

Servicios de Cómputo en la Nube

Los servicios de cómputo en la nube han transformado profundamente la manera en que las empresas y los usuarios gestionan y acceden a los recursos informáticos. A medida que las organizaciones se enfrentan a una creciente demanda de mayor flexibilidad, escalabilidad y eficiencia, la computación en la nube ha emergido como la solución ideal para adaptarse a estos desafíos.

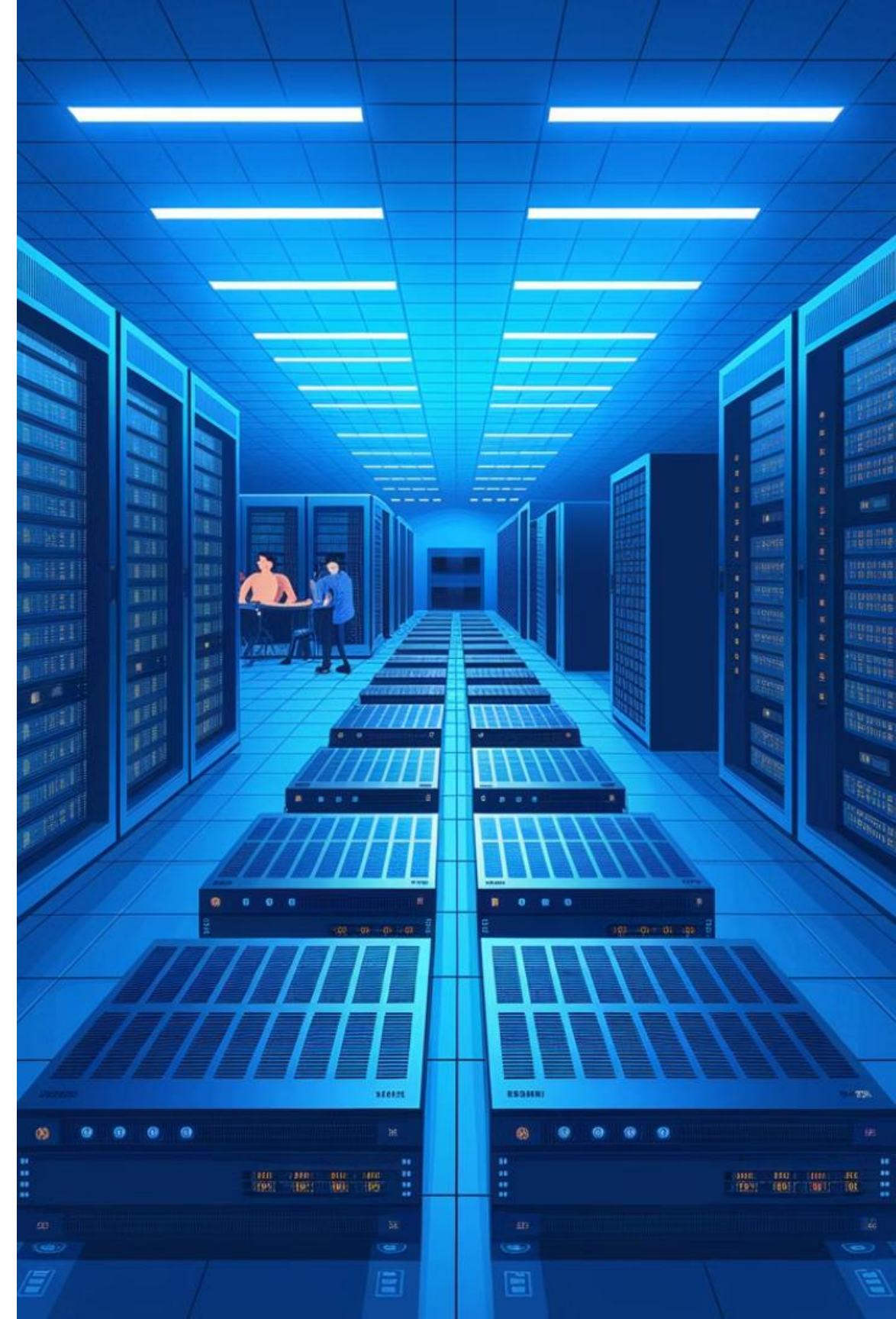
La nube no solo elimina la necesidad de poseer infraestructura física costosa y de alto mantenimiento, sino que también permite a los usuarios acceder a recursos y servicios de manera remota, a través de Internet, desde cualquier parte del mundo.

R por Kibernum Capacitación S.A.



Preguntas de activación

1. ¿Qué tipo de almacenamiento es más adecuado para manejar grandes archivos multimedia, objetos o bloques?
2. ¿Qué servicio de AWS usarías para almacenamiento a largo plazo y bajo costo?
3. ¿Por qué es importante elegir el almacenamiento adecuado para optimizar costos y rendimiento?





Transformación Digital con la Nube

Pieza Clave

La computación en la nube es fundamental en la transformación digital de muchas empresas, permitiéndoles operar con mayor eficiencia y escalar rápidamente.

Elasticidad y Agilidad

La nube ofrece elasticidad y agilidad, permitiendo a las empresas innovar, lanzar nuevos productos y servicios más rápidamente.

Impacto en Consumidores

Los consumidores individuales pueden acceder a herramientas y servicios avanzados sin necesidad de hardware especializado o gestión de infraestructura.

Definición de Servicio de Cómputo en la Nube



Modelo de Prestación

Un servicio de cómputo en la nube es un modelo donde los usuarios acceden a capacidades de procesamiento, almacenamiento, aplicaciones y redes a través de Internet.



Infraestructura Remota

Los recursos son alojados y gestionados por un proveedor de servicios de nube en centros de datos remotos, eliminando la necesidad de infraestructura física propia.

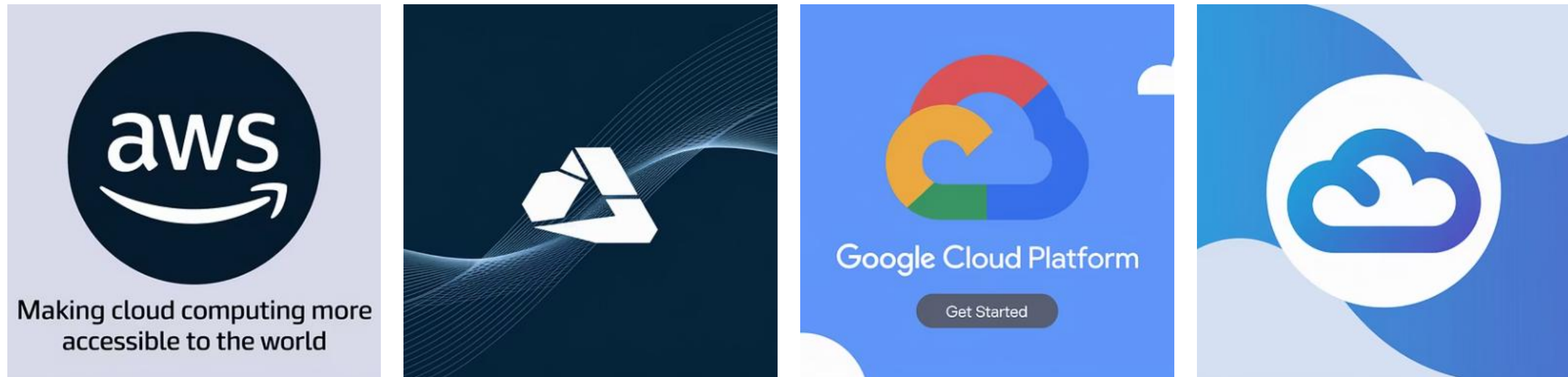


Pago por Consumo

Los usuarios solo pagan por lo que consumen, sin necesidad de inversiones iniciales en infraestructura cara, haciendo los costos más predecibles y escalables.



Proveedores Principales de Servicios en la Nube



Proveedores como Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP) y Oracle Cloud Infrastructure operan centros de datos de alto rendimiento y alta disponibilidad donde almacenan y gestionan los recursos informáticos que los usuarios solicitan. Estos servicios están disponibles en modelos como infraestructura como servicio (IaaS), plataforma como servicio (PaaS) y software como servicio (SaaS).

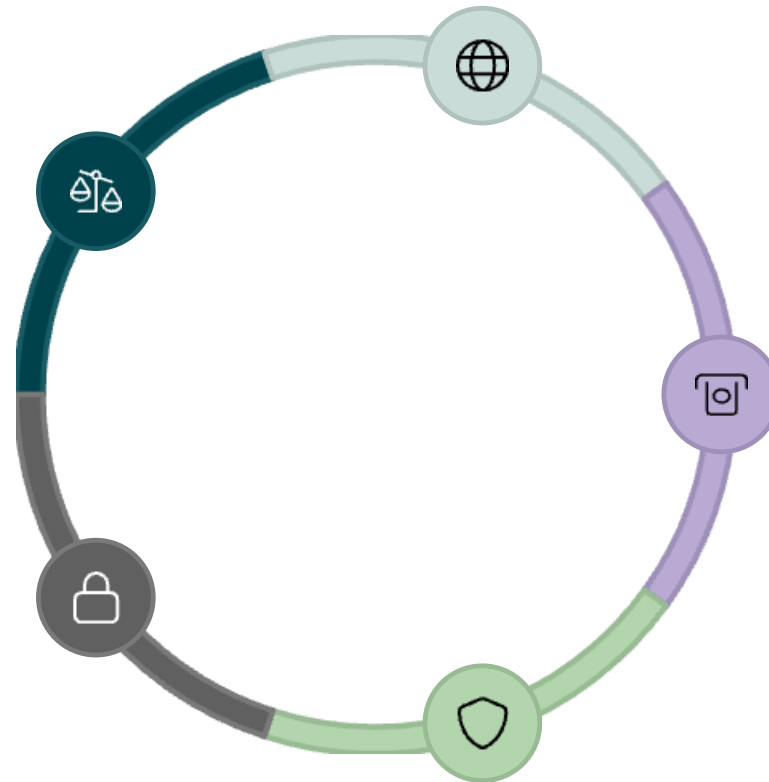
Características Clave de los Servicios en la Nube

Escalabilidad

Ajuste dinámico de recursos según necesidades, tanto vertical como horizontalmente.

Seguridad

Implementación de estrictas medidas como encriptación, firewalls y control de acceso.



Acceso Global

Disponibilidad desde cualquier lugar con conexión a Internet, promoviendo flexibilidad y trabajo remoto.

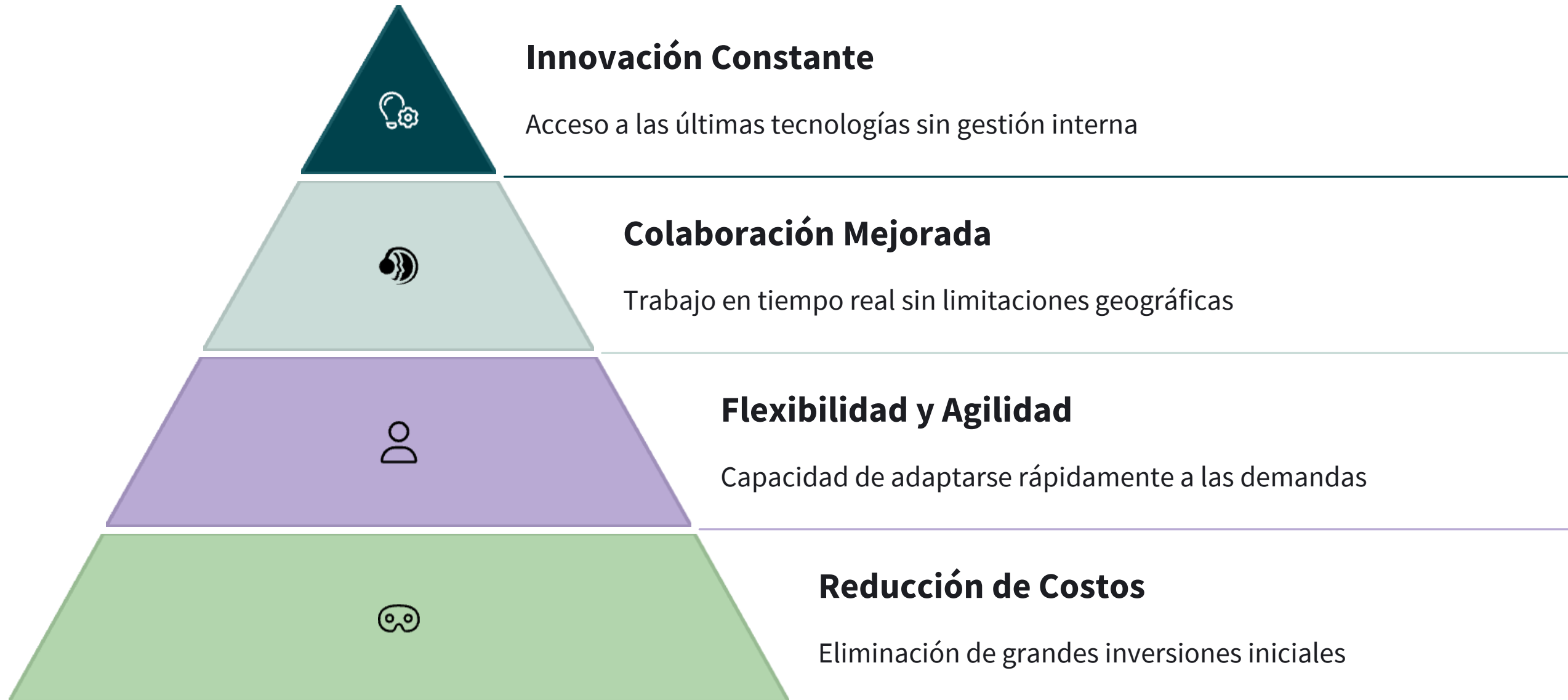
Pago por Uso

Modelo de consumo donde solo se paga por los recursos utilizados, eliminando grandes inversiones iniciales.

Alta Disponibilidad

Garantías de servicio continuo mediante redundancia y recuperación ante desastres.

Beneficios de Utilizar Cómputo en la Nube





Beneficios Adicionales del Cómputo en la Nube



Mejor Seguridad

Medidas de seguridad de clase mundial, incluyendo cifrado de datos, autenticación multifactor y protección contra DDoS.



Eficiencia Energética

Centros de datos optimizados para uso eficiente de energía, reduciendo el impacto ambiental de las infraestructuras tecnológicas.



Backup y Recuperación

Copias de seguridad continuas y automatizadas, garantizando disponibilidad de datos incluso en caso de fallos o desastres.

Tipos de Servicios de Cómputo en la Nube



Máquinas Virtuales (IaaS)

Emulación completa de sistemas computacionales



Contenedores

Unidades estandarizadas de software con dependencias



Computación sin Servidor

Ejecución de código sin gestionar infraestructura

Máquinas Virtuales (VMs)

Definición

Una máquina virtual es una emulación completa de un sistema computacional que permite ejecutar un sistema operativo y aplicaciones en un entorno virtualizado. Se ejecuta sobre un hipervisor, una capa de software que actúa como intermediario entre el hardware y las máquinas virtuales.

Tipos de Hipervisores

- Tipo 1 (bare-metal): Directamente sobre hardware
- Tipo 2 (hosted): Sobre un sistema operativo anfitrión

¿Por qué existen las VMs?

Infrautilización del Hardware

Los servidores físicos operaban muy por debajo de su capacidad máxima, desperdiciando recursos como CPU y RAM, generando costos altos sin utilización efectiva.

Alto Costo de Escalabilidad

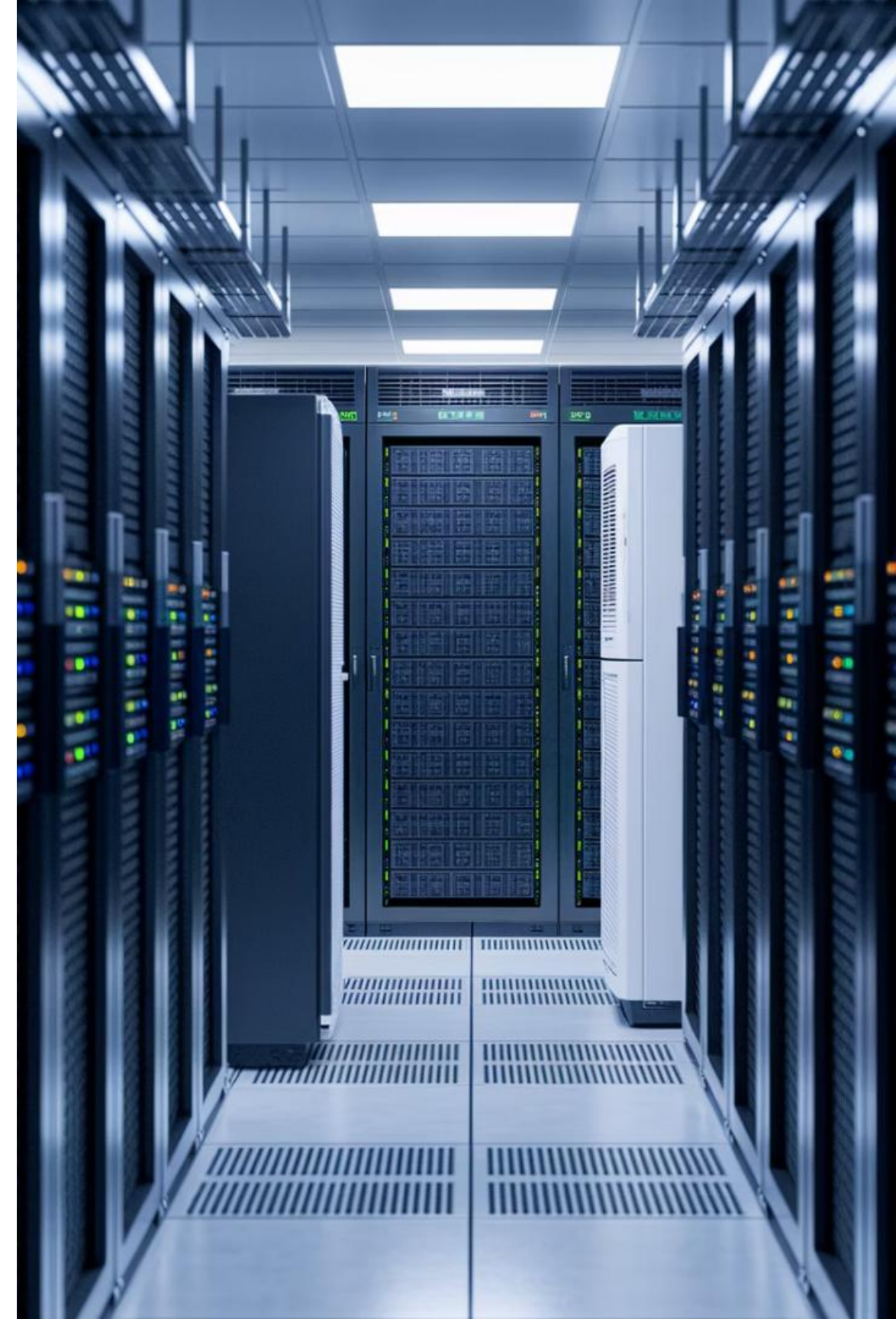
Cada nuevo entorno requería un servidor físico nuevo, implicando inversión en hardware, espacio físico, energía y personal técnico.

Falta de Flexibilidad

Implementar nuevos entornos para desarrollo o testing tomaba mucho tiempo y requería configuraciones físicas y manuales.

Requerimientos de Aislamiento

Muchas organizaciones necesitaban entornos separados para garantizar que un fallo en una aplicación no afectara a otras.



Características Clave de las Máquinas Virtuales



Sistema Operativo Completo

Cada VM contiene su propio sistema operativo, permitiendo ejecutar software que dependa de un sistema específico.



Aislamiento Total

Cada VM opera de forma independiente. Si una falla, las demás no se ven afectadas, mejorando seguridad y estabilidad.



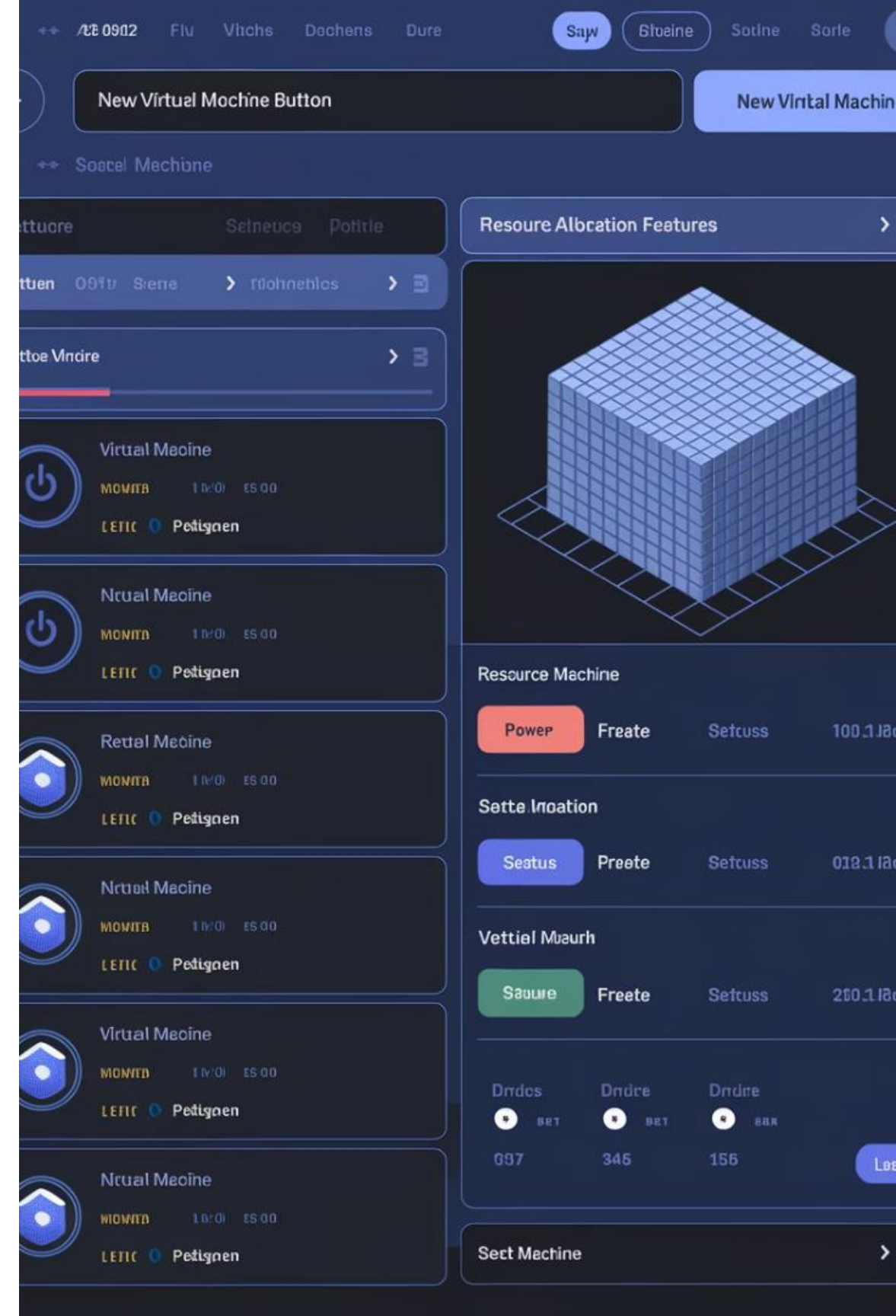
Control Total del Sistema

Acceso como root o administrador, dando libertad para instalar paquetes, modificar configuraciones y gestionar servicios.

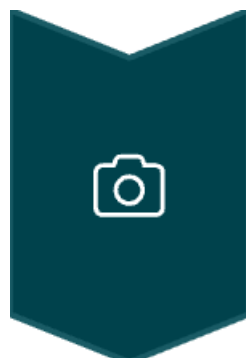


Configuración Flexible

Definición de recursos exactos asignados a cada VM, como CPU, memoria RAM, tamaño de disco e interfaces de red.

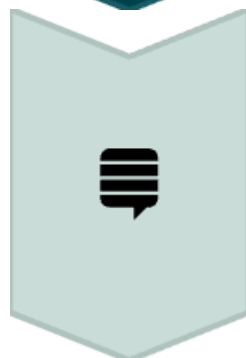


Ventajas Adicionales de las Máquinas Virtuales



Snapshot y Clonación

Capacidad de tomar "fotos" del estado actual y clonar VMs rápidamente



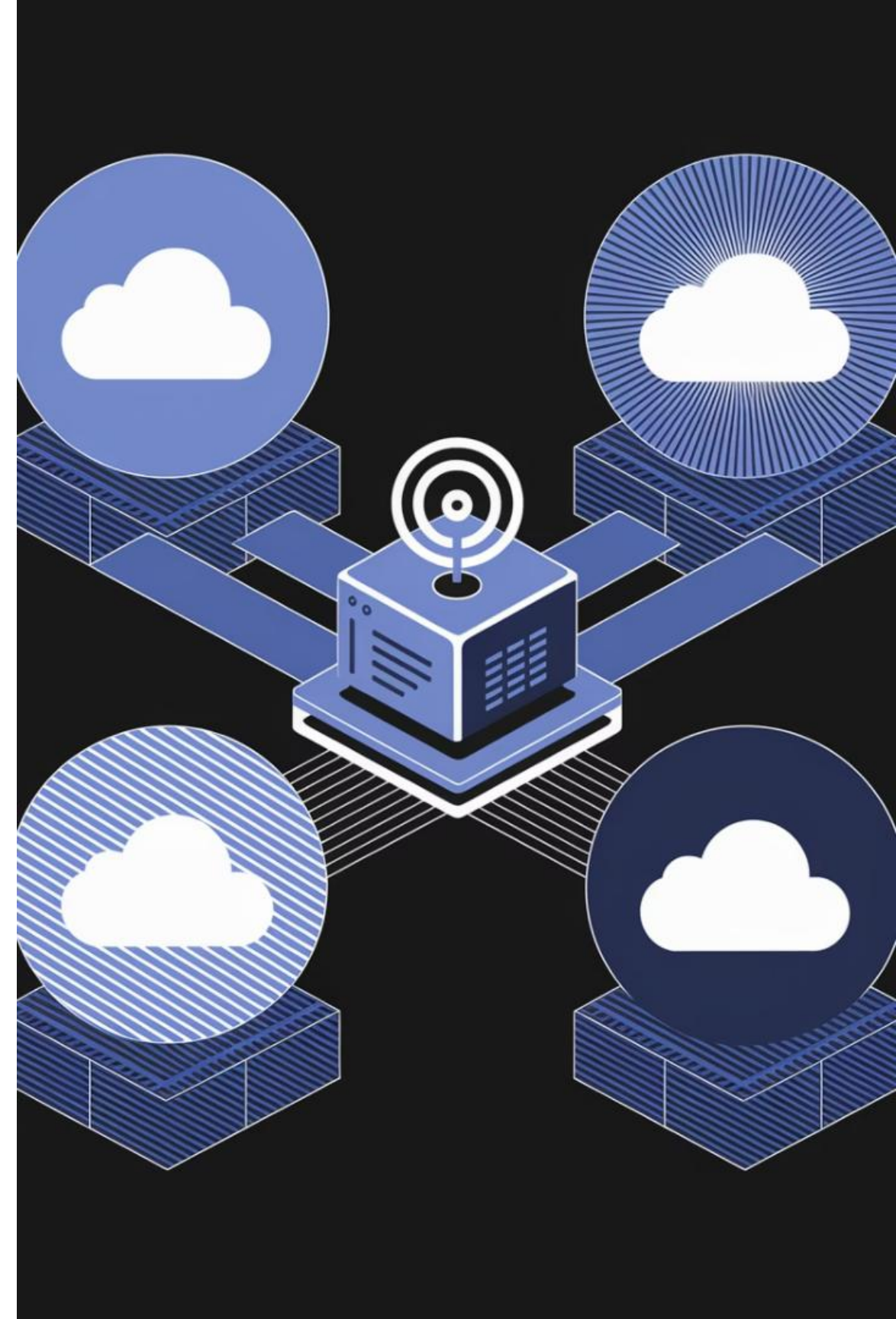
Portabilidad

Exportación y traslado entre distintos entornos manteniendo funcionamiento



Compatibilidad

Ejecución en diferentes plataformas sin cambios en la configuración



Casos de Uso de Máquinas Virtuales



Máquinas Virtuales en la Nube

Una máquina virtual en la nube es una emulación de un sistema informático que se ejecuta sobre una infraestructura física remota, gestionada por un proveedor de servicios en la nube. Funciona como si fuera un ordenador físico, pero en realidad está basada en software y se ejecuta en servidores compartidos en un centro de datos.

100%

Control

Control total sobre el sistema operativo y aplicaciones

24/7

Disponibilidad

Acceso permanente desde cualquier ubicación

0

Hardware Propio

Sin necesidad de adquirir o mantener servidores físicos



Ventajas de Utilizar Máquinas Virtuales en la Nube

Escalabilidad

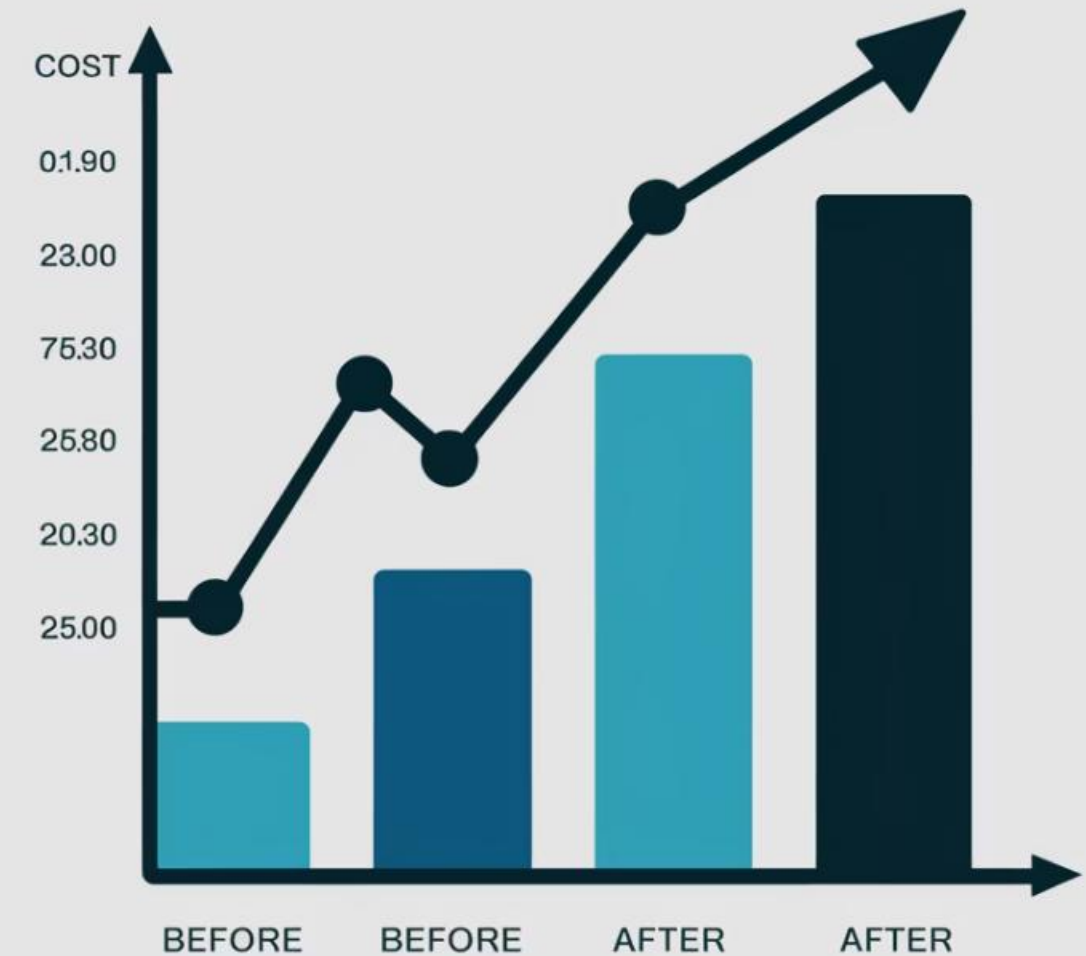
Aumento o disminución dinámica de recursos según necesidades, pagando solo por lo utilizado. Permite adaptar la infraestructura a las demandas cambiantes del negocio.

Reducción de Costos

Eliminación de inversiones en hardware físico y mantenimiento. El modelo de pago por uso permite optimizar el presupuesto y convertir gastos de capital en gastos operativos.

Flexibilidad y Rapidez

Creación, duplicación o eliminación rápida de VMs. Ideal para pruebas, desarrollo de software o despliegue de aplicaciones en tiempo récord.



Más Ventajas de las VMs en la Nube



Alta Disponibilidad

Garantía de disponibilidad continua mediante replicación en múltiples ubicaciones geográficas, asegurando operatividad incluso ante fallos.



Seguridad

Implementación de medidas avanzadas como cifrado de datos, firewalls, autenticación multifactor y monitoreo continuo, con copias de seguridad automáticas.



Acceso Remoto

Posibilidad de acceder desde cualquier lugar con conexión a internet, favoreciendo el teletrabajo, la colaboración entre equipos y la administración remota.

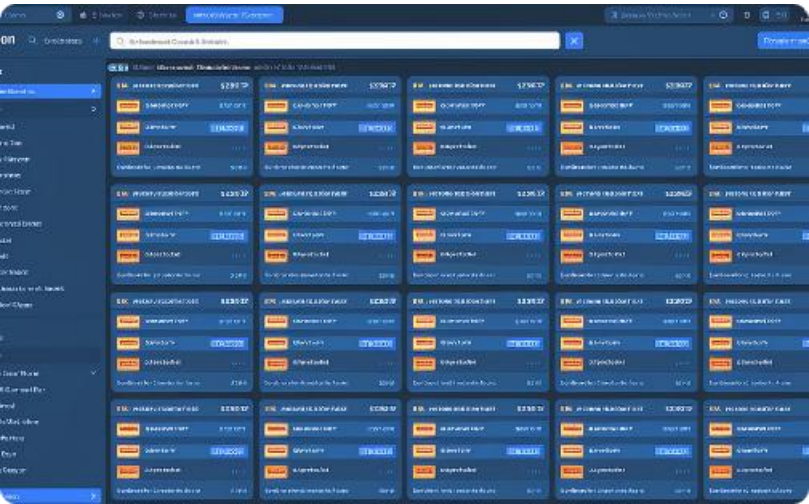


Aislamiento

Cada máquina virtual está aislada del resto, permitiendo ejecutar diferentes entornos de forma segura en un mismo servidor físico.

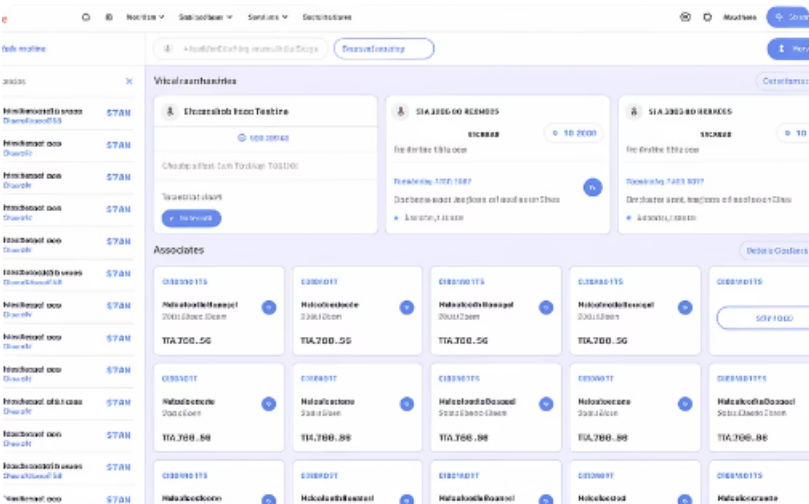


Ejemplos de Servicios de Máquinas Virtuales en la Nube



Amazon EC2 (AWS)

Permite crear instancias virtuales de diferentes tipos y tamaños para casi cualquier necesidad de cómputo, con gran flexibilidad y escalabilidad.



Google Compute Engine (GCP)

Ofrece máquinas virtuales personalizadas y optimizadas para cargas de trabajo específicas, como procesamiento de datos, inteligencia artificial o hosting web.



Azure Virtual Machines

Soporta una amplia gama de sistemas operativos y está profundamente integrado con otros servicios de Microsoft, ideal para empresas con infraestructura basada en Windows.

Casos de Uso Comunes de VMs en la Nube

- Desarrollo y pruebas de software:** Permite a los desarrolladores crear entornos de desarrollo temporales para probar nuevas versiones sin afectar los sistemas en producción.

- Hosting de aplicaciones y sitios web:** Las VMs pueden ser configuradas para alojar aplicaciones web, APIs, servidores de bases de datos, entre otros.

- Análisis de datos y procesamiento intensivo:** Gracias a la escalabilidad, es posible utilizar máquinas con gran capacidad de procesamiento para análisis masivos de datos, minería de datos o modelado estadístico.

- Formación y laboratorios virtuales:** Ideal para instituciones educativas y programas de capacitación que requieren entornos controlados para prácticas.

- Backup y recuperación ante desastres:** Las VMs se utilizan para mantener copias de seguridad de sistemas críticos, listas para activarse si el sistema principal falla.

- Inteligencia artificial y machine learning:** Se emplean VMs con GPUs potentes para entrenar modelos de IA sin invertir en costoso hardware local.

- Migración de sistemas heredados:** Muchas empresas migran sus servidores físicos a la nube utilizando VMs, lo que les permite mantener sus sistemas existentes con mayor flexibilidad y menor costo.

Contenedores en la Nube

Los contenedores han revolucionado el modo en que se crean, despliegan y administran las aplicaciones en la nube. Al ofrecer una capa de abstracción sobre el sistema operativo, permiten que el software se ejecute de manera consistente en cualquier entorno, mejorando la portabilidad, eficiencia y escalabilidad.

A diferencia de las máquinas virtuales tradicionales, los contenedores son más ligeros y rápidos, lo que los convierte en una opción ideal para arquitecturas modernas basadas en microservicios.



¿Qué es un Contenedor?

Definición Técnica

Un contenedor es una unidad estandarizada de software que empaqueta el código de una aplicación y sus dependencias, incluyendo bibliotecas y configuraciones, de manera que se pueda ejecutar de forma consistente en cualquier entorno.

Diferencia con VMs

A diferencia de una máquina virtual, que emula un sistema operativo completo, un contenedor se ejecuta sobre el sistema operativo del host, aprovechando recursos como el kernel y evitando la sobrecarga de la virtualización completa.

Explicación Sencilla

Un contenedor es como una caja de herramientas que contiene todo lo necesario para hacer un trabajo. La caja permite llevar todo a donde quieras sin que se pierda nada, y siempre tendrás las mismas herramientas listas para usar, independientemente del lugar.



Introducción a Docker



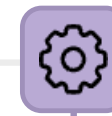
Plataforma de Código Abierto

Docker facilita la creación, despliegue y gestión de contenedores desde su lanzamiento en 2013.



Imágenes Docker

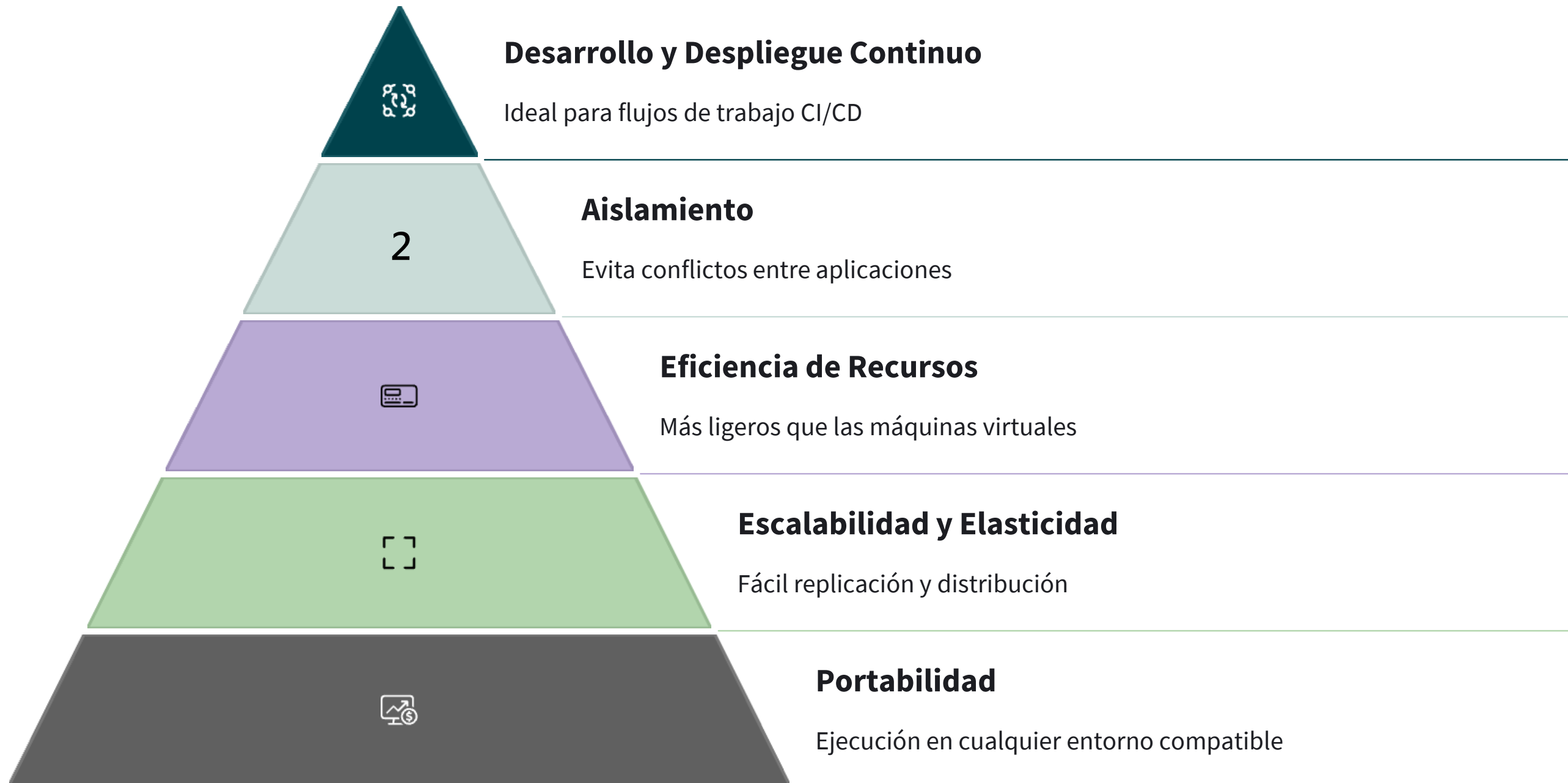
Permite empaquetar aplicaciones y dependencias en contenedores mediante la creación de "imágenes".

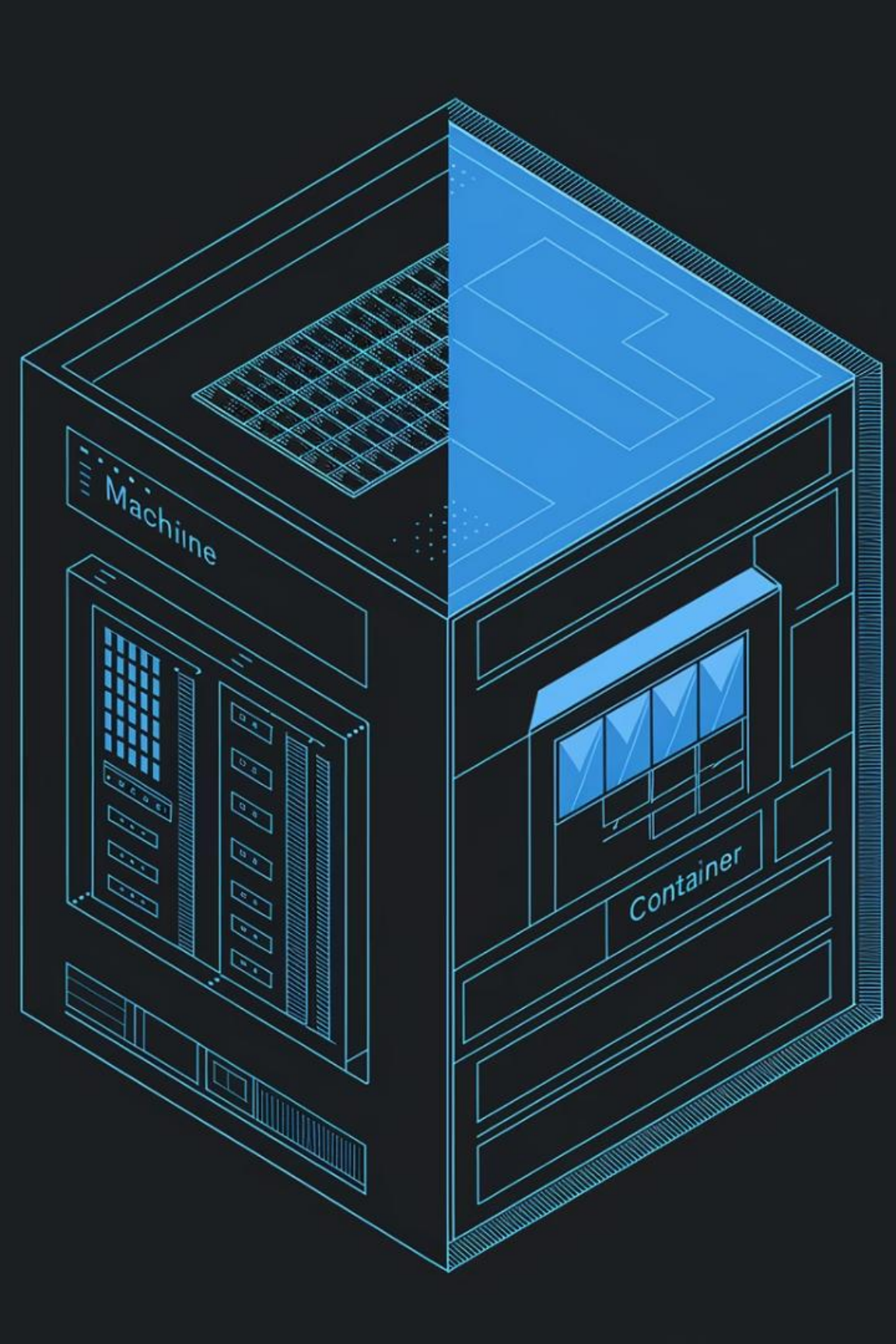


Herramientas Complementarias

Incluye Docker Compose para aplicaciones multi-contenedor, y compatibilidad con orquestadores como Docker Swarm y Kubernetes.

Ventajas de Usar Contenedores en la Nube





Comparación entre Máquinas Virtuales y Contenedores

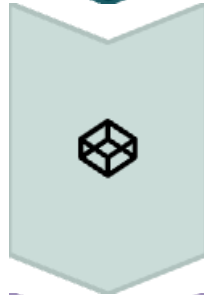
Aspecto	Máquinas Virtuales	Contenedores
Arquitectura	Incluye sistema operativo completo con kernel propio	Comparte el kernel del sistema operativo host
Rendimiento	Mayor uso de recursos y tiempo de arranque	Más eficientes y rápidos de iniciar
Escalabilidad	Más costosa y menos eficiente	Ideal para escalar rápidamente sin sobrecarga
Portabilidad	Posibles inconsistencias entre entornos	Garantía de ejecución consistente en cualquier lugar

Casos de Uso Comunes de Contenedores



Microservicios

Perfectos para arquitecturas de microservicios, permitiendo dividir aplicaciones monolíticas en servicios independientes, cada uno en su propio contenedor.



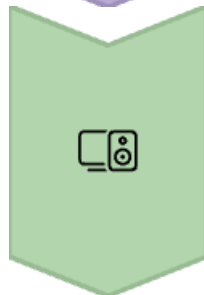
Desarrollo y Pruebas

Creación de entornos de desarrollo consistentes, donde las aplicaciones se ejecutan en la misma configuración en todas las máquinas y entornos.



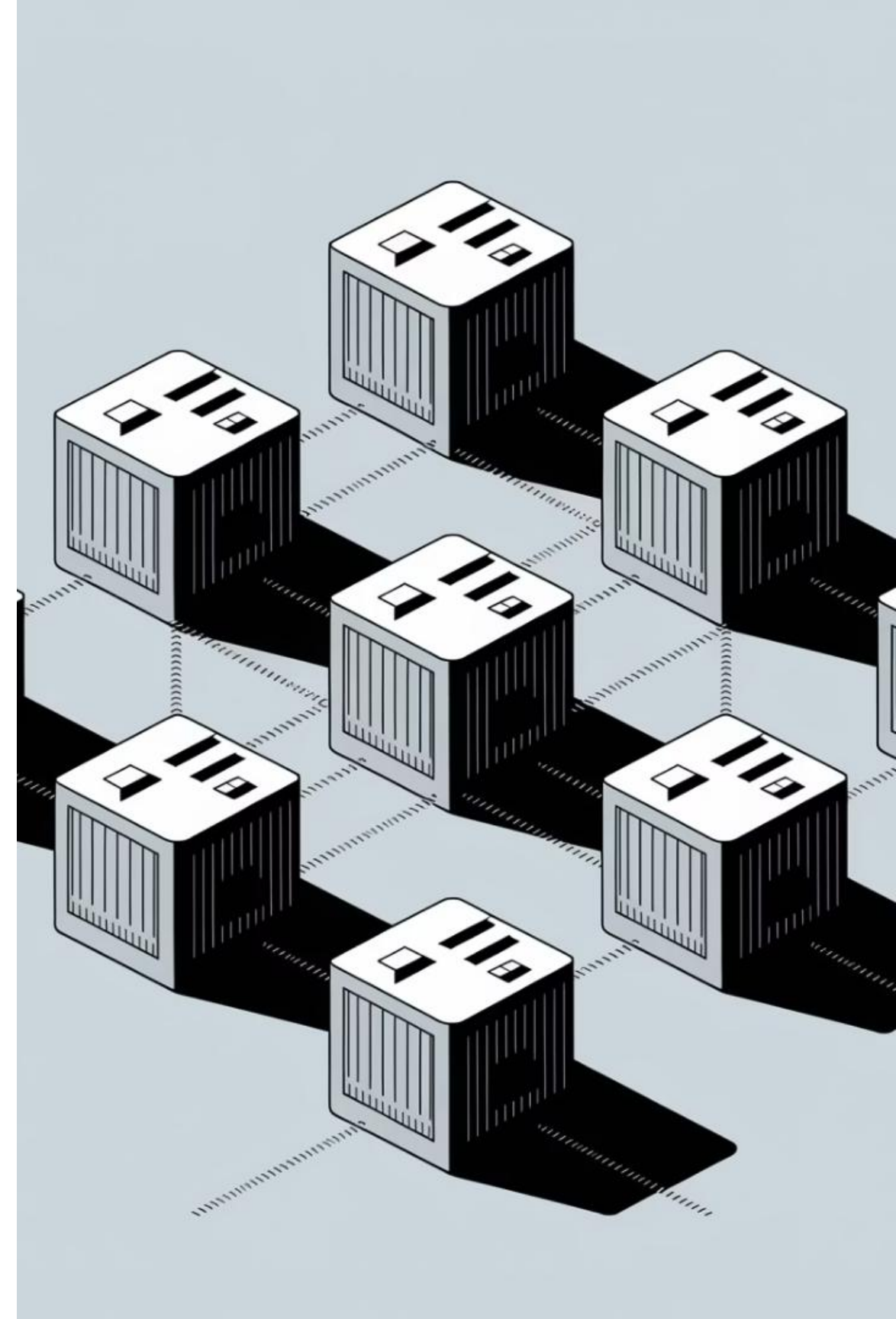
Despliegue en la Nube

Ampliamente utilizados para desplegar aplicaciones en la nube debido a su portabilidad y facilidad de escalado con herramientas como Kubernetes.



Integración y Entrega Continua

Ideales para automatizar pruebas, construcción y despliegue de aplicaciones, asegurando consistencia en cada paso del ciclo de vida del software.



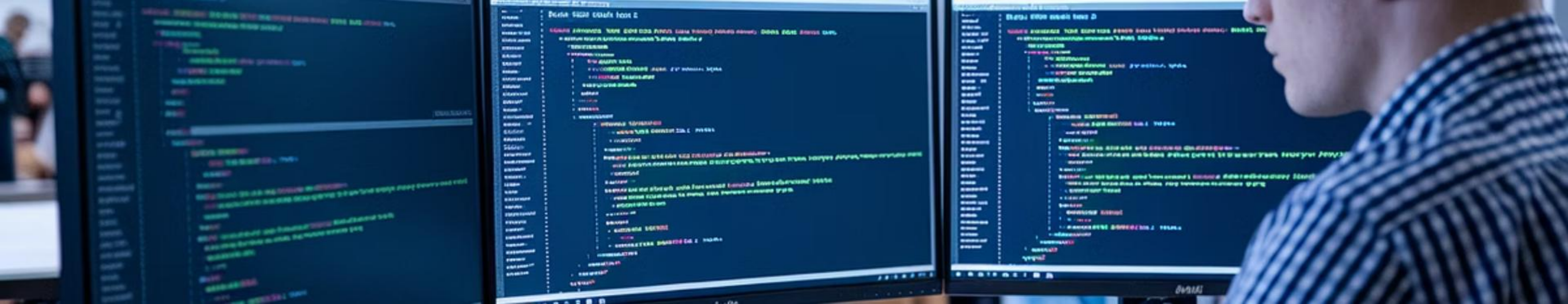
Computación sin Servidor (Serverless)

Definición

La computación sin servidor es un modelo de ejecución en la nube que permite a los desarrolladores ejecutar funciones o fragmentos de código sin preocuparse por la infraestructura subyacente, como servidores o máquinas virtuales.

Los proveedores de servicios en la nube manejan automáticamente la infraestructura, el aprovisionamiento y la gestión de recursos, permitiendo que los desarrolladores se centren exclusivamente en el código.





¿Qué es un Servicio de Cómputo sin Servidor?



Ejecución de Código

Permite ejecutar código en respuesta a eventos sin tener que gestionar servidores. Los proveedores gestionan automáticamente el escalado, aprovisionamiento y ejecución.



Gestión Automática

La infraestructura es completamente gestionada por el proveedor de la nube, eliminando la necesidad de preocuparse por servidores o su mantenimiento.



Respuesta a Eventos

Las funciones se ejecutan en respuesta a eventos específicos, como cambios en bases de datos, cargas de archivos o solicitudes HTTP.

Características y Beneficios del Serverless



Escalabilidad Automática

Los servicios sin servidor escalan automáticamente según la demanda, eliminando la necesidad de ajustar manualmente los recursos para manejar el tráfico.



Costo Eficiente

Se paga solo por el tiempo que el código se ejecuta, eliminando la necesidad de mantener servidores en ejecución constantemente, lo que reduce significativamente los costos.



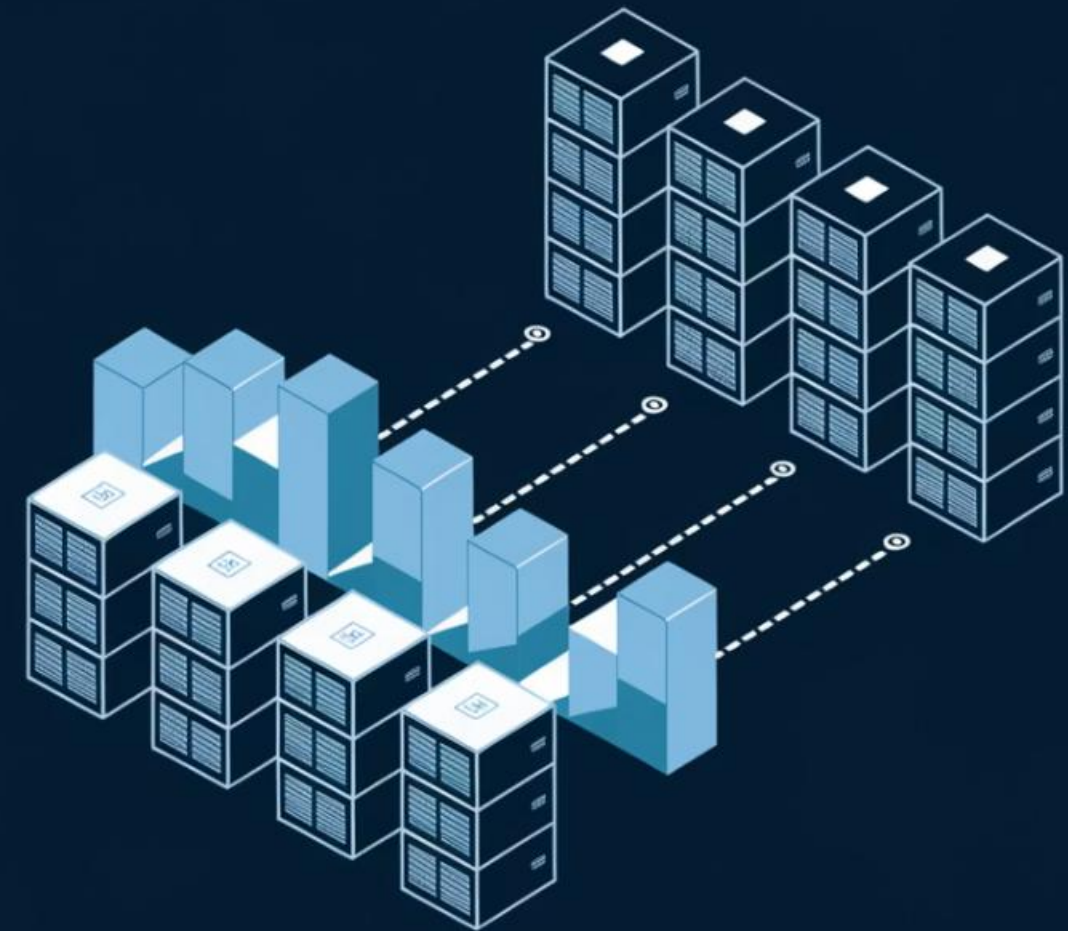
Desarrollo Rápido

Los desarrolladores pueden centrarse en escribir código mientras el proveedor maneja la infraestructura, acelerando los ciclos de desarrollo y mejorando la productividad.

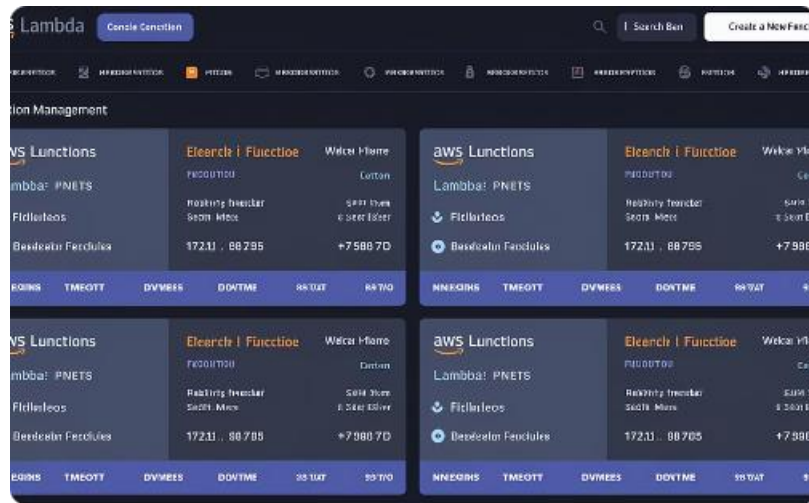


Alta Disponibilidad

Los proveedores gestionan la infraestructura de manera redundante, asegurando alta disponibilidad y tolerancia a fallos sin intervención manual.

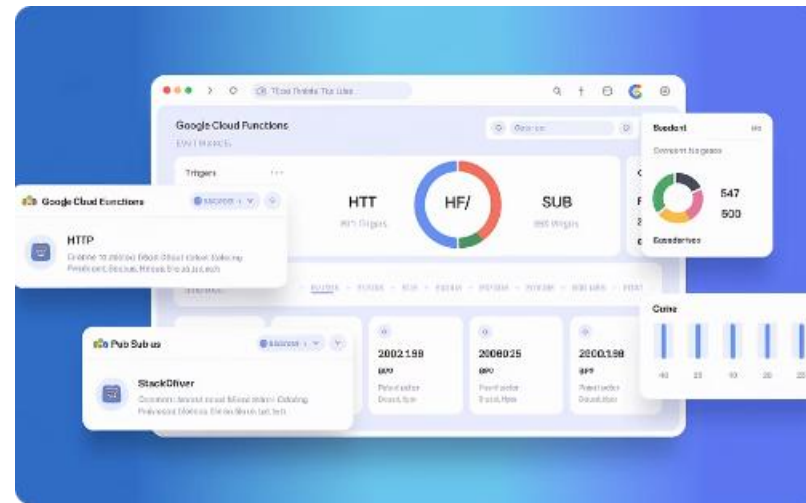


Ejemplos de Servicios sin Servidor en la Nube



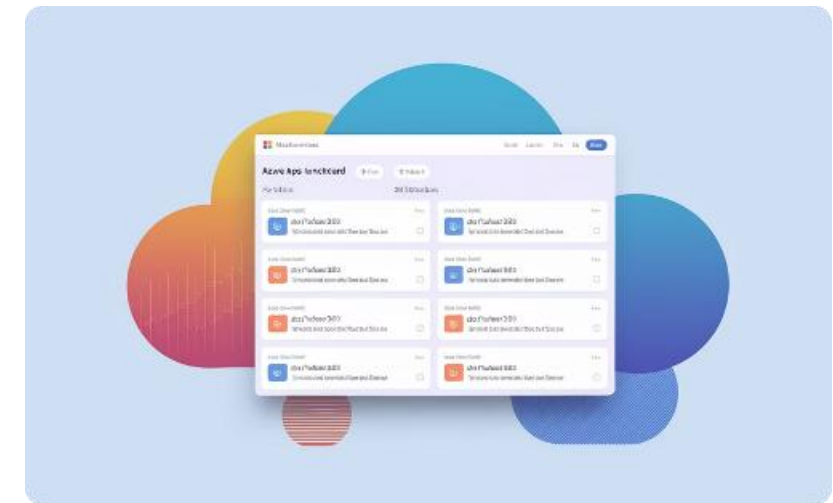
AWS Lambda

Pionero en el modelo serverless que permite ejecutar funciones en respuesta a eventos como cambios en bases de datos o solicitudes HTTP. Soporta múltiples lenguajes y se integra con el ecosistema AWS.



Google Cloud Functions







Permite ejecutar código en respuesta a eventos de Google Cloud. Orientado a simplificar flujos de trabajo en CI/CD y aplicaciones basadas en microservicios, con soporte para Node.js, Python, Go y Java.



Azure Functions

Propuesta serverless de Microsoft con profunda integración en su ecosistema. Soporta múltiples lenguajes y ofrece características avanzadas como Durable Functions para flujos de trabajo complejos.

Ejemplos de Servicios sin Servidor en la Nube

Característica	AWS Lambda	Google Cloud Functions	Azure Functions
 Proveedor	Amazon Web Services (AWS)	Google Cloud Platform (GCP)	Microsoft Azure
 Lenguajes soportados	Node.js, Python, Java, Go, C#, Ruby	Node.js, Python, Go, Java	C#, JavaScript, Python, Java, PowerShell
 Integración principal	Ecosistema AWS (S3, DynamoDB, API Gateway)	Ecosistema Google (Cloud Storage, Pub/Sub)	Ecosistema Microsoft (Azure DevOps, Office 365)
 Modelo de precios	Pago por invocación y duración	Pago por invocación y duración	Pago por invocación, duración y planes premium
 Escalabilidad	Automática	Automática	Automática, con opciones premium para control
 Casos de uso destacados	Backend de microservicios, procesamiento de datos en tiempo real	Automatización de flujos CI/CD, backend ligero	Automatización de flujos empresariales, backend en entornos Microsoft

Casos de Uso Comunes del Serverless

Procesamiento de Datos

Ideal para procesar flujos de datos entrantes, análisis de registros o transformación de archivos en tiempo real

Chatbots y Asistentes

Construcción de chatbots que procesan mensajes y responden en tiempo real



Backends para Aplicaciones

Construcción de APIs escalables para aplicaciones móviles o web

Automatización de Tareas

Automatización de procesos como CI/CD o sincronización entre sistemas

Factores a Considerar al Elegir un Servicio de Cómputo

Rendimiento

El rendimiento se refiere a la eficiencia y rapidez con la que un servicio en la nube puede ejecutar aplicaciones y procesar datos. Un rendimiento óptimo es crucial para mantener la productividad y ofrecer una experiencia satisfactoria a los usuarios.

Escalabilidad

La escalabilidad es la capacidad del servicio en la nube para adaptarse a cambios en la demanda, permitiendo aumentar o disminuir recursos según las necesidades del negocio.

Costos Asociados

Los costos asociados a los servicios de cómputo en la nube pueden variar significativamente según el proveedor, el modelo de servicio y el uso específico. Una comprensión clara de la estructura de precios es vital para evitar gastos inesperados.

Actividad Práctica Guiada: Configuración de un Servidor NGINX con Docker

Objetivo: Configurar un servidor NGINX dentro de un contenedor Docker que sirva una página con el mensaje "¡Hola Mundo Docker!" al acceder a la URL en el navegador.

Instrucción: Sigue los pasos detallados a continuación para instalar Docker, crear un archivo de configuración de NGINX, construir una imagen Docker con la configuración y ejecutar el contenedor. Al final, debes ser capaz de acceder a la página del servidor en tu navegador y ver el mensaje "¡Hola Mundo Docker!".

Detalle: En esta actividad, se aprenderá a crear un servidor básico con NGINX utilizando Docker. Se configurará un contenedor Docker que ejecutará NGINX, y se personaliza la respuesta del servidor para que muestre un mensaje en HTML simple. Esta es una actividad práctica que combina Docker y NGINX, y es ideal para aprender a trabajar con contenedores y servidores web.

Actividad Práctica Guiada: Configuración de un Servidor NGINX con Docker

Crear un directorio para el proyecto: Crea un directorio donde guardarás los archivos necesarios para la actividad:

Dentro de la carpeta crear el archivo **nginx.conf**, y pegar el siguiente contenido correspondiente a la configuración de virtualhost de Nginx:

A screenshot of a code editor window with a dark background and light-colored text. The code is Nginx configuration for a virtual host. It includes an 'events' block, an 'http' block containing a 'server' block. The 'server' block has a 'listen 80;' directive and a 'location /' block with a 'return 200 'Hola Mundo!';' directive. The code is syntax-highlighted with colors: 'events' is purple, 'http' is blue, 'server' is purple, 'listen' is blue, 'location' is purple, 'return' is yellow, '200' is red, and 'Hola Mundo!' is blue. The code is enclosed in curly braces to indicate block structure.

```
events {}  
  
http {  
    server {  
        listen 80;  
        location / {  
            return 200 'Hola Mundo!';  
        }  
    }  
}
```


Actividad Práctica Guiada: Configuración de un Servidor NGINX con Docker

Crear el archivo Dockerfile: En el mismo directorio, crea un archivo llamado Dockerfile con el siguiente contenido:

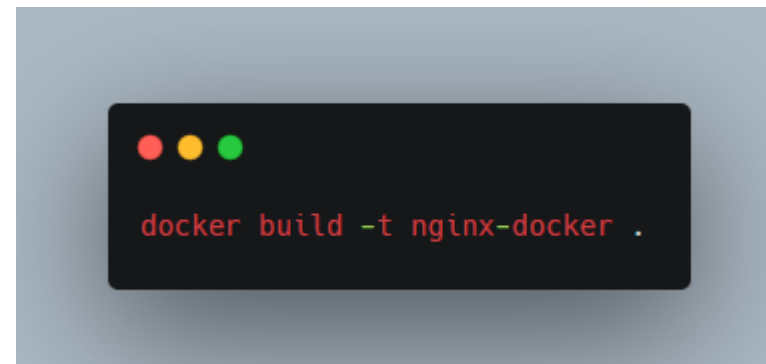
A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. It displays the following Dockerfile instructions:

```
FROM nginx:alpine
COPY nginx.conf /etc/nginx/nginx.conf
EXPOSE 80
```

Actividad Práctica Guiada: Configuración de un Servidor NGINX con Docker

Construir la imagen Docker: En el directorio donde están los archivos Dockerfile y nginx.conf, ejecuta el siguiente comando para construir la imagen Docker:

Nota: Antes de ejecutar cualquier comando en sistema windows, debe estar abierto el software de escritorio.



Actividad Práctica Guiada: Configuración de un Servidor NGINX con Docker

Luego de la ejecución de la línea de comandos anterior debería aparecer algo como esto:

```
MINGW64 ~/OneDrive/Documentos/taller
$ docker build -t nginx-docker .
#0 building with "desktop-linux" instance using docker driver

#1 [internal] load build definition from Dockerfile
#1 transferring dockerfile: 104B 0.0s done
#1 DONE 0.1s

#2 [internal] load metadata for docker.io/library/nginx:alpine
#2 DONE 0.0s

#3 [internal] load .dockerignore
#3 transferring context: 2B done
#3 DONE 0.1s

#4 [internal] load build context
#4 transferring context: 231B done
#4 DONE 0.1s

#5 [1/2] FROM docker.io/library/nginx:alpine
#5 CACHED

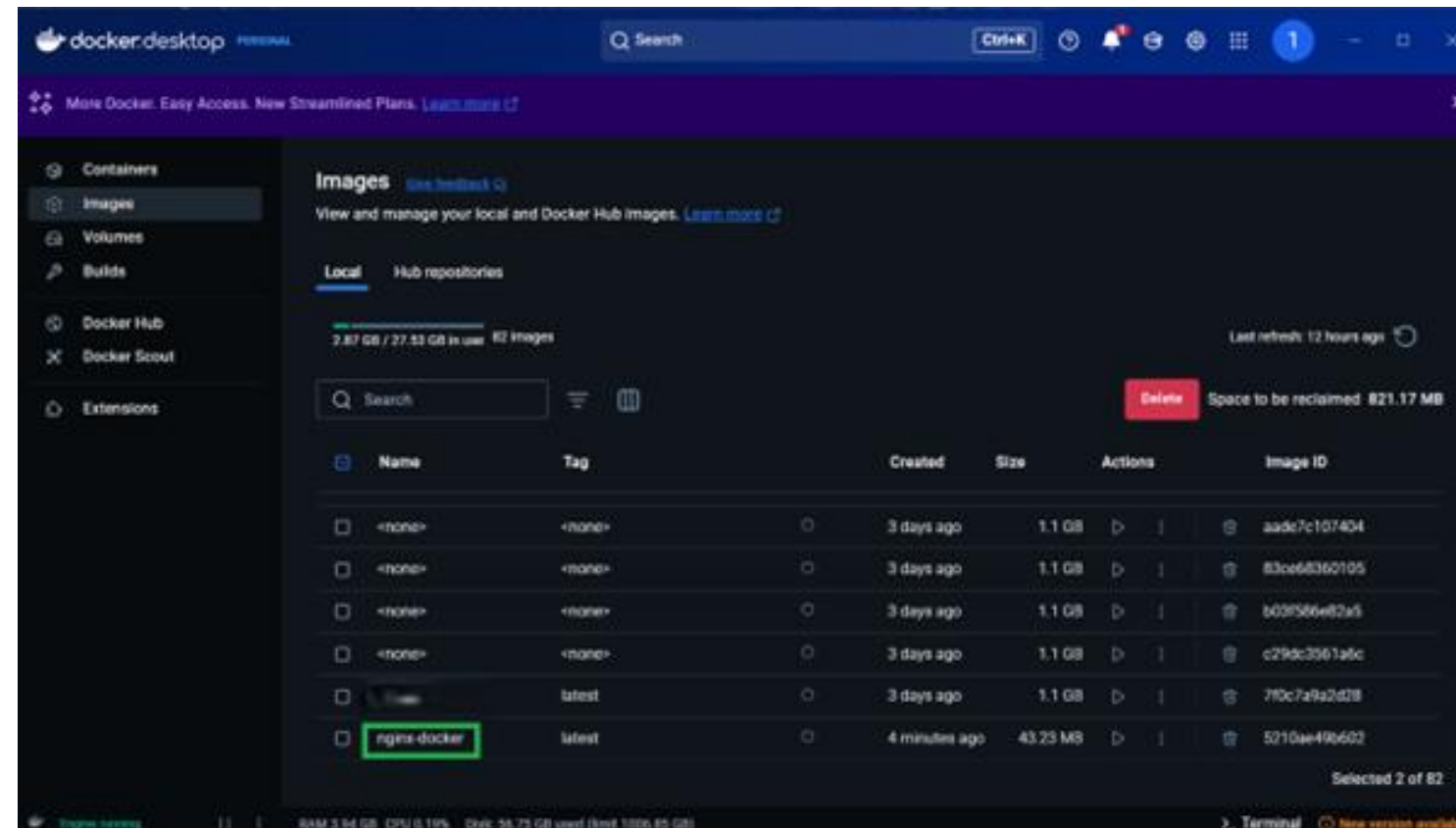
#6 [2/2] COPY nginx.conf /etc/nginx/nginx.conf
#6 DONE 0.1s

#7 exporting to image
#7 exporting layers 0.1s done
#7 writing image sha256:5210ae49b602cf053aad0e659cbc2593c9f471eb12b928cec22c00370706bed3 done
#7 naming to docker.io/library/nginx-docker 0.0s done
#7 DONE 0.1s

view build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/15x6qcoy17b7apsvxehyu6eo2
```

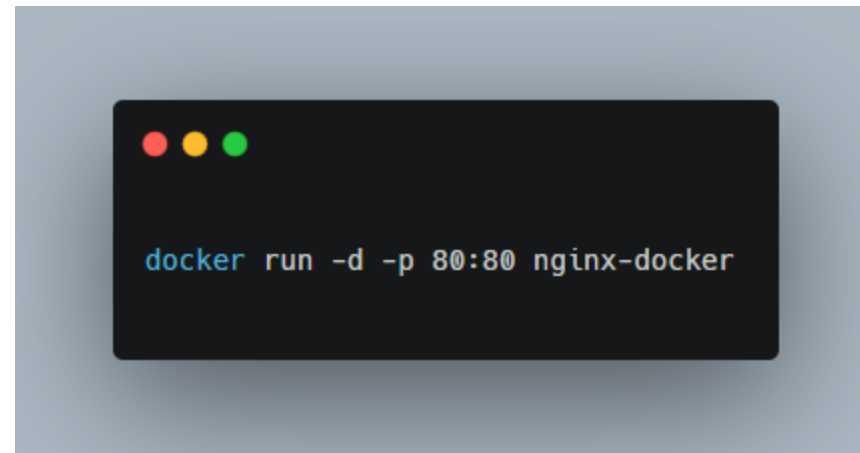

Actividad Práctica Guiada: Configuración de un Servidor NGINX con Docker

En nuestro Docker Desktop se puede observar en la sección de “imágenes” la imagen recién creada que contiene lo que configuramos en nuestro Dockerfile



Actividad Práctica Guiada: Configuración de un Servidor NGINX con Docker

Ejecutar el contenedor Docker: Teniendo la imagen ya construida, ejecuta el siguiente comando para iniciar el contenedor Docker en el puerto 80:

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. The terminal displays the command `docker run -d -p 80:80 nginx-docker` in a light blue monospace font.

```
docker run -d -p 80:80 nginx-docker
```

Nota: El formato del puerto corresponde a **<puerto_local>:<puerto_del_contenedor>**. En caso de estar ocupado el puerto 80, verifica cual tienes disponible en un máquina, ejemplo 8001.

Actividad Práctica Guiada: Configuración de un Servidor NGINX con Docker

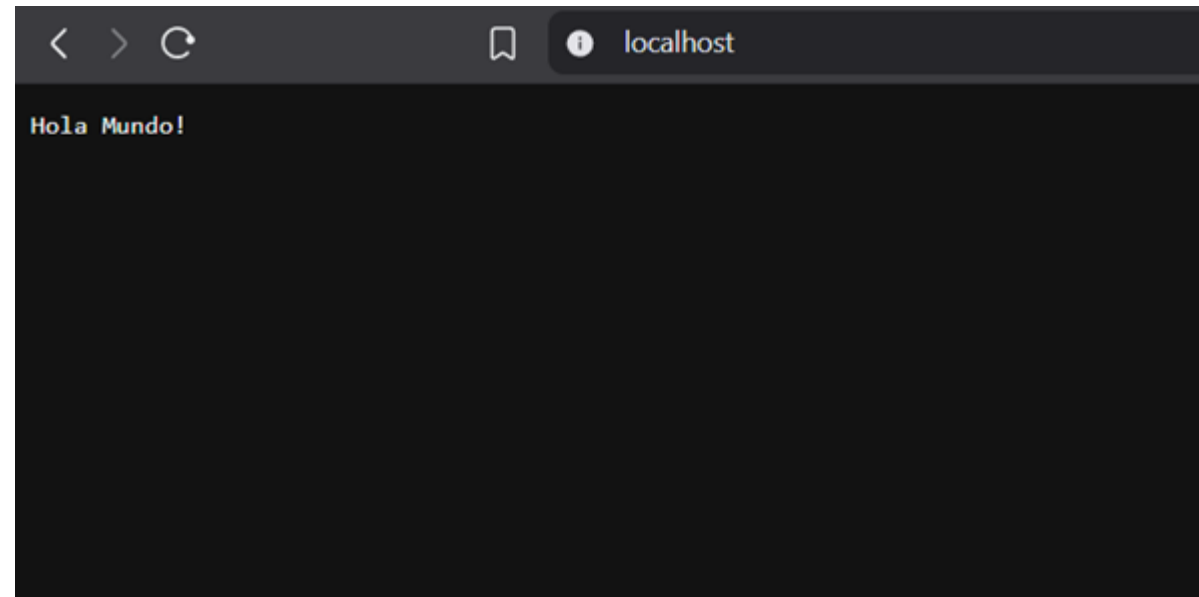
Debería aparecer algo así:

```
MINGW64 ~/OneDrive/Documentos/taller  
$ docker run -d -p 80:80 nginx-docker  
586ef78bc4df348b29f971077c834868b45303b5fae665e3867b4c0a1ed61062
```

Luego ya puedes acceder al servidor en el navegador, y verás el mensaje “**¡Hola Mundo Docker!**”

<http://localhost>

Actividad Práctica Guiada: Configuración de un Servidor NGINX con Docker



Conclusión:

Gracias a los contenedores Docker, podemos levantar un *virtualhost* con una configuración básica de manera rápida y sencilla. Este ejemplo muestra lo fácil que sería poner tu código en línea sin necesidad de realizar instalaciones en tu máquina local. Olvídate del estrés de trabajar en varios proyectos a la vez y tener que instalar los paquetes correspondientes para cada lenguaje. Docker aligera tu carga de trabajo y reduce los tiempos de desarrollo.

Material de Apoyo

- Video: [Como instalar Docker en Windows](#)
- Video: [Como instalar Docker en Linux](#)
- Enlace: [Serverless](#)
- Enlace: [Cloud Functions GCP](#)



Reflexiones sobre la computación en la nube



¿Cómo puede afectar la computación en la nube la estructura de trabajo en las empresas?



¿Qué factores son más importantes al considerar la migración a la nube?



¿Cómo imaginas que cambiará nuestra interacción diaria con la tecnología gracias a la nube?

