge diverses vegades i connectar els contenidors
- Tercera part:
 - Preparar una imatge “custom” que contingui el servidor web/balancejador de càrrega Nginx
 - Configurar-ho per a que balancegi entre diverses instàncies de Flask
 - Instanciar una imatge d’Nginx per a que balancegi càrrega entre els containers Flask

Sessió Pràctica

- Objectius de la Sessió Pràctica:
 1. Instal·lar l'entorn de Contenidors
 - L'Hypervisor (e.g. Docker)
 - Veure els repositoris d'imatges de Contenidors
 2. Desplegar contenidors
 - Usant imatges base, i modificant-les
 - Afegir serveis web com Flask i Nginx
 - Connectar els Contenidors
 3. Balanceig de Càrrega
 - Desplegar tots els serveis alhora
 - Veure com si fem peticions al balancejador, aquest reparteix la càrrega