

Cloud computing

Tema 3. Plataformas IaaS

© 2022 Javier Esparza Peidro - jesparza@dsic.upv.es

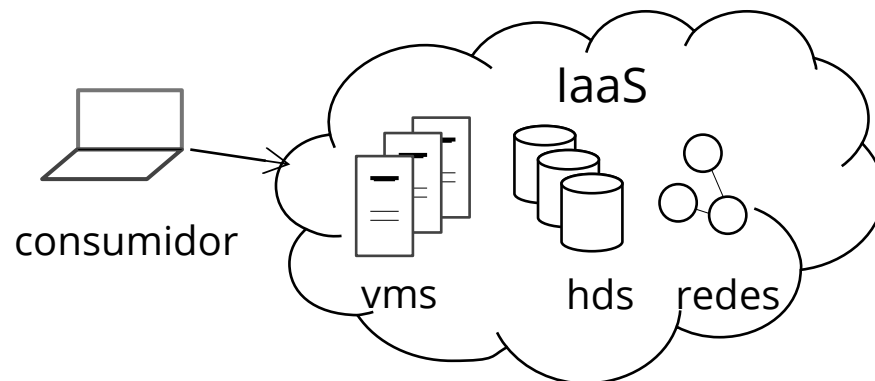
Contenido

- Introducción
- Anatomía de un IaaS
- Infraestructura física
- Infraestructura virtual
- Orquestación
- Servicios
- OpenStack

Introducción

¿Qué es?

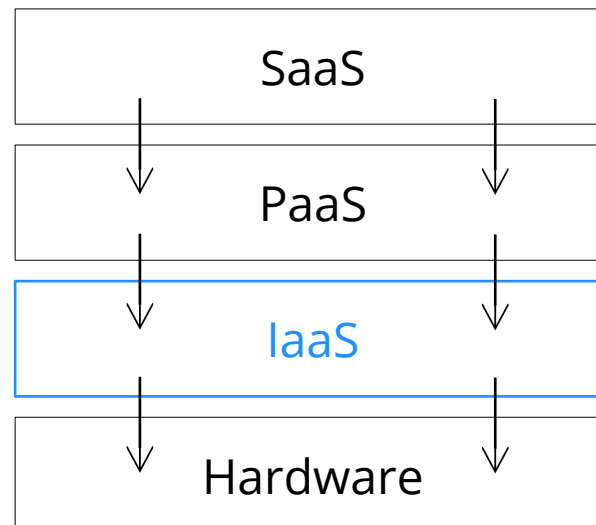
- El consumidor adquiere recursos computacionales de bajo nivel (vms, discos virtuales, redes, ...)
- El administrador/arquitecto de sistemas instala y configura el software.
- No hay control de los recursos físicos. Se gestiona una **infraestructura virtual**.



Introducción

Estructura en capas

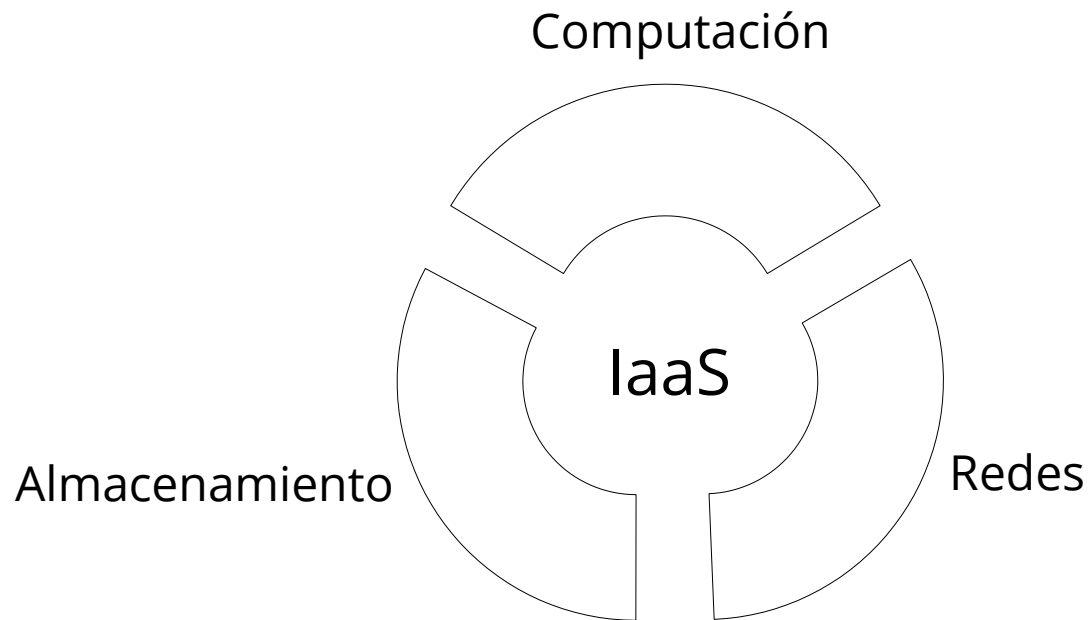
- Un entorno IaaS ocupa el nivel más bajo
- Se suele utilizar para construir servicios de más alto nivel (PaaS/SaaS)



Introducción

Recursos

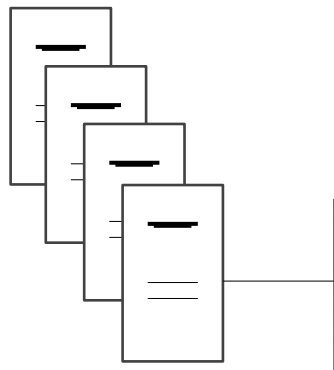
- Computación, almacenamiento, redes



Introducción

Recursos > Computación

- Máquinas virtuales: instancias
- Requisitos de CPU, GPU, memoria, ...
- Suelen existir distintas opciones preestablecidas (AWS EC2 > 50 modelos)

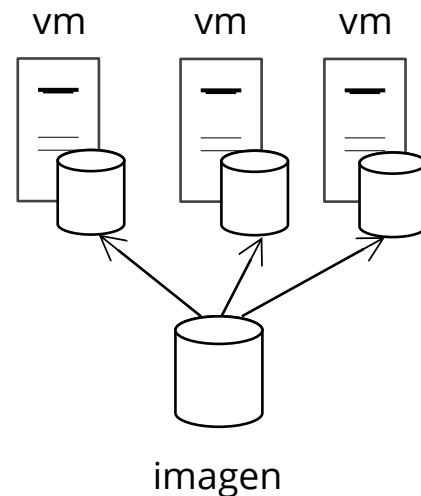


- CPU 12 núcleos
- Memoria 32 Gb
- Ancho banda red 10Gbps
- Ancho banda i/o 8000 Mbps

Introducción

Recursos > Computación

- Se suelen crear a partir de una **imagen**
- Se montan como dispositivo orientado a bloques
- Es el espacio de almacenamiento primario



Introducción

Recursos > Computación

- Las instancias se pueden arrancar, parar, pausar, reanudar, destruir
- Es posible conectarse a la vm usando SSH, etc.
- Se monitoriza el consumo de recursos
- Herramientas de escalado automático
- Reglas:
 - Condiciones que debe verificar: %CPU, %mem
 - Acciones a tomar: crear/destruir réplicas

Introducción

Recursos > Almacenamiento

- Almacenamiento tipo bloque
- Sistemas de ficheros compartido
- Almacenes SQL y NoSQL

Introducción

Recursos > Almacenamiento

- Almacenamiento tipo bloque
 - Concepto de volumen o disco virtual
 - Efímero vs persistente

Introducción

Recursos > Almacenamiento

- Almacenamiento tipo bloque
- Sistemas de ficheros compartido
 - Varias instancias comparten un directorio con ficheros
 - NFS, CIFS

Introducción

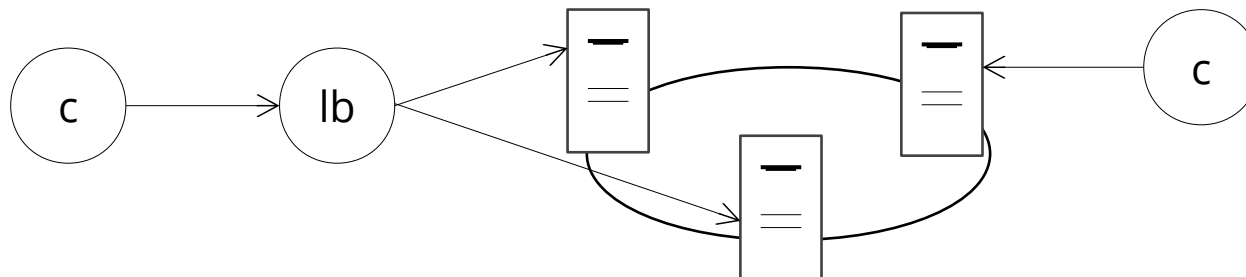
Recursos > Almacenamiento

- Almacenamiento tipo bloque
- Sistemas de ficheros compartido
- Almacenes SQL y NoSQL
 - Motores de bases de datos relacionales o no, adaptados para funcionar en un entorno cloud
 - Suelen estar replicados y optimizados para soportar grandes cargas de trabajo y volúmenes de datos

Introducción

Recursos > Redes

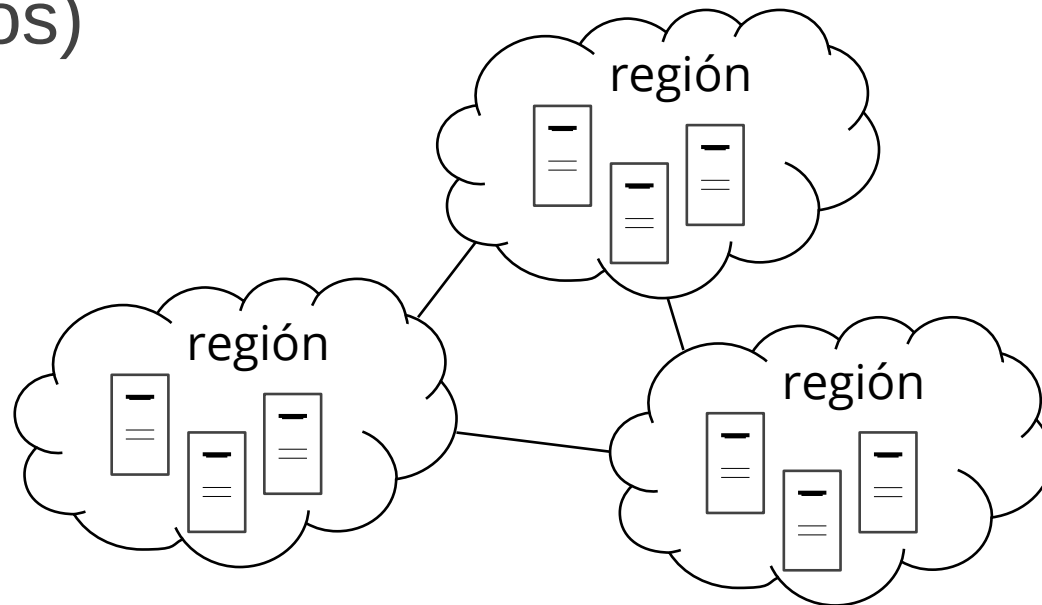
- Proporciona conectividad a las instancias
- Espacio de direccionamiento plano vs redes
- Redes es más versátil: subredes, routers, etc.
- Habitualmente por defecto las instancias no son visibles desde el exterior
- IP pública vs balanceadores de carga



Introducción

Recursos > Regiones

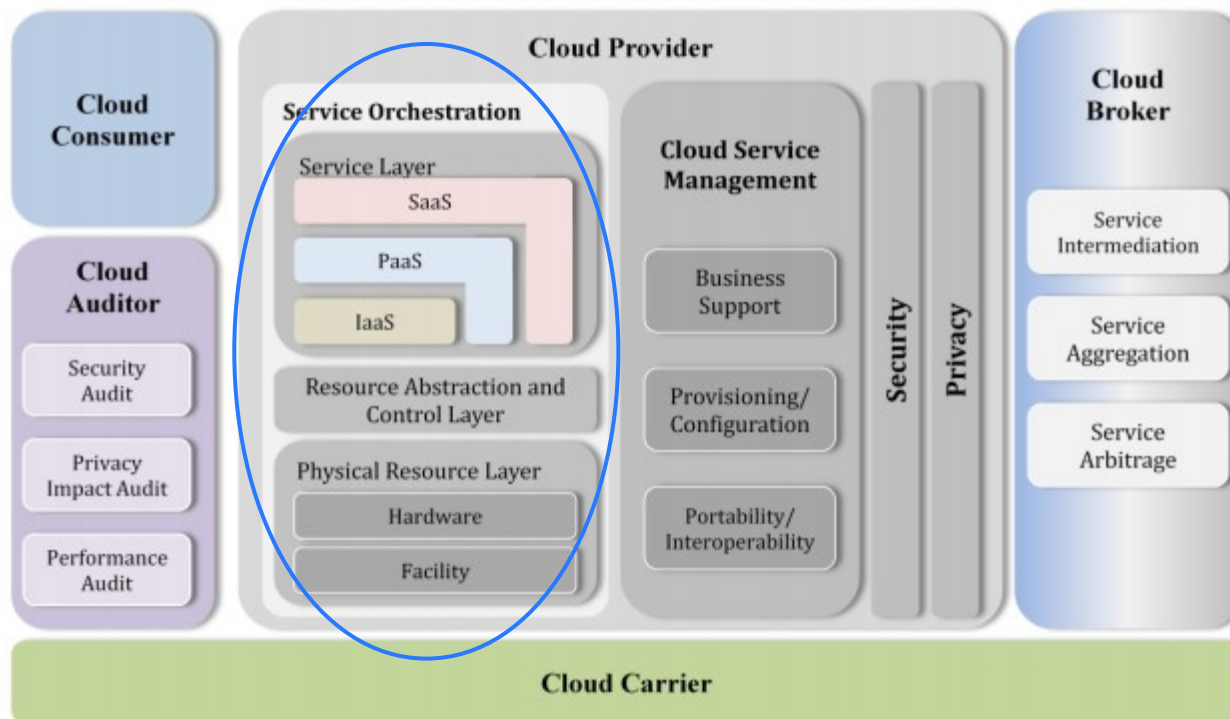
- Los recursos se pueden reservar en distintas ubicaciones geográficas (Europa, USA, China, ...)
- Incrementa la fiabilidad de la solución final
- En AWS y Google una región contiene varias zonas (centros de datos)



Anatomía de un IaaS

¿Qué es?

- Tomando con referencia la arquitectura propuesta por [NIST](#)



Anatomía de un IaaS

¿Qué es?

- Un IaaS se puede descomponer en capas

Servicios

Orquestación

Infraestructura
virtual

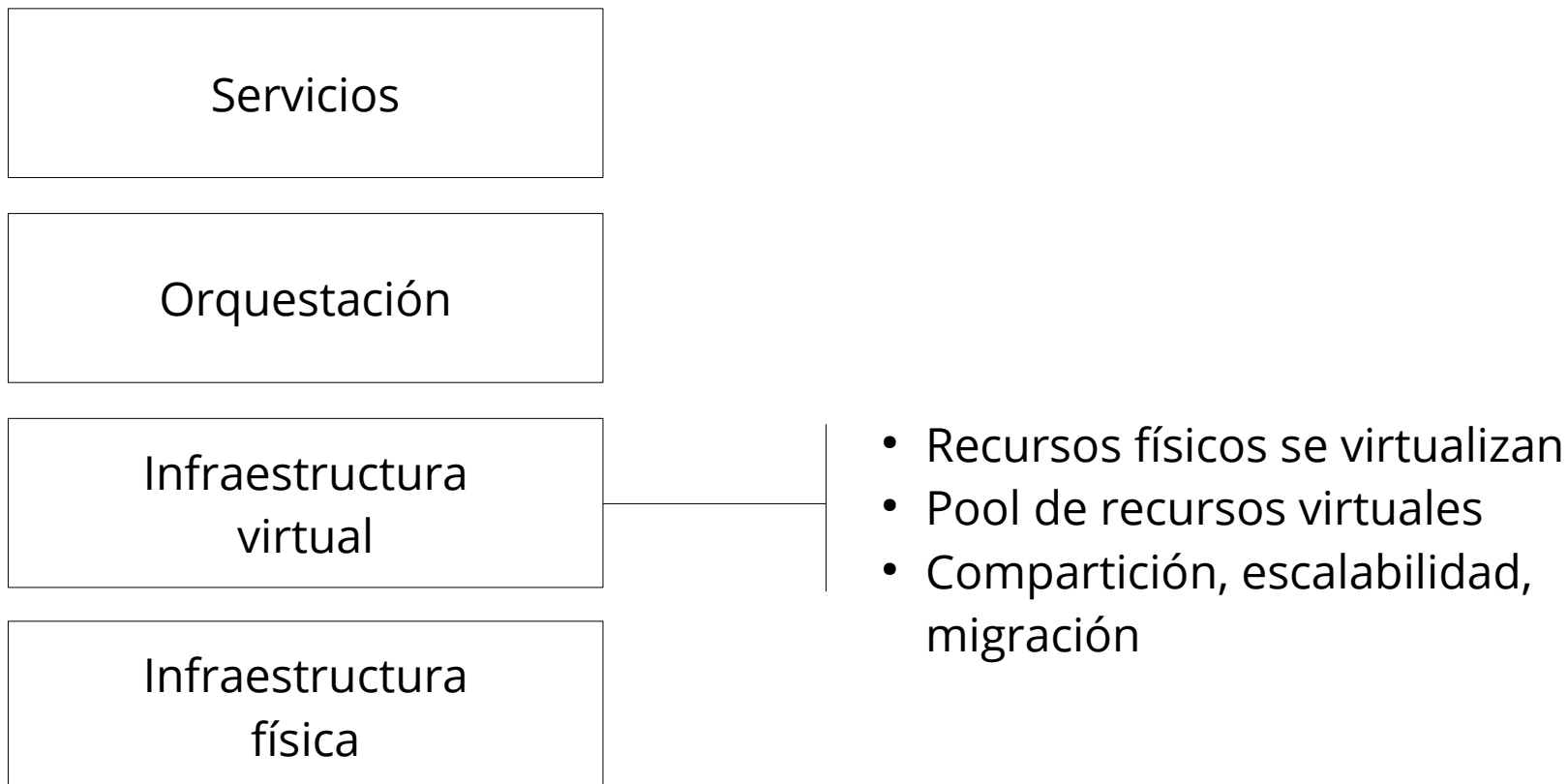
Infraestructura
física

- Recursos físicos (computadores, redes, almacenamiento, etc.)
- Infraestructura
- Suministros

Anatomía de un IaaS

¿Qué es?

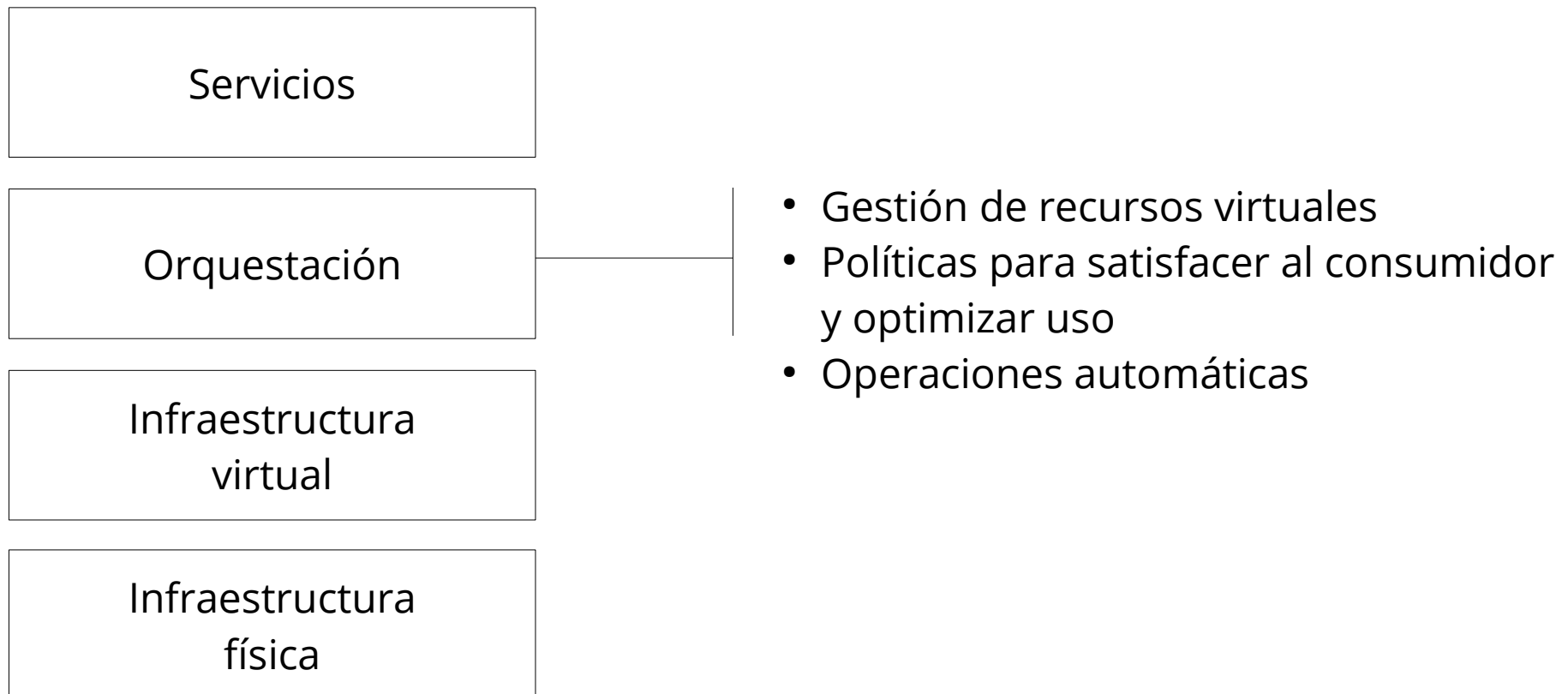
- Un IaaS se puede descomponer en capas



Anatomía de un IaaS

¿Qué es?

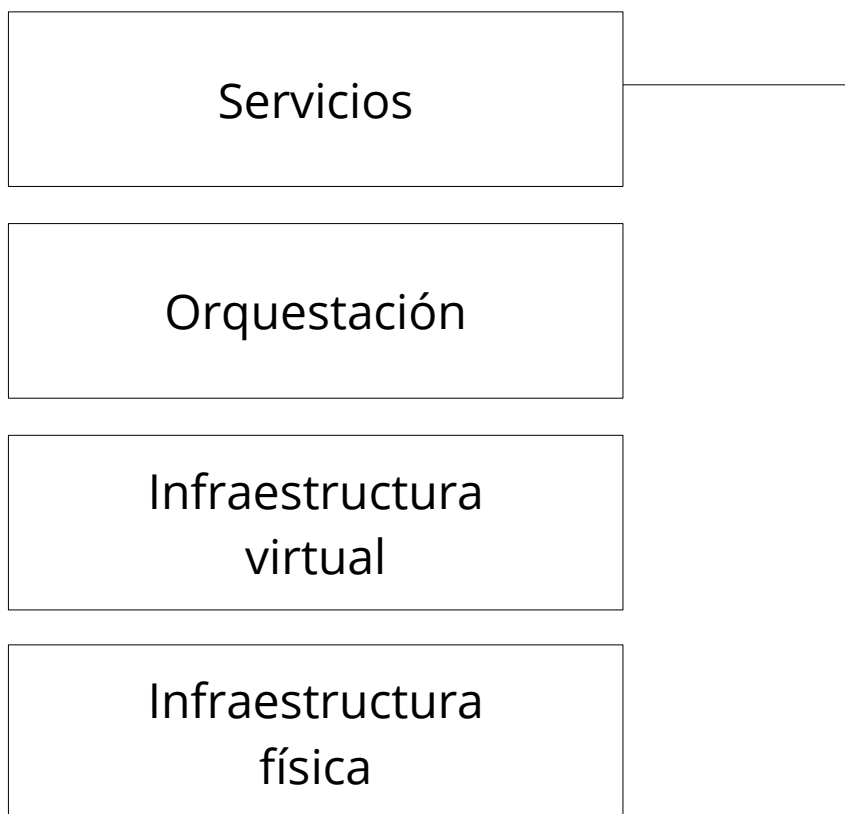
- Un IaaS se puede descomponer en capas



Anatomía de un IaaS

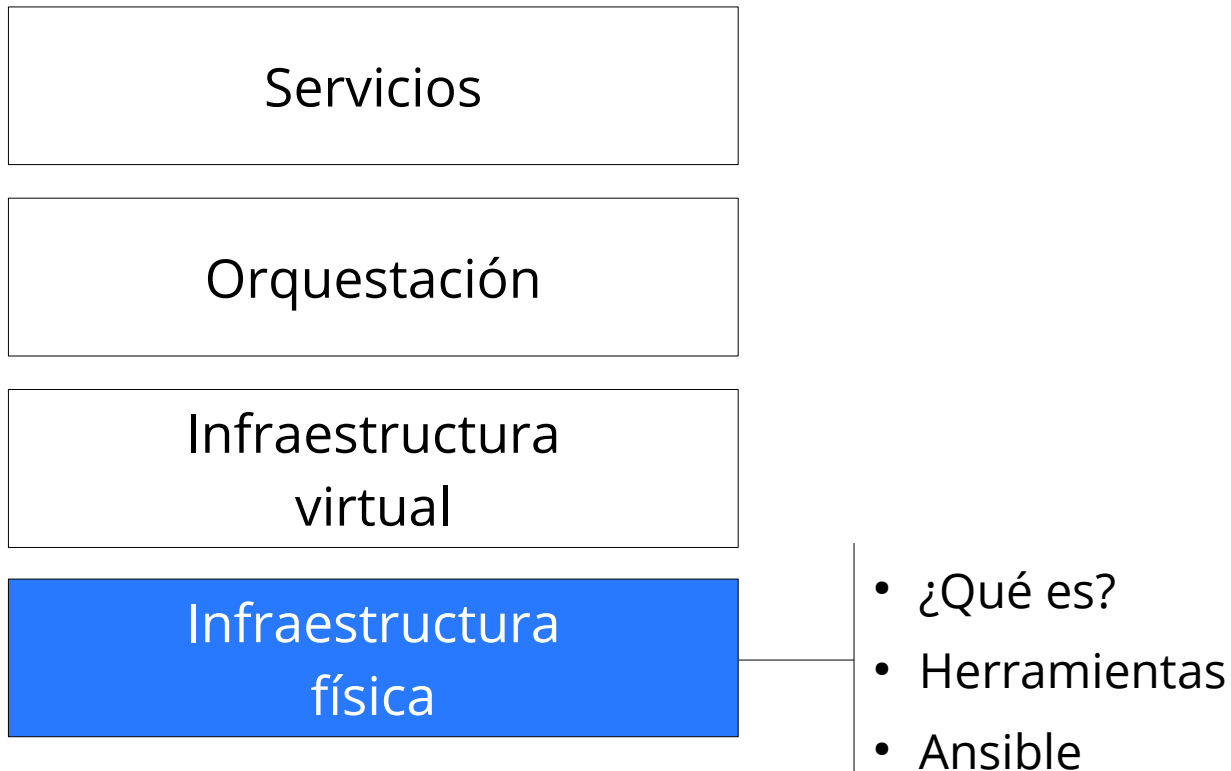
¿Qué es?

- Un IaaS se puede descomponer en capas



- Publica API para consumir recursos

Infraestructura física



Infraestructura física

¿Qué es?

- Contiene los recursos físicos del sistema
- Las labores de interconexión son manuales
- Hay un grandísimo volumen de recursos
- Hay que **automatizar** siempre que se pueda



Infraestructura física

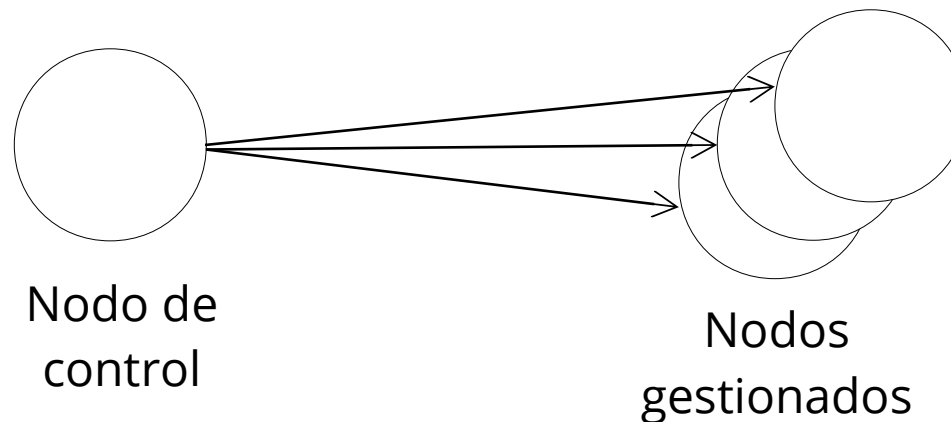
Herramientas

- Aprovisionamiento: instalación de s.o. y aplicaciones (e.g. Clober, Kickstart, ...)
- Gestión de la configuración: aplicar configuraciones distintas a las máquinas (e.g. [Ansible](#), Cfengine, Chef, ...)
- Monitorización: de recursos, logs, rendimiento, fallos (e.g. Nagios, Zabbix, ...)

Infraestructura física

Ansible

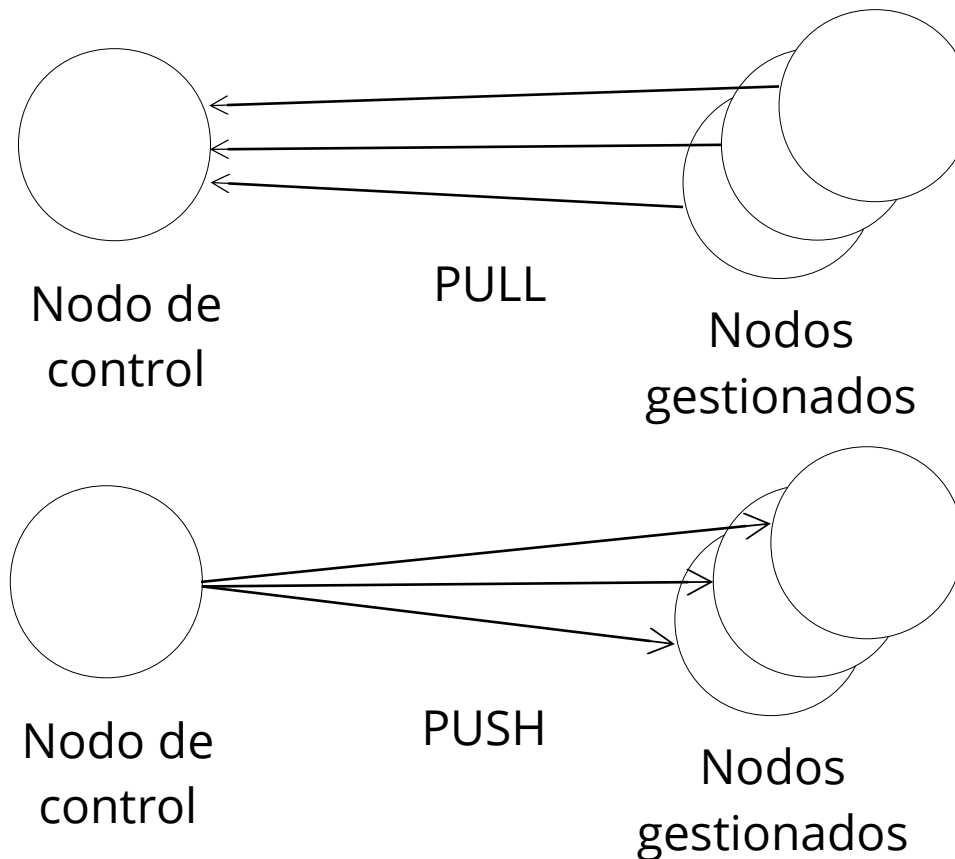
- Automatiza configuración, instalación, actualización de infraestructura física/virtual
- Un nodo de control se conecta vía SSH con cientos/miles de nodos gestionados



Infraestructura física

Ansible

- PUSH (Ansible) vs PULL



Infraestructura física

Ansible > Instalación

- Sistema de [paquetes](#) vs [pip](#)
- Nodos gestionados: ssh automático
 - > apt install openssh-server
- */etc/ssh/sshd_config*
 - PubkeyAuthentication yes
- Registrar clave pública en *~/.ssh/authorized_keys*
 - > ssh-copy-id -i .ssh/id_rsa.pub root@machine

Infraestructura física

Ansible > Configuración

- Opciones por defecto en un fichero de configuración
 - `ANSIBLE_CONFIG`, `./ansible.cfg`, `~/.ansible.cfg`,
`/etc/ansible/ansible.cfg`

```
[defaults]
inventory = hosts
remote_user = root
private_key_file = .ssh/id_rsa
host_key_checking = False
```

Infraestructura física

Ansible > Comandos ad hoc

- Útil para tareas puntuales
 - > `ansible [pattern] -m [module] -a "[options]"`
 - [pattern] determina los nodos target
 - [module] determina la acción a ejecutar
 - [options] opciones dependientes de la acción
- Módulo ping
 - > `ansible localhost -m ping`

Infraestructura física

Ansible > Comandos ad hoc

- Módulo command (por defecto)
 - > `ansible localhost -m command -a uptime`
 - > `ansible localhost -a uptime`
 - > `ansible localhost -a 'ls -la'`
- Cambiar de usuario con -u
 - > `ansible localhost -a "/sbin/reboot" -u root`
- A root con --become/-b [-K/--ask-become-pass]
 - > `ansible localhost -a "/sbin/reboot" -u root --b -K`

Infraestructura física

Ansible > Inventario

- Contiene las direcciones de los nodos gestionados ([fichero](#), [dinámico](#))
- Puede contener más información (autenticación, variables, ...)

```
www.example.com
db-[a:f].example.com
test ansible_host=192.168.122.148
test2 ansible_host=192.168.122.149
      ansible_port=22 ansible_user=root
      ansible_private_key=.ssh/id_rsa
```

Infraestructura física

Ansible > Inventario

- Los nodos se pueden incluir en grupos

```
[webservers]
```

```
www01.example.com
```

```
www[03:50].example.com
```

- Se especifica el inventario con -i

```
> ansible test -i hosts -m ping
```

- Se especifica el target con nodos, grupos y patrones

```
> ansible webservers -i hosts -m ping
```

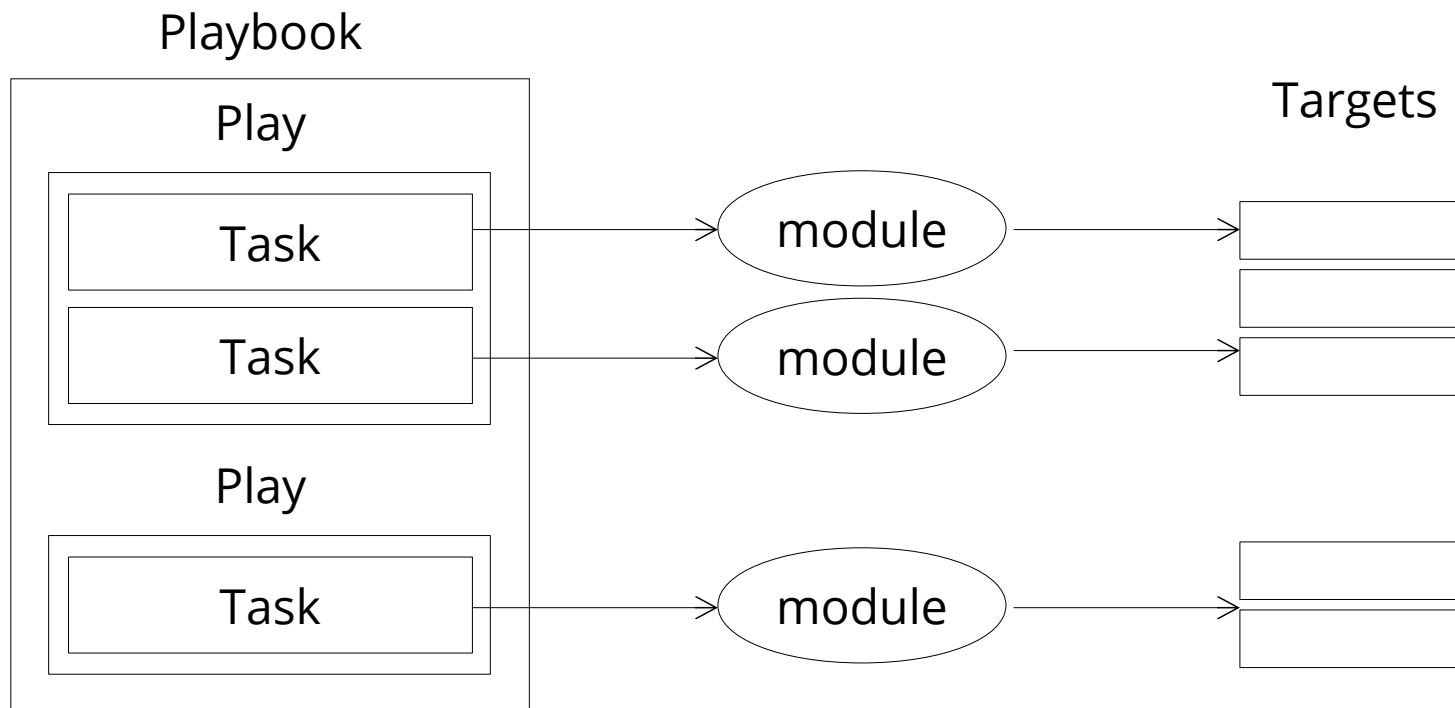
```
> ansible all -i hosts -m ping
```

```
> ansible test1,test2,test3 -i hosts -m ping
```

Infraestructura física

Ansible > Playbooks

- Lista de tareas ordenada que se puede reutilizar
- Es la manera habitual de configurar sistemas (script)



Infraestructura física

Ansible > Playbooks

- Se define utilizando [YAML](#)

```
---
```

```
# Very simple play
```

```
- name: Ping all hosts
```

```
  hosts: all
```

```
  tasks:
```

```
    - name: Make ping
```

```
      ping:
```

_____ Play

_____ Targets

_____ Tarea

_____ Módulo

- Se ejecuta con ansible-playbook

```
> ansible-playbook -i hosts ping.yml
```


Infraestructura física

Ansible > Playbooks

- Hay muchas [opciones](#) de configuración

```
---
```

```
# Multiple plays
```

```
- name: Add user
  hosts: master
  become: true
  tasks:
    - name: Create user
      user:
        name: alumno
        group: users,admin
        shell: /bin/bash
        home: /home/alumno
```

```
- name: Install python
  hosts: slaves
  become: true
  tasks:
    - name: Install Python
      apt:
        name: python3
        state: present
```

Infraestructura física

Ansible > Módulos

- Unidades de código ejecutadas por Ansible
- Devuelven resultados (que pueden utilizarse)
- Deben ser **idempotentes**
- [Catálogo](#) muy amplio: file, command, shell, apt, copy, user, service, debug, lineinfile, git, setup, template, ...
- Documentación con [ansible-doc](#)

```
> ansible-doc user
```

Infraestructura física

Ansible > Variables

- Pueden contener strings, listas, diccionarios, ...
- Se usan:
 - Como parámetros de entrada en las tareas
 - Para ejecutar tareas de manera condicional
 - En plantillas y bucles

Infraestructura física

Ansible > Variables

- En el inventario
- En un play
- En un fichero .yaml
- En línea de comandos
- A partir de los resultados de un módulo
- A partir de los facts

Infraestructura física

Ansible > Variables

- En el inventario

```
test ansible_host=192.168.1.1
```

Infraestructura física

Ansible > Variables

- En un play

```
---  
- hosts: server  
  vars:  
    user_name: pepe
```

Infraestructura física

Ansible > Variables

- En un fichero .yaml

```
---
```

```
var1: valu1
```

```
var2: valu2
```

```
---
```

```
- hosts: server
```

```
  vars_files:
```

```
    - /vars/myvars.yml
```

Infraestructura física

Ansible > Variables

- En línea de comandos

```
> ansible-playbook -i hosts  
    -e "var1:value1 var2:value2" playbook.yml
```


Infraestructura física

Ansible > Variables

- A partir de los resultados de un módulo

```
---
```

```
- hosts: web_servers
```

```
  tasks:
```

```
    - name: Register output as a variable
      ansible.builtin.shell: /usr/bin/foo
      register: foo_result
      ignore_errors: true
```

```
    - name: Use output of the previous task
      ansible.builtin.shell: /usr/bin/bar
      when: foo_result.rc == 5
```

Infraestructura física

Ansible > Variables

- A partir de los facts

```
---  
- name: ansible facts  
  hosts: all  
  tasks:  
    - debug:  
      var: ansible_facts
```

Infraestructura física

Ansible > Plantillas

- Las variables se pueden usar en plantillas [Jinja2](#)
- Se incluyen diversos [filtros](#)

```
---  
- hosts: server  
  vars:  
    user_name: pepe  
  tasks:  
    - user:  
      name: "{{ user_name }}"  
      group: users,admin  
      shell: /bin/bash  
      home: /home/alumno
```

Infraestructura física

Ansible > Condicionales

- Se ejecutan tareas de manera condicional con when
- Evalúa una expresión Jinja2

```
---  
- hosts: all  
  tasks:  
    - name: Shut down Debian flavored systems  
      ansible.builtin.command: /sbin/shutdown -t now  
      when: ansible_facts['os_family'] == "Debian"
```

Infraestructura física

Ansible > Condicionales

- Se ejecutan tareas de manera condicional con when
- Evalúa una expresión Jinja2

```
---
- hosts: all
  vars:
    epic: true
    monumental: "yes"
  tasks:
    - name: "epic" or "monumental" is true?
      ansible.builtin.shell: echo "This is epic!"
      when: epic or monumental | bool
    - name: "epic" is false
      ansible.builtin.shell: echo "This isn't epic!"
      when: not epic
```

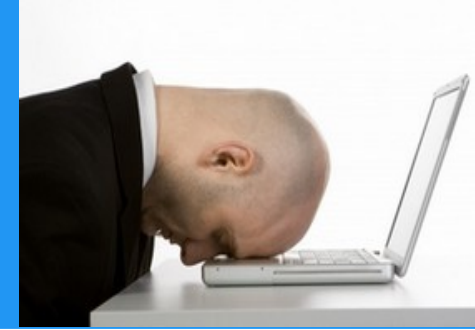
Infraestructura física

Ansible > Condicionales

- Se ejecutan tareas de manera condicional con when
- Evalúa una expresión Jinja2

```
---  
- name: Test play  
  hosts: all  
  tasks:  
    - name: Register a variable  
      ansible.builtin.shell: cat /etc/motd  
      register: motd_contents  
    - name: Use variable  
      ansible.builtin.shell: echo "yes!!"  
      when: motd_contents.stdout.find('hi') != -1
```

Infraestructura física



Ejercicio 1

- Diseñar un playbook que haga lo siguiente en los nodos gestionados:
 - 1) Crear un usuario parametrizable por variable
 - 2) Crear un directorio sync/ en el home del usuario
 - 3) Copiar ciertos archivos del nodo de control

Infraestructura virtual



- ¿Qué es?
- Virtualización
- Virtualización de la ejecución
- Virtualización de la red
- Virtualización de almacenamiento

Infraestructura virtual

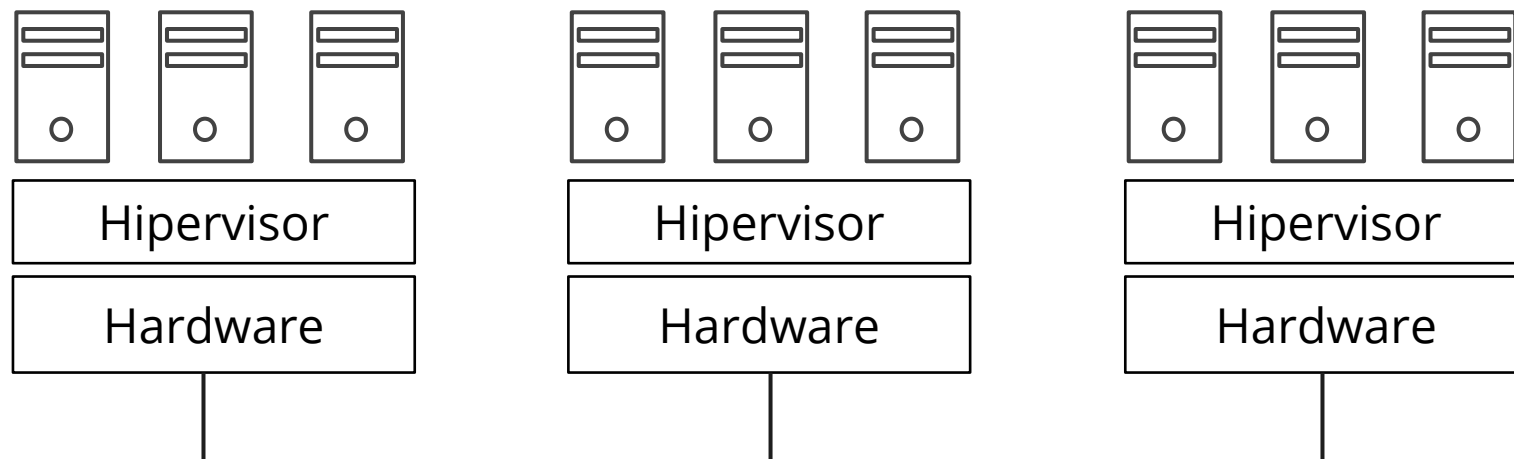
¿Qué es?

- La infraestructura física es **virtualizada**
- Se virtualizan todos los recursos (**ejecución, red, almacenamiento**)
- Mejor gestión y utilización de los recursos disponibles

Infraestructura virtual

Virtualización de la ejecución

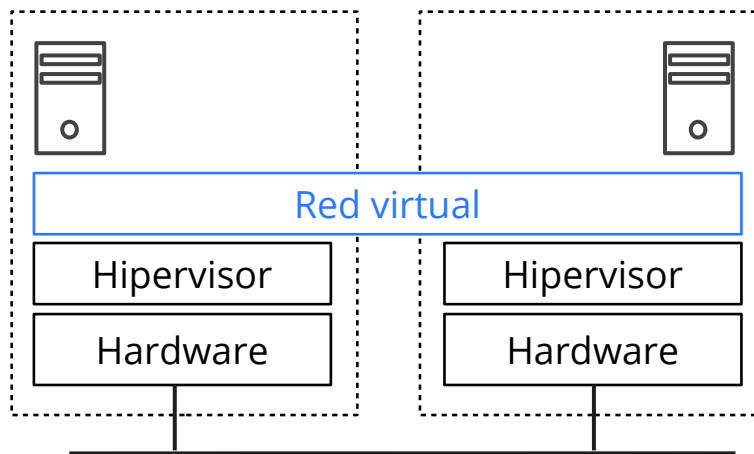
- La infraestructura virtual se construye a partir de nodos físicos con un hipervisor
- El hipervisor permite la ejecución de **instancias**
- Soluciones como **libvirt** son muy recomendables



Infraestructura virtual

Virtualización de la red

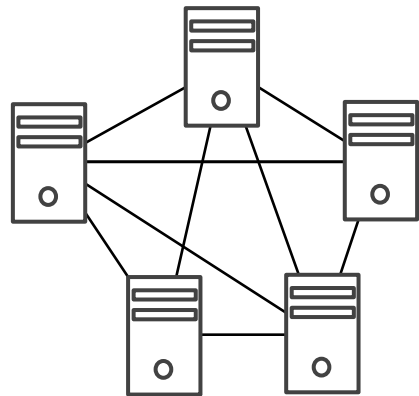
- Las instancias pueden ejecutarse en cualquier nodo físico y pueden comunicarse entre sí



Infraestructura virtual

Virtualización de la red

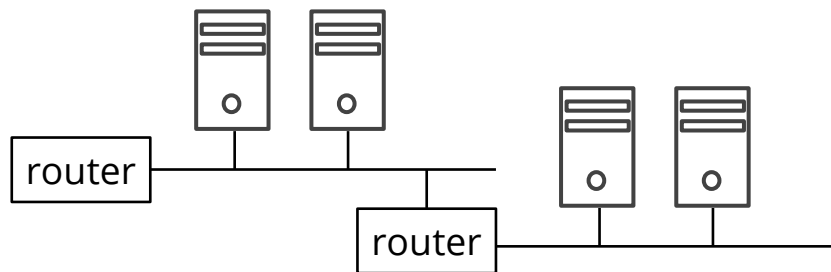
- Las instancias pueden ejecutarse en cualquier nodo físico y pueden comunicarse entre sí
- Alternativas
 - Espacio de direccionamiento plano



Infraestructura virtual

Virtualización de la red

- Las instancias pueden ejecutarse en cualquier nodo físico y pueden comunicarse entre sí
- Alternativas
 - Espacio de direccionamiento plano
 - Redes y subredes



Infraestructura virtual

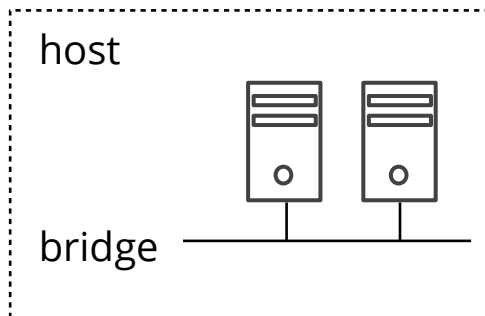
Virtualización de la red

- Las instancias pueden ejecutarse en cualquier nodo físico y pueden comunicarse entre sí
- Alternativas
 - Espacio de direccionamiento plano
 - Redes y subredes
- Existen distintas soluciones. Por ejemplo:
 - El bridge de Linux
 - Open vSwitch

Infraestructura virtual

Virtualización de la red > El bridge de Linux

- Interconexión de instancias **en el mismo host**
- Crea un puente entre dos o más redes, los paquetes se retransmiten en la capa 2 de red
- Actúa como un switch de red virtual



Infraestructura virtual

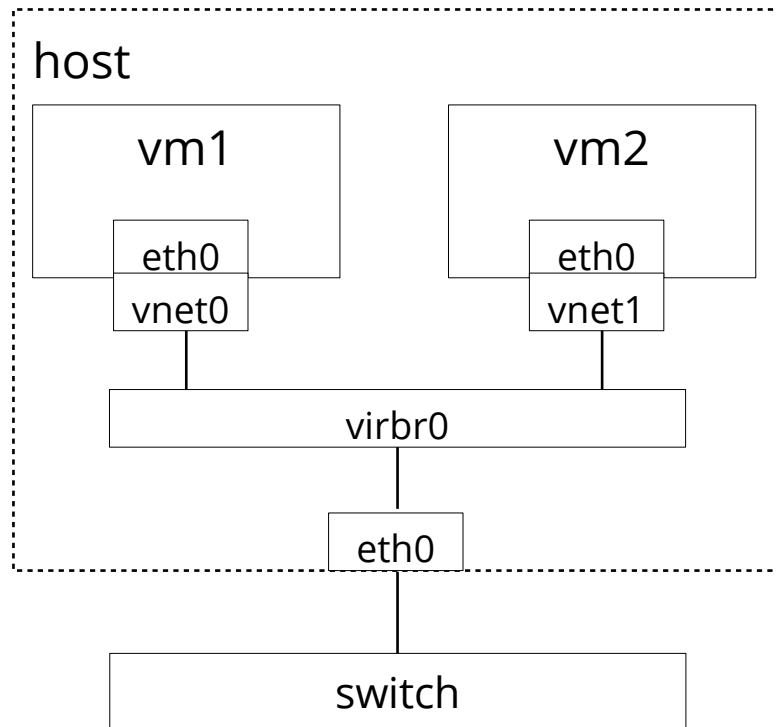
Virtualización de la red > El bridge de Linux

- Instalación
 - > `sudo apt install bridge-utils`
- Mostrar configuración
 - > `sudo brctl show`
- Crear/eliminar un puente
 - > `sudo brctl addbr/delbr <bridge>`
- Añadir/eliminar dispositivos de red al puente
 - > `sudo brctl addif/delif <bridge> <device>`

Infraestructura virtual

Virtualización de la red > El bridge de Linux

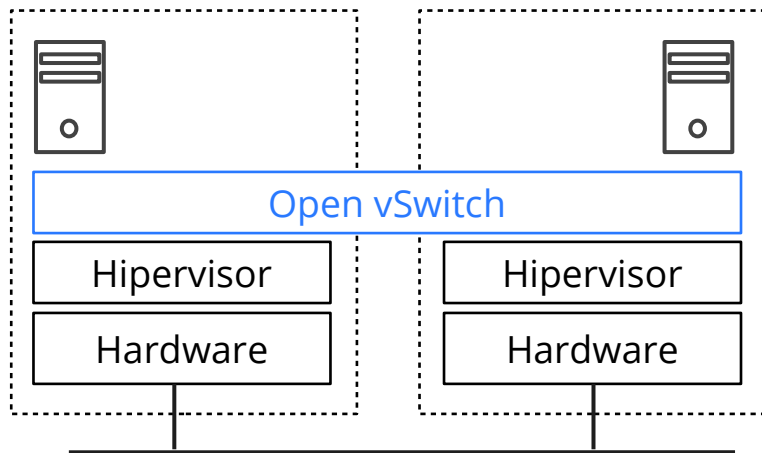
- Con máquinas virtuales (libvirt)



Infraestructura virtual

Virtualización de la red > Open vSwitch

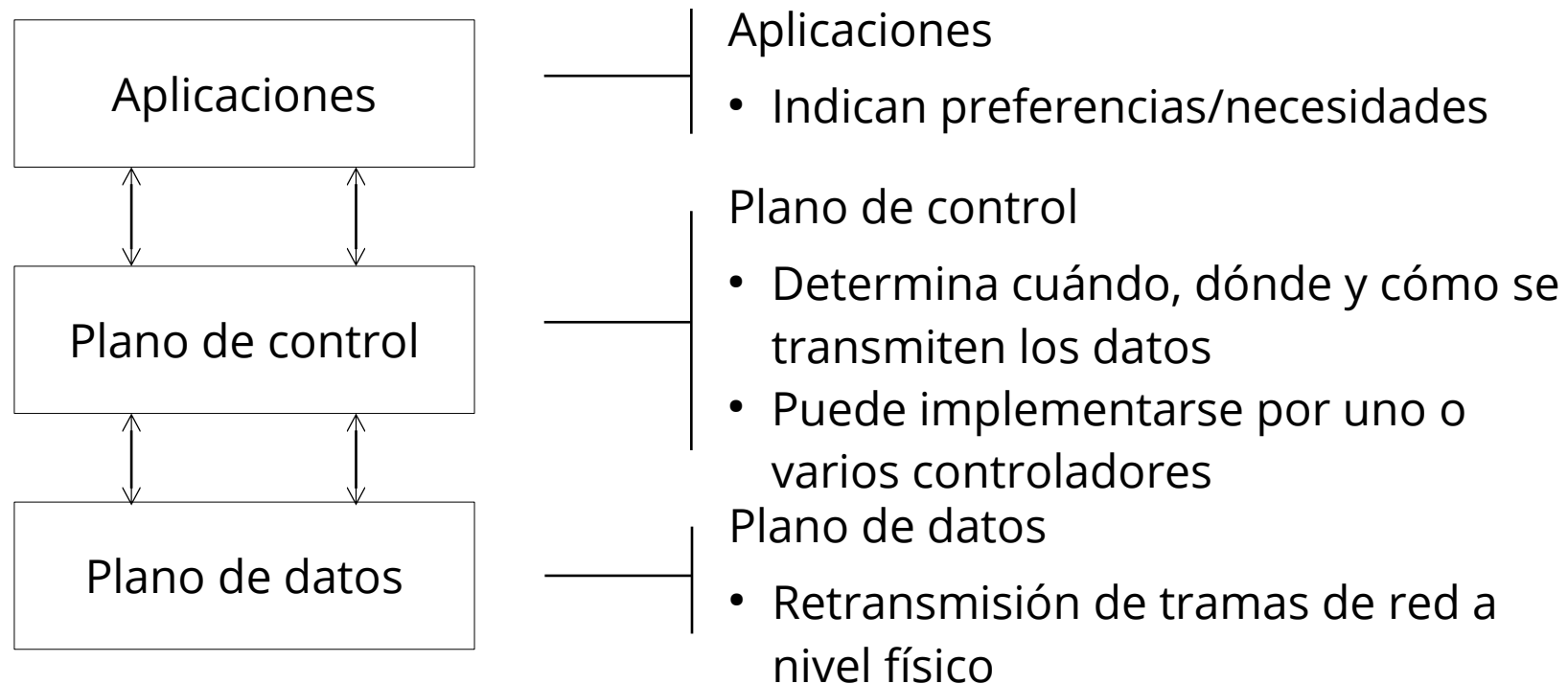
- Switch virtual inteligente, que permite la conexión de vms en el mismo o distintos hosts
- Puede ser programado (SDN – Soft-Defined Net)
- Soporta múltiples tecnologías de virtualización



Infraestructura virtual

Virtualización de la red > Open vSwitch

- Implementa protocolo [OpenFlow](#)
- Desacopla el plano de control del plano de datos

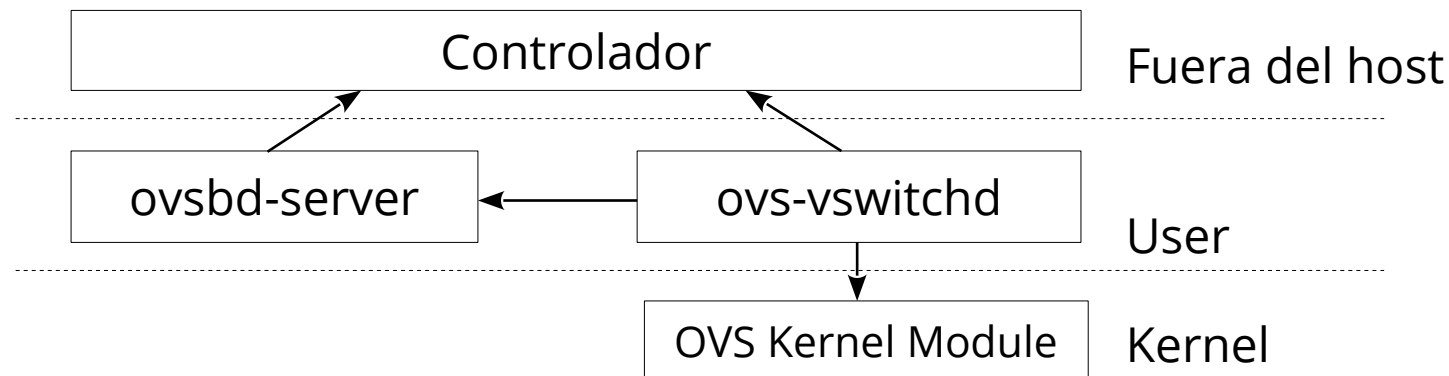


Infraestructura virtual

Virtualización de la red > Open vSwitch

Arquitectura

- Módulo de kernel: implementa el plano de datos
- Herramientas de usuario: plano de control
 - Demonio *ovs-vswitchd*: el switch
 - Demonio *ovsdb-server*: base de datos con config
 - Clientes *ovs-dpctl*, *ovs-vsctl*, *ovs-appctl*



Infraestructura virtual

Virtualización de la red > Open vSwitch

- Instalación

- > `sudo apt install openvswitch-switch`

- Mostrar configuración

- > `sudo ovs-vsctl show`

- Crear/eliminar un switch

- > `sudo ovs-vsctl add-br/del-br <switch>`

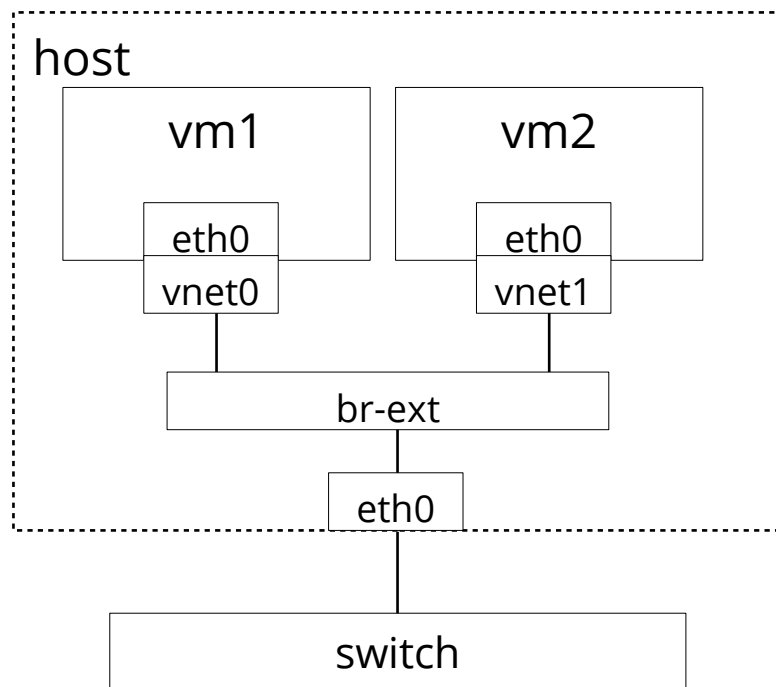
- Añadir/eliminar dispositivos de red al switch

- > `sudo ovs-vsctl add-port/del-port <switch> <dev>`

Infraestructura virtual

Virtualización de la red > Open vSwitch

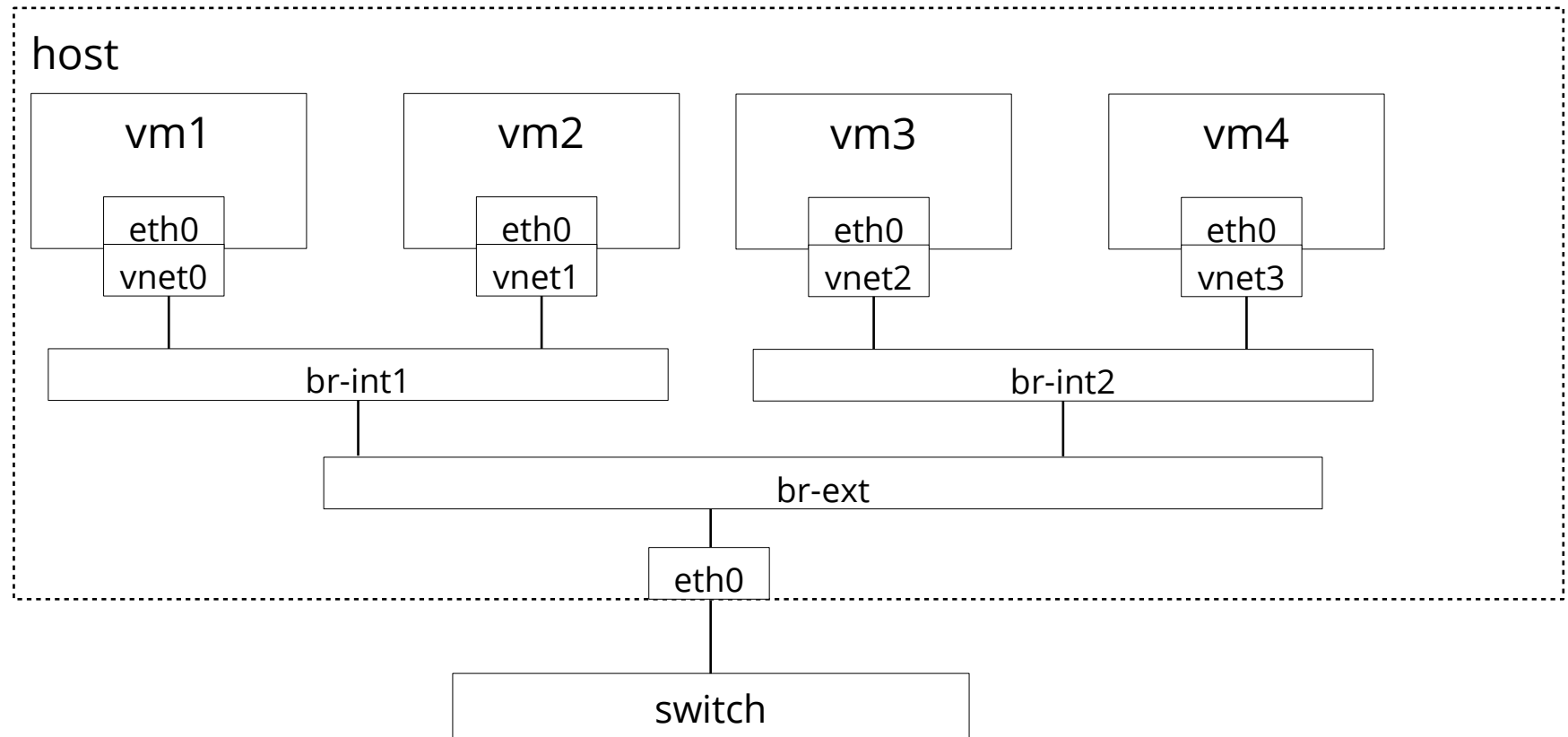
- Con máquinas virtuales en un host como bridge



Infraestructura virtual

Virtualización de la red > Open vSwitch

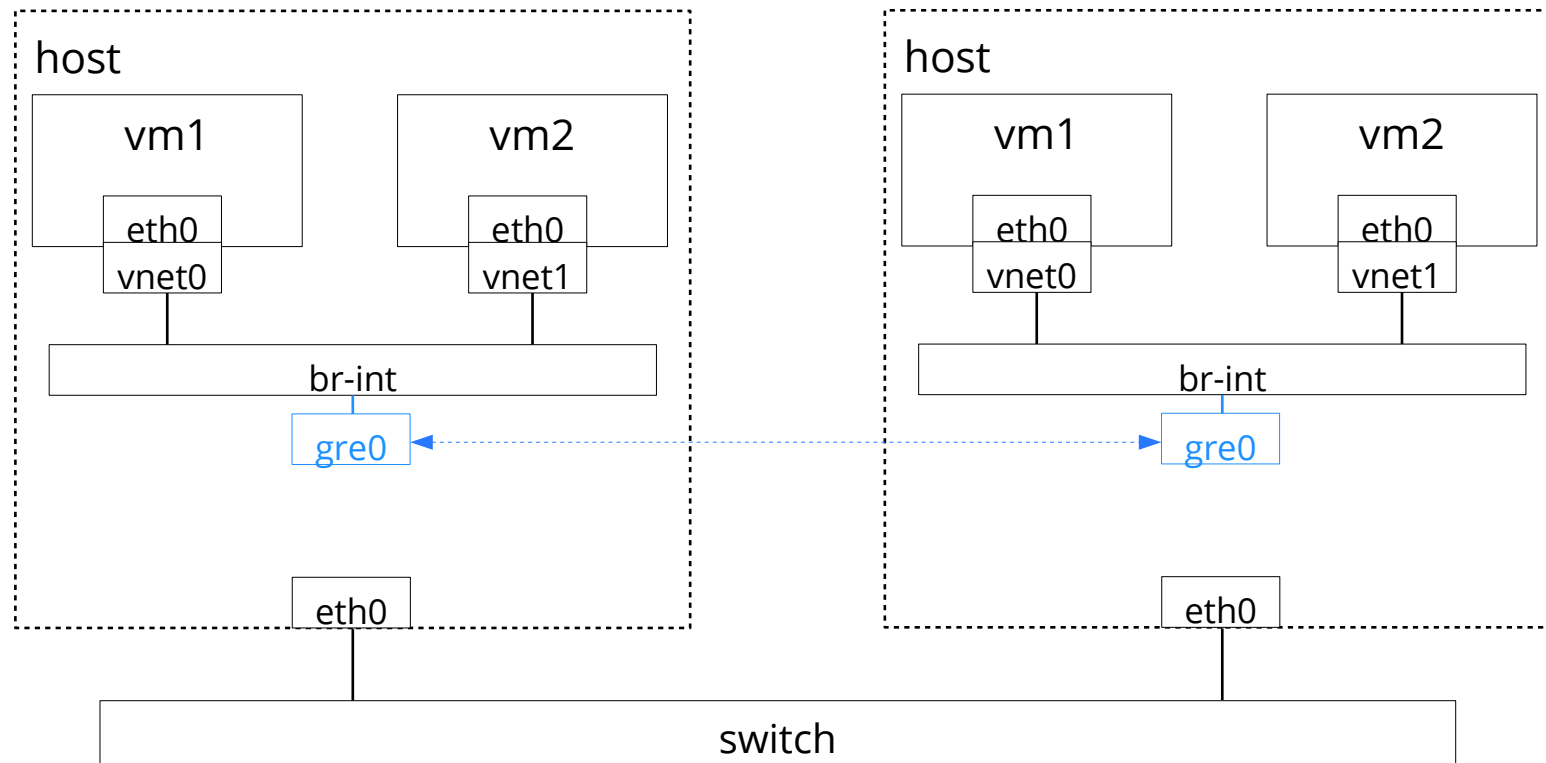
- Varias subredes



Infraestructura virtual

Virtualización de la red > Open vSwitch

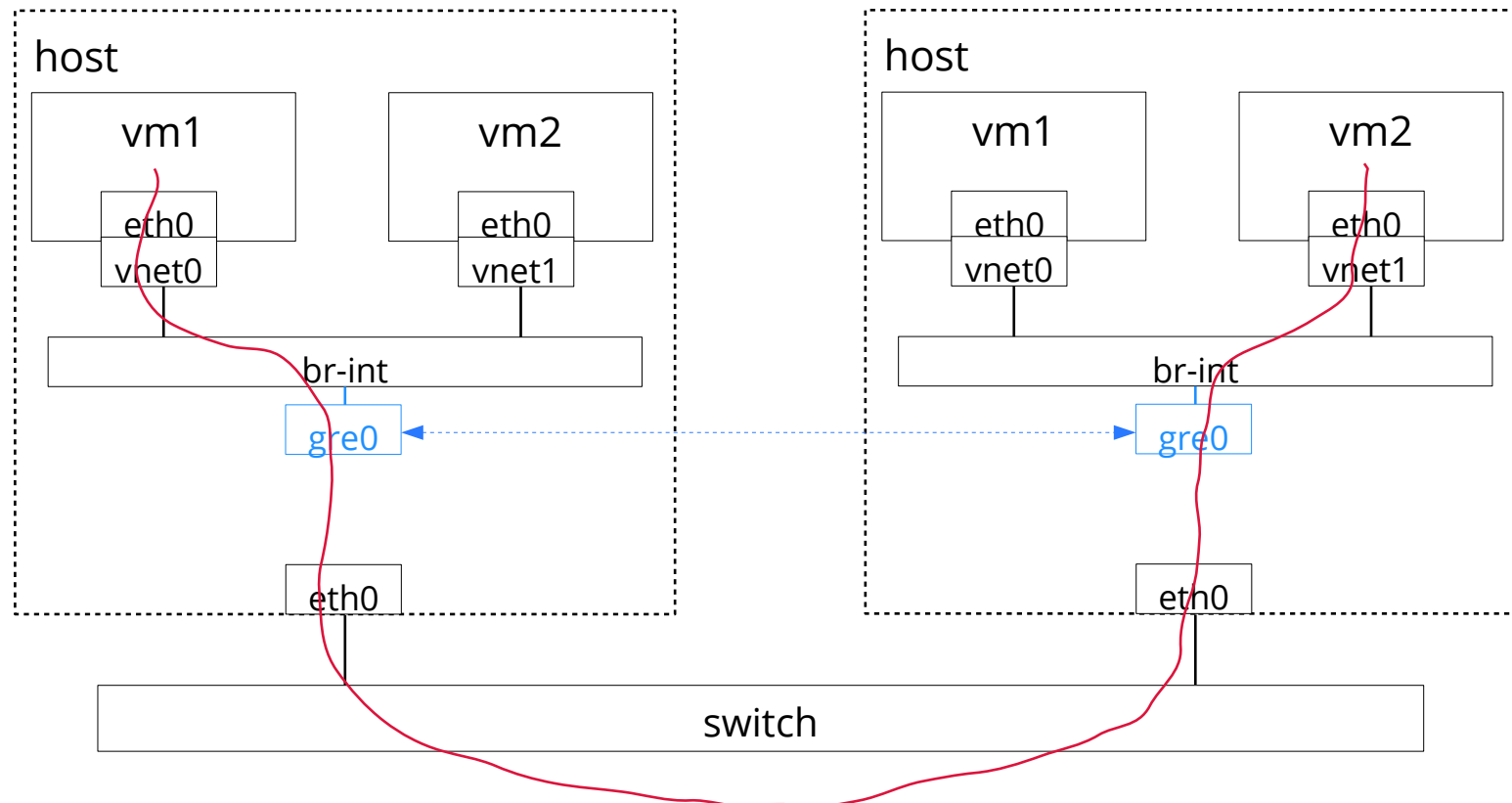
- Con máquinas virtuales en varios hosts a través de túneles



Infraestructura virtual

Virtualización de la red > Open vSwitch

- Con máquinas virtuales en varios hosts a través de túneles



Infraestructura virtual

Virtualización de la red > Open vSwitch

libvirt

- Al crear el dominio hay que definir la interfaz

```
<interface type='bridge'>  
  <source bridge='br-int' />  
  <virtualport type="openvswitch" />  
</interface>
```

- Hay que configurar la interfaz de red dentro del dominio

```
auto <iface>  
iface <iface> inet static  
address x.x.x.x  
netmask 255.255.255.0  
gateway x.x.x.x
```

Infraestructura virtual

Virtualización del almacenamiento

- A partir de todos los recursos físicos, se crea un pool lógico para:
 - Imágenes
 - Máquinas virtuales
 - Espacio de almacenamiento persistente adicional
- Puede proceder de de diversas fuentes:
 - De los nodos físicos
 - De soluciones de almacenamiento dedicadas (NAS, SAN, etc.)

Infraestructura virtual

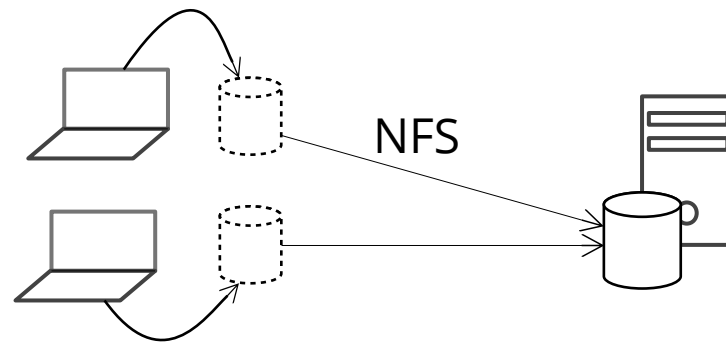
Virtualización del almacenamiento

- Distintos tipos
 - Almacenamiento tipo bloque: se monta como una unidad de disco
 - Almacenamiento de tipo fichero: directorio compartido

Infraestructura virtual

Virtualización del almacenamiento > NFS

- Network File System: protocolo de compartición de ficheros cliente-servidor
- Cliente accede a los ficheros como si fueran locales
- El servidor **exporta**, el cliente **monta**



Infraestructura virtual

Virtualización del almacenamiento > NFS

- Instalación

- > `sudo apt install nfs-kernel-server` (servidor)

- > `sudo apt install rpcbind nfs-common` (cliente)

- Servidor exporta en */etc/exports*

/dir client(rw,sync,all_squash,no_subtree_check)

- > `sudo chown nobody:nogroup /dir`

- > `sudo exportfs -a`

- > `sudo systemctl restart nfs-kernel-server`

Infraestructura virtual

Virtualización del almacenamiento > NFS

- Cliente monta

```
> mkdir dir-local
```

```
> sudo mount servidor:/dir /dir-local
```

```
> mount -t nfs
```

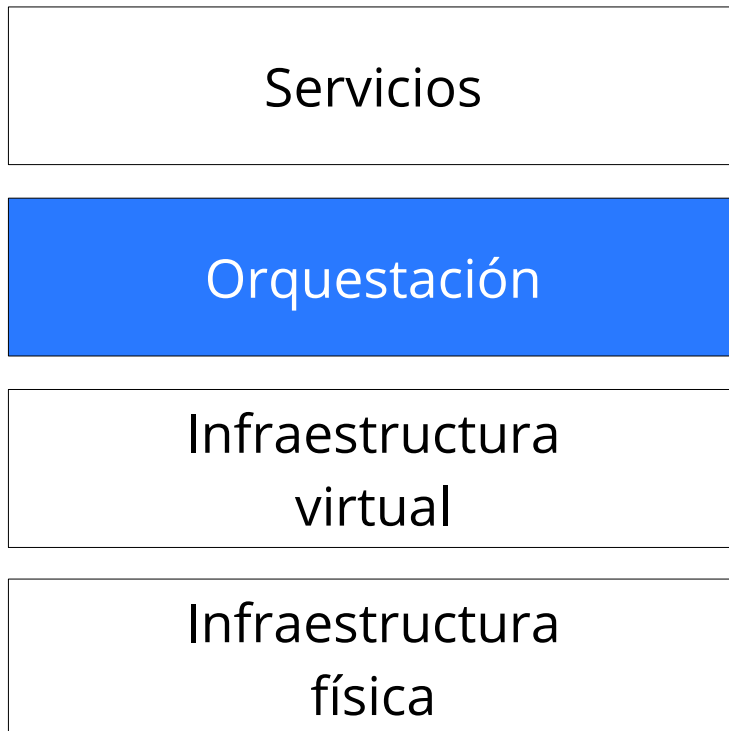
- Cliente desmonta

```
> sudo umount /dir-local
```

- Montar al reiniciar en */etc/fstab*

```
server:/dir /dir-local nfs
auto,noatime,nolock,bg,nfsvers=4,intr,tcp,actimeo=18
00 0 0
```

Orquestación

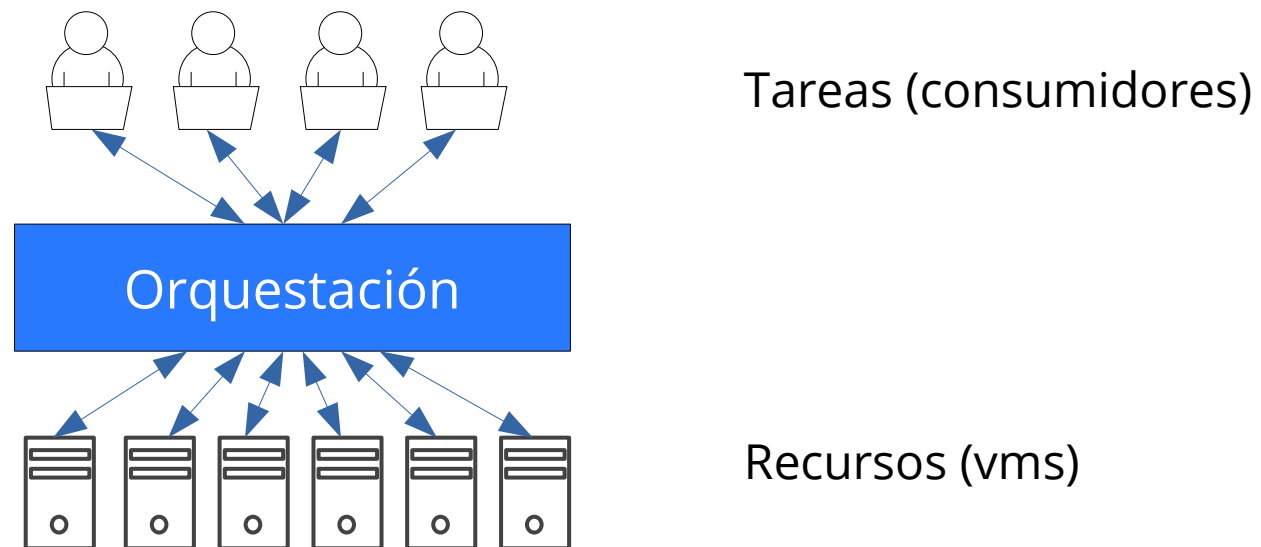


- ¿Qué es?
- Algoritmos de planificación

Orquestación

¿Qué es?

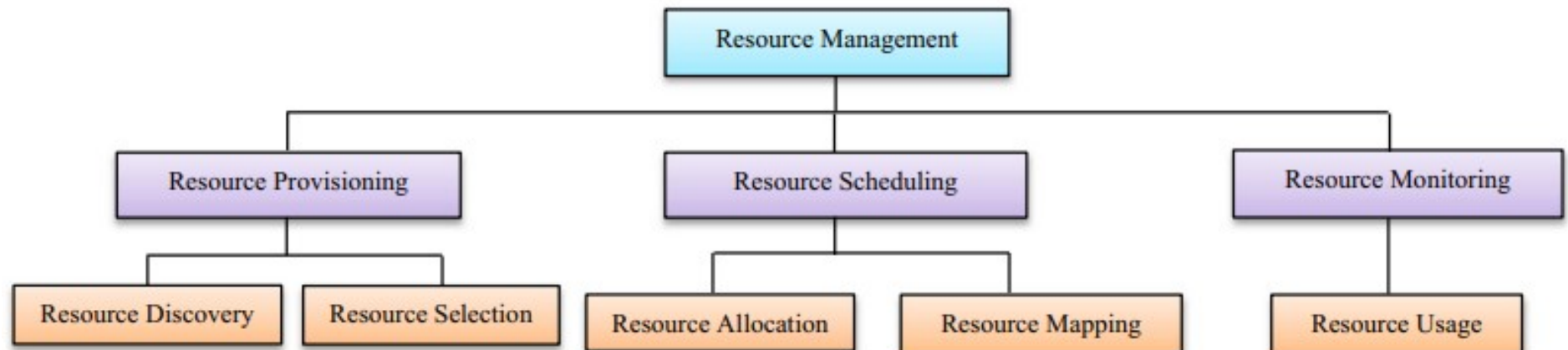
- Gestión **inteligente** de los recursos virtuales
- Proceso de asignación de recursos a un conjunto de tareas, en tiempo finito, para alcanzar la **calidad de servicio** deseada (SLAs) y optimizar los recursos



Orquestación

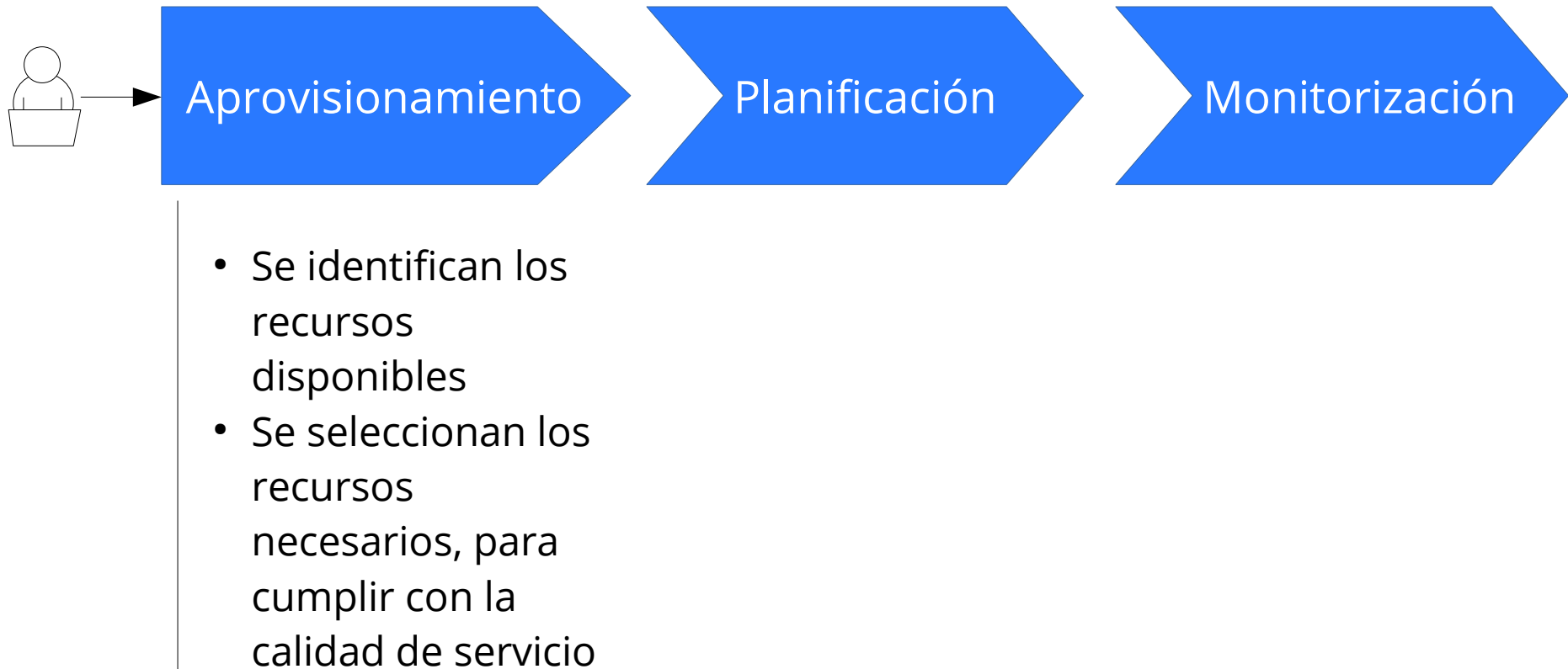
¿Qué es?

- En general, incluye tres subcomponentes principales: aprovisionamiento, planificación y monitorización

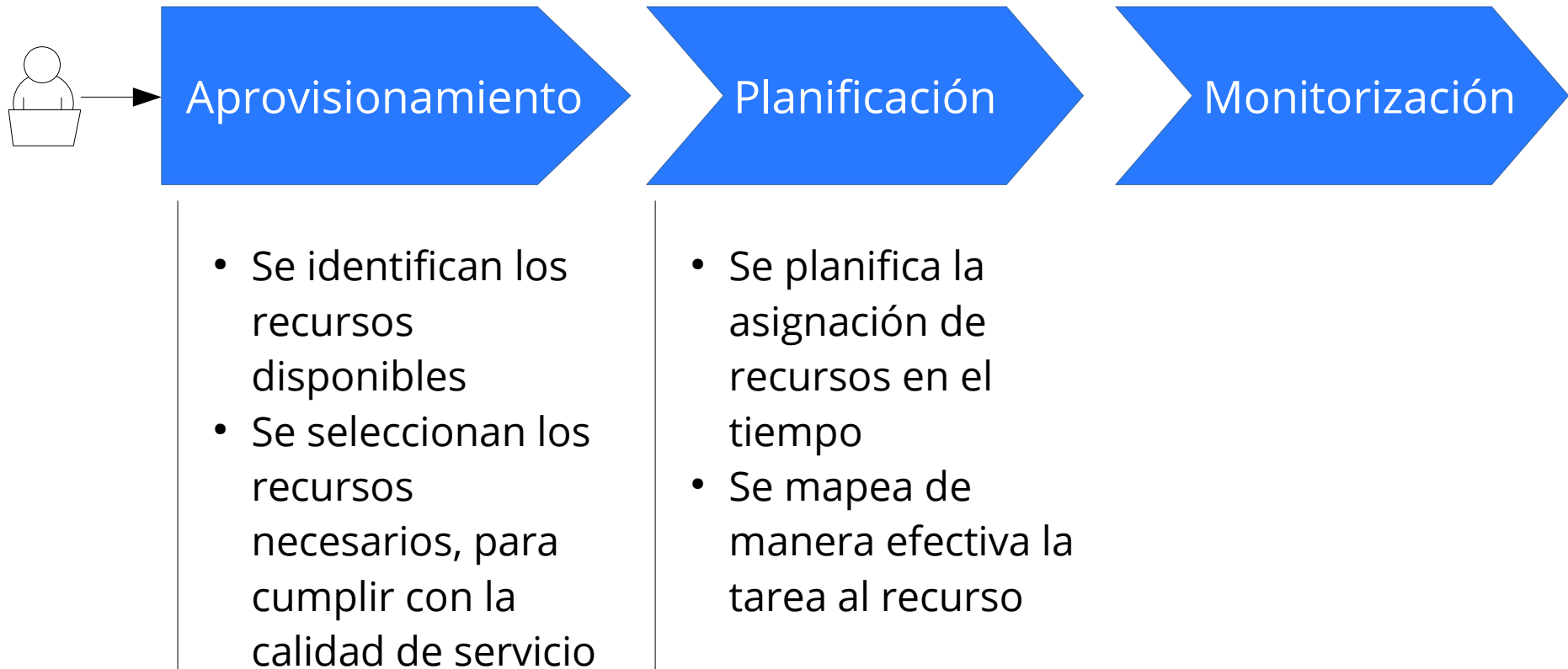


[sukhpal16]

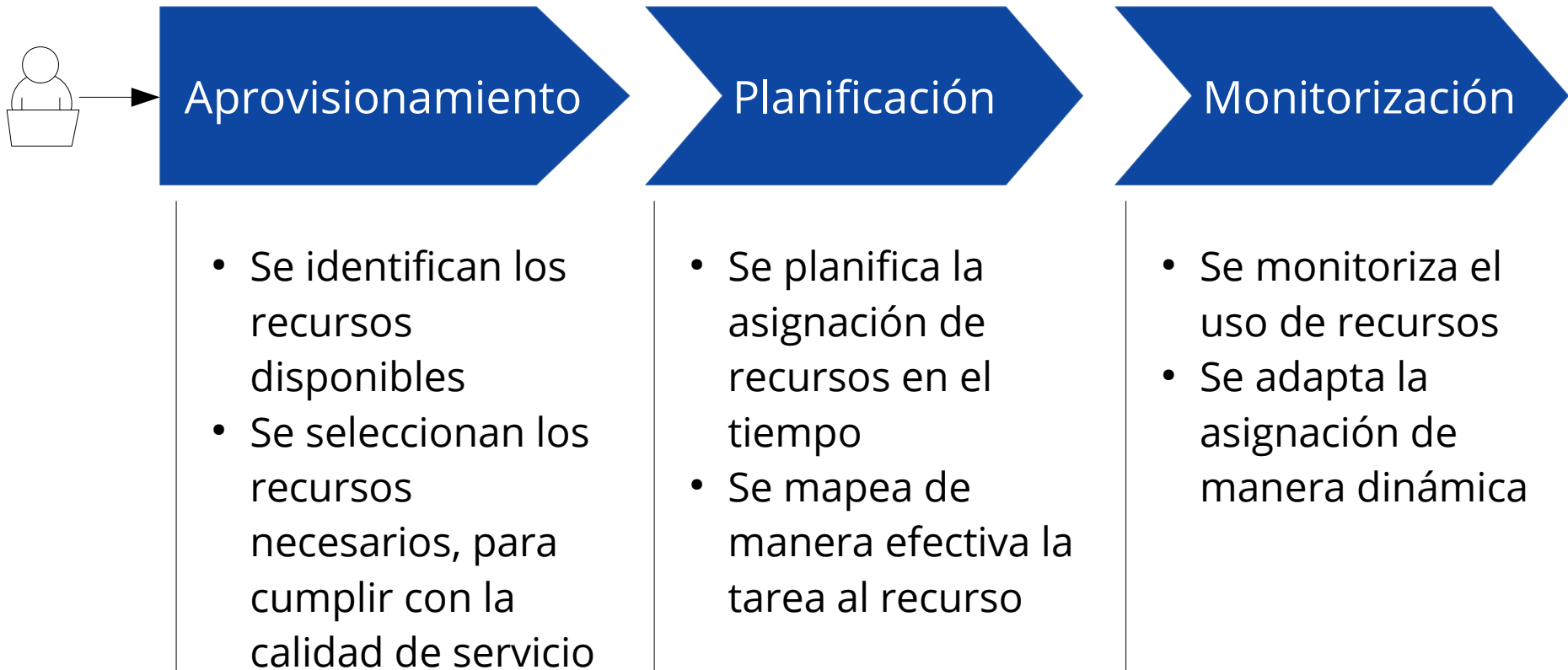
Orquestación



Orquestación



Orquestación



Orquestación

El problema de asignación de recursos

- Se dispone de un conjunto de tareas $T = \{t_i\}_{1 \leq i \leq m}$ que se desea mapear a un conjunto de recursos dinámicos y heterogéneos $R = \{r_j\}_{1 \leq j \leq n}$
- Se deben de cumplir los niveles de calidad de servicio (QoS) especificados en el SLA (tiempo respuesta, coste, tasa de fallos, makespan, productividad, etc.)
- Se desea obtener un máximo beneficio por parte del proveedor (coste, % utilización, energía, etc.)

Orquestación

El problema de asignación de recursos

- Es un problema de **optimización**
- Es un problema **NP-duro** que no se puede resolver en tiempo lineal
- No se desea obtener una solución exacta, sino una **aproximación** “lo suficientemente buena”, en menor tiempo

Orquestación

Estrategias de planificación

- **Estáticas:** tiene en cuenta un estado global anterior del sistema (First Come First Served (FCFS), Round Robin (RR), Min-Min, Max-Min, etc.)
- **Dinámicas:** tiene en cuenta el estado actual del sistema (Algoritmos genéticos, particle swarm optimization (PSO), ant colony optimization (ACO), etc.)
- En la práctica, se tiene en cuenta la historia para predecir, pero se planifica dinámicamente

Servicios



- ¿Qué es?
- Arquitecturas SOA
- Servicios RESTful

Azure



Azure

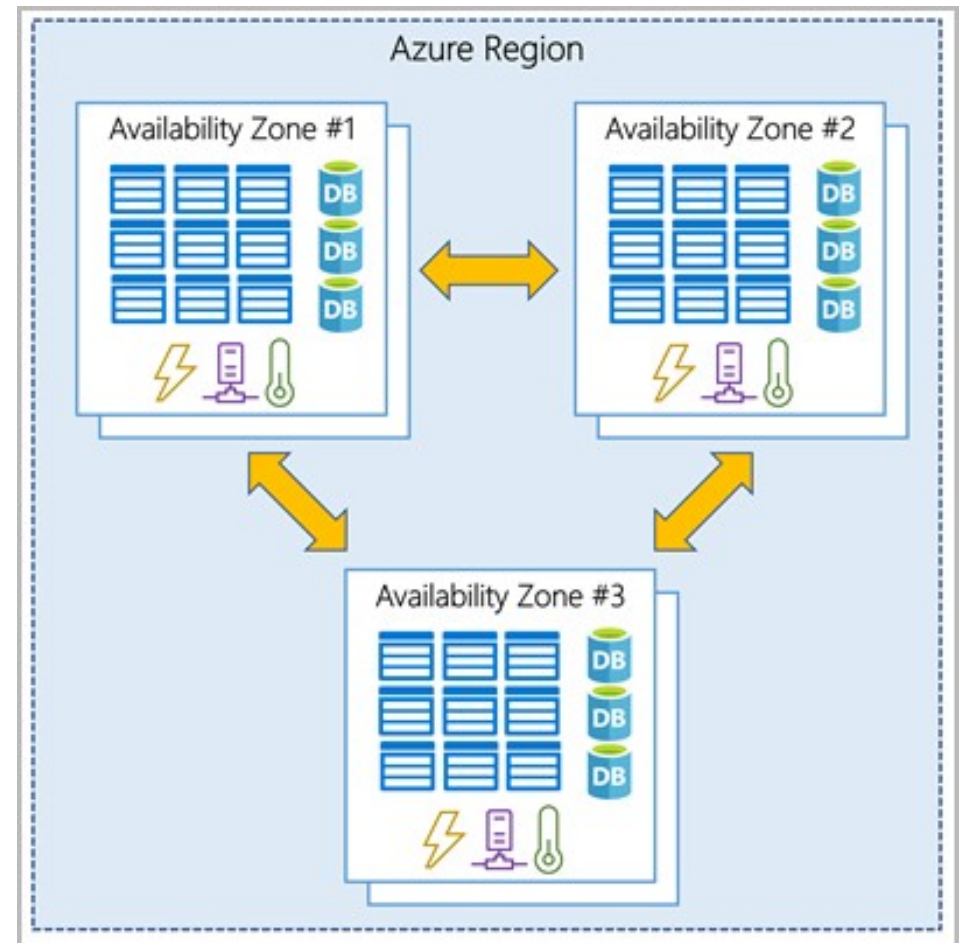
¿Qué es?

- <https://azure.microsoft.com>
- Proveedor cloud público ofrecido por Microsoft, con más de 200 productos y servicios
- Proporciona servicios IaaS, PaaS y SaaS
- Muchas [categorías](#): AI + machine learning, Analytics, Compute, Containers, Databases, Developer tools, DevOps, Hybrid + multicloud, Identity, Integration, Internet of Things, Management and governance, Media, Migration, Mixed reality, Mobile, Networking, Security, Storage, Virtual desktop infrastructure, Web

Azure

Infraestructura física

- Centros de datos por todo el mundo
- Se agrupan en regiones
- Se agrupan en zonas de disponibilidad para garantizar resiliencia
- Explorar la infraestructura física [aquí](#)



Azure

Puesta en marcha

- Primero hay que crearse una [cuenta](#), con 200\$ de crédito para consumir en el primer mes
- Los estudiantes pueden obtener 100\$ de crédito para consumir durante un año
- Además, hay servicios que son [gratuitos](#). Algunos sólo los primeros 12 meses. Otros para siempre
- Se puede acceder a Azure a través de Azure Portal, Azure CLI, Azure API, librerías
- <https://portal.azure.com>

Azure

Infraestructura de gestión

- Todo lo que se crea, aprovisiona, despliega, etc. es un recurso. El recurso es la pieza básica
- Para crear y usar recursos, primero necesitamos una suscripción. Podemos tener cualquier número de suscripciones (personal, empresa, departamento, etc.)
- Los recursos se organizan en grupos. Una operación efectuada en un grupo se aplica a todos sus miembros
- Los recursos se gestionan por proveedores de recursos. Para usar un recurso tenemos que registrar la suscripción en el recurso (en general es automático)

Azure

Creación de máquinas virtuales

- A partir de una imagen (marketplace)
- Tamaño de máquina virtual
- Autenticación (SSH vs password)
- Visible desde fuera o no
- Discos
- Redes (redes virtuales, IP pública)
- Balanceo de carga



The End