

PRÁCTICA 2: DOCKER

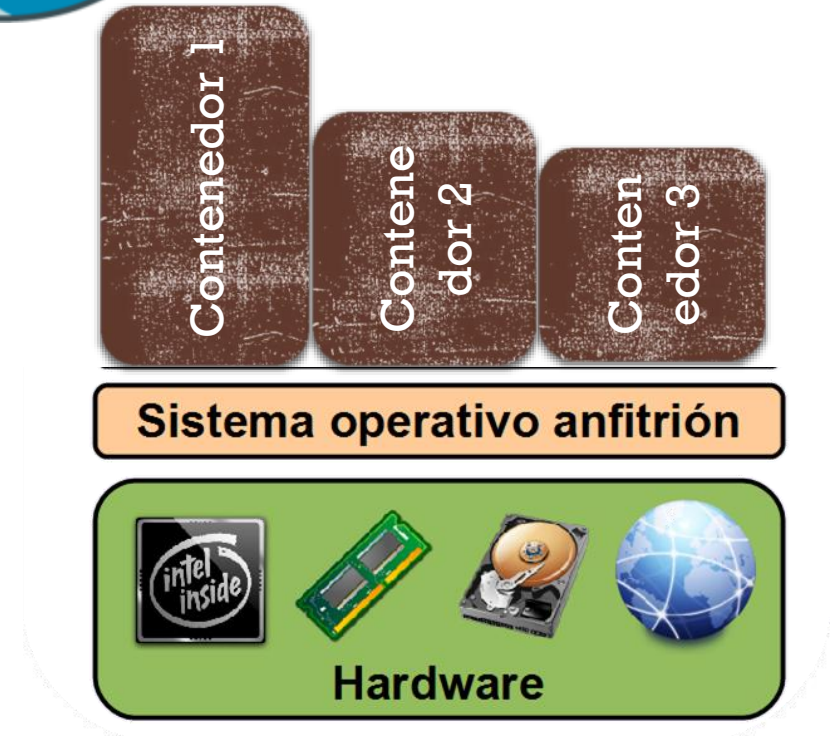
Daniel Cascado Caballero, M^a José Morón Fernández

1

INTRODUCCIÓN



- Docker es la plataforma de contenedores más popular.
- Originalmente construido sobre LXC (*Linux Container*)
- Docker NO es una tecnología de virtualización:
 - Los contenedores Docker interactúan directamente con el kernel de Linux del Host.
 - Permite desplegar, replicar, migrar un servicio más rápida y sencillamente que con MV
 - Facilita la coordinación básica de múltiples contenedores



DOCKER



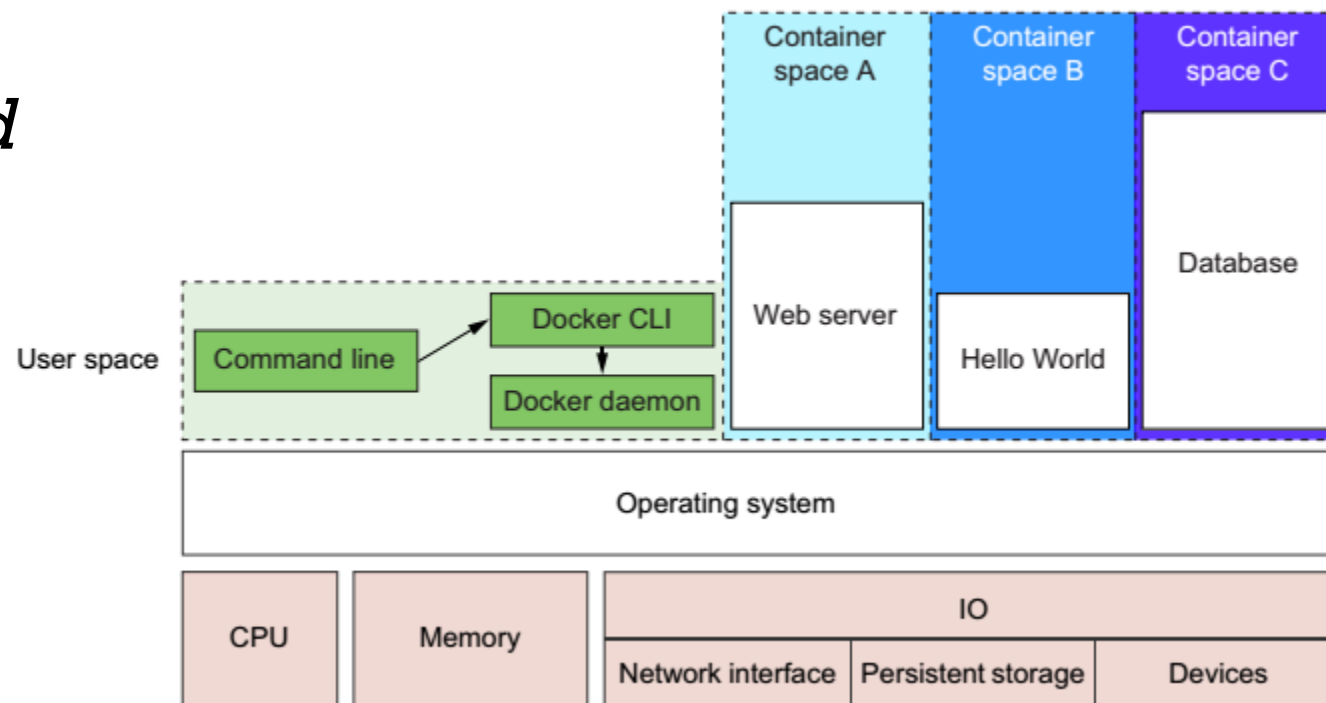
➤ Componentes de Docker:

- En el espacio de Usuario:

- ❑ Interfaz de línea de comandos (*Command Line Interface*, Docker CLI), con el que interactúa el usuario para arrancar, detener o instalar SW

- ❑ *daemon en background* (*Background Daemon*)

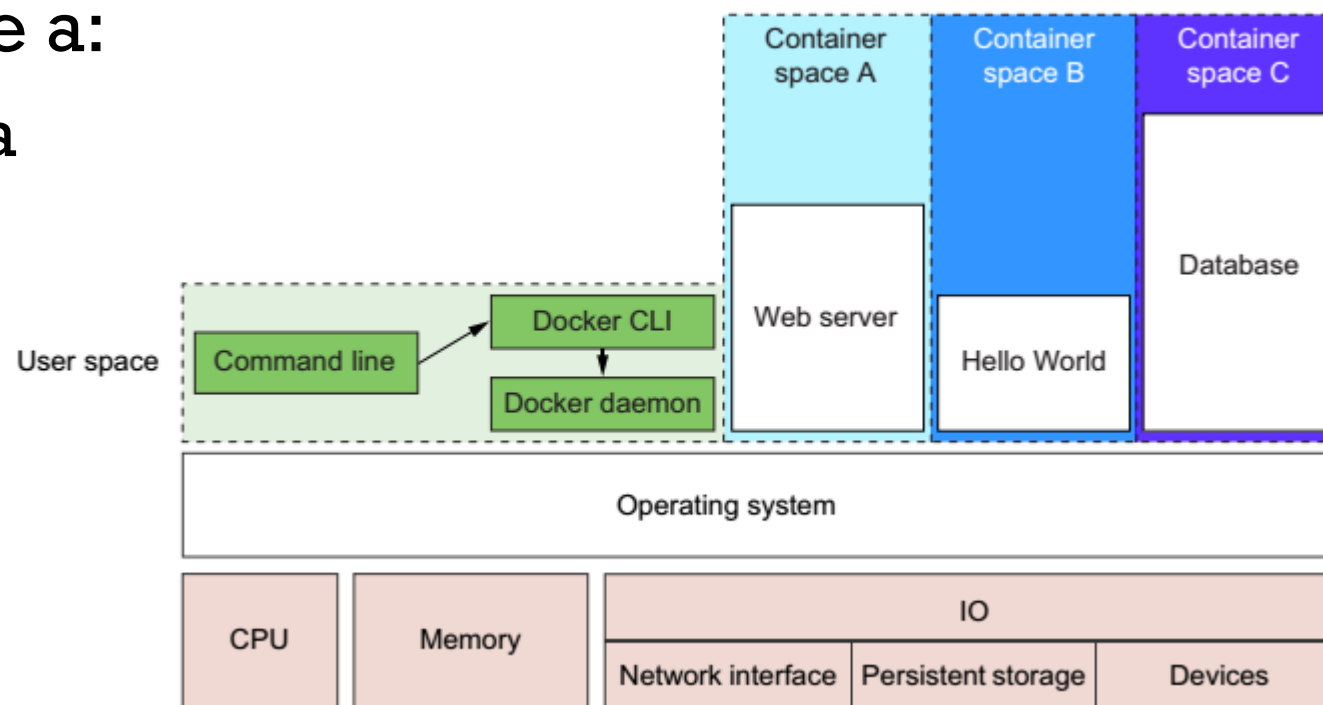
- Un conjunto de servicios remotos



DOCKER



- Cada contenedor se ejecuta como un proceso hijo (*child process*) del *Docker Daemon*, en su propio subespacio de memoria del espacio de Usuario.
- Los programas en ejecución dentro de un contenedor pueden acceder únicamente a:
 - su propia memoria
 - los recursos del ámbito del contenedor



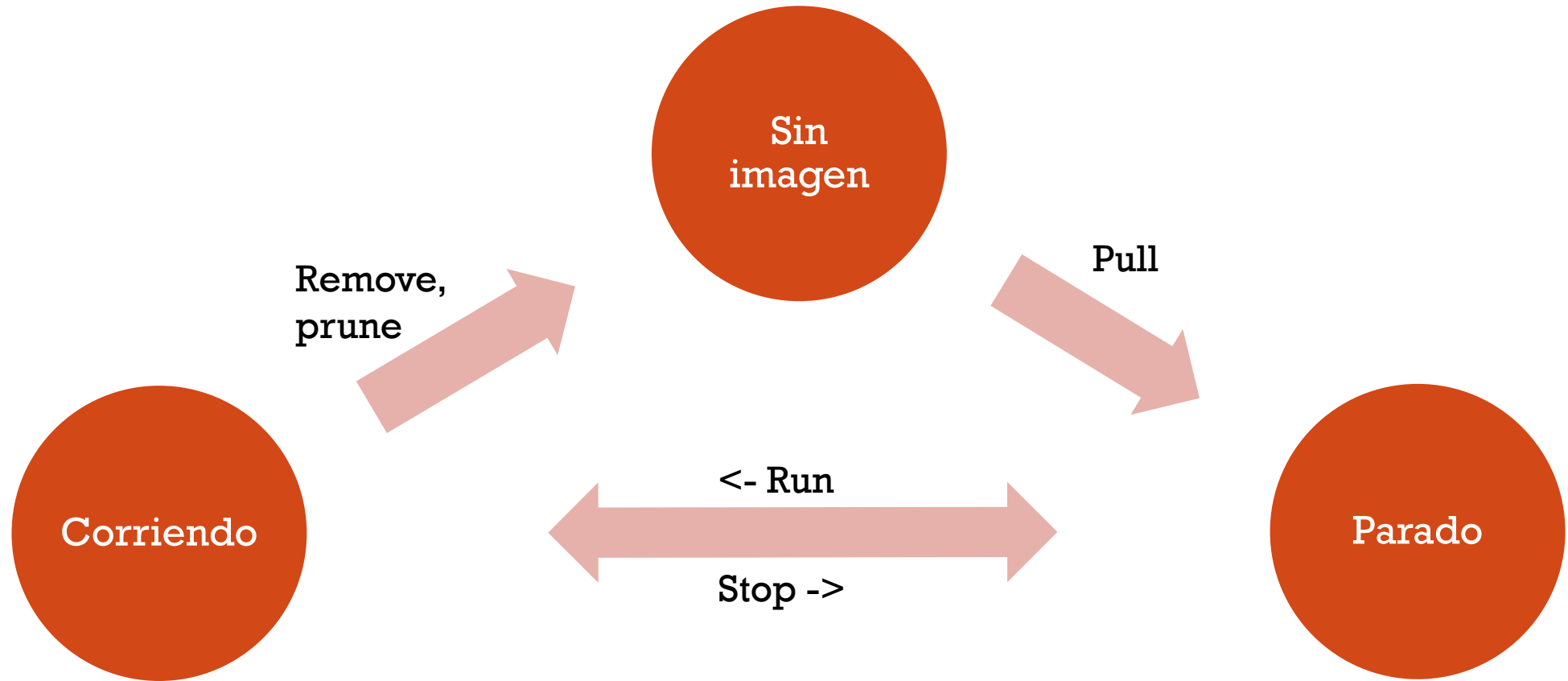
CONTENEDORES DOCKER



Los contenedores que Docker genera proporciona aislamiento en 8 aspectos:

1. Identificadores de Proceso (*PID namespace*)
2. Host y nombre de dominio (*UTS namespace*)
3. Acceso y estructura del Sistema de Fichero (*MNT namespace*)
4. Comunicación de procesos sobre memoria compartida (*IPC namespace*)
5. Acceso y estructura de red (*NET namespace*)
6. Nombres de usuario e identificadores (*USR namespace*)
7. Control de la ubicación del raíz del sistema de archivos (*chroot()*)
8. Protección de recursos (*cgroups*)

CICLO DE VIDA DE UN CONTENEDOR



REDES Y ALMACENAMIENTO

- Se pueden crear subredes para conectar contenedores entre si
- DNS propio. Cada contenedor se puede referir a otro:
 - por su nombre
 - Por su dirección IP: se puede saber inspeccionando las propiedades de un contenedor
- Los volúmenes de almacenamiento permiten salvaguardar los datos de un contenedor después de que el contenedor se haya parado.
 - Residen en una carpeta protegida del SSOO que los aloja

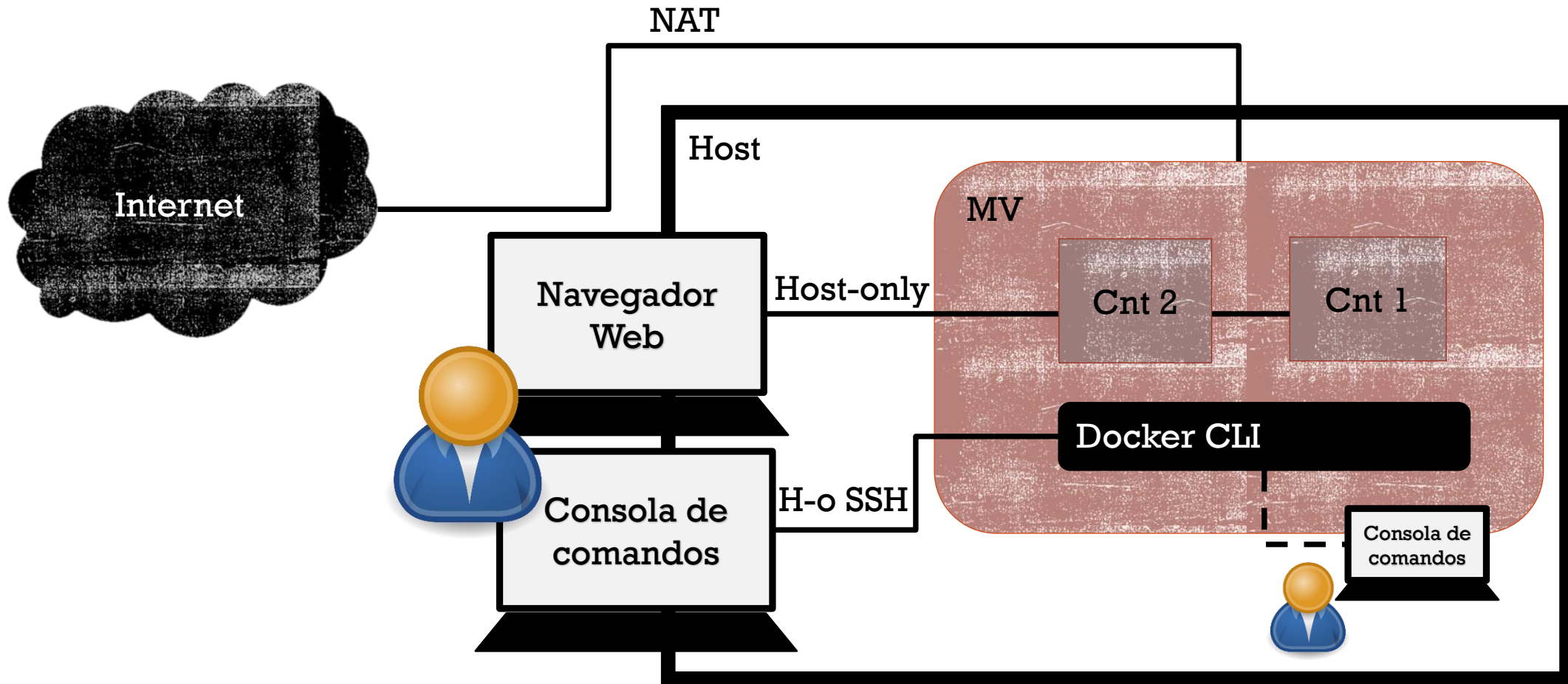
ORQUESTACIÓN BÁSICA: SWARM

- Es una herramienta de clusterización y organización de contenedores
- Permite hacer clusters de contenedores en réplica de alta disponibilidad y escalado de servicios con cierta facilidad
- Fichero de definición del cluster: `docker-compose.yml`
 - Imágenes
 - Número de réplicas
 - Puertos
 - Redes
 - Recursos de CPU, memoria...
- Se pueden tener varios clusters a la vez, mediante varios nodos trabajadores, cada uno con su cluster

ESCENARIO PRÁCTICO (I)

- Se dispone de una máquina virtual Linux con Docker YA instalado.
- Host en Ubuntu si no se dispone de acceso admin.
 - Sólo para los PCs de prácticas.
- Se han dispuesto dos interfaces de red:
 - NAT: para comunicación con el exterior (internet y servicios remotos)
 - Host-Only:
 - Acceso por consola de comandos remota SSH.
 - Comando: `ssh usuario@dirección_ip`
 - Acceso a los servicios de los contenedores
 - Acceso del navegador web del host a los servicios de la máquina virtual
 - Quitar el proxy de la configuración del navegador

ESCENARIO PRÁCTICO (II)



TAREAS PARA EL HITO

- Hacer funcionar un contenedor con Apache Web Server
- Generar un fichero index.html
- Copiarlo al contenedor, en la carpeta raíz de documentos web:
 - /usr/local/apache2/htdocs/ ...
 - o algo así, dependiendo del contenedor escogido
- Visualizar por un cliente web, como CURL, desde la máquina virtual.
 - curl http://<ip_del_contenedor>