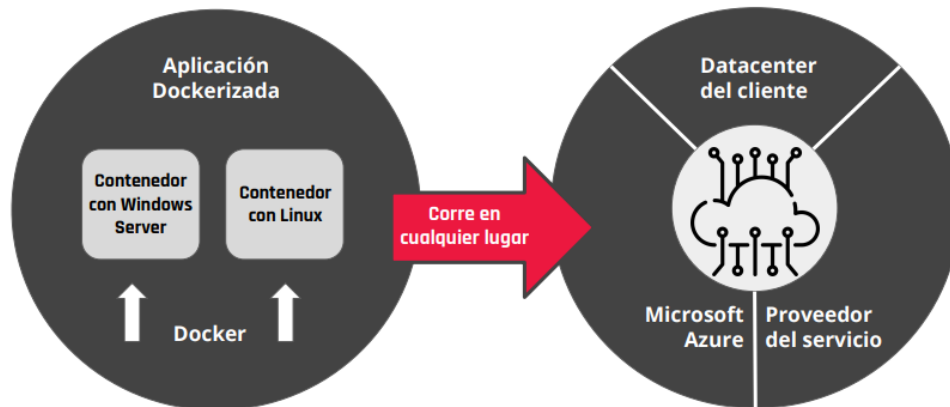


¿Por qué Docker?



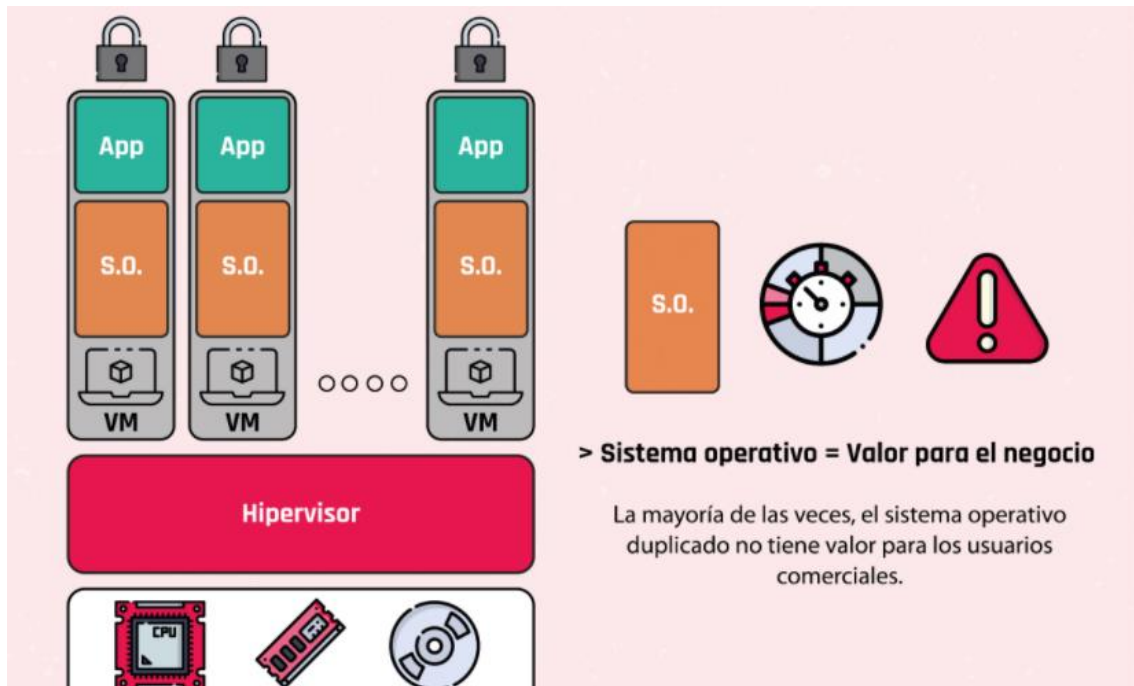
¿Por qué Docker?

- Pueden crear cualquier aplicación en cualquier idioma usando cualquier stack (sistema operativo).
- Las aplicaciones dockerizadas pueden ejecutarse en cualquier lugar y sobre cualquier cosa.
- No más al dicho: “Funciona en mi máquina”.
- Encapsulando dependencias los desarrolladores y los administradores de sistemas trabajan mejor en conjunto

Vocabulario de Docker

- *Host*: Una máquina virtual que ejecuta Docker daemon para alojar una colección de contenedores Docker.
- *Cliente*: aquí se ejecutan los comandos que están siendo ejecutados. (cliente-servidor).
- *Imagen*: Una colección ordenada de sistemas de archivos (capas) que se utilizarán al crear una instancia de un contenedor.
- *Contenedor*: Una instancia en tiempo de ejecución de una imagen.
- *Registro*: Una colección de imágenes de Docker.

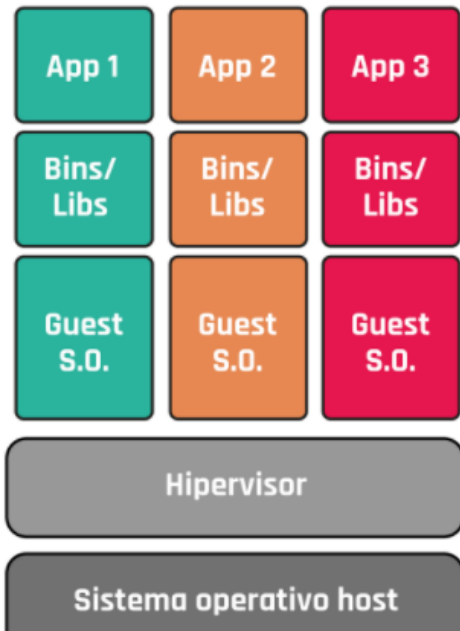
Desafíos con la virtualización.



Comparativa entre máquinas virtuales y containers.

Máquina Virtual vs. Contenedores

Máquina Virtual



Contenedores



Contenedores

Entornos físicos

- Aplicaciones construidas e implementadas tradicionalmente en sistemas físicos con relación 1:1.
- Las nuevas aplicaciones a menudo requieren nuevos sistemas físicos para el aislamiento de recursos.

Entornos virtuales

- Mejor utilización e implementación de aplicaciones, más rápidas que en un entorno físico tradicional.
- Las aplicaciones implementadas en máquinas virtuales son muy compatibles.

Entornos físicos/ virtuales

- Aceleran aún más la implementación de la aplicación.
- Reducen el esfuerzo para implementar aplicaciones.
- Optimizan el desarrollo y las pruebas.
- Menores costos asociados con la implementación de aplicaciones.
- Incrementan la consolidación de servidores

Plataforma de Docker



Docker Engine



Docker Hub



Docker Trusted Registry



Docker Client



Docker Images

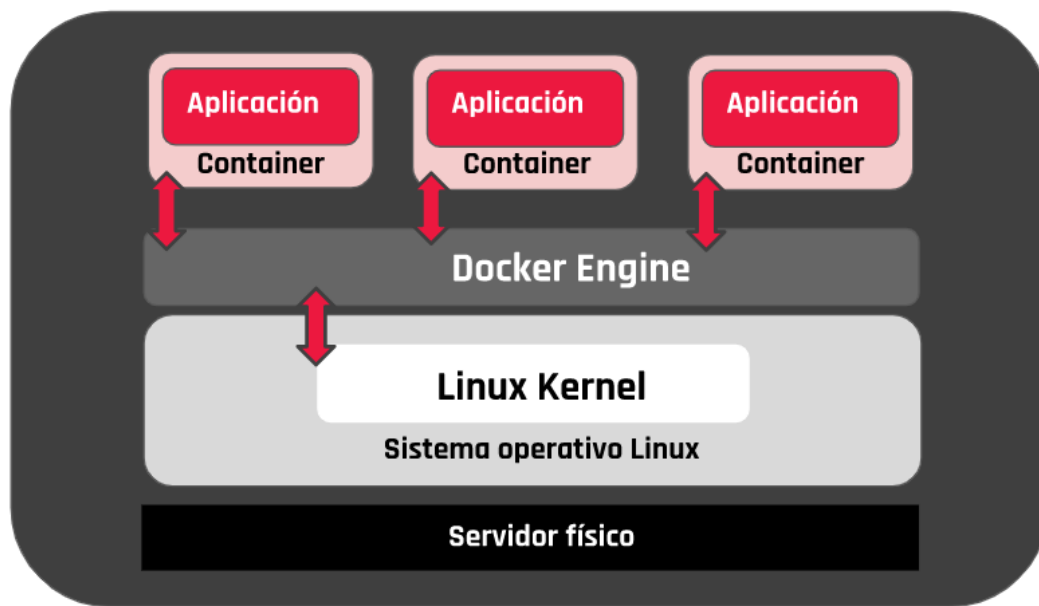


Docker Containers

Docker Engine, también conocido como **Docker Daemon**, es el programa que permite construir, enviar y ejecutar contenedores. Utiliza espacios de nombres y grupos de control del kernel de Linux para proporcionar un entorno de tiempo de ejecución aislado para cada aplicación.

Docker Hub Es un registro en línea de imágenes de Docker.

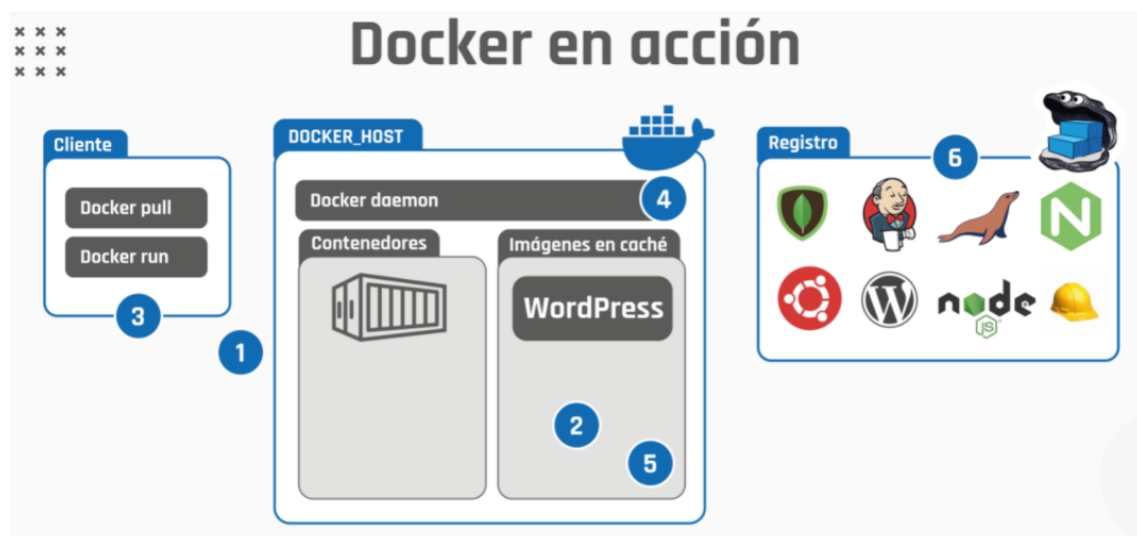
Docker Trusted Registry es un registro privado en el sitio para imágenes de Docker.



Docker Client es el que toma las entradas del usuario y las envía al daemon. El cliente y el daemon pueden ejecutarse en el mismo host o en diferentes hosts.

Docker Images es una plantilla de solo lectura utilizada para crear contenedores. Contiene un conjunto de instrucciones para crear los contenedores.

Por último, **Docker Containers** es una plataforma de aplicación aislada basada en una o más imágenes que contiene todo lo necesario para ejecutar una aplicación.

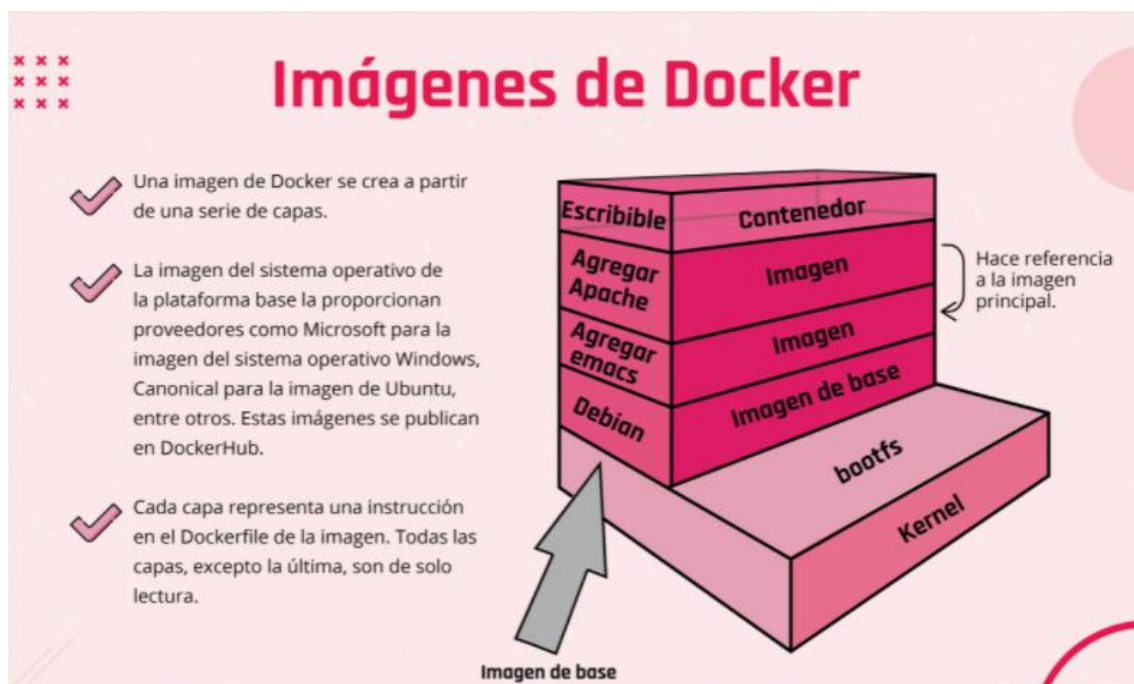


¿Qué es un registro Docker?

Los registros Docker sirven para almacenar las diversas imágenes Docker que utilizemos en nuestro sistema. De esta forma podremos subir imágenes nuevas a los registros o descargarlas cuando las necesitemos en alguna máquina Docker.

El registro es como una estantería donde las imágenes se almacenan y están disponibles para extraerlas con el fin de compilar contenedores que ejecuten servicios o aplicaciones web. Hay registros de Docker privados a nivel local y en la nube pública. Docker Hub es que un registro público mantenido por Docker; junto con *Docker Trusted Registry*, una solución a nivel empresarial.

Colocar imágenes en un registro permite almacenar fragmentos de la aplicación que son estáticos e inmutables, incluidas todas sus dependencias a nivel de marco. Después, esas imágenes se pueden versionar e implementar en varios entornos y, por lo tanto, proporcionar una unidad de implementación coherente.



Taxonomía básica en Docker



¿Qué es Dockerfile?

Es un archivo de texto simple con un conjunto de comandos o instrucciones. Estos comandos / instrucciones se ejecutan sucesivamente para realizar acciones en la imagen base para crear una nueva imagen de la ventana acoplable.



Instrucciones básicas de Dockerfile

DE: Define la imagen base para usar e iniciar el proceso de construcción.

CORRER: Toma el comando y sus argumentos para ejecutarlo desde la imagen.

CMD: Función similar a un comando run, pero se ejecuta solo después de que se crea una instancia del contenedor.

PUNTO DE ENTRADA: Se dirige a su aplicación predeterminada en la imagen cuando se crea el contenedor.

AÑADIR: Copia los archivos de origen a destino (dentro del contenedor).

ENV: Establece variables de entorno.

Etiquetas de imagen

Los tags o etiquetas sirven para identificar las versiones de las imágenes, a la hora de listar las imágenes se listan con su tag o etiqueta asociado. Pueden agrupar sus imágenes usando nombres y etiquetas (si no proporcionan ninguna etiqueta, se asume el valor predeterminado de la última).

Docker Hub

Docker Hub es un servicio de registro de repositorios proporcionado por Docker Inc. Compartir y colaborar son sus premisas

¿Para qué sirve Docker Hub?

Nos permite extraer y enviar imágenes de la ventana acoplable hacia y desde Docker Hub. Podemos tratar esto como un GitHub, donde obtenemos y enviamos nuestro código fuente, pero en el caso de Docker Hub, descargamos o publicamos nuestras imágenes de contenedor. Es un repositorio en línea basado en la nube que almacena ambos tipos de repositorios, es decir, el repositorio público y el privado. Los repositorios públicos son accesibles para todos, pero el privado es accesible para el propietario interesado de los repositorios; también hay un costo asociado si almacenamos más de un cierto número de repositorios como privado.

Características de Docker Hub

1. Repositorios de imágenes: Nos ayuda a encontrar y extraer imágenes de contenedores de Docker Hub.
2. Equipo y organizaciones: Nos permite crear grupos de trabajo e impulsar los repositorios privados, que están disponibles para su uso únicamente dentro de nuestra organización. De esta forma, hemos gestionado el acceso a nuestros repositorios privados de imágenes de contenedores.
3. Integración de GitHub y Bitbucket: Permite la integración con repositorios de código fuente como GitHub y BitBucket.
4. Construcciones automatizadas: Si se ha enviado algún cambio en el código fuente a los repositorios de código fuente, automáticamente detecta y crea imágenes de contenedor desde GitHub o BitBucket y las envía a Docker Hub.
5. Webhooks: Una vez que hemos enviado nuestras imágenes con éxito, con la ayuda de un webhook, desencadena una acción para integrar Docker Hub con otros servicios.
6. Imágenes oficiales y del editor: Las imágenes de alta calidad proporcionadas por los dockers se consideran imágenes oficiales y se pueden extraer y utilizar. Del mismo modo, las imágenes de alta calidad proporcionadas por proveedores externos son imágenes del editor, también llamadas imágenes certificadas, que brindan soporte y garantía de compatibilidad con Docker Enterprise.

Hay dos formas de buscar imágenes y repositorios públicos desde Docker Hub, es decir, podemos buscarlo en el sitio web de Docker Hub o usar la herramienta de línea de comandos y ejecutar el siguiente comando. Consideremos que queremos buscar en la imagen del repositorio de MySQL.

Descargar una imagen

Podemos descargar una imagen del comando Docker Hub usando pull comando de la siguiente manera:

```
# docker pull mysql
```

Si ya tenemos mysql image en nuestra máquina, el comando anterior actualizará automáticamente la imagen a la última versión. Una cosa a tener en cuenta aquí es que si notamos la salida del comando de búsqueda de la ventana acoplable, hay muchas imágenes de MySQL en Docker Hub, y eso se debe a que cualquiera puede enviar una imagen. Pero depende de nosotros, cuál usar en función de nuestro caso de uso; y necesitamos el apropiado. Digamos que queremos extraer una imagen bitnami / mysql:

```
# docker pull bitnami/mysql
```

Crear una imagen El proceso de crear una imagen requiere un Dockerfile. Podemos pensar un Dockerfile como un manual de instrucciones que le dice a Docker qué ensamblar. En resumen, es un archivo de configuración que sigue ensamblando instrucciones. ¿Cómo funciona? Docker lee las instrucciones de un Dockerfile y crea imágenes automáticamente. La imagen de Docker es un sistema de archivos en capas y consta de varias capas de solo lectura y cada capa de una imagen de Docker representa las instrucciones de un Dockerfile. Sigamos los pasos a continuación para crear una imagen usando Dockerfile que especifica la configuración de nuestra aplicación

```
# sudo vim Dockerfile
```

Crear una imagen

FROM (de): define la imagen base que se utilizará.

MAINTAINER (mantenedores): persona que va a mantener esa imagen.

RUN (correr): se utiliza para ejecutar la instrucción dada para la imagen. En nuestro caso, primero actualice el sistema y luego instale MySQL.

CMD: se utiliza para ejecutar un comando una vez que se ha lanzado el contenedor.

COPY (copiar): se utiliza para copiar un archivo de nuestro sistema operativo host al contenedor de la ventana acoplable.

EXPOSE (exponer): se utiliza para especificar el número de puerto en el que el contenedor ejecutará su proceso

Empujar una imagen.

Una vez que nuestra imagen se ha creado correctamente y se está ejecutando, podemos enviarla a Docker Hub mediante el comando push.

```
docker push asadali08527/first-repo
```


Imágenes certificadas por Docker

Estas son las imágenes oficiales impulsadas por proveedores o contribuyentes. Una imagen solo puede ser certificada por Docker Hub si su contenido cumple con las reglas, estándares y leyes proporcionadas por Docker Hub. En resumen, esa imagen debe pasar ciertas pruebas de referencia. Docker Hub proporciona `inspectDockerImage`, herramienta a través de la cual un proveedor puede autocertificar las imágenes y los complementos (por lo general, el proveedor o contribuyente publica sus complementos para registrar volúmenes y redes).

Imágenes populares en Docker Hub. Hay muchas imágenes seleccionadas y optimizadas disponibles en Docker Hub. La popularidad de estas imágenes depende de varios factores, presencia en el mercado, calificaciones, puntajes de satisfacción, entre otros. Para obtener una lista detallada de los repositorios más populares, pueden navegar hasta el Docker Hub sitio web. El uso de una imagen también depende del sistema operativo y su arquitectura.

Imágenes populares en Docker Hub. Si conocemos para qué sistema operativo y arquitectura se utilizarán las imágenes, debemos considerar los factores clave a continuación antes de extraer una imagen.

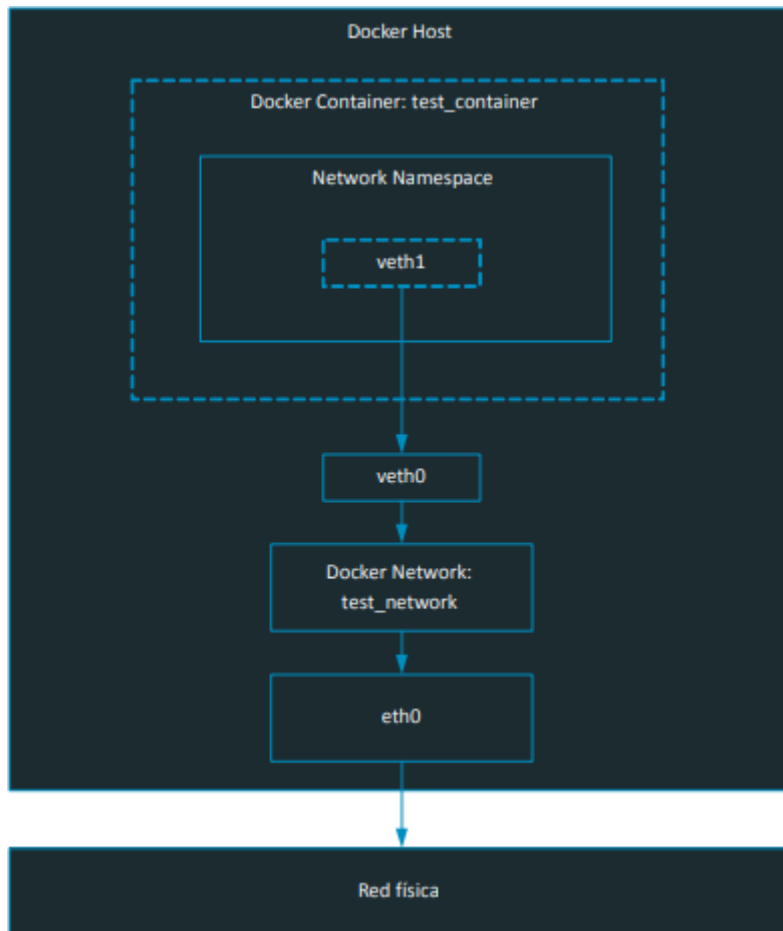
- Buscar una versión específica utilizando etiquetas (principalmente la última).
- Optar por el que tenga máxima cantidad descargas y estrellas.
- Buscar la fecha de su última actualización.
- Si es posible, verificar su tipo, ya sea del editor verificado u oficial (Docker Certified).

Redes de Docker

Nos vamos a enfocar en el driver llamado bridge, ya que con el nuevo enfoque de Docker orientado a habilitar a los desarrolladores es el que mayoritariamente le darán uso en sus computadoras. Todos los otros drivers están orientados a ambientes productivos, pero ese ámbito hoy está dominado por Kubernetes, que tienen otro modelo de Networking completamente diferente.

- Las redes de tipo “bridge” son locales y exclusivas del host en donde fueron creadas.
- Son el tipo de red por defecto en Docker.
- Simula la creación de switches o hubs, de nivel 2 (en el modelo OSI). Podemos utilizar herramientas como ‘brctl’ en Linux para ver el funcionamiento interno.
- Luego de la instalación de Docker se crea por defecto una red de tipo bridge llamada “bridge”, las buenas prácticas indican que esta no debe ser utilizada y en su lugar se deben crear nuevas redes para usos específicos.

Bridge Driver en acción



Resolución de nombres

Las IPs son efímeras y más aún en el mundo de los containers —estos también deberían serlo. De modo que necesitamos una forma que le permita a un container hablar con otro sin conocer necesariamente la IP que le fue asignada. ¿Qué hace Docker por nosotros? Para todas las redes que creemos —las redes creadas por defecto no nos proveen esta ventaja—, y para todos los containers a los que se le asigne un nombre de manera deliberada —usando `--name` en `docker create`— conectados a estas, Docker nos provee con un servicio de resolución de nombres dentro del ámbito de la red misma utilizando el nombre del container para identificarlo por medio del servicio DNS.

`docker network - create`: nos permite crear una red.

`--internal`: agrando esta opción la red creada será del tipo 'privada'.

`- connect`: nos permite conectar un container existente a una red.

`docker run --network` : nombre de la red a la cual conectaremos el container.

`--name`: nombre del container, necesario para que funcione la resolución de nombres.

`-p ::` nos permite publicar ciertos puertos de un container en el host para poder acceder al servicio dentro del container.

-P: nos permite publicar todos los puertos definidos en la especificación del container (Dockerfile) en puertos aleatorios del host.

Docker Compose

Aprender a utilizar Docker Compose

Docker Compose es una herramienta que permite simplificar el uso de Docker. A partir de archivos YAML es más sencillo crear contenedores, conectarlos, habilitar puertos, volúmenes, etc. Con Compose podemos crear diferentes contenedores y al mismo tiempo, en cada contenedor, diferentes servicios, unirlos a un volumen común, iniciarlos y apagarlos, etc. Es un componente fundamental para poder construir aplicaciones y microservicios.

En vez de utilizar Docker vía una serie inmemorable de comandos Bash y scripts, Docker Compose te permite mediante archivos YAML para poder instruir al Docker Engine a realizar tareas, programáticamente. Y esta es la clave, la facilidad para dar una serie de instrucciones y luego repetirlas en diferentes ambientes.

¿Por qué utilizar Docker con Docker Compose?

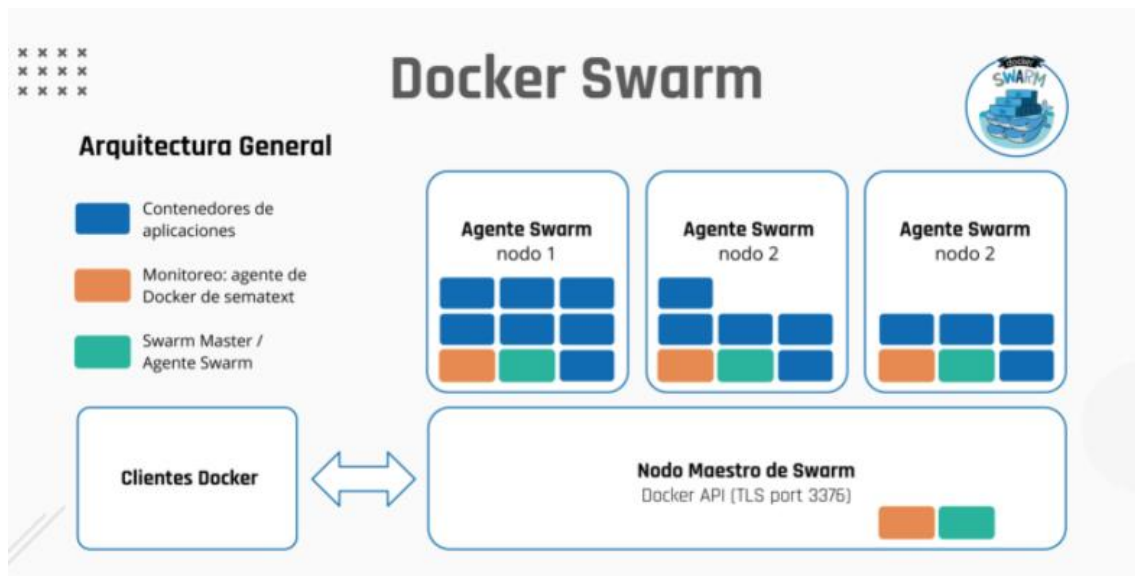
Docker y Docker Compose nos brindan algunos beneficios:

- No necesitamos instalar ni mantener software adicional en nuestro equipo.
- Podemos tener todo nuestro entorno de desarrollo en un único repositorio, por ejemplo, tenemos el back end, el front end y las configuraciones de la base de datos en un mismo repositorio lo que facilita a los desarrolladores el poder colaborar de mejor manera en el proyecto.
- Levantar todo el entorno de desarrollo se limita a un solo comando `docker-compose up`.

Pero no todo es color de rosa, también tenemos una desventaja que está enfocada al rendimiento, ya que a pesar de que los contenedores están enfocados a ser eficientes, siguen consumiendo recursos de la máquina anfitriona tales como procesador y memoria por lo que si la cantidad de contenedores que están corriendo al mismo tiempo es grande o si los contenedores son pesados al momento de crearse y levantarse, podrían llevar a cuelgues del sistema y cosas similares.

Conclusión

Podemos probar con distintos entornos y lenguajes, la idea es conseguir optimizar al máximo nuestro entorno de desarrollo y enfocarnos en el principal objetivo que es escribir código de calidad.



Swarm es una herramienta integrada en Docker —aplicación para la creación de contenedores— que permite agrupar una serie de hosts de Docker en un clúster y gestionarlos de forma centralizada, así como orquestar contenedores.

Mejores prácticas en Docker para desarrolladores

Utilizar imágenes oficiales: En Docker Hub vamos a encontrar dos tipos de imágenes. Las oficiales, publicadas por organizaciones (por ejemplo, Ubuntu) o las publicadas por usuarios individuales. ¿Cómo diferenciarlas? Las imágenes oficiales están etiquetadas como tales en el sitio de Docker Hub, si buscamos Ubuntu vamos a ver que las imágenes publicadas por Canonical están etiquetadas como Docker Official Image.

Pero la manera más práctica para entender si una imagen es oficial o no, es su nombre. Las imágenes no oficiales incluyen el nombre del usuario que las publicó, mientras que las oficiales no. Veamos la diferencia a la hora de ejecutar 'docker pull': Para una imagen oficial: `docker pull ubuntu:latest`

Mientras que para una imagen que no lo es, el comando se vería así: `docker pull usuario/nginx:latest` En el caso de la imagen no oficial, seguramente contenga un servidor NGINX, pero no sabemos —por lo menos sin realizar un análisis más detallado— qué otras modificaciones podría haber incluido el usuario que publicó la imagen.

Utilizar el comando COPY en lugar de ADD:

Ambos permiten agregar o copiar un archivo de una fuente externa a una imagen de Docker. ADD fue el primero en ser creado para la especificación de los Dockerfiles. La gran diferencia reside en que ADD soporta copiar desde distintos orígenes. Por ejemplo, puede agregar un archivo hosteado en la web o un archivo que está en un filesystem de la máquina que está compilando el dockerfile; y también puede desempaquetar un archivo .tar (tarball, un formato de empaquetado nativo de Unix / Linux). El problema es que ADD al hacer un montón de cosas puede generar confusión, incluso algunos inconvenientes técnicos —qué pasa si quiero copiar un archivo .tar a la imagen en lugar de descomprimirlo— y hasta generar problemas de performance. COPY, por el contrario, fue introducido más tarde para cumplir la función específica de copiar un archivo desde un filesystem local a la imagen. Incluso COPY, al tener un propósito más específico, facilita la lectura del dockerfile.

En términos generales se recomienda:

- Usar COPY para copiar archivos locales a la imagen en tiempo de compilación.

- Usar RUN para ejecutar curl y encadenar con otros comandos para bajar un archivo de una fuente web y copiarlo, instalarlo o descomprimirlo.
- Usar ADD cuando se tiene un archivo .tar y se lo quiere desempaquetar dentro de la imagen que se está construyendo.

Usar multi-stage builds:

Esta es una práctica un tanto compleja, pero vale la pena saber que existe. La misma nos permite generar imágenes más pequeñas incluyendo explícitamente solo aquellos componentes necesarios para ejecutar una aplicación.

Hasta ahora vimos Dockerfiles que incluyen un solo 'FROM'. En los multi-stage builds los Dockerfiles tienen dos o más ocurrencias de 'FROM'. Esto lo que produce es varias imágenes etéreas durante el proceso de 'build' y nos permite copiar elementos entre ellas hasta generar una imagen definitiva con solo aquello estrictamente necesario para ejecutar la aplicación.

En esta práctica, por lo general, se usa una primera imagen de base (especificada en el primer FROM), en la que copiamos el código fuente de nuestra aplicación y contamos con las herramientas necesarias para compilarla.

Luego indicamos un segundo dentro del mismo Dockerfile indicamos un segundo FROM, haciendo referencia a una imagen de base que tiene solo lo necesario para ejecutar la aplicación, pero no así las herramientas de compilación. Y copiamos el artefacto generado producto de la compilación en el primer container, al segundo.

No generar dependencias externas:

Los contenedores que construyamos tienen que ser trasladables. No importa en donde corran, deben poder contar con todo lo necesario para ejecutar las aplicaciones que viven en ellos.

Es natural que los desarrolladores se sientan tentados a montar directorios del equipo en el que desarrollan dentro de un container para así acelerar el ciclo de desarrollo. En lugar de tener que compilar la imagen cada vez que desean probar su aplicación y copiar los artefactos necesarios dentro de la imagen. Es más sencillo mapear un directorio local y dejar allí los componentes actualizados. Esta práctica puede llevar a que a la hora de correr las aplicaciones en producción haya dependencias no resueltas.

Concatenar comandos:

Muchas veces nos vamos a encontrar en Dockerfiles con líneas como ésta: RUN apt-get update && apt-get upgrade -y

En lugar de algo como:

RUN apt-get update

RUN apt-get upgrade -y

¿Por qué? Cada línea en un Dockerfile produce una nueva capa en nuestra imagen, de modo que ejecutar estos dos comandos por separado, produciría dos capas.

En este caso, ambos comandos sirven al mismo propósito, actualizar los paquetes instalados en el sistema operativo. Siendo así, tiene sentido concatenarlos, ya que están estrechamente relacionados y ejecutar uno sin ejecutar el otro no tendría sentido. De esta manera optimizamos el uso de recursos y tiempo de compilación.

Más información al respecto de esta práctica en: <https://docs.docker.com/develop/dev-best-practices/#how-to-keep-your-images-small>

Más información al respecto: <https://docs.docker.com/develop/dev-best-practices/>

Introducción cloud computing

Existen diferentes tipos de nubes, dependiendo de la forma de implementación y de la cantidad de clientes que puedan acceder a la información que existe en la nube. Múltiples servicios de los que estamos acostumbrados a usar, como iCloud o Google Drive, están basadas en la computación en la nube.

La computación en la nube —también conocida como servicios en la nube o informática en la nube, entre otros nombres— es la tecnología que permite tener todos nuestros archivos e información en la nube de manera que podamos acceder a ellos desde múltiples dispositivos y sin necesidad de preocuparnos por poseer la capacidad suficiente de almacenamiento. La computación en la nube se ofrece como un servicio web por demanda, que significa que la aplicación software corre en una infraestructura o servidor proveedor para dar servicio a múltiples clientes —estos son los dispositivos mediante los cuales accedemos a la información que se encuentra en la nube—. Este tipo de aplicaciones es accesible mediante cualquier navegador web y el usuario no tiene control sobre él, aunque puede modificar algunos parámetros de configuración.

Historia de cloud computing

Durante la década de los años sesenta, con la aparición del software abierto y la formación de clusters, fundamentalmente en universidades y centros de investigación, surgió la idea de la red de computadores universal. No fue hasta la década de los noventa— a partir de la cual Internet fue accesible para una mayor cantidad de personas— cuando esta tecnología empezó realmente a desarrollarse. Durante esta época comienzan a popularizarse las tecnologías de virtualización, que permiten crear máquinas virtuales haciendo “independiente” el software del hardware. Esto supuso un gran impulso para la computación en la nube. Se eliminaron los problemas de la computación grid (acceso \neq servicio). No es hasta finales de la década de los noventa y principios de este siglo cuando aparecen las primeras empresas —Salesforce.com, Amazon Web Services, etc.— que ofrecen servicios en la nube. El impulso definitivo al cloud computing se produjo gracias a la inmersión en esta tecnología de dos gigantes de la informática: Google y Microsoft.

Definición de cloud computing

La informática en la nube es la entrega bajo demanda de potencia de cómputo, bases de datos, almacenamiento, aplicaciones y otros recursos de IT, a través de Internet con un sistema de precios de pago por uso. Estos recursos se ejecutan en equipos de servidores ubicados en grandes centros de datos en diferentes partes del mundo. Se deja de considerar la infraestructura como hardware y en cambio, se empieza verla (y usarla) como software.

¿Cómo funciona?

Cloud computing se sustenta en 3 pilares fundamentales, esenciales y diferentes que cubren distintas áreas de producción y servicios, ya sea para una persona o empresa.

Software como
servicio

Plataforma como
servicio

Infraestructura
como servicio

Tipos de nubes

En función de su privacidad existen:

Nubes privadas: Accesibles únicamente desde una determinada organización. Gestionadas por la propia organización o por un tercero.

Nubes públicas: Abiertas al público y son propiedad de un proveedor de cloud computing que, adicionalmente, se encarga de gestionarlas. Todas las garantías de privacidad, seguridad y disponibilidad, así como las penalizaciones por incumplimiento, deben estar expresadas en el contrato de servicio.

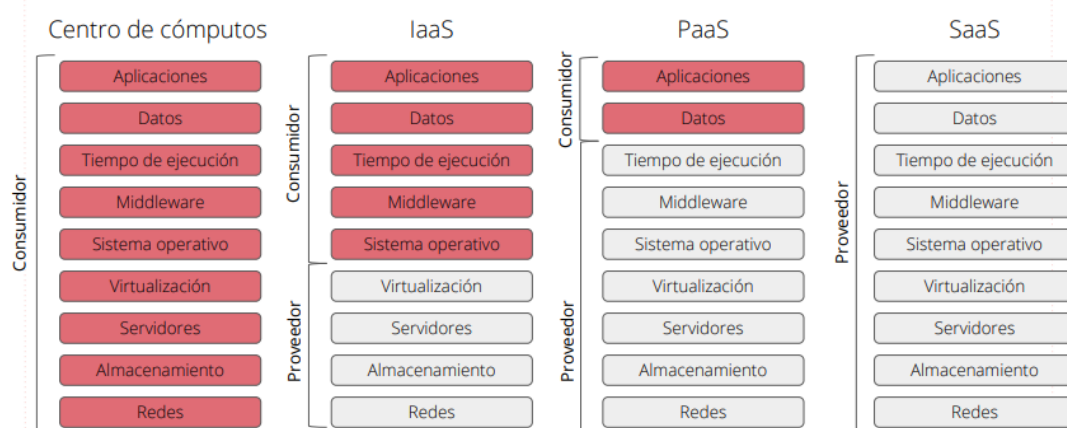
Nubes híbridas: Mezcla de los dos anteriores y tienen la capacidad de portabilidad de aplicaciones y datos como característica principal.

Modelo de responsabilidades

¿Por qué es necesario?

La necesidad - Como ya vimos, en cloud computing lo que hacemos es utilizar recursos de un tercero. Básicamente nos convertimos en inquilinos de aquel que es propietario de los activos informáticos. El modelo de responsabilidades es un **estándar que nos permite definir diferentes tipos de contratación de los servicios al dueño de la nube, qué tareas quedan delegadas a la nube y cuáles serán nuestra responsabilidad.**

El modelo



Analicemos el modelo Si miramos el modelo podemos identificar:

- **Columnas: centro de cómputos, IaaS, PaaS y SaaS.**
- **Roles:** consumidor y proveedor
- **Cajas** que definen un **conjunto de tareas, tecnologías y/o disciplinas** con las que ya estamos familiarizados, como pueden ser: redes, virtualización o middleware.
- Y, finalmente, **colores** que tiñen cada una de ellas, dependiendo la columna en la que nos encontremos.

Pero, ¿qué significan?

Los roles: consumidor y proveedor

Utilizamos estos dos roles para **definir quién es responsable**. Ya vimos que en cloud computing, lo que hacemos es en realidad contratar recursos de manera flexible y con interfaces bien definidas —como pueden ser APIs— para la administración de los mismos a un tercero. Ese tercero, dueño de la nube o de los centros de cómputos en los que decidamos correr nuestras aplicaciones, es a quien vamos a llamar proveedor.

¿Y quién es el consumidor? La respuesta es sencilla, ¡nosotros! Que decidimos contratar los servicios del proveedor para ejecutar nuestras aplicaciones en sus centros de cómputos.

Las columnas: centro de cómputos

Cuando hablamos de centro de cómputos, lo que hacemos, es definir de qué tareas nosotros seríamos responsables si somos quienes alojamos en su totalidad los servicios de tecnologías. En otras palabras, que no contratamos los servicios de ningún proveedor para que albergue o administre nuestros sistemas.

Pros: Tenemos control total sobre los activos, podemos decidir cómo, para qué, dónde y cuándo los utilizamos.

Contras: Tenemos responsabilidad total por la salud, utilización y mantenimiento del parque informático. No podemos escalar de manera flexible.

Las columnas: IaaS

Cuando hablamos de IaaS nos referimos a un acrónimo en inglés: Infrastructure as a Service (infraestructura como servicio, en castellano). En este modelo de responsabilidad, el proveedor nos da la posibilidad de **instanciar máquinas virtuales** en su centro de cómputos sin nosotros tener la necesidad de administrar la infraestructura subyacente.

Pros: Seguimos teniendo control sobre el sistema operativo, un ambiente ideal para instalar aplicaciones legacy. Nos permite escalar la infraestructura de forma más dinámica.

Contras: Seguimos siendo responsables por la administración y salud del sistema operativo y todos los servicios y/o aplicaciones que se ejecuten encima de este.

Las columnas: PaaS

Cuando hablamos de PaaS nos referimos a un acrónimo en inglés: Platform as a Service (en español, plataforma como servicio). En este modelo de responsabilidad, el proveedor nos ofrece

una tecnología específica, expuesta por medio de interfaces definidas, ya sean gráficas o APIs, y nos abstrae de toda la gestión de los recursos subyacentes.

Pros: Nos abstraemos completamente de la administración de los recursos de las capas inferiores. Delegamos en el proveedor la gestión, monitoreo y escalabilidad de la tecnología. Permittiéndonos a nosotros enfocarnos plenamente en la entrega de valor.

Contras: Tiende a ser una **opción costosa**, ya que nos abstrae de todo el músculo requerido para administración y monitoreo de la infraestructura subyacente. Las configuraciones disponibles están limitadas a aquellas que el proveedor haya decidido exponer o disponibilizar.

Ejemplo de columna PaaS

El modelo PaaS tiende a ser difícil de visualizar cuando no tenemos experiencia. Tomemos un ejemplo para ayudarnos: Amazon DynamoDB. DynamoDB es una base de datos no estructurada, similar a MongoDB. Pero mientras que si queremos utilizar MongoDB, debemos instanciar la infraestructura necesaria y luego instalarla; Amazon DynamoDB, la podemos consumir, sencillamente, como un servicio. ¿Qué significa esto? Que una vez suscrito al servicio mediante el sitio web de Amazon AWS, nos van a proveer:

- Una interfaz web a través de la cual podremos hacer algunas configuraciones del servicio de Dynamo.
- Una API, que nos permitirá configurar los mismos parámetros que la interfaz web, pero de manera programática.
- Y una cadena de conexión, que utilizaremos en nuestra aplicación para conectarnos contra la base de datos de Dynamo.

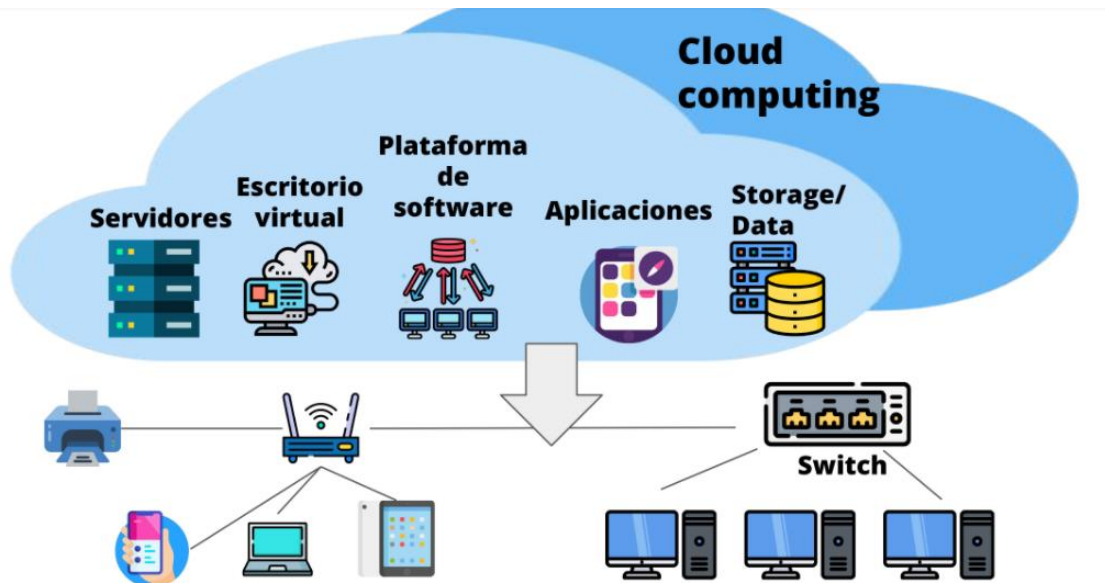
De esta manera, no tuvimos la necesidad de instalar ningún recurso. Sencillamente suscribirnos al servicio, y la plataforma o tecnología está disponible para nuestro uso, Internet mediante.

Las columnas: SaaS

Cuando hablamos de SaaS nos referimos a un acrónimo en inglés: Software as a Service (en castellano, software como servicio). En este modelo de responsabilidad, el proveedor nos ofrece una **aplicación**. La mejor manera de entender este modelo de responsabilidad es pensar en ejemplos: Trello, Salesforce o GMail son ejemplos de SaaS.

Pros: Con tan solo una tarjeta de crédito podemos acceder a una aplicación lista para utilizar y resolver una necesidad. Y sin tener que cubrir el costo de la infraestructura y operación de la misma.

Contras: Poco a nulo control sobre la configuración del sistema contratado. Dependiendo la flexibilidad de software puede que tengamos que adaptar nuestros procesos a lo que el software permite hacer. No hay control sobre las nuevas funcionalidades del sistema y cuándo se liberan.



¿Cómo puedo crear máquinas virtuales en la nube?

Gracias a la popularidad creciente de la nube y el gran desarrollo de la conectividad a internet en todo el mundo, las empresas recurren cada vez más al uso de máquinas virtuales en la nube y aplicaciones móviles.

Pero de crear estas máquinas en equipos físicos hasta hace poco tiempo, ahora se pasó a crear máquinas virtuales en la nube, aprovechando el equipamiento de altísimo nivel que ofrecen algunas empresas tecnológicas mundiales.

La virtualización no es otra cosa que el proceso de creación, a través de un software o programa especial, de una versión virtual de algún recurso tecnológico, como un sistema operativo, una plataforma de hardware, un periférico, un disco de almacenamiento o cualquier otro equipamiento que exista en la vida real.

Con el creciente acceso a internet en todos los países del mundo, las mayores velocidades de conexión y la cada vez mayor capacidad de almacenamiento que ofrecen muchas empresas, tanto de forma gratuita como pagando por servicios especiales, es posible también disponer de máquinas virtuales que funcionarán en la nube.

¿Por qué usar una máquina virtual en la nube?

Por ejemplo, podríamos necesitar un servidor conectado las 24 horas a internet, donde se alojan páginas web y servicios online de venta o distribución de algún producto.

En este caso solo tienen dos opciones. La primera sería comprar un costoso equipo, o disponer de uno ya existente, para montar un servidor en nuestra casa o negocio, además de tener que comprar dispositivos para conectarlo a internet y un plan de datos de alta velocidad, configurarlo, instalar sistemas de seguridad y contra incendios para protegerlo, comprar e instalar sistemas de respaldo de datos, instalar sistemas de energía por si hay un apagón o, en fin, atender cualquier problema físico o de configuración característicos de la administración de un servidor.

La segunda opción sería contratar un servicio para crear una máquina virtual en la nube, que podría actuar como servidor y a la cual solo tendrían que subir sus bases de datos y programas usados para operar sus negocios. Microsoft Azure, Amazon Web Services y Google Platform son de los grandes servicios existentes para crear sistemas virtualizados en la nube.

Además, esta máquina sería accesible en poco tiempo a partir de la contratación; y su mantenimiento quedaría a cargo de la prestataria del servicio.

¿Cómo puedo crear una máquina virtual en la nube?

Actualmente hay dos grandes sistemas que ofrecen servicios de máquinas virtuales en la nube, operados por las gigantes Amazon y Microsoft —que poseen grandes centros de datos en varios países del mundo—.

El sistema de Amazon es parte de su plataforma Amazon Web Services (AWS) y se denomina Amazon Elastic Compute Cloud (EC2). Este servicio crea y ejecuta máquinas virtuales en la nube a las cuales denomina “instancias”.

Además, ofrece plantillas para que el cliente configure y cree en línea las máquinas que desee, cada una basada en un sistema operativo como Linux SuSE, Linux Amazon, Linux Ubuntu y Red Hat Linux, así como Windows Server 2012.

Estas plantillas también incluyen aplicaciones y configuraciones predefinidas de memoria RAM, número de procesadores virtuales y otros detalles “físicos” de cada máquina a crear.

Por su parte, Microsoft ofrece su plataforma Azure —que es parte de los servicios de su tecnología IaaS (*Infrastructure as a Service*)—. Estos servicios incluyen desde la instalación de un servidor web hasta bases de datos y la creación de máquinas virtuales de cualquier tipo y potencia.

Ventajas de las máquinas virtuales en la nube

Existen buenas razones para hacer uso de una máquina virtual en la nube. La principal ventaja es la económica, ya que solo tenemos que pagar el servicio a un proveedor y generalmente este ofrecerá atractivos paquetes diseñados en función de cuántas máquinas queremos tener, cantidad de procesadores, memoria RAM, espacio en disco duro, entre otras características.



Tendremos menos gasto en hardware: Con una máquina virtual en la nube es el proveedor del servicio quien asume el costo del mantenimiento de los servidores y demás equipos reales requeridos, así como también del tiempo y costo de mantener el “hardware virtual” que se configuró según los requerimientos.



Software actualizado constantemente: El proveedor tendrá siempre actualizado el software usado para configurar y mantener las máquinas virtuales de los clientes.



Solo pagamos por el uso: Al contratar una máquina virtual en la nube solo pagaremos por el uso que le dan a la misma y cuando no la necesiten, pueden desconectarse para reducir costos.



Accesibilidad desde cualquier lugar: No importa dónde se encuentren, siempre podemos acceder y usar sus máquinas virtuales en la nube desde un equipo que tenga conexión a internet.



Mayor rapidez de implementación: El costo y tiempo necesario desde que contratan una máquina virtual hasta que pueden usarla es mucho menor que lo que tardaríamos en armar y configurar un servicio similar en un equipo físico en nuestra casa u oficina.



Menores costos operativos: Contratar una máquina virtual en la nube nos evita tener que pagar facturas de electricidad, sistemas antirrobo, aire acondicionado o calefacción y alquiler del espacio físico o infraestructura donde se instalarán equipos servidores reales.



Fiabilidad: La mayoría de las empresas que ofrecen máquinas virtualizadas en la nube operan en centros de datos muy modernos y equipados con lo mejor en cuanto a conectividad a internet, abastecimiento eléctrico las 24 horas, seguridad y, sobre todo, poseen equipos de muy alto nivel y con gran capacidad de almacenamiento de datos.



Potencia escalable: Si lo requieren, pueden aumentar o disminuir la potencia de su conjunto de máquinas virtuales o modificar los parámetros de los servicios que estas requieren. Entre estos cambios tenemos el aumento de la capacidad de procesamiento de datos, la necesidad de cambiar a un software más adecuado para nuestras necesidades, pero más exigente en recursos, un mayor o menor almacenamiento en disco o incluso la asignación de una máquina virtual más potente para una tarea específica durante un período de tiempo.



Desventajas de usar máquinas virtuales ubicadas en la nube

Hay varias razones por las cuales debemos pensarlo bien antes de usar este tipo de recurso tecnológico.



Menor rendimiento: Entre una máquina virtual y un equipo anfitrión físico hay una capa de programación que influye ligeramente en el rendimiento de los programas y sistemas operativos virtualizados. Este problema es mayor cuando la máquina virtual opera desde la nube porque se agregan más capas entre el hardware real del servidor de la empresa que presta el servicio de alojamiento y virtualización, el sistema operativo del servidor y el equipo usado por el cliente que accede a la máquina virtual desde su casa u oficina.



Su tiempo de respuesta o latencia es mayor: Al depender de internet, como acceso a la máquina virtual, se enfrentan al problema del lag o retraso en la transmisión de datos y respuesta a cada comando durante las horas del día en que hay mayor tráfico de datos en las redes. Eso implica que será imposible tener una respuesta instantánea de parte de la máquina virtual cuando el flujo de internet esté muy saturado o si los servidores de la empresa que maneja la nube tienen mucha carga de trabajo.

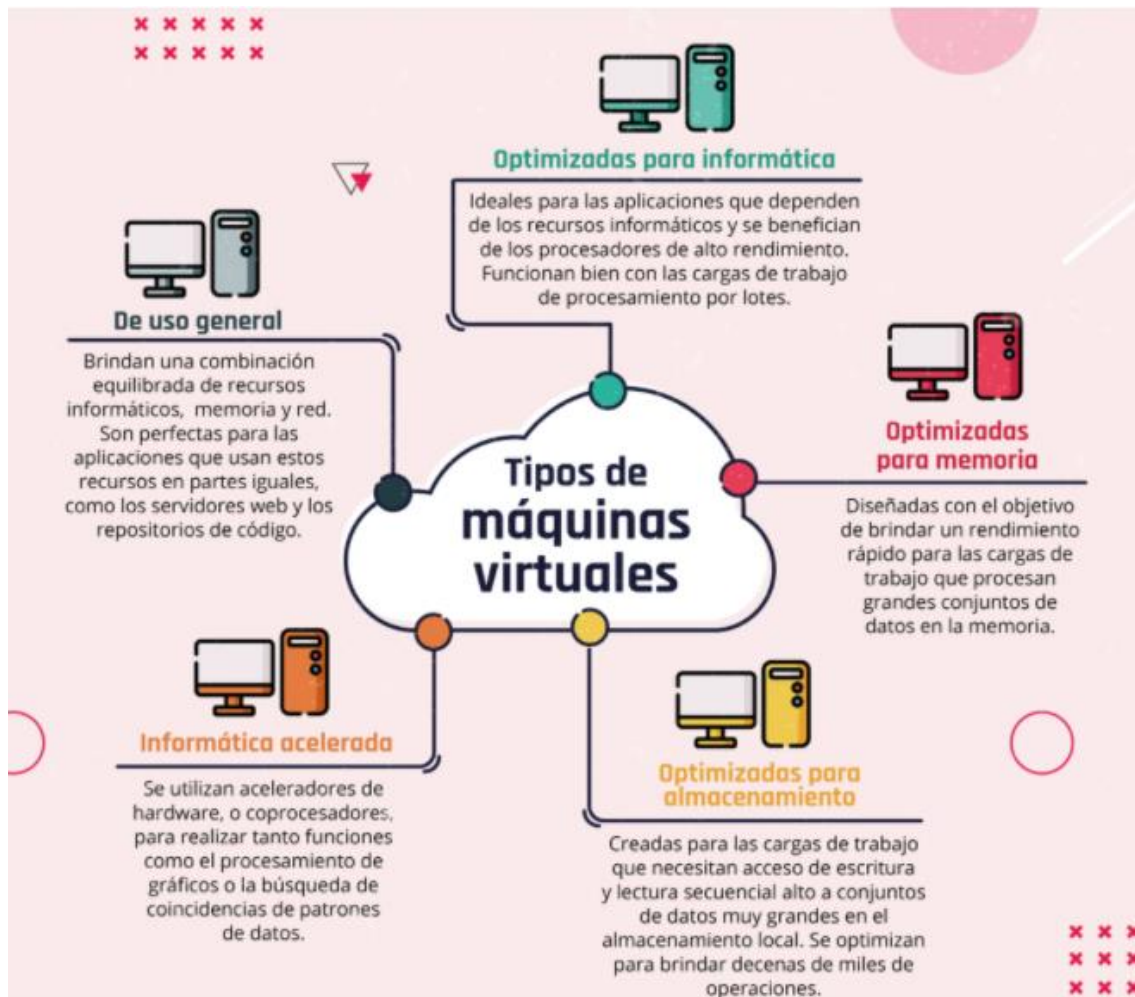
Además, si vivimos en una zona o país con conexión a internet lenta o irregular, enfrentaremos el riesgo de llegar a quedar desconectado de nuestras máquinas virtuales y los datos alojados en ellas durante los momentos en que el servicio sea más lento o se interrumpa.

En muchas ocasiones, esto se resuelve contratando dos o más conexiones con proveedores distintos, pero a su vez esto implica un mayor gasto económico para los particulares o pequeños emprendedores.

A esto se suma que la transferencia de grandes volúmenes de datos nunca será tan rápida por internet como lo sería a través del disco duro de una máquina anfitriona física y una máquina virtual instalada directamente en ella.

Además, al depender del acceso a la nube a través de terceros, se corre el riesgo de que los datos del usuario puedan ser vulnerados por un atacante que logre acceso al sistema del proveedor del servicio de virtualización.

En este caso su seguridad dependerá de la capacidad tecnológica del proveedor, por lo que deberán confiar que aplique medidas como instalación de cortafuegos físicos en sus líneas de datos, servicios automatizados de respaldo de datos, instalación de sus servidores en espacios protegidos contra robos y fuego, entre otras.



¿Qué es la escalabilidad de la nube?

Se refiere a la capacidad de crecer en capacidad en demanda. Por ejemplo, si tenemos 1000 usuarios y de repente crecemos a 2000, la capacidad de escalamiento de la nube se refiere a su habilidad para incrementar su capacidad. Pero no solamente crecer, sino también aumentar su capacidad para ahorrar recursos. A este concepto de crecer y reducir su tamaño se le llama elasticidad, a una infraestructura que permite este comportamiento se la llama elástica. Cuando se utilizan servicios elásticos, la nube debe ser capaz de realizar cobros en base al uso de estos, de forma que no se realicen cargos por recursos que no se encuentran en uso.

¿Qué es la tolerancia al fallo de la nube?

Cuando un servicio es tolerante a los fallos, dicho servicio es capaz de responder a cierto grado de errores sin dejar de ser funcional, es decir que tiene la capacidad de solucionar los problemas que existen o reemplazar elementos que no funcionan de forma correcta por otros que si lo hacen. Un ejemplo puede ser una red local conectada a un switch de muchos puertos alámbricos. Si uno de estos puertos falla, la red puede seguir operando con el resto de los puertos existentes.

¿Qué es la disponibilidad de la nube?

Cuando, por ejemplo, tenemos múltiples recursos y uno de éstos falla, pero hay otros que hagan su trabajo, podemos considerar que existe alta disponibilidad. Si por el contrario, este recurso es único y no puede ser reemplazado, se dice que existe una baja disponibilidad.

Proveedores

Amazon Elastic Compute Cloud (Amazon EC2) Es un servicio web que proporciona capacidad informática en la nube segura y de tamaño modificable. Está diseñado para simplificar el uso de la informática en la nube a escala web para los desarrolladores. La sencilla interfaz de servicios web de Amazon EC2 permite obtener y configurar capacidad con una fricción mínima. Proporciona un control completo sobre los recursos informáticos y puede ejecutarse en el entorno informático acreditado de Amazon.

A su vez, ofrece la plataforma informática más amplia y profunda con elección de procesador, almacenamiento, red, sistema operativo y modelo de compra. Cuentan con las instancias de GPU más poderosas para la capacitación de machine learning y las cargas de trabajo gráficas, así como las instancias de costo por inferencia más bajas de la nube. En AWS se ejecutan más cargas de trabajo de SAP, HPC, machine learning y Windows que en cualquier otra nube.

Máquinas virtuales en Microsoft Azure

En Microsoft Azure tienen soporte para sistemas operativos Windows y Linux, e incluso soporte a Windows Server 2003. Podemos configurar, a través de diversas opciones, la memoria RAM asignada y la CPU que va a tener asociada.

Componentes de la máquina virtual

Disco virtual: incluye el sistema operativo instalado. Gracias a este disco virtual podemos iniciar el equipo y guardar información en forma persistente

Placa de red virtual: al igual que en un equipo físico, es la que me facilitará la conexión con una o más redes.

Direcciones IP: debido a ellas podré conectarme al equipo virtual (pueden ser privadas o públicas).

El sistema operativo: lo que ejecutará como core el equipo virtual. No se puede modificar salvo operaciones avanzadas sobre la VM.

Grupos de seguridad de red: que nos ayudarán a definir desde qué orígenes nos podemos conectar y hacia qué destinos podemos acceder, teniendo en cuenta protocolos, puertos, etc. Los network security groups son una manera ágil de gestionar los permisos de red, para una o más máquinas. Todos estos componentes son definidos por software. No somos los responsables de mantener los componentes de bajo nivel que permiten su funcionamiento (hardware).

En Microsoft Azure podremos parametrizar dos elementos fundamentales:

El nombre de la máquina virtual: una vez elegido, no podrá cambiar luego de su creación.

El tamaño: Nos brindan diversas opciones de tamaños preconfigurados para la CPU, memoria, cantidad de discos soportados y calidad de componentes. Se puede cambiar en cualquier momento, y a eso se le llama “cambio vertical” en el hardware virtual. A veces, estos cambios requieren que el equipo se reinicie.

Instancias de máquina virtual en Google Cloud

Una instancia es una máquina virtual alojada en la infraestructura de Google. Podemos crear una instancia si utilizamos Google Cloud Console, la herramienta de línea de comandos de gcloud o la API de Compute Engine. Las instancias de Compute Engine pueden ejecutar las imágenes públicas de Linux y Windows Server que proporciona Google, así como las imágenes personalizadas privadas que podemos crear o importar desde sus sistemas existentes. También tenemos la posibilidad de implementar contenedores de Docker, que se inician de forma automática en instancias que ejecutan la imagen pública de Container-Optimized OS.

Si usamos un conjunto de tipos predefinidos de máquinas o creamos nuestros propios tipos personalizados de máquinas, podemos elegir las propiedades de máquina de nuestras instancias, como la cantidad de CPU virtuales y de memoria.

Instancias y proyectos

Cada instancia pertenece a un proyecto de Google Cloud Console y cada proyecto puede tener una o más instancias. Cuando creamos una instancia en un proyecto, especificamos la zona, el sistema operativo y el tipo de máquina de esa instancia. Cuando la borramos, se quita del proyecto. Instancias y opciones de almacenamiento De forma predeterminada, cada instancia de Compute Engine tiene un pequeño disco de arranque persistente que contiene el sistema operativo. Cuando las aplicaciones que se ejecutan en sus instancias requieren más espacio de almacenamiento, pueden agregar opciones de almacenamiento adicionales a sus instancias.

Herramientas para gestionar instancias

Como desarrollamos anteriormente, podemos usar diversas herramientas para crear y administrar instancias, como Google Cloud Console, la herramienta de línea de comandos de gcloud y la API de REST. Si queremos configurar aplicaciones en las instancias, conectémonos a la instancia con Secure Shell (SSH) para instancias de Linux o Remote Desktop Protocol (RDP) para instancias de Windows Server.

¿Qué es el almacenamiento en la nube?

El almacenamiento en la nube —del inglés cloud storage— permite guardar cualquier tipo de archivos en una infraestructura digital con la comodidad de la administración y backup remotos. Cada día, más personas y empresas almacenan archivos en servicios como Microsoft OneDrive, Google Docs o Amazon. De hecho, la transformación que ha ocurrido con servicios de reproducción de música o videos en línea también se basa en la nube, la cual reemplaza archivos físicos con la posibilidad de acceder a sus versiones digitales. Adicionalmente, la información estará siempre almacenada y disponible desde cualquier lugar.

Ventajas de almacenar los archivos en la nube

1- Tiene más espacio físico

Cuando los servidores físicos eran la norma, el principal problema surgía cuando no había espacio suficiente. Por ejemplo, si una empresa tenía un hardware con 100 terabytes de almacenamiento, cuando llegaba al límite de espacio, solo había dos soluciones: borrar archivos —una solución temporal— o comprar otro hardware —una solución cara—. Con la nube, es posible contratar más espacio si llegamos al límite de almacenamiento y expandirlo de manera simple e inmediata. En otras palabras, almacenar documentos en la nube es perfectamente ajustable a las necesidades de cada empresa, por lo tanto, fácilmente escalable.

2- Compartir archivos con mayor facilidad.

Administrar archivos en la nube es más práctico y menos complicado que en un hardware. De hecho, las empresas que usan papel pierden mucho tiempo con trabajo repetitivo: impresión, revisión, plazos de aprobación y autenticación hasta la versión final de un documento. Los documentos digitales hacen los procesos más ágiles porque todas las etapas son realizadas desde un único sistema en línea. Estos documentos pueden ser firmados electrónicamente con seguridad, integridad y conformidad.

3- Acceso a la información desde cualquier dispositivo.

Si incorporamos documentos, videos, fotos o cualquier tipo de archivos en la nube, tendremos acceso a ellos desde casi cualquier dispositivo y, además, la mayoría de los proveedores nos permite subir, descargar y modificarlos en tiempo real. Esto dependerá del sistema operativo y entre algunos de ellos encontramos:

- Windows;
- OS X;
- iOS;
- Google Chrome OS;
- Android.

La tecnología de la nube protegerá los archivos en caso de que falle nuestro dispositivo, pudiendo acceder a los documentos con nuestro usuario y contraseña de forma segura.

4- Diversidad de archivos.

Cada empresa tiene necesidades de almacenamiento distintas, ya sea porque maneja archivos muy pesados por la naturaleza del trabajo o porque tiene operaciones que requieren mucho orden y estructura. Normalmente las plataformas ofrecen una cantidad de gigabytes gratuitos, por lo que no es necesario que todos los usuarios tengan que pagar una renta para disfrutar de un espacio acorde a sus requerimientos.

5- Optimización automática.

Esta es una de las ventajas más importante. Se trata de la optimización automática que brinda cada proveedor en su plataforma. Ante la enorme competencia dentro de este sector, es común encontrar cada vez más aspectos que los diferencien entre ellos para atraer mayor cantidad de usuarios y afianzar la fidelidad quienes ya lo utilizan.

EJEMPLOS:

- Dropbox: si utilizamos este servicio, nos habremos dado cuenta de que es muy sencillo añadir atajos a las carpetas para que se parezca a nuestro escritorio, pudiendo acceder a los documentos de la nube de forma más rápida y sencilla.
- iCloud: este servicio se encuentra preinstalado en todos los dispositivos de Apple, haciendo un backup de la mayoría de tu información de forma automática, respaldando tus documentos en la nube para brindarnos mayor accesibilidad y seguridad.
- Google Drive: ofrece todo el pack de Google Docs, Slide, Sheets, fotos, entre otros.

¿Qué puedo guardar en la nube?

¿Qué es un Blob?

Los BLOB (Binary Large Object) son elementos utilizados en las bases de datos para almacenar datos de gran tamaño que cambian de forma dinámica. No todos los sistemas de gestión de bases de datos son compatibles con los BLOB. Generalmente, estos datos son imágenes, archivos de sonido y otros objetos multimedia; a veces se almacenan como códigos de binarios BLOB. Un objeto BLOB representa un objeto tipo fichero de datos planos inmutables. Los BLOBS representan datos que no necesariamente se encuentran en un formato nativo.

¿Qué es un file share?

Los file share son archivos para compartir. El intercambio de archivos es el acto de distribuir o proveer acceso a información almacenada digitalmente, como programas informáticos, obras multimedia (audio, video), documentos o libros electrónicos. Puede ser implementado con distintos tipos de almacenamiento, transmisión y modelos de distribución. Algunos de los métodos más comunes son la distribución manual mediante el uso de medios extraíbles (CD, DVD, disquetes, cintas magnéticas, memorias flash), las instalaciones centralizadas de servidores de archivos en redes informáticas, los documentos enlazados de la World Wide Web y el uso de redes peer-to-peer (P2P) distribuidas.

¿Qué es una tabla en la nube?

Una tabla en la nube es similar a la de una bases de datos, se refiere al tipo de modelado de datos donde se guardan los datos recogidos por un programa. Su estructura general se asemeja a la vista general de un programa de tablas. Las tablas se componen de dos estructuras: Campo: Corresponde al nombre de la columna. Debe ser único y además de tener un tipo de dato asociado. Registro: Corresponde a cada fila que compone la tabla. Ahí se componen los datos y los registros. Eventualmente pueden ser nulos en su almacenamiento. Cada tabla creada debe tener un nombre único en cada base de datos, haciéndola accesible mediante su nombre o su seudónimo (alias), dependiendo del tipo de base de datos elegida.

Almacenamiento público y privado

En el almacenamiento público, los datos se guardan en una nube pública, abierta al uso a todas las personas que lo deseen. Este servicio puede ser gratuito o de pago, pudiendo adquirir

diferentes planes en función de la capacidad de almacenamiento que necesitemos, el ancho de banda y las diferentes facilidades que nos pueda aportar el proveedor del servicio. Este tipo de almacenamiento lo ofrece Amazon, Azure de Microsoft y Google Engine.

2- Almacenamiento privado

Cuando la empresa es muy grande o compleja, lo más recomendable es decidirse por una nube privada. Los datos se almacenan en la nube, pero dentro de un entorno local de difícil acceso a todos aquellos que no sean de la empresa, por razones de seguridad. Uno de los sistemas de almacenamiento privado más destacable es Openstack, solución open source.

3- Almacenamiento híbrido.

Este tipo de nube combina las dos anteriores, las empresas pueden aprovecharse de las características de la nube privada, pero recurrir al uso de una nube pública, por ejemplo, cuando contrata servicios a terceros que no están dentro de la propia empresa. Este tipo de soluciones tiene mucho potencial, ya que permite hacer crecer el sistema de almacenamiento de la empresa en base a las necesidades, sin dejar de centralizar los recursos de la misma.

Principales proveedores

Microsoft

Amazon

IBM

Salesforce

SAP

En este apartado veremos las soluciones que nos ofrecen los dos principales cloud providers: Microsoft y Amazon.

1- Microsoft

Windows Azure Storage

La plataforma principal de Azure Storage es la solución de almacenamiento en la nube de Microsoft. Proporciona almacenamiento para objetos de datos, que presenta una alta disponibilidad, es seguro, durable y redundante y se puede escalar de forma masiva. La plataforma de Azure Storage incluye los servicios de datos siguientes:

- Blobs de Azure: un almacén de objetos que se puede escalar de forma masiva para datos de texto y binarios. También incluye compatibilidad con el análisis de macrodatos a través de Data Lake Storage Gen2.
- Azure Files: recursos compartidos de archivos administrados para implementaciones locales y en la nube.
- Colas de Azure: un almacén de mensajería para mensajería confiable entre componentes de aplicación.
- Tablas de Azure: un almacén NoSQL para el almacenamiento sin esquema de datos estructurados.

- Azure Disks: volúmenes de almacenamiento en el nivel de bloque para máquinas virtuales de Azure.

Características	Descripción	Cuándo se usa
Archivos de Azure	Ofrece recursos compartidos de archivos en la nube totalmente administrados a los que se puede acceder desde cualquier lugar a través del protocolo de bloque de mensajes del servidor (SMB) estándar del sector. Los recursos compartidos de archivos de Azure se pueden montar desde implementaciones de Windows, Linux y macOS en la nube o locales.	Cuando se desee migrar mediante lift-and-shift, una aplicación a la nube que ya usa las API del sistema de archivos nativo para compartir datos entre ella y otras aplicaciones que se ejecutan en Azure. Cuando se quiera reemplazar o complementar los servidores de archivos locales o los dispositivos NAS. Cuando se desee almacenar herramientas de desarrollo y depuración a las que es necesario acceder desde muchas máquinas virtuales.
Azure Blobs	Permite que los datos no estructurados se almacenen y accedan a una escala masiva en Blobs en bloques. También admite Azure Data Lake Storage Gen2 para soluciones de análisis de macrodatos empresariales.	Cuando se desea que la aplicación admita escenarios de streaming y de acceso aleatorio. Cuando se desea poder tener acceso a datos de la aplicación desde cualquier lugar. Cuando se desea crear una instancia empresarial de Data Lake en Azure y realizar análisis de macrodatos.
Azure Disks	Permite que los datos se almacenen y se acceda a ellos desde un disco duro virtual conectado de manera persistente.	Cuando se desea migrar mediante lift-and-shift aplicaciones que usan las API del sistema de archivos nativo para leer y escribir datos en discos persistentes. Cuando se desea almacenar datos a los que no se necesita acceder desde fuera de la máquina virtual a la que está conectado el disco.
Colas de Azure	Permite la puesta en cola de mensajes asíncronos entre los componentes de la aplicación.	Cuando se quiere desacoplar los componentes de la aplicación y usar la mensajería asíncrona para comunicarse entre ellos.
Tablas de Azure	Permite almacenar datos NoSQL estructurados en la nube, lo que proporciona un almacén de claves y atributos con un diseño sin esquema.	Cuando se quiere almacenar conjuntos de datos flexibles, como datos de usuarios para aplicaciones web, libretas de direcciones, información de dispositivos u otros tipos de metadatos que el servicio requiera.

2- Amazon

Servicios de almacenamiento de AWS

- Almacenamiento de objetos:



Almacenamiento de objetos
creado para almacenar y recuperar
cualquier volumen de datos desde
cualquier ubicación

- Almacenamiento de archivos:

 Amazon Elastic File System Sistema de archivos NFS escalable, elástico y nativo en la nube	 Amazon FSx for Windows File Server Almacenamiento de archivos completamente administrado basado en Windows Server	 Amazon FSx for Lustre Sistema de archivos de alto rendimiento completamente administrado e integrado con Amazon S3
--	---	--

- Almacenamiento en bloque:



Almacenamiento en bloque de
alto rendimiento y con facilidad de
uso a cualquier escala

- Transferencias de datos:



AWS Storage Gateway

Almacenamiento en la nube híbrida que proporciona acceso local al almacenamiento prácticamente ilimitado en la nube



AWS DataSync

Transfiere los datos fácilmente hacia y desde AWS hasta 10 veces más rápido



AWS Transfer Family

Transferencia de archivos, simple y sin problemas a Amazon S3 con protocolos SFTP, FTPS y FTP



Familia de productos AWS Snow

Dispositivos físicos para migrar datos hacia y desde AWS

- Almacenamiento e informática de borde:



Familia de productos AWS Snow

Dispositivos de almacenamiento e informática de borde físicos para entornos duros o desconectados

Conclusión

Actualmente tanto Azure Storage como Amazon Storage soportan sitios web estáticos, en otras palabras, un sitio web estático se compone de HTML, CSS, JavaScript y otros archivos estáticos, como imágenes o fuentes. Un sitio estático suele ser una aplicación de página única (SPA) escrita con diversos marcos de JavaScript, como Angular, React o Vue. Independientemente de cómo se diseñe la aplicación, puede servir los archivos directamente desde el storage en lugar de usar un servidor web. El hospedaje en el almacenamiento es más sencillo y mucho más económico que el mantenimiento de un servidor web; por lo general, el hospedaje estático cuesta solo unos céntimos al mes. Si necesita procesamiento en el lado servidor, a menudo puede satisfacer esas necesidades mediante funciones sin servidor como las admitidas por cada proveedor.

Amazon S3

Amazon Simple Storage Service (Amazon S3) es un servicio de almacenamiento para Internet. Está diseñado para facilitar la informática de escalada web. Tiene una interfaz de servicios web simple que podemos utilizar para almacenar y recuperar cualquier cantidad de datos, en cualquier momento y desde cualquier parte de la web. Ofrece a cualquier desarrollador acceso a la misma infraestructura de almacenamiento de datos económica, fácilmente escalable, fiable, segura y rápida que utiliza Amazon para mantener su propia red global de sitios web. Este servicio tiene como fin maximizar los beneficios de escalar.

Amazon S3 se ha desarrollado de forma deliberada con un conjunto mínimo de características que se centran en la simplicidad y robustez.

Ventajas:

- Creación de buckets: podemos crear y nombrar un bucket que almacena datos. Los buckets son los contenedores fundamentales en Amazon S3 para el almacenamiento de datos.
- Almacenamiento de datos en buckets: tenemos la posibilidad de almacenar una cantidad ilimitada de datos en un bucket. Cargamos la cantidad de objetos que deseamos en un bucket de Amazon S3. Cada objeto puede contener hasta 5 TB de datos. Cada objeto se almacena y recupera con una clave única asignada por el desarrollador.
- Descarga de datos: permite descargar nuestros datos, o que otros lo hagan.
- Permisos: brindemos o denegamos acceso a otras personas que desean cargar o descargar datos en nuestro bucket de Amazon S3. Concedemos permisos para cargar y descargar a tres tipos de usuarios. Los mecanismos de autenticación pueden ayudar a proteger los datos del acceso no autorizado.
- Interfaces estándar: utilicemos las interfaces REST y SOAP basadas en estándares diseñados para trabajar con cualquier conjunto de herramientas de desarrollo de Internet.

S3 Bucket

Un bucket es un contenedor para objetos almacenados en Amazon S3, es decir, cada objeto está almacenado en un bucket.

Los buckets sirven a diversos fines:

- Organizan el espacio de nombres de Amazon S3 al más alto nivel.
- Identifican la cuenta responsable para los cargos de almacenamiento y transferencia de datos.
- Juegan un papel en el control de acceso.
- Sirven como la unidad de agregación para informes de uso.

Objetos

Los objetos son las entidades fundamentales almacenadas en Amazon S3, se componen de datos de objetos y metadatos. Los metadatos son conjuntos de pares nombre-valor que describen el objeto. Los objetos incluyen algunos metadatos predeterminados, como la fecha de la última

modificación y los metadatos HTTP estándar, como Content-Type. También podemos especificar metadatos personalizados en el momento en que se almacena el objeto.

Un objeto se identifica de forma exclusiva dentro de un bucket con una clave (nombre) y un ID de versión.

Claves

Una clave es el identificador único de un objeto dentro de un bucket. Cada objeto de un bucket tiene una clave. La combinación de un bucket, clave e ID de versión identifican de forma única cada objeto. Por lo tanto, se puede pensar en Amazon S3 como una asignación de datos básica entre "bucket + clave + versión" y el objeto en sí. Se puede acceder a cada objeto de Amazon S3 de forma exclusiva a través de la combinación de punto de enlace de servicio web, nombre del bucket, clave y, de forma opcional, una versión.

Regiones

Podemos elegir la región geográfica de AWS donde Amazon S3 almacenará los buckets que creamos. Esta elección puede optimizar la latencia, minimizar los costos o cumplir con requisitos legales. Los objetos almacenados en una región nunca la abandonan, a menos que se transfieran expresamente a otra región. Por ejemplo, los objetos almacenados en la región UE (Irlanda) nunca salen de ella.

Identity and Access Management

Utilicemos AWS Identity and Access Management (IAM) para administrar el acceso a nuestros recursos de Amazon S3. Por ejemplo, podemos usar IAM con Amazon S3 para controlar el tipo de acceso que tiene un usuario o un grupo de usuarios a partes concretas de un bucket de Amazon S3 que es propiedad de nuestra cuenta de AWS.

¿Qué otros servicios se ofrecen más allá de compute y storage?



¿Qué es la informática sin servidor?

Se trata de un modelo de desarrollo nativo de la nube que permite que los desarrolladores diseñen y ejecuten aplicaciones sin tener que gestionar servidores. Si bien se utilizan servidores, se encuentran aislados de la etapa de desarrollo de las aplicaciones. El proveedor de nube se encarga de las tareas rutinarias de preparación, mantenimiento y adaptación de la infraestructura de los servidores. Los desarrolladores solo deben empaquetar el código en contenedores para implementarlo.

Una vez que se instalan las aplicaciones sin servidor, responden a la demanda y se amplían o reducen automáticamente en función de las necesidades. Por lo general, los proveedores de nube pública ofrecen estas tecnologías a un costo que se mide a través de un modelo de ejecución basado en eventos, según se soliciten. Por eso, cuando una función permanece inactiva, no se genera ningún cargo.

Lo que distingue a la informática sin servidor de los demás modelos de cloud computing es que el proveedor de nube se encarga de gestionar tanto la infraestructura de nube como la expansión de las aplicaciones. Aquellas sin servidor se implementan en contenedores que se inician automáticamente cuando se solicitan. En el caso de los modelos estándar de cloud computing de infraestructura como servicio (IaaS), los usuarios adquieren cierta capacidad por adelantado, lo cual significa que pagan a un proveedor de nube pública por ciertos elementos del servidor que funcionan de forma permanente para ejecutar sus aplicaciones. El usuario debe encargarse de ampliar la capacidad durante los períodos de mayor demanda y reducirla cuando ya no haga falta. La infraestructura de nube requerida para ejecutar las aplicaciones permanece activa incluso cuando no se utilizan.

En cambio, la arquitectura sin servidor permite que se inicien solo cuando sea necesario. Cuando un evento activa el funcionamiento del código de las aplicaciones, el proveedor de nube pública le asigna los recursos correspondientes de forma dinámica, y el usuario deja de pagar cuando termina de ejecutarse. La informática sin servidor no solo ofrece beneficios en cuanto a los costos y la eficiencia, sino que también permite que los desarrolladores ya no deban encargarse de las tareas rutinarias y poco relevantes.

¿Qué función desempeña el proveedor de nube en la informática sin servidor?

Los proveedores de nube ejecutan los servidores físicos y asignan los recursos de forma dinámica para los usuarios que implementan el código directamente en la producción. Normalmente, las ofertas de informática sin servidor se clasifican en dos grupos: Back end como servicio (BaaS) y Función como servicio (FaaS). El primero otorga a los desarrolladores acceso a diferentes aplicaciones y servicios externos. Por ejemplo, un proveedor de nube puede ofrecer servicios de autenticación, cifrado adicional, bases de datos a las que se puede acceder desde la nube e información muy confiable sobre el uso. Las funciones sin servidor se suelen solicitar mediante llamadas a las interfaces de programación de aplicaciones (API).

Ventajas y desventajas

Ventajas	Desventajas
<ul style="list-style-type: none">• Reduce el costo operativo y aumenta la productividad del desarrollador.• Se reducen los costos operativos, ya que puede pagar por el tiempo de procesamiento en la nube según sea necesario, en lugar de ejecutar y gestionar sus propios servidores todo el tiempo.	<ul style="list-style-type: none">• Los proveedores de nube tienen restricciones estrictas en cuanto a la interacción con sus elementos, lo cual a su vez determina el grado de flexibilidad y personalización de sus propios sistemas.• Probablemente, la decisión de cambiar de proveedor también implique un costo de actualización de los sistemas, para cumplir con las especificaciones de la nueva empresa.

¿Qué es una nube híbrida?

La nube híbrida es una arquitectura de IT que incorpora cierto grado de gestión, organización y portabilidad de las cargas de trabajo en dos o más entornos. Según a quién le consulte, es posible que esos entornos deban incluir lo siguiente:

- Al menos una nube privada y una pública.
- Dos o más nubes privadas.
- Dos o más nubes públicas.
- Un entorno virtual o sin sistema operativo conectado a al menos una nube, ya sea pública o privada.

Funciones de las nubes híbridas

Todas las nubes híbridas deben poder realizar lo siguiente:

- Conectar varias computadoras a través de una red.
- Consolidar los recursos de IT.
- Escalar horizontalmente e implementar los recursos nuevos con rapidez.
- Poder trasladar las cargas de trabajo entre los entornos.
- Incorporar una sola herramienta de gestión unificada.
- Organizar los procesos con la ayuda de la automatización.

¿Cómo funcionan las nubes híbridas?

La forma en que las nubes públicas y privadas funcionan como parte de una nube híbrida es similar a cómo lo hacen de forma independiente:

- Una red de área local (LAN), una red de área amplia (WAN), una red privada virtual (VPN) y las interfaces de programación de aplicaciones (API) conectan varias computadoras entre sí.
- La virtualización, los contenedores o el almacenamiento definido por software extraen los recursos, que pueden agruparse en lagos de datos.
- El sistema de software de gestión asigna esos recursos a entornos donde las aplicaciones pueden ejecutarse, los cuales luego se implementan, según se solicite, con la ayuda de un servicio de autenticación.

Las nubes independientes se vuelven híbridas cuando esos entornos se conectan de la forma más sencilla posible. Esa interconectividad es lo único que permite que las nubes híbridas funcionen, y es por eso que estas nubes son la base del edge computing. Además, determina la forma en que se trasladan las cargas de trabajo, se unifica la gestión y se organizan los procesos. La calidad de las conexiones tiene un efecto directo sobre el funcionamiento de su nube híbrida.

Arquitectura moderna de la nube híbrida

Actualmente, las nubes híbridas ya no necesitan una red amplia de API para trasladar las cargas de trabajo de una nube a otra. Para diseñar nubes híbridas, los equipos modernos de IT ejecutan el mismo sistema operativo en todos los entornos de IT; desarrollan e implementan aplicaciones como grupos de servicios pequeños, independientes y sin conexión directa; y gestionan todo con una PaaS unificada.

Si se utiliza el mismo sistema operativo, se extraen todos los requisitos del sistema de hardware, mientras que la plataforma de organización extrae todos los de las aplicaciones. Esto genera un entorno informático uniforme e interconectado en el que las aplicaciones pueden trasladarse de un entorno a otro sin tener que mantener un mapa complejo de las API que falle cada vez que se actualicen las aplicaciones o que cambie de proveedores de nube. Esta interconectividad permite que los equipos de desarrollo y operaciones trabajen juntos en un modelo de DevOps, que es un proceso con el cual los equipos trabajan en conjunto en entornos integrados utilizando una arquitectura de microservicios compatible con los contenedores.

¿Son seguras las nubes híbridas?

Una nube híbrida diseñada, integrada y gestionada de forma correcta puede ser tan segura como una infraestructura de IT local. Aunque hay algunos desafíos exclusivos de la seguridad de la nube híbrida (como la migración de datos, el aumento de la complejidad y una mayor superficie de ataque), la presencia de varios entornos puede constituir una de las defensas más sólidas contra los riesgos de seguridad. Gracias a todos esos entornos interconectados, las empresas pueden elegir dónde colocar los datos confidenciales en función de los requisitos, y los equipos de seguridad pueden adoptar de manera uniforme un sistema de almacenamiento en la nube que sea redundante y pueda aumentar los esfuerzos de recuperación ante desastres.

x x x x x
x x x x x
x x x x x

¿Cómo conecto mi centro de cómputos a la nube?



Point-to-Site

Una VPN Point-to-Site nos permite crear una conexión segura a la red virtual desde un equipo cliente individual. Se establece una conexión de punto a sitio al iniciarla desde el equipo cliente. Esta solución resulta útil para los teletrabajadores que deseen conectarse a redes virtuales desde una ubicación remota, por ejemplo, desde una casa. La conexión VPN de punto a sitio también es una solución útil en comparación con la conexión VPN de sitio a sitio cuando solo necesitan conectarse a la red virtual algunos clientes.

Site-to-Site

Una VPN Site-to-Site (de sitio a sitio) se utiliza principalmente en las empresas. Las organizaciones con instalaciones en diferentes ubicaciones geográficas utilizan esta VPN para conectar la red de una instalación a la red en otra.

x x x x x
x x x x x