

Programación Distribuida y Tiempo Real

Virtualización

Contenidos Conceptuales

- Contenidos de los videos 02/03/04
 - Recuerden que pueden acelerar vel.



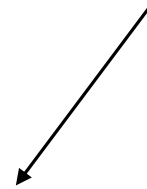
Contenidos Conceptuales

- Contenidos de los videos 02/03/04
 - Recuerden que pueden acelerar vel.
- Contenidos de Redes en TP1 y TP2
 - Relacionar con PDyTR



Ambientes de Ejecución

- Ambientes Virtuales (AV) - Virtualización
 - Usuarios de Vbox/VMware/WSL (¿?)...



Preguntar a alguna IA...

<https://learn.microsoft.com/en-us/windows/wsl/faq>

Ambientes de Ejecución

- Ambientes Virtuales (AV) - Virtualización
 - Usuarios de Vbox/VMware/WSL (¿?)...
 - Gestión (automatización) de mv
 - “Empaquetados” (boxes)
 - Instalación y configuración ¡con código!
-

Ambientes de Ejecución

- Ambientes Virtuales (AV) - Virtualización
 - Usuarios de Vbox/VMware/WSL (¿?)...
 - Gestión (automatización) de mv
 - “Empaquetados” (boxes)
 - Instalación y configuración ¡con código!
 - Gestión (“*deploy*”) de una aplicación
 - Usual: contenedor
-

Ambientes de Ejecución

- Ambientes Virtuales (AV) - Virtualización
 - Usuarios de Vbox/VMware/WSL (¿?)...
 - Gestión (automatización) de mv
 - “Empaquetados” (boxes)
 - Instalación y configuración ¡con código!
 - Gestión (“*deploy*”) de una aplicación
 - Gestión de contenedores... “*orquestración*”... *quizás*
-

Ambientes de Ejecución

- Ambientes Virtuales (AV) - Virtualización
 - Usuarios de Vbox/VMware/WSL (¿?)...
 - Gestión (automatización) de mv
 - “Empaquetados” (boxes)
 - Instalación y configuración ¡con código!
 - Gestión (“*deploy*”) de una aplicación
 - Gestión de contenedores... “*orquestación*”... *quizás*



vagrant

Ambientes de Ejecución

- Ambientes Virtuales (AV) - Virtualización
 - Usuarios de Vbox/VMware/WSL (¿?)...
 - Gestión (automatización) de mv
 - “Empaquetados” (boxes)
 - Instalación y configuración ¡con código!
 - Gestión (“*deploy*”) de una aplicación
 - Gestión de contenedores... “*orquestración*”... *quizás*



vagrant

docker

Ambientes de Ejecución

- Ambientes Virtuales (AV) - Virtualización
 - Usuarios de Vbox/VMware/WSL (¿?)...
 - Gestión (automatización) de mv
 - “Empaquetados” (boxes)
 - Instalación y configuración ¡con código!
 - Gestión (“*deploy*”) de una aplicación
 - Gestión de contenedores... “*orquestración*”... *quizás*



vagrant

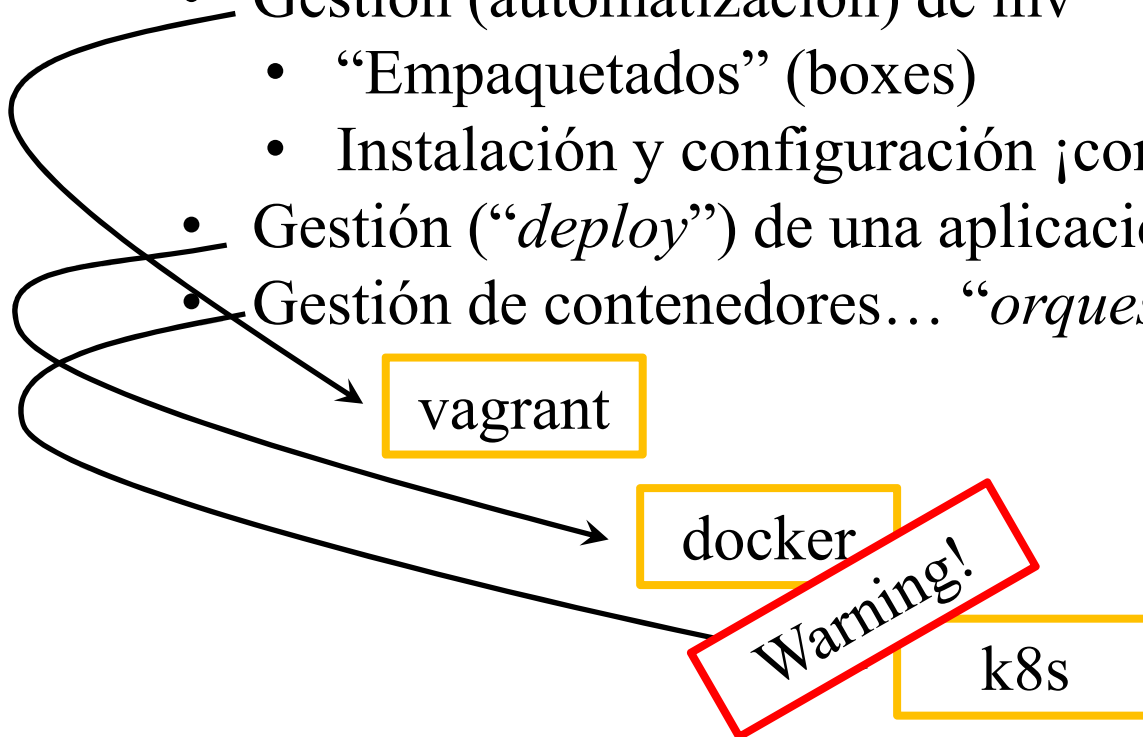
The diagram consists of three yellow rectangular boxes with black text, arranged diagonally from top-left to bottom-right. The boxes are labeled 'vagrant', 'docker', and 'k8s'. Three curved black arrows originate from the list of items above. The first arrow starts from the 'Gestión (automatización) de mv' item and points to the 'vagrant' box. The second arrow starts from the 'Gestión (“deploy”) de una aplicación' item and points to the 'docker' box. The third arrow starts from the 'Gestión de contenedores... “orquestración”... quizás' item and points to the 'k8s' box.

docker

k8s

Ambientes de Ejecución

- Ambientes Virtuales (AV) - Virtualización
 - Usuarios de Vbox/VMware/WSL (¿?)...
 - Gestión (automatización) de mv
 - “Empaquetados” (boxes)
 - Instalación y configuración ¡con código!
 - Gestión (“*deploy*”) de una aplicación
 - Gestión de contenedores... “*orquestración*”... *quizás*



Virtualización

- Qué hay “por debajo” (< nivel de abstracción)
 - Terminología: bare metal - host - guest
 - ¿Bare metal = Hardware?
-

Virtualización

- Qué hay “por debajo” (< nivel de abstracción)
 - Terminología: bare metal - host - guest
 - ¿Bare metal = Hardware?
 - Bare metal: “real hardware”
 - Hardware: lo que “ve”/administra el SO (*quizás* “hal”)
 - Complejidad del bare metal ==> “hal”
-

Virtualización

- Qué hay “por debajo” (< nivel de abstracción)
 - Terminología: bare metal - host - guest
 - ¿Bare metal = Hardware?
 - Bare metal: “real hardware”
 - Hardware: lo que “ve”/administra el SO (*quizás* “hal”)
 - Host: lo que contiene al *guest*
 - Usualmente OS completo con capacidad para *guest*
 - Escala mayor: bare metal + gestión de guests...
 - OS completo ==> “sistema de cómputo operativo”
 - Usualmente solo importa que contenga al *guest*...
-

Virtualización

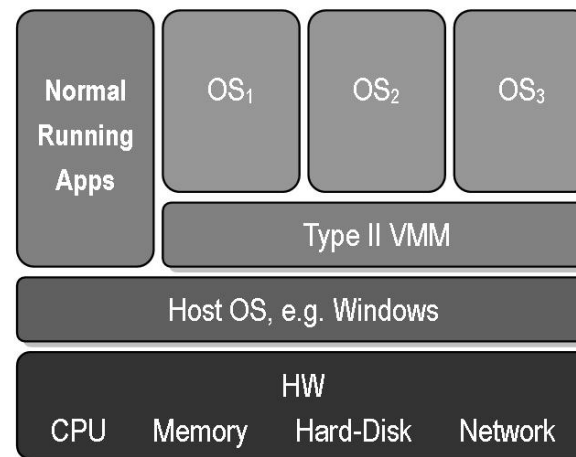
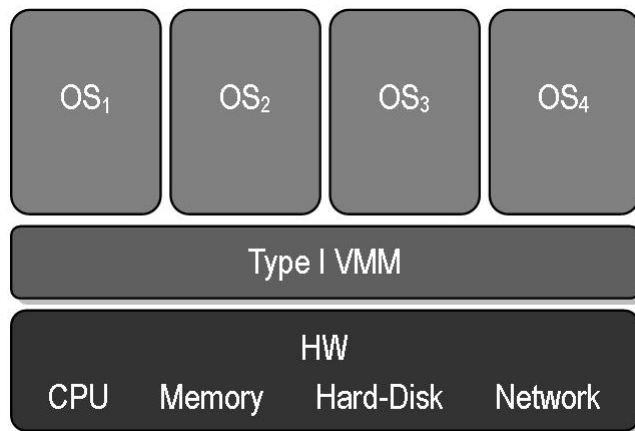
- Qué hay “por debajo” (< nivel de abstracción)
 - Terminología: bare metal - host - guest
 - ¿Bare metal = Hardware?
 - Bare metal: “real hardware”
 - Hardware: lo que “ve”/administra el SO (*quizás* “hal”)
 - Host: lo que contiene al guest
 - Guest: sistema virtualizado
 - Sobre el que se ejecuta-foco
-

Virtualización

- Qué hay “por debajo” (< nivel de abstracción)
 - Terminología: bare metal - host - guest
 - ¿Bare metal = Hardware?
 - Bare metal: “real hardware”
 - Hardware: lo que “ve”/administra el SO (*quizás* “hal”)
 - Host: lo que contiene al guest
 - Guest: sistema virtualizado
 - Sobre el que se ejecuta-foco
 - Implementación
 - Asistencia de hardware
 - Hipervisor/administrador host \Leftrightarrow guest
 - Hipervisor-VMM: Virtual Machine Monitor
-

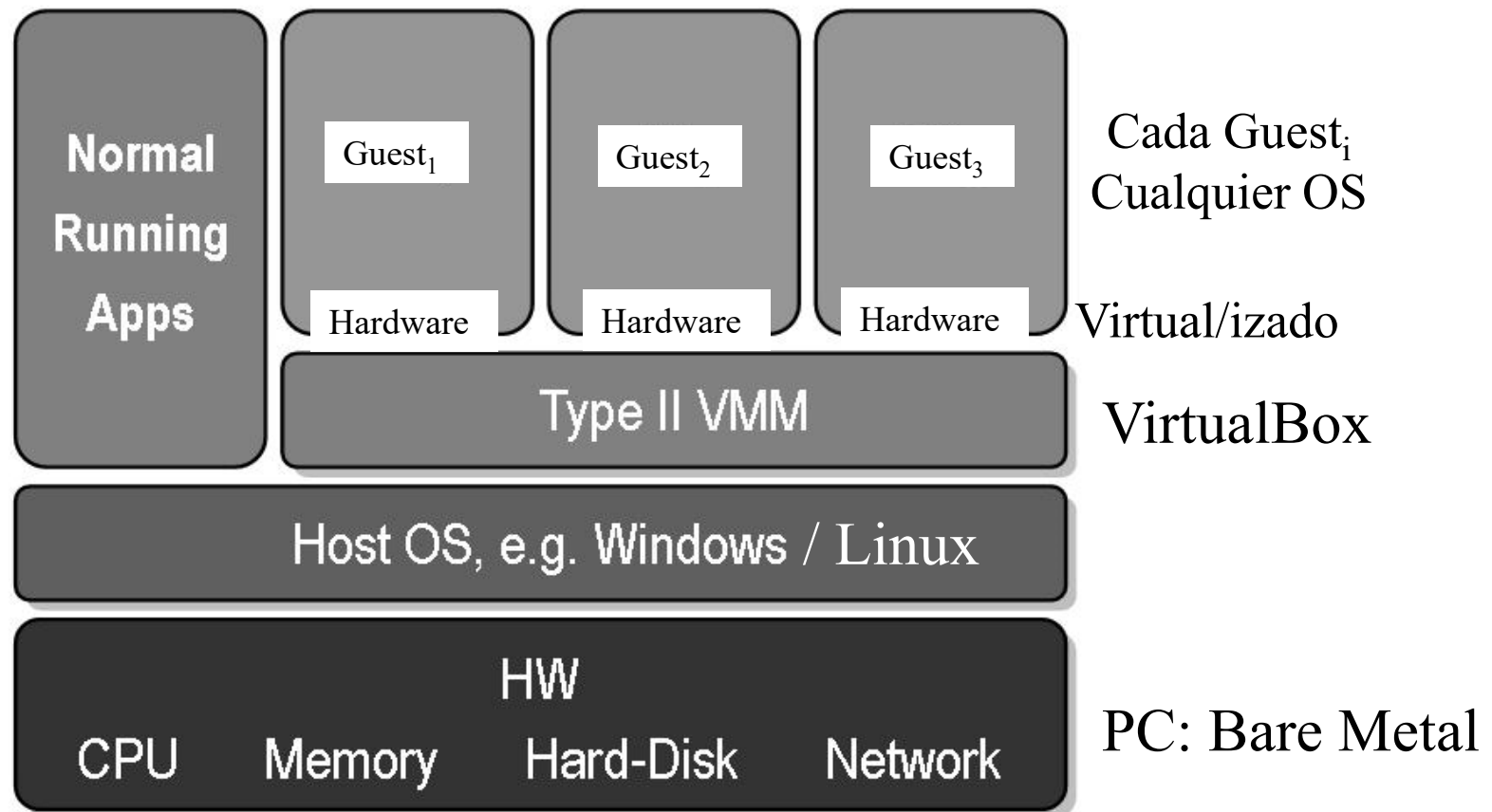
Virtualización

- Hipervisores



Virtualización

- VirtualBox - Tipo II



Vagrant

- VirtualBox
 - Usuario “plano”, ok
 - Gestión manual/“artesanal”: igual a la de una PC...
 - Gestión automatizada... quizás de algunos detalles
 - Vagrant: automatización “completa” de VMs
 - En particular, puede ser usando VirtualBox
 - Boxes: un ambiente/OS virtualizado en particular
 - OS “predefinido”
 - Se puede considerar como una vm “lista para usar”
 - Configurar recursos de hardware (ej: RAM)
 - Instalar y configurar software
 - Programar instalación y configuración de VMs
 - IaC... (¿I?)
-

Vagrant

- VirtualBox ya instalado
 - Instalar Vagrant
 - Trivial con instalador/paquete
 - Buscar Box
 - Usaremos el más básico, bajos recursos requeridos
 - Directamente a través de ejemplo/s
-

Vagrant

- Administrar/Ejecutar una vm de un vmm/hipervisor
 - vmm => “provider” para vagrant
- La base de la vm usualmente es una “box”
 - Se puede instalar “desde cero”-OS

Vagrant

- Administrar/Ejecutar una vm de un vmm/hipervisor
 - vmm => “provider” para vagrant
- La base de la vm usualmente es una “box”
 - Se puede instalar “desde cero”-OS
- En nuestro caso: box =>
 - Ubuntu sin interfaz gráfica
 - Al menos una interfaz de red
 - 1 CPU, 512MB RAM
- El comando vagrant interpreta el vagrantfile
 - Todo lo que define la vm, lo que gestiona vagrant
 - Obtiene de los repositorios lo que sea necesario
 - Verifica actualizaciones
 - Al menos en parte: *declarativo (atómico, idempotente)*

Vagrant

- Comando vagrant, en todos los casos a partir del vagrantfile
 - up, halt, status, destroy
 - Ej. de vagrantfile: vagrantfile.minubuntu1 (cp a vagrantfile)
 - Sección de configuración
 - Sub-sección de “aprovisionamiento” (provision)
 - Ver la interfaz de VBox antes de ejecutar (y después)
cp vagrantfile.minubuntu1 vagrantfile
vagrant up
 - Entre toooodo lo reportado:
 - ==> default: Preparing network interfaces based on configuration...
default: Adapter 1: nat
default: SSH username: vagrant
 - ==> default: Mounting shared folders...
default: /vagrant => <wd> (donde se ejecuta vagrant up)
-

Vagrant

- Si se cambia algo del vagrantfile conviene hacer
vagrant reload --provision
 - Ejecución en la vm:
vagrant ssh
 - Detalles de la vm
 - Usuario: vagrant
 - Clave: vagrant
 - El usuario es sudoer
 - Ver la salida de los comandos
 - free, ls /vagrant, ifconfig -a
 - Cambiar a red pública...
-

Vagrant

- Otros comandos

`vagrant status`

`vagrant halt`

Terminar una vm que no termina con "halt"

`vagrant suspend`

`vagrant halt`

Destruir/borrar completamente una mv

`vagrant destroy`

...
