

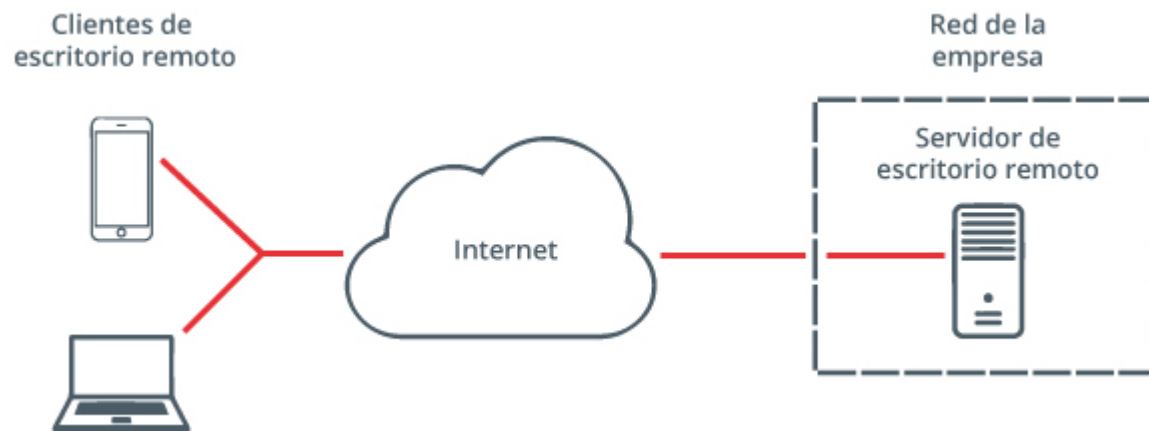
Programación Científica y HPCI

Máster Universitario en Ingeniería Matemática y Computación

Tema 10 / Programación Heterogénea- HPC

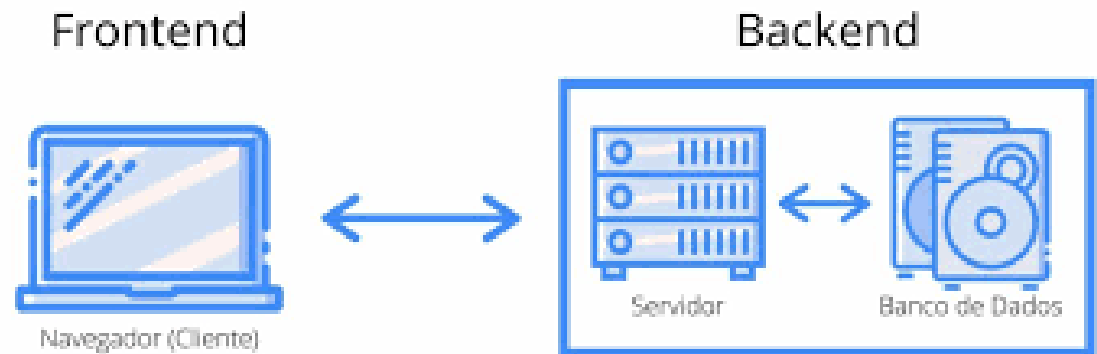
Introducción y Objetivos

- La computación en la nube, también conocida como cloud computing, es una forma avanzada de computación basada en la red y llevada a cabo a través de Internet.
- Permite el acceso remoto a servicios y software, almacenamiento de archivos y procesamiento de datos en procesadores en la nube.
- Proporciona servicios y recursos escalables sin necesidad de instalaciones o mantenimiento en el equipo personal.
- Ahorra costes de inversión en equipos, almacenamiento y software, ya que se alquilan cuando se necesitan.
- Ofrece una interfaz sencilla o API que oculta la complejidad de la infraestructura subyacente.



Componentes de la Computación en la Nube

- Interfaz de usuario o front end: Permite el acceso al servicio en la nube a través de la red y la aplicación.
- Entorno o back end: Constituido por sistemas de almacenamiento de información, aplicaciones disponibles y un servidor central para administrar el sistema.
- Se utiliza una capa de red y protocolos para facilitar las conexiones entre los equipos de la red.



Modelos de Funcionamiento de la Computación en la Nube

- Modelos de despliegue: Se refiere a cómo se implementa la infraestructura en la nube. Puede ser público, privado, híbrido o comunitario.
- Modelos de servicio: Se refiere a los tipos de servicios ofrecidos en la nube. Los principales modelos son: infraestructura como servicio (IaaS), plataforma como servicio (PaaS) y software como servicio (SaaS).



Ventajas de la Computación en la Nube



- Escalabilidad: Permite ajustar los recursos según las necesidades del usuario.
- Acceso remoto: Permite acceder a los servicios y datos desde cualquier ubicación con conexión a Internet.
- Ahorro de costes: Elimina la necesidad de invertir en hardware, almacenamiento y software.
- Actualizaciones automáticas: Las actualizaciones de software y seguridad son manejadas por el proveedor de la nube.
- Mayor colaboración: Facilita el trabajo en equipo y el intercambio de información de forma rápida y eficiente.

Modelos de Despliegue - Nube Pública

- En la nube pública, los recursos, sistemas y servicios son accesibles para cualquier tipo de usuario.
- La infraestructura informática se encuentra en el sitio del proveedor.



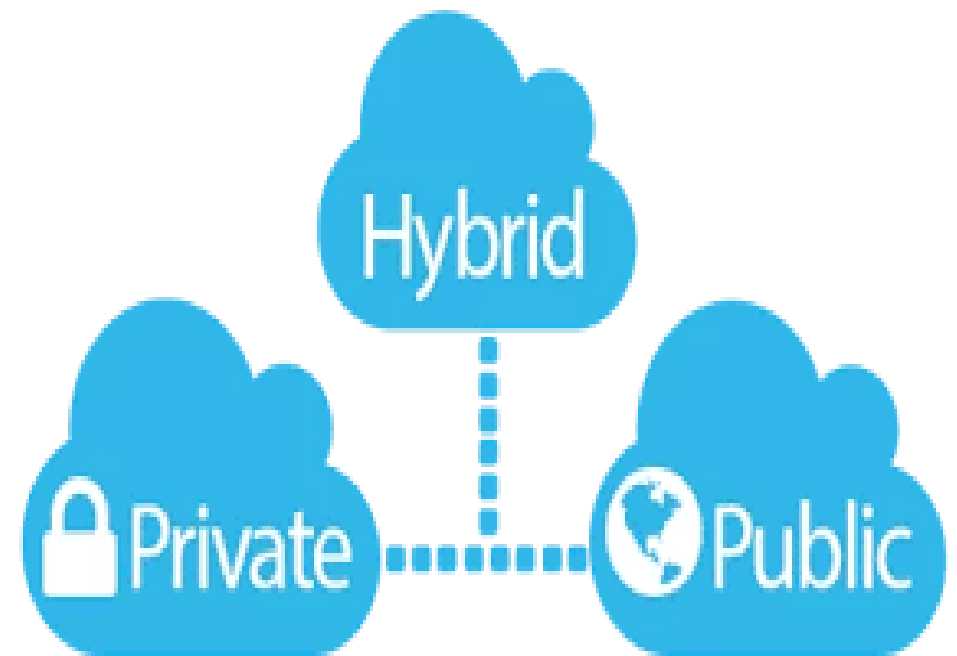
Modelos de Despliegue - Nube Privada



- La nube privada proporciona servicios de uso exclusivo para cada cliente, generalmente organizaciones de gran volumen.
- Los sistemas y servicios son accesibles desde la propia organización o el centro de datos del proveedor.
- Ofrece un nivel más alto de seguridad y control.

Modelos de Despliegue - Nube Híbrida

- La nube híbrida es una solución que combina la nube privada y la nube pública.
- Los clientes alojan aplicaciones críticas en la nube privada para mayor seguridad y control, mientras que las aplicaciones no críticas se alojan en la nube pública.



Categorización de Servicios - Infraestructura como Servicio (IaaS)

- IaaS permite el acceso a recursos fundamentales como máquinas físicas, virtuales y almacenamiento de datos y redes a demanda.
- El cliente es responsable de la instalación, configuración, protección y mantenimiento del software instalado en los recursos de la infraestructura.
- Orientado a administradores de sistemas y arquitectos de red.



Categorización de Servicios - Plataforma como Servicio (PaaS)



- PaaS proporciona al cliente el entorno de ejecución de aplicaciones y herramientas de desarrollo y despliegue.
- Incluye la infraestructura y soluciones para soportar el ciclo de vida de las aplicaciones.
- Orientado a desarrolladores de aplicaciones.

Categorización de Servicios - Software como Servicio (SaaS)



- SaaS aloja las aplicaciones del cliente en la nube y proporciona una infraestructura de cómputo actualizada y gestionada por el proveedor.
- Permite ampliar o reducir los servicios contratados según las necesidades.
- Orientado al usuario final.

Categorización de Servicios - Funciones como Servicio (FaaS)

- FaaS permite la ejecución de funciones a través de contenedores efímeros.
- Favorece el desarrollo de arquitecturas basadas en microservicios y despliegues continuos.
- Conocido como arquitectura serverless.



Más Ventajas de la computación en la nube

- Flexibilidad: Atender las demandas de los usuarios de forma inmediata y poder variar los servicios contratados en cualquier momento.
- Escalabilidad: Capacidad de crecimiento y adaptación a las necesidades cambiantes de los clientes.
- Confianza: Arquitecturas con redundancia que garantizan un alto porcentaje de acceso a los servidores incluso en caso de fallos de hardware.



Más Ventajas de la computación en la nube

- Copias de seguridad y recuperación: Posibilidad de realizar copias de seguridad periódicas y gestionar la recuperación eficientemente.
- Actualizaciones: Proveedores encargados de controlar las actualizaciones de software para asegurar el uso de las versiones más modernas.
- Hiperconvergencia: Combinación de redes y almacenamiento en un solo sistema, facilitando el trabajo colaborativo y el acceso compartido a la información.

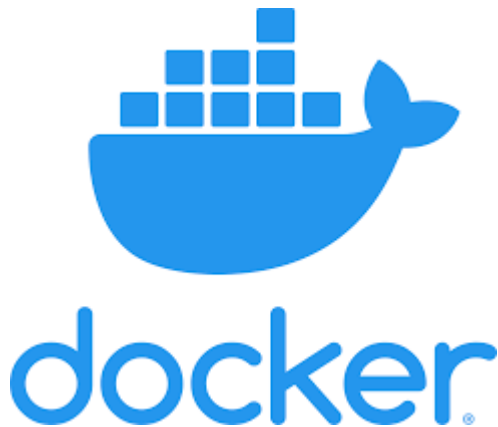


Más Ventajas de la computación en la nube



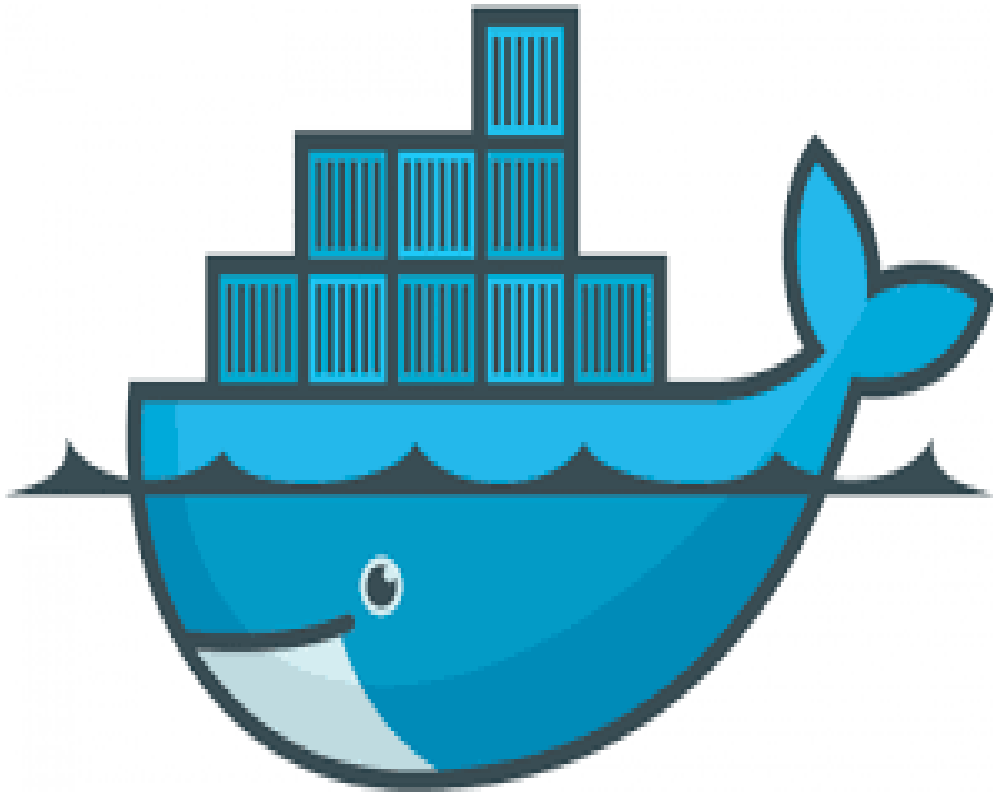
- Seguridad: Protección de aplicaciones, datos e infraestructuras con enfoque en confidencialidad, mantenimiento y borrado de datos.
- Reducción de costes: Eliminación de grandes inversiones en hardware y software, ya que son proporcionados por la infraestructura de la nube.
- Fomento de la innovación: Acceso a nuevas aplicaciones y servicios que permiten el uso de tecnologías como inteligencia artificial e internet de las cosas.

Introducción a Docker para aplicaciones en Python



- Herramienta de creación de contenedores para entornos de aplicación aislados y reproducibles.
- Elevada popularidad entre los desarrolladores de Python.
- Uso de Docker para desarrollar y desplegar aplicaciones localmente y en la nube, acelerando el proceso de paso a producción.

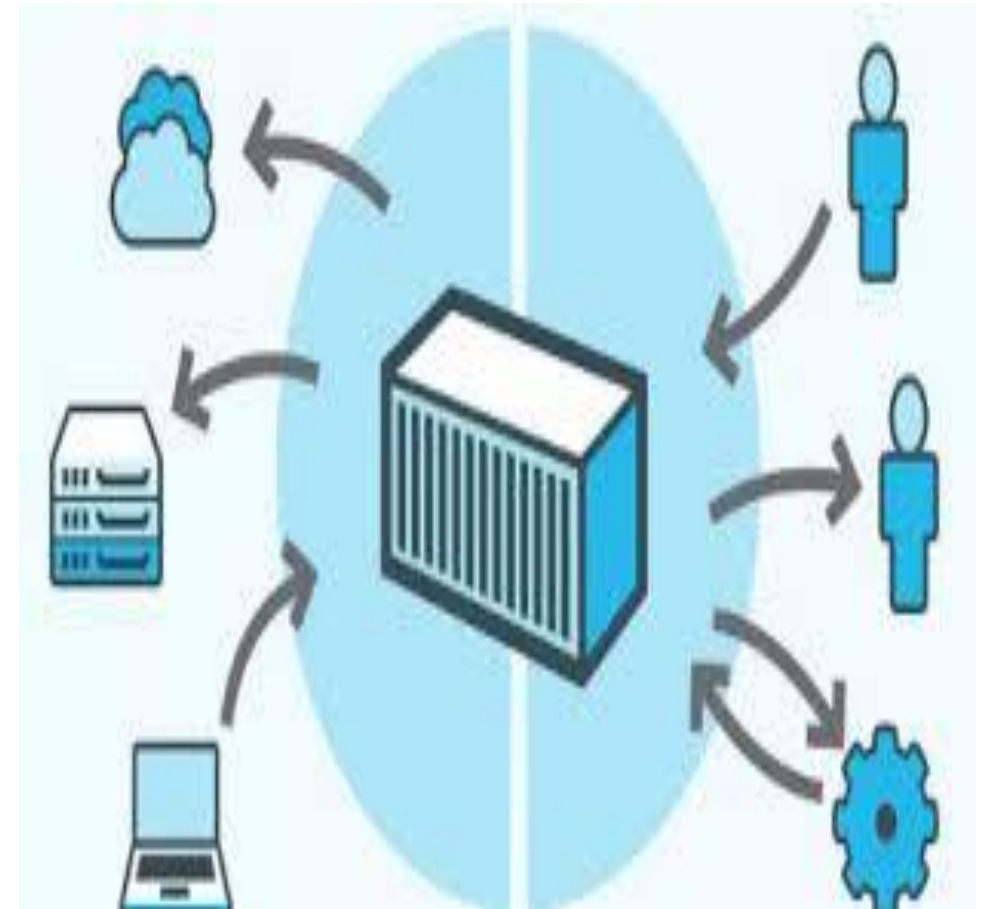
Propósito de los contenedores en Docker



- Separación de procesos en contenedores para ejecutar de manera independiente sobre una infraestructura y mantener la seguridad.
- Los contenedores permiten ejecutar aplicaciones en sistemas separados.

Docker como plataforma para ejecutar contenedores

- Plataforma que permite ejecutar contenedores con aplicaciones preempaquetadas.
- Beneficios de empaquetar aplicaciones con todas sus dependencias para su ejecución en diferentes plataformas.
- Alta rapidez y repetibilidad del despliegue de aplicaciones.



Instalación de Docker

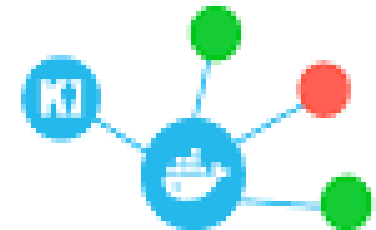
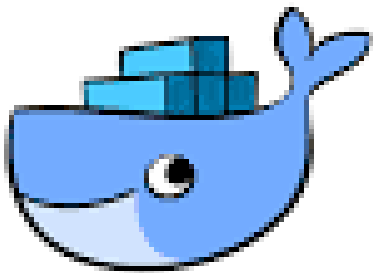
- Docker debe ser instalado previamente para su uso conveniente.
- Componentes de Docker: el motor de Docker (Docker Engine), el cliente Docker y Docker Compose.

Docker Engine

Docker Compose

Docker Machine

Kitematic



Componentes principales de Docker Engine

- Docker Engine como una arquitectura cliente-servidor.
- Elementos principales del motor: el servidor (proceso demonio), la API REST y el cliente de interfaz de línea de comandos (CLI).



Órdenes básicas en Docker

- Órdenes básicas que se ejecutan en la línea de comandos de Docker.
- Dockerfile para construir una imagen.
- Recomendación de uso de la opción: -rm para eliminar el contenedor después de su uso.

Construir una imagen:

```
docker build -t nombre_imagen .
```

Ejecutar una imagen:

```
docker run --rm nombre_imagen
```

Listar imágenes:

```
docker images
```

Listar contenedores:

```
docker ps -a
```

Borrar una imagen:

```
docker rmi imagen_id
```

Borrar un contenedor:

```
docker rm container_id
```

Listado y eliminación de imágenes y contenedores

- Uso de las órdenes `docker images` y `docker ps -a` para listar imágenes y contenedores.
- Borrado de imágenes y contenedores usando sus identificadores.

Ejemplo de salida de '`docker images`':

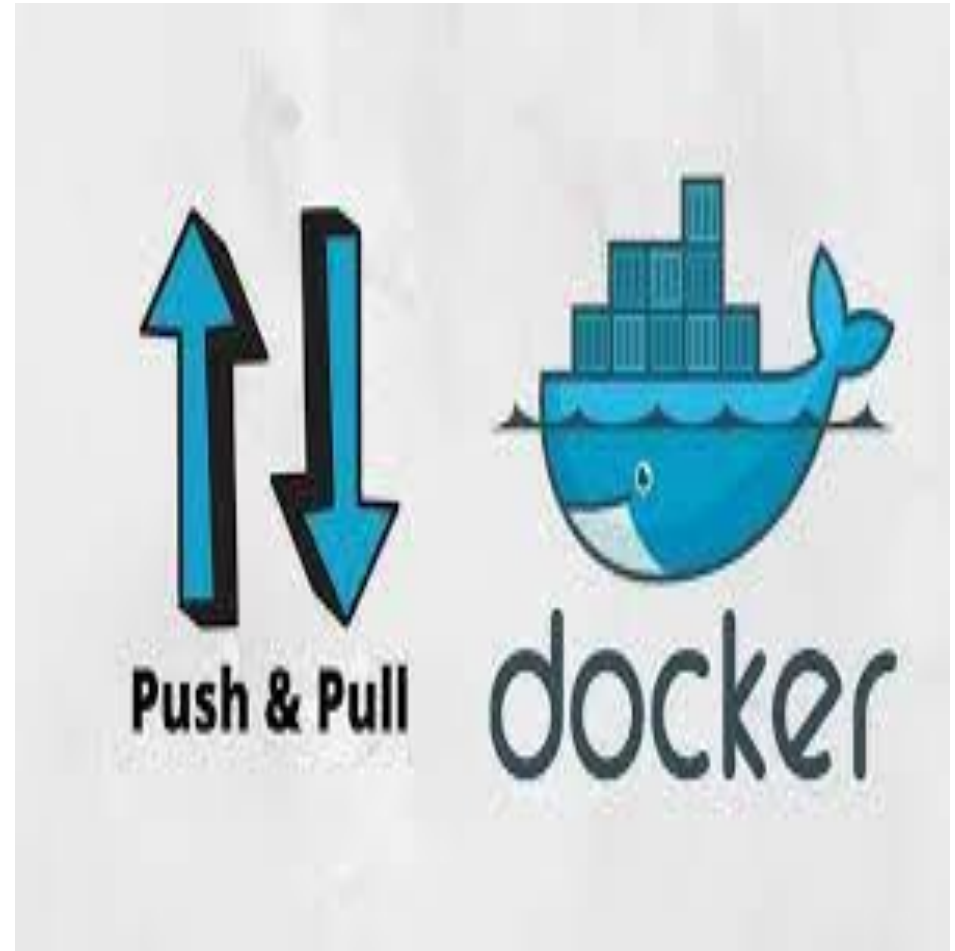
REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
python	rc	1a2b3c4d5e6f	2 weeks ago	800MB
nginx	latest	7a8b9c0d1e2f	3 weeks ago	400MB
mysql	5.7	3a4b5c6d7e8f	1 month ago	1.5GB

Ejemplo de salida de '`docker ps -a`':

CONTAINER ID	IMAGE	COMMAND	CREATED
a1b2c3d4e5f6	python:rc	"python"	2 days ago
e7f8g9h0i1j2	nginx:latest	"nginx -g 'daemon...'"	1 week ago
k3l4m5n6o7p8	mysql:5.7	"mysqld"	1 month ago

Ejecución de código Python en un contenedor Docker

- 1. Abrir el terminal.
- 2. Ejecutar el siguiente comando:
 - `docker run -it --rm python:rc python /path/to/script.py`
 - - ``docker run``: Ejecuta un contenedor.
 - - ``-it``: Habilita la interactividad con el contenedor.
 - - ``--rm``: Elimina el contenedor después de su uso.
 - - ``python:rc``: La imagen de Python utilizada.
 - - ``python /path/to/script.py``: El comando para ejecutar el script de Python dentro del contenedor.
- 3. El script de Python se ejecutará dentro del contenedor Docker y mostrará los resultados en el terminal.



Programación Científica y HPCI

Máster Universitario en Ingeniería Matemática y Computación

Preparación Examen

Explicación examen

- **Los 10 temas tendrán preguntas tipo test.**
 - No permitidos apuntes.
 - No permitido auriculares, ni aplicaciones tipo Whatsapp.
 - Cada pregunta contestada correctamente vale 0,5 puntos.
 - Cada pregunta contestada erróneamente resta 0,15 puntos





www.unir.net