

Sistemas Operativos Distribuidos

30221 - Sistemas Distribuidos

Unai Arronategui - Rafael Tolosana

Dpto. Informática e Ing. de Sistemas

Lectura Recomendada

- Coulouris, Dollimore, Kindberg and Blair, Distributed Systems: Concepts and Design Edn. 5, 2012. Capítulo 7
- Tanenbaum and Van Steen, Distributed Systems: Principles and Paradigms, 3e, (c) 2017 Prentice- Hall. Capitulo 3

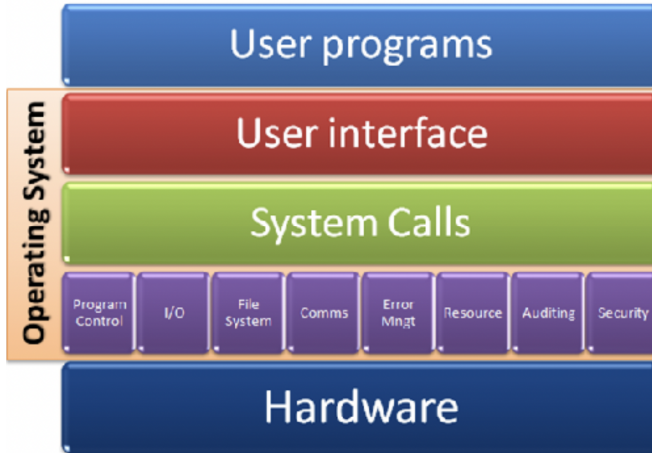
Motivación

Motivación

¿Qué es un Sistema Operativo?

Motivación

¿Qué es un Sistema Operativo?



Motivación

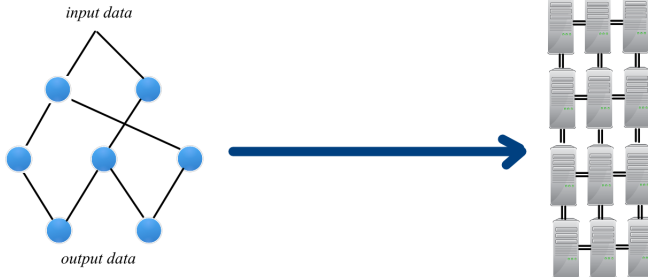
¿Qué es un Sistema Operativo?

Un sistema operativo:

- es un gestor de recursos
 - CPU, memoria, sistema de ficheros, red, ratón, teclado...
- proporciona una interfaz abstracta (API)
- gestiona el acceso concurrente a los recursos: múltiples procesos

Motivación

Problema Fundamental de un SD



Sistemas Operativos Distribuidos

Sistemas Operativos Distribuidos

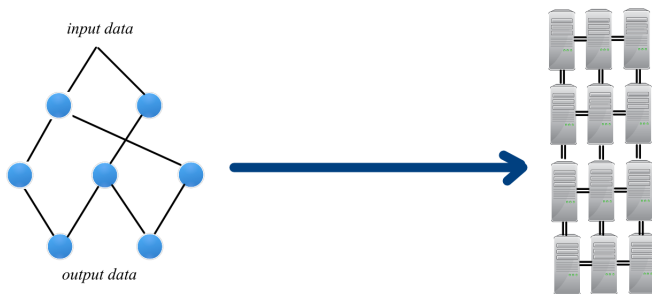
¿Qué es un Sistema Operativo Distribuido?

¹Tanenbaum, Andrew S (September 1993). "Distributed operating systems anno 1992. What have we learned so far?". Distributed Systems Engineering. 1 (1): 3–10.

Sistemas Operativos Distribuidos

¿Qué es un Sistema Operativo Distribuido?

A distributed operating system is system software over a collection of *independent software, networked*, communicating, and physically separate computational nodes. They handle jobs which are serviced by multiple CPUs. ¹



¹Tanenbaum, Andrew S (September 1993). "Distributed operating systems anno 1992. What have we learned so far?". Distributed Systems Engineering. 1 (1): 3–10.

Sistemas Operativos Distribuidos

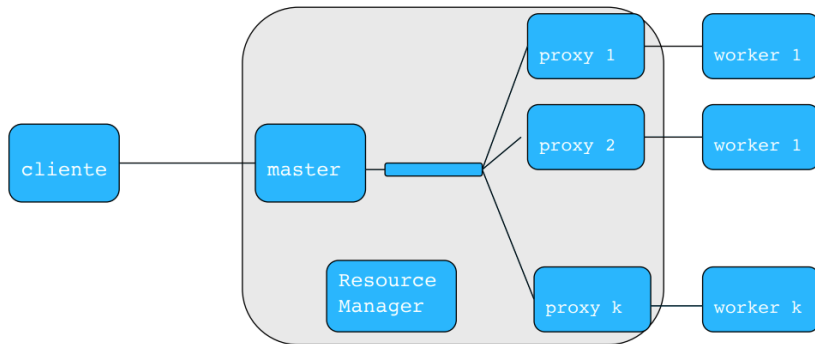
A veces las aplicaciones distribuidas utilizan servicios:

- Nombres (registro RMI, DNS,...)
- Persistencia (Ficheros distribuidos -NFS, Ceph, GlusterFS)
- Seguridad (Kerberos, Infraestructuras de Clave Pública,...)
- Registro y búsqueda de entidades distribuidas (registro RMI, DNS,...)
- Proxies con balanceadores de carga
- Gestión de despliegue de aplicaciones distribuidas
- Gestión de eventos y notificaciones distribuidas
- Control de concurrencia
- Detección de bloqueos
- Gestión de transacciones

Sistemas Operativos Distribuidos

Arquitectura Máster-Worker

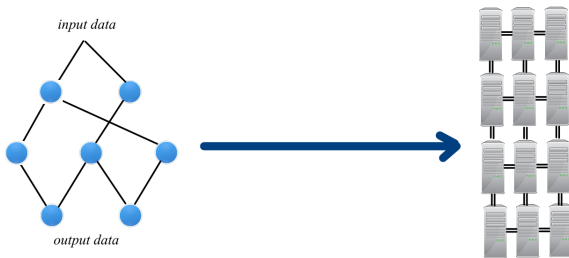
¿Qué **retos** nos encontramos?



Modelo de Aplicación

Retos para un SSDD

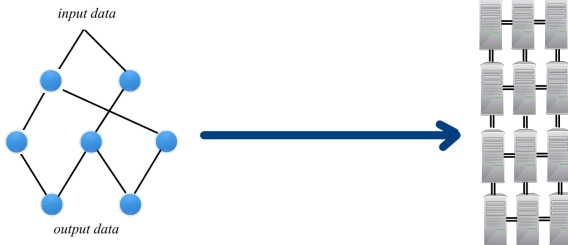
¿Cómo es la **aplicación**?



Modelo de Aplicación

Retos para un SSDD

¿Cómo es la **aplicación**?

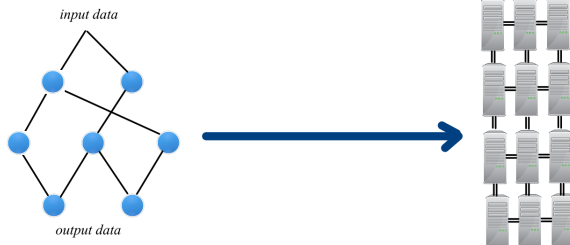


- monolítica: 1 tarea, v.gr. findPrimes

© 2004 Blackwell Publishing Ltd, *Journal of Internal Medicine* 255: 115–122

¿Cómo es la **aplicación**?

¿Cómo es la **aplicación**?

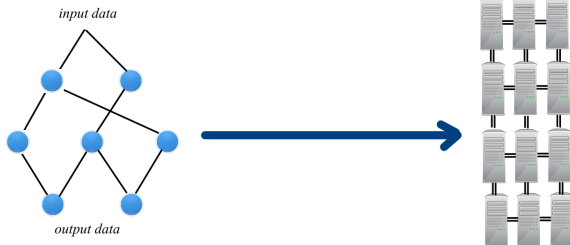


- monolítica: 1 tarea, v.gr. findPrimes
- secuencia de tareas, v.gr. programa secuencial

Modelo de Aplicación

Retos para un SSDD

¿Cómo es la **aplicación**?



- monolítica: 1 tarea, v.gr. findPrimes
- secuencia de tareas, v.gr. programa secuencial
- grafo

Despliegue (Deployment) del software

Software Deployment (Definición IBM)

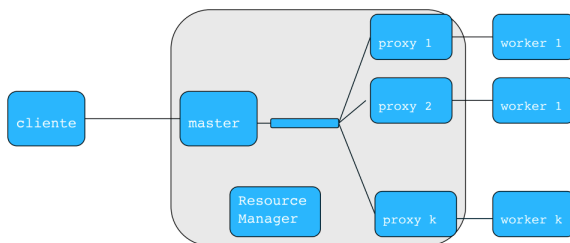
- Software deployment is the process of making software available to be used on a system by users and other programs ²
- making software available — > llevar fuentes, configurar, bibliotecas...

²<https://www.ibm.com/docs/en/zos/2.4.0?topic=task-deploying-software>

Despliegue (Deployment) del software

Arquitectura Máster-Worker

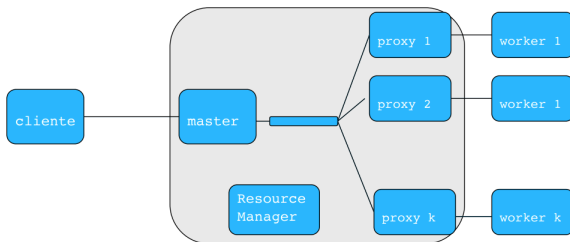
¿Qué características / requisitos tiene el software, v.gr. findPrimes?



Despliegue (Deployment) del software

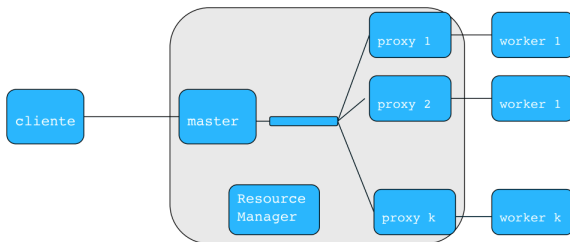
Arquitectura Máster-Worker

¿Qué características / requisitos tiene el software, v.gr. findPrimes?



- Sistema Operativo: cada tarea puede requerir un sistema operativo

¿Qué características / requisitos tiene el software, v.gr. findPrimes?

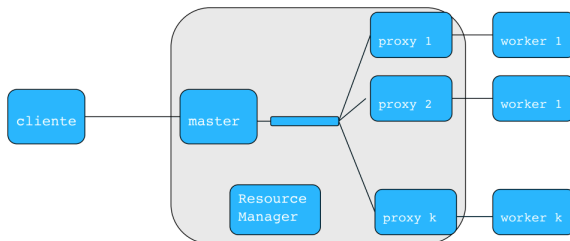


- Sistema Operativo: cada tarea puede requerir un sistema operativo
- Bibliotecas: necesarias para efectuar una tarea

Despliegue (Deployment) del software

Arquitectura Máster-Worker

¿Qué características / requisitos tiene el software, v.gr. findPrimes?



- Sistema Operativo: cada tarea puede requerir un sistema operativo
- Bibliotecas: necesarias para ejecutar una tarea
- Compilador

Gestión de Configuraciones

³Roger S. Pressman (2009). Software Engineering: A Practitioner's Approach (7th International ed.). New York: McGraw-Hill.

Gestión de Configuraciones

In software engineering, **software configuration management** (SCM) is the task of tracking and controlling changes in the software, part of the larger cross-disciplinary field of configuration management.[1] SCM practices include revision control and the establishment of baselines. If something goes wrong, SCM can determine the "what, when, why and who" of the change. If a configuration is working well, SCM can determine how to replicate it across many hosts. ³

³Roger S. Pressman (2009). Software Engineering: A Practitioner's Approach (7th International ed.). New York: McGraw-Hill.

Gestión de Configuraciones

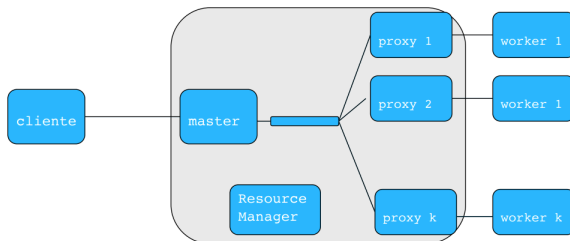
In software engineering, **software configuration management** (SCM) is the task of tracking and controlling changes in the software, part of the larger cross-disciplinary field of configuration management.[1] SCM practices include revision control and the establishment of baselines. If something goes wrong, SCM can determine the "what, when, why and who" of the change. If a configuration is working well, SCM can determine how to replicate it across many hosts. ³

Herramienta actual: **github**

³Roger S. Pressman (2009). Software Engineering: A Practitioner's Approach (7th International ed.). New York: McGraw-Hill.

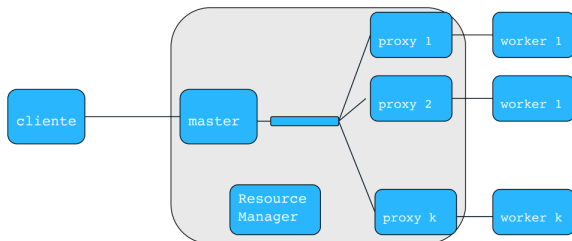
Asignación de recursos

Scheduling (planificación)



Asignación de recursos

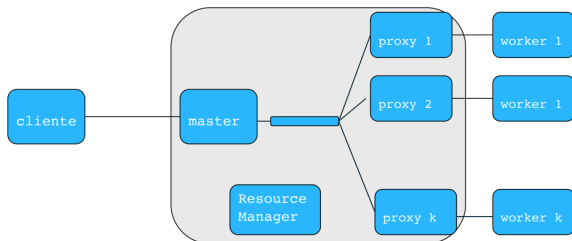
Scheduling (planificación)



- Requisitos tareas — > recursos disponibles

Asignación de recursos

Scheduling (planificación)



- Requisitos tareas — $>$ recursos disponibles
- Compartir recursos: maximizar el uso (%CPU) de los recursos, garantizando el QoS

Asignación de recursos

¿Cómo pueden afrontarse estos retos?

Cloud computing

Cloud computing

Definición de Cloud NIST ⁴

- *Cloud computing* is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be *rapidly* provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models.

⁴<https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-145.pdf>

Cloud computing

Definición de Cloud NIST ⁵

- *Cloud computing* is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be *rapidly* provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of **five essential characteristics**, three service models, and four deployment models.

⁵<https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-145.pdf>

Cloud computing

Características esenciales (NIST)

- On-Demand self-service
 - Los usuarios pueden provisionar recursos hardware

Cloud computing

Características esenciales (NIST)

- On-Demand self-service
 - Los usuarios pueden provisionar recursos hardware
- Broad network access
 - Acceso ubicuo desde cualquier dispositivo móvil

Cloud computing

Características esenciales (NIST)

- On-Demand self-service
 - Los usuarios pueden provisionar recursos hardware
- Broad network access
 - Acceso ubicuo desde cualquier dispositivo móvil
- Resource pooling
 - Los proveedores ofrecen recursos transparentemente a múltiples clientes simultáneamente

Cloud computing

Características esenciales (NIST)

- On-Demand self-service
 - Los usuarios pueden provisionar recursos hardware
- Broad network access
 - Acceso ubicuo desde cualquier dispositivo móvil
- Resource pooling
 - Los proveedores ofrecen recursos transparentemente a múltiples clientes simultáneamente
- Measured service
 - El uso de los recursos computacionales / servicios se mide

Cloud computing

Características esenciales (NIST)

- On-Demand self-service
 - Los usuarios pueden provisionar recursos hardware
- Broad network access
 - Acceso ubicuo desde cualquier dispositivo móvil
- Resource pooling
 - Los proveedores ofrecen recursos transparentemente a múltiples clientes simultáneamente
- Measured service
 - El uso de los recursos computacionales / servicios se mide
- Rapid elasticity

Elasticidad

Definición: Elasticidad

- La *elasticidad* es la capacidad de un sistema de adaptarse a los cambios de la carga de trabajo mediante el aprovisionamiento y desaprovisionamiento de recursos de manera automática, de modo que en cada momento los recursos disponibles coincidan lo más posible con la demanda actual. ⁶

⁶Elasticity in Cloud Computing: What It Is, and What It Is Not. Herbst, Kounev, Reussner: ICAC 2013
<https://sdqweb.ipd.kit.edu/publications/pdfs/HeKoRe2013-ICAC-Elasticity.pdf>

Elasticidad

Definición: Elasticidad

- La *elasticidad* es la capacidad de un sistema de adaptarse a los cambios de la carga de trabajo mediante el aprovisionamiento y desaprovisionamiento de recursos de manera automática, de modo que en cada momento los recursos disponibles coincidan lo más posible con la demanda actual. ⁶

Pero puede haber variabilidad en:

- Carga de trabajo

⁶Elasticity in Cloud Computing: What It Is, and What It Is Not. Herbst, Kounev, Reussner: ICAC 2013
<https://sdqweb.ipd.kit.edu/publications/pdfs/HeKoRe2013-ICAC-Elasticity.pdf>

Elasticidad

Definición: Elasticidad

- La *elasticidad* es la capacidad de un sistema de adaptarse a los cambios de la carga de trabajo mediante el aprovisionamiento y desaprovisionamiento de recursos de manera automática, de modo que en cada momento los recursos disponibles coincidan lo más posible con la demanda actual.⁶

Pero puede haber variabilidad en:

- Carga de trabajo
- Recursos computacionales

⁶Elasticity in Cloud Computing: What It Is, and What It Is Not. Herbst, Kounev, Reussner: ICAC 2013
<https://sdqweb.ipd.kit.edu/publications/pdfs/HeKoRe2013-ICAC-Elasticity.pdf>

Elasticidad

Definición: Carga de Trabajo

- Las operaciones que tiene que realizar un sistema computacional sobre unos datos por unidad de tiempo

Elasticidad

Definición: Carga de Trabajo

- Las operaciones que tiene que realizar un sistema computacional sobre unos datos por unidad de tiempo
- Las operaciones involucran **tres tipos** de **recursos**:
 - CPU, almacenamiento y red de comunicación

Elasticidad

Definición: Carga de Trabajo

- Las operaciones que tiene que realizar un sistema computacional sobre unos datos por unidad de tiempo
- Las operaciones involucran **tres tipos** de **recursos**:
 - CPU, almacenamiento y red de comunicación
- Puede ser
 - De frecuencia (tiempo entre operaciones) **regular** o **variable**
 - **Predecible** o **impredecible** (aleatoria)
 - **Batch** (respuesta con holgura) vs **Online** (respuesta inmediata)

Elasticidad

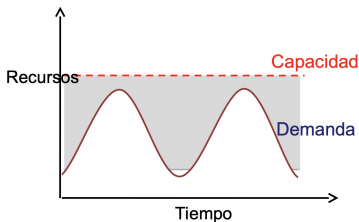
Escalabilidad Vertical

- Incrementar / Decrementar la capacidad de los Recursos
 - Por ejemplo, se añade una CPU adicional a una máquina; se añade más memoria, etc.

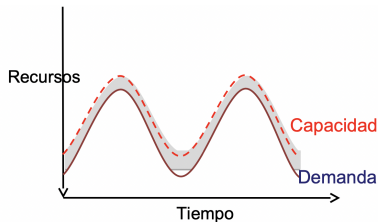
Escalabilidad Horizontal

- Incrementar / Decrementar el número de Recursos (máquinas físicas, virtuales, contenedores)

Elasticidad

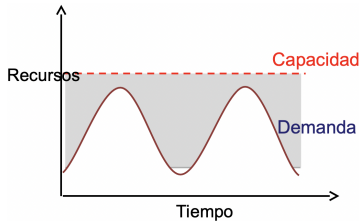


Centro de datos
“provisto estáticamente”

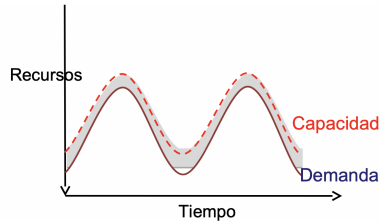


Centro de datos virtual en Cloud

Elasticidad



Centro de datos
“provisto estáticamente”

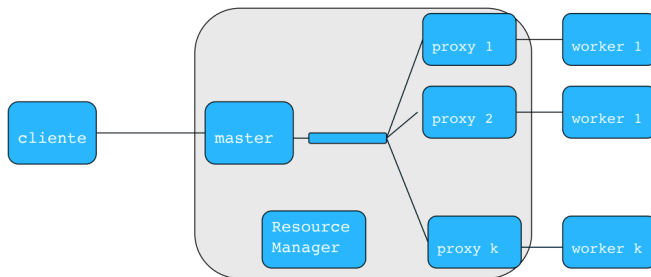


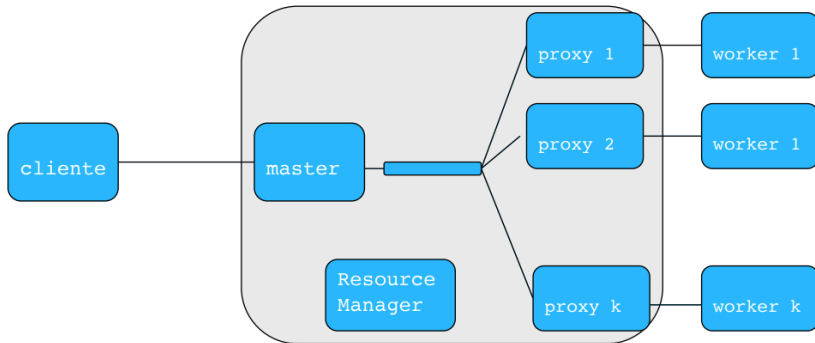
Centro de datos virtual en Cloud

- Aprovisionar más recursos de los necesarios es ineficiente

Elasticidad

Elasticidad en el Máster Worker de la Práctica 1





Componente Costar de Recursos (Resource Manager)

Elasticidad

Ciclo de Vida del Componente Gestor de Recursos: MAPE

- 1 Monitorización
- 2 Análisis
- 3 Planificación
- 4 Ejecución

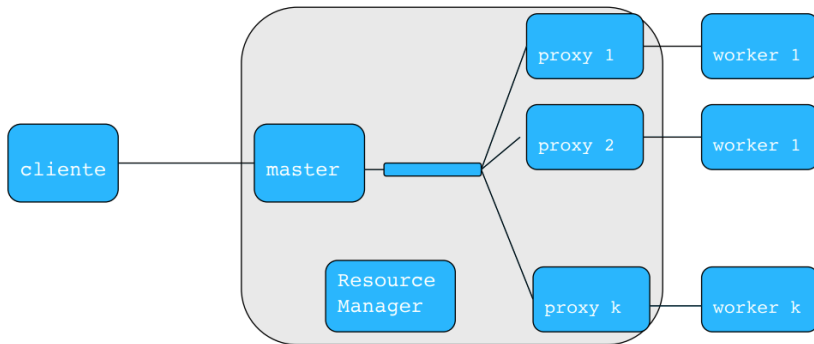
Elasticidad

Monitorización

- Se trata de obtener **métricas** del sistema para saber cuáles son las prestaciones, el QoS y la carga de trabajo
- Es deseable que tenga **poco overhead** y que tenga información de **precisión**
- Muchas veces tiene que realizarse en **tiempo real**: compromiso entre métricas fiables, precisas, en tiempo real
- El **tamaño** de las métricas de monitorización puede ser enorme

Elasticidad

Arquitectura Máster Worker



Monitorización

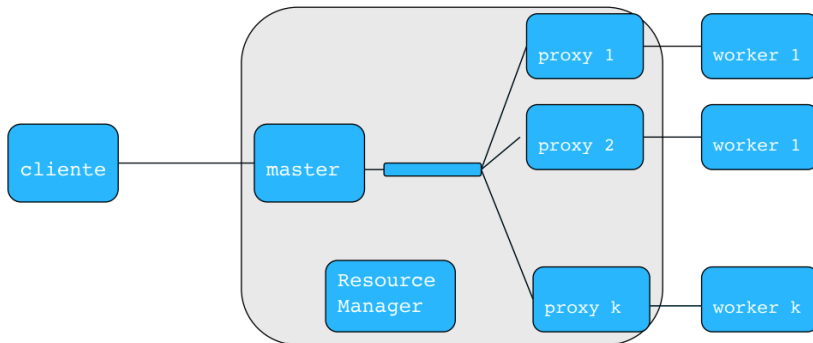
Elasticidad

Análisis

- Se trata de analizar cómo de lejos está el sistema del **objetivo**
- ¿Qué **objetivos** se pueden tener?
 - Time deadline, Throughput, Energía, Coste económico
- ¿Cómo puede ser el **análisis**?
 - Reactivo: ¿**es** la capacidad computacional adecuada?
 - Predictivo: ¿**será** la capacidad computacional adecuada?

Elasticidad

Arquitectura Máster Worker



Análisis

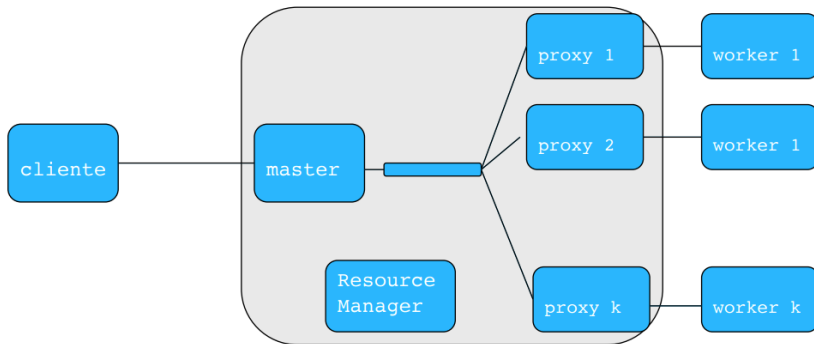
Elasticidad

Planificación

- **Acción** o **Acciones** para alcanzar el objetivo de forma rápida
- Si la acción genera un efecto demasiado fuerte, puede provocar oscilaciones e inestabilidad
- Si la acción genera un efecto demasiado débil, tardará mucho en revertirse el problema

Elasticidad

Arquitectura Máster Worker



Planificación

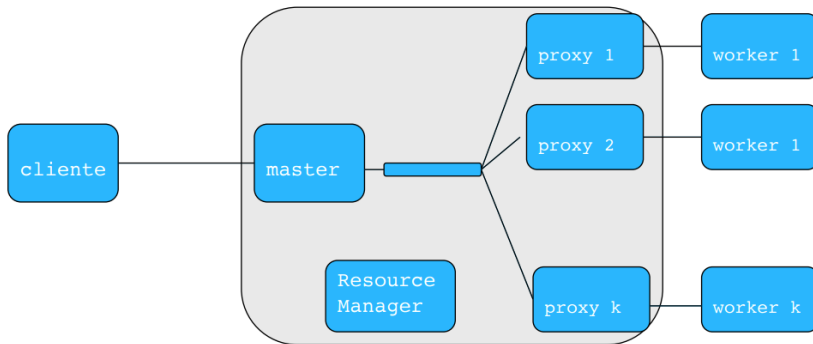
Elasticidad

Ejecución

- La fase de ejecución corresponde al desencadenamiento de las acciones planificadas para conseguir el objetivo
- Estos sistemas tienen **inercia**:
 - Inercia es la tendencia que muestra un sistema a permanecer sin cambio durante un tiempo, una vez que se han lanzado las acciones.
 - Arrancar / quitar máquinas no es instantáneo
 - El tiempo de procesamiento, si añadido capacidad computacional, se verá reflejado una vez haya terminado de procesarse
- Un controlador reactivo *sufrirá* la inercia:
 - ¿Cuánto tiempo hay que esperar a que surta efecto las acciones ejecutadas?

Elasticidad

Arquitectura Máster Worker



Ejecución

Elasticidad

Controlador Reactivo

- Acciones exclusivamente en función de la observación de la entrada

Claves

- ¿Cada cuánto se ejecuta el bucle MAPE?
- ¿Cuánto tiempo esperar para analizar si una acción ha surtido efecto?
- ¿Cómo detectar recursos no utilizados?

Modelos de servicio

Definición de Cloud NIST ⁷

- *Cloud computing* is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be *rapidly* provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, **three service models**, and four deployment models.

⁷<https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-145.pdf>

Modelos de servicio

Software as a Service (SaaS)

Platform as a Service (PaaS)

Infrastructure as a Service (IaaS)

Modelos de servicio

Software as a Service (SaaS) ⁸

- La capacidad que se ofrece al consumidor es la de **utilizar las aplicaciones del proveedor** que se ejecutan en una infraestructura en la nube.

⁸<https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-145.pdf>

Modelos de servicio

Software as a Service (SaaS) ⁸

- La capacidad que se ofrece al consumidor es la de **utilizar las aplicaciones del proveedor** que se ejecutan en una infraestructura en la nube.
- Las aplicaciones son accesibles desde varios dispositivos cliente a través de una interfaz de cliente ligero, como un navegador web.

⁸<https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-145.pdf>

Modelos de servicio

Software as a Service (SaaS) ⁸

- La capacidad que se ofrece al consumidor es la de **utilizar las aplicaciones del proveedor** que se ejecutan en una infraestructura en la nube.
- Las aplicaciones son accesibles desde varios dispositivos cliente a través de una interfaz de cliente ligero, como un navegador web.
- El consumidor no gestiona ni controla la infraestructura subyacente de la nube, incluyendo la red, los servidores, los sistemas operativos, el almacenamiento o incluso las capacidades individuales de la aplicación, con la posible excepción de los ajustes de configuración de la aplicación específicos del usuario.

⁸<https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-145.pdf>

Modelos de servicio

Platform as a Service (PaaS) ⁹

- La capacidad que se proporciona al consumidor es la de **desplegar en la infraestructura** de la nube aplicaciones creadas o adquiridas por el consumidor y creadas utilizando lenguajes de programación, bibliotecas, servicios y herramientas soportadas por el proveedor.
- El consumidor no gestiona ni controla la infraestructura subyacente de la nube, incluyendo la red, los servidores, los sistemas operativos o el almacenamiento, pero sí controla las aplicaciones desplegadas y, posiblemente, los ajustes de configuración del alojamiento de aplicaciones.

⁹<https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-145.pdf>

Modelos de servicio

Infrastructure as a Service (IaaS) ¹⁰

- La capacidad que se proporciona al consumidor es la de **suministrar procesamiento, almacenamiento, redes** y otros recursos informáticos fundamentales en los que el consumidor es capaz de desplegar y ejecutar software arbitrario, que puede incluir sistemas operativos y aplicaciones.
- El consumidor no gestiona ni controla la infraestructura subyacente de la nube, pero tiene control sobre los sistemas operativos, el almacenamiento y las aplicaciones desplegadas; y posiblemente un control limitado de determinados componentes de red (por ejemplo, cortafuegos de host).

¹⁰<https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-145.pdf>

Modelos de despliegue

Definición de Cloud NIST ¹¹

- *Cloud computing* is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be *rapidly* provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and **four deployment models**.

¹¹<https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-145.pdf>

Modelos de despliegue

Modelos de Despliegue

- Cloud privado
- Cloud comunitario
- Cloud público
- Cloud híbrido

Modelos de despliegue

Modelos de Despliegue: Cloud privado¹²

- La infraestructura de la nube se aprovisiona **para el uso exclusivo de una sola organización** que comprende múltiples consumidores (por ejemplo, unidades de negocio).
- Puede ser propiedad, estar gestionada y la organización, un tercero, o una combinación de ellos, y puede existir dentro o fuera de las instalaciones.

¹²<https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-145.pdf>

Modelos de despliegue

Modelos de Despliegue: Cloud comunitario

- La infraestructura de la nube se suministra para el **uso exclusivo de una comunidad** específica de consumidores de organizaciones que tienen intereses compartidos (por ejemplo, la misión).
- Puede ser propiedad, gestionada y operada por una o más organizaciones de la comunidad, por un tercero o por una combinación de ellas.

Modelos de despliegue

Modelos de Despliegue: Cloud público

- La infraestructura de la nube está provista **para su uso abierto por el público en general**.
- Puede ser ser propiedad, gestionada y operada por una organización empresarial, académica o gubernamental, o alguna combinación de ellas.

Modelos de despliegue

Modelos de Despliegue: Cloud híbrido

- La infraestructura de nube es una **composición de dos o más** infraestructuras de nube distintas (privadas, comunitarias o públicas) que siguen siendo entidades únicas, pero que están unidas tecnología estandarizada o propietaria que permite la portabilidad de datos y aplicaciones portabilidad de datos y aplicaciones.

Modelos de despliegue

¿Cómo son los centros de datos de la nube?

Modelos de despliegue

¿Cómo son los centros de datos de la nube?

- Construcción de centros de datos muy grandes (10.000 a 100.000 ordenadores corrientes)
 - Economías de escala: 5 a 10 veces más barato que un centro de datos de 1000 máquinas.

Modelos de despliegue

¿Cómo son los centros de datos de la nube?

- Construcción de centros de datos muy grandes (10.000 a 100.000 ordenadores corrientes)
 - Economías de escala: 5 a 10 veces más barato que un centro de datos de 1000 máquinas.
 - En Google, por ejemplo, observaron que la mayor parte de los fallos se producía por el software, no por el hardware, ¿para qué comprar hardware caro y sofisticado?

Modelos de despliegue

¿Cómo son los centros de datos de la nube?

- Construcción de centros de datos muy grandes (10.000 a 100.000 ordenadores corrientes)
 - Economías de escala: 5 a 10 veces más barato que un centro de datos de 1000 máquinas.
 - En Google, por ejemplo, observaron que la mayor parte de los fallos se producía por el software, no por el hardware, ¿para qué comprar hardware caro y sofisticado?
 - Interferencia hardware en prestaciones: dirigido por la demanda de los usuarios

Modelos de despliegue

¿Cómo son los centros de datos de la nube?

- Construcción de centros de datos muy grandes (10.000 a 100.000 ordenadores corrientes)
 - Economías de escala: 5 a 10 veces más barato que un centro de datos de 1000 máquinas.
 - En Google, por ejemplo, observaron que la mayor parte de los fallos se producía por el software, no por el hardware, ¿para qué comprar hardware caro y sofisticado?
 - Interferencia hardware en prestaciones: dirigido por la demanda de los usuarios
 - Virtualización rápida

Modelos de despliegue

¿Cuál es la tecnología que permite desarrollar cloud computing?

Modelos de despliegue

¿Cuál es la tecnología que permite desarrollar cloud computing?

Virtualización

Virtualización

- La *virtualización* es el proceso de ejecución de una instancia virtual (no física) de un recurso computacional (CPU, almacenamiento o red) en una capa que se ha abstraído del hardware subyacente.

Virtualización

- La *virtualización* es el proceso de ejecución de una instancia virtual (no física) de un recurso computacional (CPU, almacenamiento o red) en una capa que se ha abstraído del hardware subyacente.
- La virtualización ya se utilizó en los años 60, para dividir los recursos proporcionados por un mainframe, pero recientemente los avances en la tecnología han posibilitado el desarrollo del cloud.

Virtualización

Tipos de virtualización

- Virtualización completa

Virtualización

Tipos de virtualización

- **Virtualización completa**
- Una capa adicional (hypervisor) provee servicio multiplexado total del hardware real (CPU; memoria, disco, red,..) entre S.Os. completamente *diferentes*, pero se ejecutan las instrucciones nativas del procesador (salvo las excepciones e interrupciones).
- Ejemplos : KVM, VirtualBox, Xen, Vmware...

Virtualización

Tipos de virtualización

- **Virtualización completa**
- Una capa adicional (hypervisor) provee servicio multiplexado total del hardware real (CPU; memoria, disco, red,..) entre S.Os. completamente *diferentes*, pero se ejecutan las instrucciones nativas del procesador (salvo las excepciones e interrupciones).
- Ejemplos : KVM, VirtualBox, Xen, Vmware...
- Normalmente se necesita instrucciones máquina en procesador físico para acelerar ejecución de excepciones e interrupciones. AMD64 y ARM64 disponen de ellas.

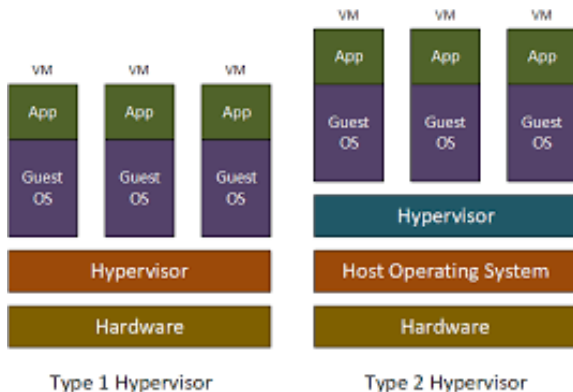
Virtualización

Tipos de virtualización

- **Virtualización completa**
- Una capa adicional (hypervisor) provee servicio multiplexado total del hardware real (CPU; memoria, disco, red,..) entre S.Os. completamente *diferentes*, pero se ejecutan las instrucciones nativas del procesador (salvo las excepciones e interrupciones).
- Ejemplos : KVM, VirtualBox, Xen, Vmware...
- Normalmente se necesita instrucciones máquina en procesador físico para acelerar ejecución de excepciones e interrupciones. AMD64 y ARM64 disponen de ellas.
- Permite **ejecutar con rapidez S.Os. dispares** (Linux, BSDs, Windows, Mac OSX), de forma simultanea, en una misma máquina.

Virtualización

Virtualización Completa



Virtualización

Virtualización de nivel sistema operativo (Contenedores)

- Se **comparte núcleo de S.O.**, pero se separan diferentes instancias de sistema: servidores virtuales (Docker, OpenVZ, FreeBSD Jails, Solaris Zones,...)

Virtualización

Virtualización de nivel sistema operativo (Contenedores)

- Se **comparte núcleo de S.O.**, pero se separan diferentes instancias de sistema: servidores virtuales (Docker, OpenVZ, FreeBSD Jails, Solaris Zones,...)
- Es más rápido y con mayor aprovechamiento de recursos (eficiente), pero sólo un tipo de S.O.

Virtualización

Virtualización de nivel sistema operativo (Contenedores)

- Se **comparte núcleo de S.O.**, pero se separan diferentes instancias de sistema: servidores virtuales (Docker, OpenVZ, FreeBSD Jails, Solaris Zones,...)
- Es más rápido y con mayor aprovechamiento de recursos (eficiente), pero sólo un tipo de S.O.
- Modelo implementado en la infraestructura de Google desde hace, aproximadamente, 15 años. Generalizado recientemente y su utilización está incrementandose de forma significativa con Kubernetes (CoreOS, Redhat Openshift,...)

Virtualización

Emulación completa

- Sistemas operativos y aplicaciones compilados para una arquitectura de procesador se ejecutan sobre otro procesador diferente (Qemu, VM de Java, VM Erlang, ...)

Contenedores

Definición

- Un contenedor es un entorno de ejecución virtualizado utilizado en el desarrollo de aplicaciones.
- Se utiliza para crear, ejecutar y desplegar aplicaciones que están aisladas del hardware subyacente.

Contenedores

Definición

- Un contenedor es un entorno de ejecución virtualizado utilizado en el desarrollo de aplicaciones.
- Se utiliza para crear, ejecutar y desplegar aplicaciones que están aisladas del hardware subyacente.
- Utiliza la tecnología cgroups del kernel de Linux, para aislamiento de recursos (procesos, métodos IPC, interfaces de red, sockets, puntos montaje, etc).
- Plataformas de manipulación de contenedores locales: Docker, rkt (CoreOS)

Contenedores

Docker

- Herramienta para automatizar el despliegue de aplicaciones en un conjunto de imágenes autosuficientes, portables y ligeros que puedan ejecutarse en una máquina.

Contenedores

Docker

- Herramienta para automatizar el despliegue de aplicaciones en un conjunto de imágenes autosuficientes, portables y ligeros que puedan ejecutarse en una máquina.

Características principales

- Encapsulación de servicios, aplicaciones y sus dependencias
- Ligero de operar, mover y manipular
- Separación entre implementación del modelo y la infraestructura
- Aislamiento de recursos, red y contenido entre aplicaciones

Contenedores

Repositorio / Catálogo de contenedores

- Un contenedor se especifica, pero
- Su imagen se puede almacenar en un catálogo / repositorio
- Se pueden definir repositorios privados

Contenedores



Contenedores

– Ejemplo utilización Docker :

```
$ docker pull nginx
```

```
...a7767ce9ec2c: Pull complete....
```

```
$ docker run -t -i nginx /bin/bash
```

```
[ root@055dbcc687d8]$ ls -al
```

```
[ root@055dbcc687d8]$ exit
```

```
$ docker run --name mynginx1 -P -d nginx
```

```
$ docker ps
```

<u>CONTAINER ID</u>	<u>IMAGE</u>	<u>COMMAND</u>	<u>CREATED</u>	<u>STATUS</u>
40e8591b6325	<u>nginx</u>	" <u>nginx -g 'daemon off'</u> "	7 minutes <u>ago</u>	Up 7
minutes				
<u>PORTS</u>			<u>NAMES</u>	
0.0.0.0:32769->80/tcp, 0.0.0.0:32768->443/tcp		mynginx1		

```
$ curl http://localhost:32769
```

Kubernetes

Kubernetes

Documentación de Kubernetes

- <https://kubernetes.io/docs/tutorials/kubernetes-basics/>
- <https://kubernetes.io/docs/getting-started-guides/minikube/>

Kubernetes

¿Qué es Kubernetes?

- Automatizar despliegue y gestión de aplicaciones (distribuidas o no) en múltiples máquinas (+100) mediante contenedores

Kubernetes

¿Qué es Kubernetes?

- Automatizar despliegue y gestión de aplicaciones (distribuidas o no) en múltiples máquinas (+100) mediante contenedores
 - utiliza un runtime de contenedores

Kubernetes

¿Qué es Kubernetes?

- Automatizar despliegue y gestión de aplicaciones (distribuidas o no) en múltiples máquinas (+100) mediante contenedores
 - utiliza un runtime de contenedores
- Inspirado en sistemas y experiencia de Google

Kubernetes

¿Qué es Kubernetes?

- Automatizar despliegue y gestión de aplicaciones (distribuidas o no) en múltiples máquinas (+100) mediante contenedores
 - utiliza un runtime de contenedores
- Inspirado en sistemas y experiencia de Google
- Eficiente y robusto

Kubernetes

¿Qué es Kubernetes?

- Automatizar despliegue y gestión de aplicaciones (distribuidas o no) en múltiples máquinas (+100) mediante contenedores
 - utiliza un runtime de contenedores
- Inspirado en sistemas y experiencia de Google
- Eficiente y robusto
- Escalable

Kubernetes

¿Qué es Kubernetes?

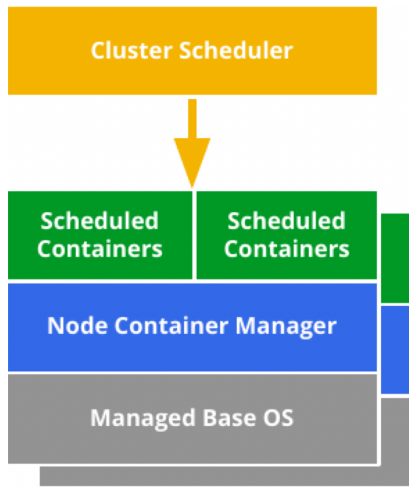
- Automatizar despliegue y gestión de aplicaciones (distribuidas o no) en múltiples máquinas (+100) mediante contenedores
 - utiliza un runtime de contenedores
- Inspirado en sistemas y experiencia de Google
- Eficiente y robusto
- Escalable
- Soporte de tolerancia a fallos y monitorización

Kubernetes

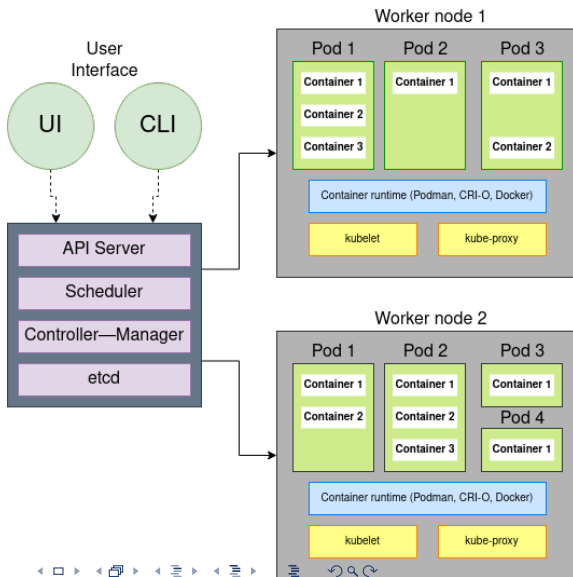
¿Qué es Kubernetes?

- Automatizar despliegue y gestión de aplicaciones (distribuidas o no) en múltiples máquinas (+100) mediante contenedores
 - utiliza un runtime de contenedores
- Inspirado en sistemas y experiencia de Google
- Eficiente y robusto
- Escalable
- Soporte de tolerancia a fallos y monitorización
- Código de dominio público
 - Implementado en Golang, disponible en Github

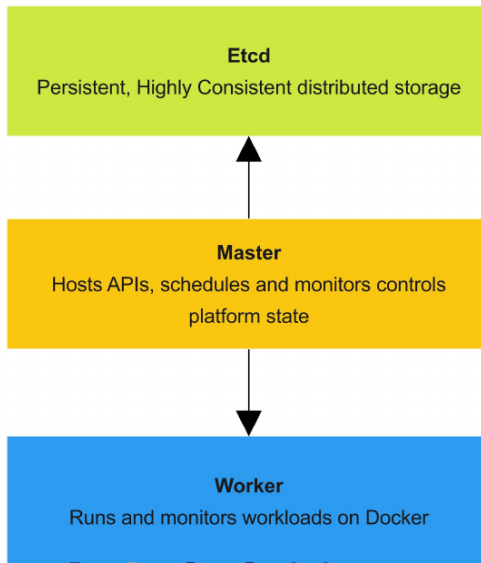
Arquitectura de Kubernetes



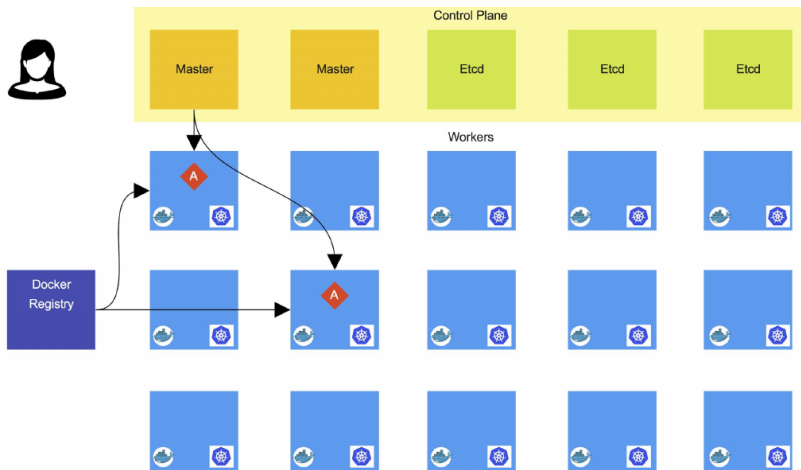
Arquitectura de Kubernetes



Arquitectura de Kubernetes



Arquitectura de Kubernetes



Elementos Básicos

Pod

- Es la unidad de ejecución en Kubernetes
 - Un Pod se utiliza para ejecutar una instancia de una aplicación

Elementos Básicos

Pod

- Es la unidad de ejecución en Kubernetes
 - Un Pod se utiliza para ejecutar una instancia de una aplicación
- Un Pod puede contener 1 ó múltiples contenedores con fuerte interrelación (hardware: CPU, red)
 - Típicamente un pod (sus contenedores) se van a ejecutar dentro de la misma máquina (virtual o física)

Elementos Básicos

Pod

- Es la unidad de ejecución en Kubernetes
 - Un Pod se utiliza para ejecutar una instancia de una aplicación
- Un Pod puede contener 1 ó múltiples contenedores con fuerte interrelación (hardware: CPU, red)
 - Típicamente un pod (sus contenedores) se van a ejecutar dentro de la misma máquina (virtual o física)
 - Lo más habitual 1 app - 1 Pod - 1 contendor

Elementos Básicos

Pod

- Es la unidad de ejecución en Kubernetes
 - Un Pod se utiliza para ejecutar una instancia de una aplicación
- Un Pod puede contener 1 ó múltiples contenedores con fuerte interrelación (hardware: CPU, red)
 - Típicamente un pod (sus contenedores) se van a ejecutar dentro de la misma máquina (virtual o física)
 - Lo más habitual 1 app - 1 Pod - 1 contendor
 - Hay apps que requieren múltiples contenedores que se comunican entre sí: 1 pod - N contenedores

Elementos Básicos

Pod

- Es la unidad de ejecución en Kubernetes
 - Un Pod se utiliza para ejecutar una instancia de una aplicación
- Un Pod puede contener 1 ó múltiples contenedores con fuerte interrelación (hardware: CPU, red)
 - Típicamente un pod (sus contenedores) se van a ejecutar dentro de la misma máquina (virtual o física)
 - Lo más habitual 1 app - 1 Pod - 1 contendor
 - Hay apps que requieren múltiples contenedores que se comunican entre sí: 1 pod - N contenedores
- Sistema de ficheros compartido entre contenedores

Elementos Básicos

Pod

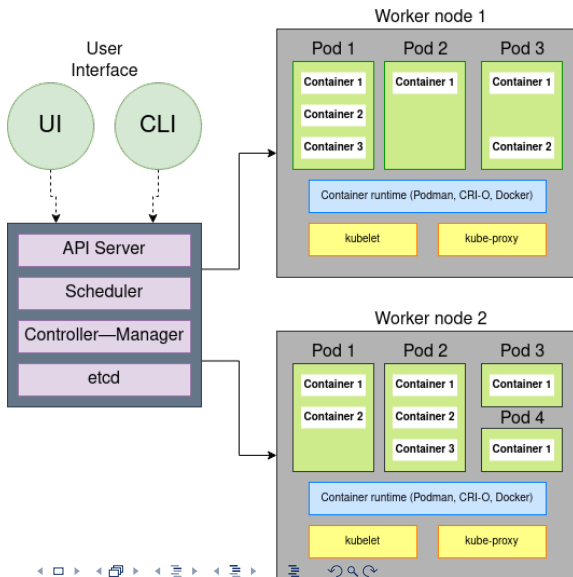
- Es la unidad de ejecución en Kubernetes
 - Un Pod se utiliza para ejecutar una instancia de una aplicación
- Un Pod puede contener 1 ó múltiples contenedores con fuerte interrelación (hardware: CPU, red)
 - Típicamente un pod (sus contenedores) se van a ejecutar dentro de la misma máquina (virtual o física)
 - Lo más habitual 1 app - 1 Pod - 1 contendor
 - Hay apps que requieren múltiples contenedores que se comunican entre sí: 1 pod - N contenedores
- Sistema de ficheros compartido entre contenedores
- Cada contenedor tiene su hardware delimitado

Elementos Básicos

Pod

- Es la unidad de ejecución en Kubernetes
 - Un Pod se utiliza para ejecutar una instancia de una aplicación
- Un Pod puede contener 1 ó múltiples contenedores con fuerte interrelación (hardware: CPU, red)
 - Típicamente un pod (sus contenedores) se van a ejecutar dentro de la misma máquina (virtual o física)
 - Lo más habitual 1 app - 1 Pod - 1 contendor
 - Hay apps que requieren múltiples contenedores que se comunican entre sí: 1 pod - N contenedores
- Sistema de ficheros compartido entre contenedores
- Cada contenedor tiene su hardware delimitado
- El Pod es la unidad de escalado

Elementos Básicos



Elementos Básicos

Red

- Cada Pod tiene su propia IP única en el cluster
 - La IP se comparte entre los contenedores del Pod

Elementos Básicos

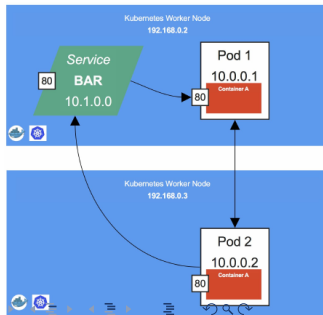
Red

- Cada Pod tiene su propia IP única en el cluster
 - La IP se comparte entre los contenedores del Pod
 - Todos los Pods de un cluster se pueden comunicar entre ellos sin NAT

Elementos Básicos

Red

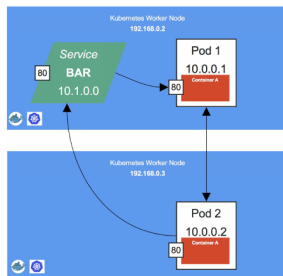
- Cada Pod tiene su propia IP única en el cluster
 - La IP se comparte entre los contenedores del Pod
 - Todos los Pods de un cluster se pueden comunicar entre ellos sin NAT
 - Todos los Pods pueden comunicarse y participar en Servicios



Elementos Básicos

Servicio

- Expone un nombre, un puerto y una IP estable para un grupo de Pods
- Balancea la carga entre un grupo de Pods
- Se provee a los Pods mediante DNS o una variable de entorno
- Construido utilizando un selector sobre los labels de Pods



Elementos Básicos

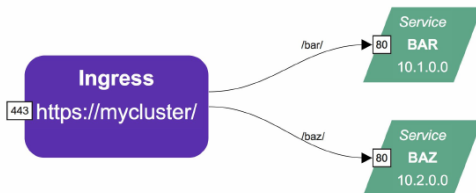
Controladores

- Mantienen estado de aplicaciones (como herramientas estilo Puppet). Se basa en manifiestos declarativos para definir estado objetivo
- **Monitorización:** Lee estado actual workers
- **Análisis:** Compara estado con objetivo en Etcd
- **Acción:** Efectúa acciones correctivas si necesario

Elementos Básicos

Ingress

- Mecanismo que permite definir acceso de red desde internet a recursos del cluster
 - Trabaja con balanceadores de carga
 - Presenta un mapeado con una URL de una sola raíz
 - Expone redes privadas de forma pública al exterior
 - Permite conexiones TLS/SSL



Tipos de Aplicaciones

Job

- crea uno o más Pods y asegura que un determinado n^o de ellos termina con éxito, el trabajo termina, entonces, correctamente

Tipos de Aplicaciones

Job

- crea uno o más Pods y asegura que un determinado n^o de ellos termina con éxito, el trabajo termina, entonces, correctamente
- Se rearrancan Pods, si fallan o se suprimen.

Tipos de Aplicaciones

Job

- crea uno o más Pods y asegura que un determinado n^o de ellos termina con éxito, el trabajo termina, entonces, correctamente
- Se rearrancan Pods, si fallan o se suprimen.

Daemon Set

- todos (o algunos) nodo ejecutan copia de un Pod

Tipos de Aplicaciones

Job

- crea uno o más Pods y asegura que un determinado n^o de ellos termina con éxito, el trabajo termina, entonces, correctamente
- Se rearrancan Pods, si fallan o se suprimen.

Daemon Set

- todos (o algunos) nodo ejecutan copia de un Pod
- Se rearrancan Pods, si fallan o se suprimen.

Tipos de Aplicaciones

Job

- crea uno o más Pods y asegura que un determinado n° de ellos termina con éxito, el trabajo termina, entonces, correctamente
- Se rearrancan Pods, si fallan o se suprimen.

Daemon Set

- todos (o algunos) nodo ejecutan copia de un Pod
- Se rearrancan Pods, si fallan o se suprimen.

ReplicaSet

- garantiza que un determinado n° de réplicas esta siempre en marcha.

Tipos de Aplicaciones

Deployment

- Gestión estilo Puppet de puesta en marcha de Pods o ReplicaSets. Se declara un estado de configuración a alcanzar a un determinado ritmo.

Tipos de Aplicaciones

Deployment

- Gestión estilo Puppet de puesta en marcha de Pods o ReplicaSets. Se declara un estado de configuración a alcanzar a un determinado ritmo.

StatefulSet

- Como los deployments, pero mantiene garantías de unicidad de Pods y su orden de puesta en marcha y eliminación.

Tipos de Aplicaciones

Deployment

- Gestión estilo Puppet de puesta en marcha de Pods o ReplicaSets. Se declara un estado de configuración a alcanzar a un determinado ritmo.

StatefulSet

- Como los deployments, pero mantiene garantías de unicidad de Pods y su orden de puesta en marcha y eliminación.

Services

- Conjunto lógico de Pods y política de acceso a ellos
 - IPs ligadas a Pods desaparecen con ellos, pero el servicio necesita punto de acceso estable eliminación.

Mecanismos organizacionales

Namespaces

- Mecanismo para agrupar recursos (cada uno con nombre único dentro de un namespace) y definir misma política de control de accesos.
 - Engloba la mayor parte de recursos salvo algunos básicos (nodos, Volúmenes Persistentes)

Mecanismos organizacionales

Labels

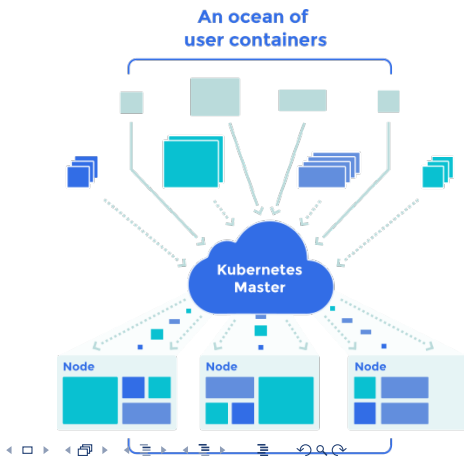
- Parejas clave valor ligadas a objetos (recursos) en Kubernetes, como Pods o nodos
- Método más genérico de organizar grupos de recursos y acceder a ellos

Scheduling en Kubernetes

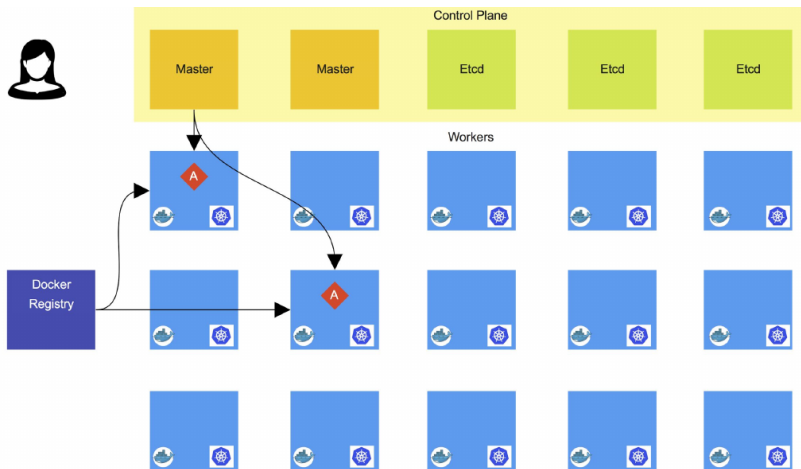
Scheduling en Kubernetes

Planificación (Scheduling)

- Decidir cómo y cuándo se asignan Pods a Nodos distribuidos, según criterios de disponibilidad e idoneidad.



Scheduling en Kubernetes

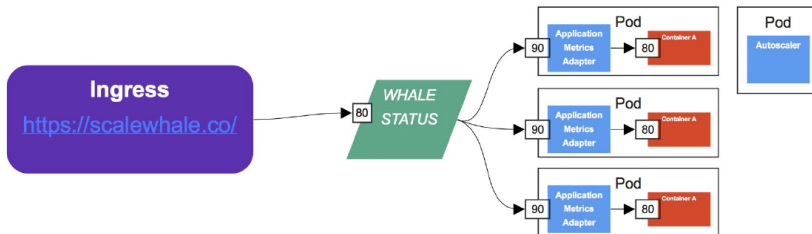


Scheduling en Kubernetes

Diferentes Posibilidades de Planificación (Scheduling)

- **Por defecto:** selecciona nodos, los prioriza y elige al más prioritario
 - Filtra mediante predicados (diferentes disponibles)
 - A continuación los prioriza según funciones de priorización seleccionables
 - Elige al nodo con prioridad más alta
- Multi schedulers
 - Se pueden disponer de diferentes schedulers que implementan políticas diferentes y seleccionarlos de forma explícita para cada Pod

Esquema de despliegue de aplicación con autoescalado



Administración de Kubernetes

Kuberneters API

- Existe una herramienta, **kubectl**, para administrar un cluster de Kubernetes
- Aplicación de un manifiesto de aplicación de kubernetes
- Ejemplo despliegue:
 - `kubectl apply -f nginx-deployment.yaml`

Administración de Kubernetes

Kuberneters API

- Existe una herramienta, **kubectl**, para administrar un cluster de Kubernetes
- Aplicación de un manifiesto de aplicación de kubernetes
- Ejemplo despliegue:
 - `kubectl apply -f nginx-deployment.yaml`
- Ejemplo autoescalado horizontal:
 - `kubectl autoscale deployment nginx --min=2 --max=5 --cpu-percent=80`
- Obtener información del sistema
 - `kubectl get pods`

Administración de Kubernetes

Ejemplo de despliegue de aplicaciones

- `https://kubernetes.io/docs/tutorials/kubernetes-basics/deploy-app/deploy-interactive/`

Sistemas Operativos Distribuidos

30221 - Sistemas Distribuidos

Unai Arronategui - Rafael Tolosana

Dpto. Informática e Ing. de Sistemas