Conceptos básicos sobre virtualización y computación en la nube

Aprenderás a distinguir las distintas formas de realizar despliegues de aplicaciones y a conocer la gama de servicios ofrecidos por las distintas plataformas.

- 1.- Conceptos de la computación en la nube y tipos de servicios
- 2.- Tipos de virtualización
- 3.- Infraestructuras
- 4.- Administración de sistemas en la nube
- 5.- Orquestación de contenedores para DevOps
- 6.- Licencia y autoría de este material

Licencia Creative Commons BY-NC-SA.

1.- Conceptos de la computación en la nube y tipos de servicios

La computación en la nube, también conocida como "**cloud computing**", es un modelo de servicios que nos permite acceder a recursos informáticos a través de Internet de forma flexible y bajo demanda. En lugar de tener que poseer y mantener nuestros propios servidores y equipos, podemos utilizar los recursos de computación, almacenamiento y redes de un proveedor de servicios en la nube.

- **Proveedor de servicios en la nube**: Es la empresa que ofrece los servicios de computación en la nube, como Amazon Web Services (AWS), Microsoft Azure o Google Cloud Platform.
- **Recursos informáticos**: Son los componentes necesarios para ejecutar aplicaciones y almacenar datos. Incluyen servidores, redes, almacenamiento y otros recursos necesarios para el funcionamiento de los sistemas.
- **Acceso a través de Internet**: Podemos acceder a los recursos en la nube utilizando cualquier dispositivo con conexión a Internet, como computadoras, teléfonos móviles o tabletas

La computación en la nube ofrece varias ventajas que la hacen atractiva para empresas y usuarios:

Escalabilidad

Podemos aumentar o disminuir los recursos según nuestras necesidades. Esto permite adaptar la infraestructura a la demanda, evitando gastos innecesarios por recursos infrautilizados.

Flexibilidad

Podemos acceder a nuestros datos y aplicaciones desde cualquier lugar y dispositivo con conexión a Internet. Esto facilita el trabajo remoto y la colaboración en equipo.

Costos reducidos

Licencia Creative Commons BY-NC-SA.

Al no tener que invertir en infraestructura propia, podemos ahorrar en la compra de servidores y equipos. Además, los servicios en la nube suelen tener modelos de pago según el uso, lo que permite ajustar los costos a medida que se utilizan los recursos.

Actualizaciones automáticas

Los proveedores de servicios en la nube se encargan de mantener y actualizar la infraestructura, lo que nos permite utilizar las últimas tecnologías sin tener que preocuparnos por la gestión de los equipos.

Sin embargo, también hay algunas consideraciones a tener en cuenta:

Dependencia de la conexión a Internet

Para acceder a los recursos en la nube, necesitamos una conexión a Internet estable. Si la conexión falla, puede afectar la disponibilidad de los servicios.

Seguridad

Al almacenar nuestros datos en servidores remotos, es importante asegurarnos de que se implementen medidas de seguridad adecuadas para proteger la información.

Privacidad

Al utilizar servicios en la nube, nuestros datos pueden estar sujetos a las políticas de privacidad y protección de datos del proveedor. Es importante leer y comprender estas políticas antes de utilizar los servicios.

En resumen, **la computación en la nube nos brinda la posibilidad de acceder a recursos informáticos de forma flexible y bajo demanda**, lo que nos permite escalar nuestros sistemas de manera eficiente. Sin embargo, es importante considerar tanto las ventajas como las desventajas antes de adoptar esta tecnología en nuestras organizaciones.

Para saber más

Amazon Web Services (AWS), Microsoft Azure o Google Cloud Platform

Licencia Creative Commons BY-NC-SA.

Licencia Creative Commons BY-NC-SA.

2.- Tipos de virtualización

En el campo de la computación en la nube, la virtualización desempeña un papel fundamental al permitir la creación de entornos virtuales que aprovechan eficientemente los recursos de hardware. Existen diferentes tipos de virtualización, y en esta sección vamos a explorar tres de ellos: Emulación, virtualización y paravirtualización. A continuación, exploraremos estos tipos de virtualización y proporcionaremos ejemplos de cada uno:

- **Emulación:** La emulación es un enfoque de virtualización que permite ejecutar un sistema operativo o una aplicación en un entorno que simula el hardware original. En este caso, el hardware virtualizado es diferente al hardware físico subyacente. Un ejemplo común de emulación es el software de emulación de videojuegos arcade antiguos, donde se recrean las características de una consola de juegos con hardware de los año 80.
  La emulación proporciona una gran flexibilidad, ya que permite ejecutar sistemas operativos y aplicaciones diseñadas para un hardware específico en un hardware diferente. Sin embargo, debido a la naturaleza de la emulación, es menos eficiente en términos de rendimiento que otros enfoques de virtualización, ya que hay que virtualizar por completo el hardware emulado.

- **Virtualización tradicional:** La virtualización tradicional, también conocida como virtualización completa, se basa en la creación de máquinas virtuales (VM) que ejecutan sistemas operativos y aplicaciones en un hardware virtualizado e independiente unas de otras. En este caso, cada máquina virtual tiene que tener su propio sistema operativo independiente del SO anfitrión. Ejemplos populares de software de virtualización tradicional son VMware, Microsoft Hyper-V, VirtualBox, etc. Las máquinas virtuales creadas necesitan ser ejecutadas sobre un software especial que las interprete y gestione, llamado hipervisor, también conocido como VMM (Virtual Machine Monitor), que es el que se encarga de realizar las peticiones hardware al sistema operativo anfitrión o emular el hardware que no esté disponible en la máquina real. Su función principal es dividir los recursos físicos de la máquina anfitrión en múltiples entornos virtuales.
  La virtualización tradicional permite ejecutar múltiples sistemas operativos en un único servidor físico y proporciona un alto nivel de aislamiento entre las máquinas virtuales. Cada máquina virtual se comporta como si estuviera ejecutándose en su propio hardware dedicado. Esto facilita la consolidación de servidores y la gestión de infraestructuras complejas.
  Existen dos tipos principales de hipervisores (gestores e interpretes de VM):

Licencia Creative Commons BY-NC-SA.

- **Hipervisor de tipo 1** (o "bare metal"): Este tipo de hipervisor se ejecuta directamente sobre el hardware físico del servidor. No requiere de un sistema operativo anfitrión adicional. Los hipervisores de tipo 1 ofrecen un mayor rendimiento y eficiencia, ya que pueden acceder directamente a los recursos del hardware. Algunos ejemplos populares de hipervisores de tipo 1 son VMware ESXi, Microsoft Hyper-V y Xen.
- **Hipervisor de tipo 2** (o "hosted"): Este tipo de hipervisor se ejecuta como un software dentro de un sistema operativo anfitrión. El sistema operativo anfitrión proporciona los controladores de dispositivo y los servicios básicos, mientras que el hipervisor se ejecuta como una aplicación dentro de él. Los hipervisores de tipo 2 son más fáciles de instalar y configurar, pero pueden tener un rendimiento ligeramente inferior debido a la capa adicional del sistema operativo anfitrión. Algunos ejemplos de hipervisores de tipo 2 son VMware Workstation, VirtualBox, QEMU y Parallels Desktop.
- **Hipervisor híbrido**: Además de estos dos tipos principales, también existe una variante llamada "hipervisor híbrido", que combina características de los hipervisores de tipo 1 y tipo 2. Estos hipervisores se ejecutan directamente sobre el hardware pero también requieren de un sistema operativo anfitrión mínimo para proporcionar servicios adicionales. Un ejemplo de este tipo híbrido puede ser Proxmox, que necesita de una distribución Debian con el kernel de Linux y el hipervisor de tipo 2 de código abierto QEMU, como base para la gestión de las máquinas virtuales, y KVM (Kernel-based Virtual Machine) para la gestión de contenedores LXC.

- **Paravirtualización:** La paravirtualización es un enfoque de virtualización en el que el sistema operativo invitado se modifica para ser consciente de que se está ejecutando en un entorno virtualizado. A diferencia de la virtualización tradicional, donde se emula el hardware, en la paravirtualización se utilizan interfaces especiales para acceder al hardware subyacente de manera más eficiente.
  En la paravirtualización, el sistema operativo invitado y el sistema operativo anfitrión colaboran para mejorar el rendimiento y la eficiencia general. Esto permite una comunicación más rápida y directa entre el sistema operativo invitado y el hardware real, lo que resulta en un mejor rendimiento en comparación con la virtualización tradicional.
  Las máquinas paravirtualizadas comparten el kernel del SO anfitrión y por tanto, no se puede virtualizar máquinas con sistemas operativos incompatibles.
  El hipervisor ofrece un interfaz especial para acceder a los recursos. En ocasiones, es necesario la adaptación del sistema operativo de la máquina virtual. Ofrecen el máximo rendimiento, pero no se pueden usar sistemas operativos sin modificaciones o hardware especifico.

Licencia Creative Commons BY-NC-SA.

- **Paravirtualización ligera:** También llamada virtualización a nivel de sistema operativo, o virtualización basada en contenedores. Es un método de virtualización en el que, sobre el núcleo del sistema operativo se ejecuta una capa de virtualización que permite que existan múltiples instancias aisladas de espacios de usuario. A cada espacio de usuario aislado lo llamamos contenedor. Por lo tanto, un contenedor es un conjunto de procesos aislado, que se ejecuta en un servidor, y que accede a un sistema de ficheros propio, tiene una configuración red propio y accede a los recursos del host (memoria y CPU). Podemos hacer la siguiente clasificación de contenedores:
  - **Los contenedores hardware o de sistema**.
  - **Los contenedores software o de aplicación**.

2.1.- Contenedores de sistemas (LXC)

La virtualización ligera a nivel de hardware, también conocida como contenedores hardware, es una tecnología que permite ejecutar múltiples instancias aisladas de sistemas operativos en un único servidor físico pero aprovechando el sistema operativo anfitrión para acceder al hardware. Un ejemplo común de esta tecnología es LinuX Containers (LXC) que aprovecha el núcleo de Linux del SO del servidor anfitrión (o host).

> Evidentemente, no podemos crear un contenedor hardware LXC con un núcleo del SO diferente al de Linux.

Los contenedores a nivel de hardware ofrecen una gran flexibilidad y agilidad, ya que se pueden iniciar y detener rápidamente, y ocupan menos recursos en comparación con las máquinas virtuales convencionales. Esto los hace ideales para implementar aplicaciones aisladas en entornos de desarrollo, pruebas o producción. Permite una mayor densidad de utilización de recursos, al compartir eficientemente el hardware del sistema, lo que resulta en un menor consumo energético y reserva de recursos en comparación con la virtualización completa. Además, los contenedores LXC son rápidos de crear, clonar y desplegar, lo que facilita la escalabilidad y la replicación de distintos escenarios.

Son una opción más interesante que los contenedores de aplicaciones, como por ejemplo Docker, cuando tenemos la necesidad de emular un hardware específico pero sin sobrecargar al servidor con la virtualización completa de las tradicionales máquinas virtuales (un ejemplo de máquina

Licencia Creative Commons BY-NC-SA.

virtual tradicional son las creadas con VirtualBox)

La virtualización LXC se basa en la creación de contenedores ligeros y aislados de los errores producidos en otros LXC. Estos contenedores permiten ejecutar múltiples instancias de sistemas operativos Linux de forma eficiente y compartiendo los recursos del sistema anfitrión, como CPU, memoria y almacenamiento.

Cada contenedor LXC incluye su propio entorno de usuario y su espacio de archivos aislado, lo que permite la ejecución de aplicaciones y servicios de manera independiente dentro del contenedor. Los contenedores pueden comunicarse entre sí y con el sistema operativo anfitrión mediante mecanismos de comunicación definidos, como sockets UNIX o redes virtuales utilizando la pila del protocolo TCP/IP.

El hardware de los contenedores LXC puede ser configurado, cómo ejemplo podemos reservar memoria máxima de RAM para utilizar, número de núcleos o hilos del procesador a utilizar, etc... Es decir, **podemos hacer lo mismo que podías hasta ahora hacer con las máquinas virtuales pero sin tener que emular el hardware por completo.**

> En resumen, los contenedores de sistema **LXC** es una tecnología que permite la virtualización a nivel de sistema operativo en entornos Linux, a través de paravirtualización ligera para compartir los recursos del sistema anfitrión. Esta tecnología **ofrece beneficios en términos de eficiencia de recursos** y escalabilidad en el desarrollo, pruebas y despliegue continuo de aplicaciones.

2.2.- Contenedores de aplicación (Docker)

La virtualización a nivel de aplicación se basa en el uso de contenedores para encapsular una aplicación y sus dependencias en un entorno aislado. Docker es una de las tecnologías más populares que se utiliza en este enfoque pero no la única.

Un contenedor Docker permite empaquetar una aplicación junto con todas las bibliotecas y archivos necesarios para su ejecución en un entorno aislado. Los contenedores Docker son livianos, portátiles y se pueden ejecutar en cualquier sistema operativo compatible con Docker (Linux, Windows sobre emulación del kernel de Linux, Mac, etc).

Licencia Creative Commons BY-NC-SA.

La virtualización a nivel de aplicación ofrece ventajas en términos de portabilidad, eficiencia y escalabilidad. **Permite a los desarrolladores crear, probar y desplegar aplicaciones de manera más rápida y sencilla, asegurando que la aplicación se ejecute de la misma manera en diferentes entornos sin la necesidad de preocuparse del hardware concreto subyacente.**

Tanto los contenedores sistemas como los contenedores de aplicación son tecnologías importantes en el campo de la computación en la nube. Cada una tiene sus propias características y casos de uso, y es importante comprender sus diferencias y ventajas para tomar decisiones informadas al diseñar y administrar infraestructuras en la nube.

Con Docker, puedes agrupar una aplicación en una especie de "paquete" estándar que se utiliza para el desarrollo y despliegue de software. Este paquete enjaula la aplicación y sus dependencias (librerías) necesarias para que la aplicación funcione sin problemas en cualquier entorno informático.

Antes de Docker, las empresas solían usar máquinas virtuales (VM) para ejecutar aplicaciones aisladas. Estas máquinas virtuales permiten a los desarrolladores dividir un servidor físico en varios. Sin embargo, este enfoque tiene algunas desventajas:

Cada máquina virtual contiene una copia completa del sistema operativo y de la aplicación, así como otros archivos necesarios para su funcionamiento. Estos archivos pueden ocupar mucho espacio en el disco duro, a veces decenas de gigabytes. Además, la virtualización del hardware para cada máquina virtual puede requerir muchos recursos y ralentizar el rendimiento.

> El arranque de las máquinas virtuales tradicionales suele ser incluso más lento que el arranque del SO de la máquina anfitriona.

Cuando cae un contenedor por un fallo del software de la propia aplicación o de alguna de sus dependencias, puede volverse a inicializar el contenedor completo en segundos, sin producirse apenas corte en el servicio.

Sin embargo, los contenedores de aplicaciones no nos sirven para emular escenarios concretos con hardware concretos, ya que la filosofía es la abstracción completa del hardware de la máquina anfitriona. No están diseñados, por tanto, para configurar o limitar el tipo de hardware del contenedor, al contrario de los contenedores LXC, aunque se podrían hacer utilizando comandos especiales.

Los contenedores Docker, originalmente, son un derivado de los contenedores LXC pero pensados para ser utilizados por desarrolladores software y usuarios, utilizando en principio, todos los recursos disponibles por el sistema operativo del host anfitrión.

Licencia Creative Commons BY-NC-SA.

Es muy común utilizar contenedores Docker sobre un contenedor LXC para limitar el hardware que pueda utilizar las aplicaciones ejecutadas en Docker.

> En resumen, **los contenedores de aplicaciones (Docker) encapsulan aplicaciones y sus dependencias, abstrayéndose por completo del hardware del servidor real sin perder la velocidad nativa de una instalación tradicional** y optimizando los recursos, pudiendo ser compartidos por otros contenedores o procesos. Las ventajas en la seguridad, escalabilidad, velocidad de cómputo y economía de recursos hacen que esta tecnología sea la base del éxito de la computación en la nube.

Licencia Creative Commons BY-NC-SA.

3.- Infraestructuras

En el mundo de la computación en la nube, existen diferentes infraestructuras que proporcionan servicios y recursos para almacenar, procesar, administrar datos y aplicaciones. Estas infraestructuras se clasifican en tres modelos principales: **nube pública, nube privada y nube híbrida.**

A continuación, exploraremos cada uno de ellos:

- **Nube pública:**
  La nube pública es una infraestructura en la que los servicios y recursos se ofrecen a través de proveedores de servicios accesibles a través de Internet. Estos proveedores, como Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform, entre otros, poseen y administran la infraestructura subyacente, incluidos los servidores, redes y sistemas de almacenamiento.
  En la nube pública, los usuarios pueden acceder a servicios y **recursos bajo demanda**, como almacenamiento, servidores virtuales, bases de datos y herramientas de desarrollo. Esto permite a las empresas y a los individuos escalar sus operaciones sin necesidad de invertir en infraestructura física. Un ejemplo práctico sería una empresa que utiliza la nube pública para alojar su sitio web y almacenar sus datos en servidores externos.
- **Nube privada:**
  La nube privada es una infraestructura en la que los servicios y recursos se implementan y operan exclusivamente para una única organización. La infraestructura de la nube privada puede estar ubicada en las instalaciones de la organización o ser gestionada por un proveedor de servicios externo. En este modelo, la organización tiene un control total sobre la infraestructura y puede adaptarla a sus necesidades específicas.
  Las nubes privadas son utilizadas principalmente por organizaciones que requieren un alto nivel de seguridad y control sobre sus datos y aplicaciones. Por ejemplo, un banco puede optar por utilizar una nube privada para almacenar información confidencial de sus clientes, garantizando así un mayor nivel de protección y cumplimiento normativo.
- **Nube híbrida:**
  La nube híbrida es una combinación de la nube pública y la nube privada. En este modelo, las organizaciones utilizan una infraestructura mixta que integra servicios y recursos de la nube pública y la nube privada. Esta combinación permite a las organizaciones aprovechar los beneficios de ambos modelos, equilibrando la flexibilidad y escalabilidad de la nube pública con el control y la seguridad de la nube privada.
  En una nube híbrida, las aplicaciones y los datos pueden moverse de forma transparente entre la nube pública y la nube privada según las necesidades y los requisitos específicos de la organización. Por ejemplo, una empresa puede utilizar una nube privada para

Licencia Creative Commons BY-NC-SA.

almacenar datos sensibles, mientras que utiliza la nube pública para escalar sus recursos de manera rentable durante períodos de alta demanda.

> En resumen, en la computación en la nube existen tres modelos principales de infraestructura: nube pública, nube privada y nube híbrida. Cada uno de estos modelos ofrece distintas características y beneficios. **La elección de la infraestructura adecuada depende de las necesidades y requisitos de la organización en términos de control, seguridad, escalabilidad y costos.**

3.1.- Plataformas de Cloud Computing

En el mercado de la computación en la nube, existen diversas plataformas que ofrecen servicios y recursos para respaldar las necesidades de empresas y usuarios individuales. A continuación, analizaremos algunas de las principales alternativas y compararemos sus ventajas y desventajas



1. **Amazon Web Services (AWS):**
   Ventajas: AWS es una de las plataformas líderes en el mercado de la nube, ofreciendo una amplia gama de servicios y soluciones altamente escalables. Cuenta con una gran variedad de servicios, como almacenamiento, cómputo, bases de datos y herramientas de desarrollo. Además, tiene una amplia presencia global con múltiples regiones y zonas de disponibilidad.
   Desventajas: La complejidad de AWS puede resultar abrumadora para los usuarios sin experiencia previa en la nube. Algunos servicios pueden tener un costo elevado, y su amplia gama de opciones puede dificultar la selección de los servicios adecuados para las necesidades específicas.

2. **Microsoft Azure:**

Licencia Creative Commons BY-NC-SA.

Ventajas: Azure es otra plataforma líder que ofrece una amplia gama de servicios en la nube. Es conocida por su integración con otras herramientas y servicios de Microsoft, como Office 365 y Active Directory. Además, Azure proporciona una experiencia fluida para aquellos que ya están familiarizados con el entorno de Windows.
Desventajas: Al igual que con AWS, la complejidad de Azure puede dificultar la comprensión para los principiantes. Algunos servicios pueden tener un aprendizaje inicial más costoso y su presencia global puede ser menor en comparación con AWS.

3. **Google Cloud Platform (GCP):**
   Ventajas: GCP ofrece una amplia gama de servicios en la nube y se destaca por su enfoque en la inteligencia artificial y el análisis de datos. Google tiene una infraestructura global sólida y ofrece servicios como Google Kubernetes Engine para la administración de contenedores.
   Desventajas: Aunque GCP ha ganado popularidad, aún se encuentra en un segundo plano en comparación con AWS y Azure. La documentación puede ser menos completa y algunos servicios pueden tener menos características en comparación con sus competidores.

Es importante tener en cuenta que estas son solo algunas de las muchas plataformas de computación en la nube disponibles en el mercado. Al seleccionar una plataforma, es esencial considerar las necesidades específicas del proyecto, los costos asociados, el soporte y la comunidad de usuarios, así como la curva de aprendizaje requerida.

3.1.1.- Comparativa de los servicios ofrecidos por las tres principales plataformas de Cloud Computing

	Amazon Web Services (AWS)	Microsoft Azure	Google Cloud Platform (GCP)
Almacenamiento	Amazon S3, Amazon EBS, Amazon Glacier	Azure Blob Storage, Azure Files	Google Cloud Storage, Google Cloud Filestore
Cómputo	Amazon EC2, AWS Lambda, AWS Batch	Azure Virtual Machines, Azure Functions	Google Compute Engine, Google Cloud Functions
Bases de datos	Amazon RDS, Amazon DynamoDB	Azure SQL Database, Azure Cosmos DB, Azure Database for MySQL	BigQuery

Licencia Creative Commons BY-NC-SA.

Redes	Amazon VPC, Amazon CloudFront	Azure Virtual Network, Azure ExpressRoute	Virtual Private Cloud (VPC), Cloud Load Balancing
Herramientas de desarrollo	AWS CloudFormation, AWS CLI, AWS SDK	Azure Resource Manager, Azure DevOps	Google Cloud Deployment Manager, Cloud SDK
IA y analítica de datos	Amazon Redshift, Amazon Athena, AWS Glue	Azure Machine Learning, Azure Cognitive Services	BigQuery ML, AI Platform, TensorFlow, AutoML
Seguridad	AWS Identity and Access Management, AWS CloudTrail	Azure Active Directory, Azure Security Center, Azure Sentinel	Google Cloud Identity and Access Management, Cloud IAM

Esta tabla es solo una muestra de los servicios ofrecidos por cada plataforma y que cada proveedor de servicios en la nube tiene una amplia gama de opciones disponibles. Al seleccionar una plataforma, es recomendable investigar más a fondo y considerar las necesidades específicas del proyecto.

> En resumen, AWS, Azure y GCP son algunas de las principales plataformas de computación en la nube. Cada una tiene sus ventajas y desventajas, y la elección depende de los requisitos del proyecto y las preferencias individuales.

3.2.- Tipos de virtualización utilizados por las plataformas AWS, Microsoft Azure y Google Cloud Platform

En la computación en la nube, los proveedores de servicios como AWS, Azure y GCP ofrecen diferentes tipos de virtualización para aprovechar al máximo los recursos y garantizar un entorno eficiente y escalable. A continuación, analizaremos los principales tipos de virtualización utilizados por estos proveedores:

1. **Virtualización basada en hardware (virtualización tradicional):**
   AWS: Amazon EC2 (Elastic Compute Cloud) de AWS utiliza virtualización basada en hardware para asignar recursos virtuales a instancias virtuales en servidores físicos. Esto

Licencia Creative Commons BY-NC-SA.

permite a los usuarios ejecutar múltiples instancias virtuales en un solo servidor físico, maximizando la utilización de recursos y facilitando la escalabilidad.
Azure: Azure Virtual Machines también utiliza virtualización basada en hardware para proporcionar instancias virtuales a los usuarios. Al igual que AWS, Azure aprovecha esta tecnología para optimizar la utilización de los recursos físicos y permitir la escalabilidad según la demanda.
GCP: Google Compute Engine utiliza un enfoque similar de virtualización basada en hardware para asignar recursos virtuales a las instancias de máquinas virtuales. Esto permite a los usuarios ejecutar sus cargas de trabajo en servidores físicos compartidos de manera eficiente.

2. **Contenedores (paravirtualización):**
   AWS: Amazon Elastic Container Service (ECS) y Amazon Elastic Kubernetes Service (EKS) ofrecen servicios de contenedores gestionados. Los contenedores son unidades de software que enjaulan las aplicaciones y sus dependencias, lo que permite una implementación más rápida y eficiente, así como una mayor portabilidad y seguridad de las aplicaciones.
   Azure: Azure Container Instances y Azure Kubernetes Service (AKS) brindan servicios de contenedores en Azure. Estos servicios permiten a los usuarios implementar y administrar fácilmente contenedores en la nube.
   GCP: Google Kubernetes Engine (GKE) es el servicio de contenedores administrados de GCP. Permite a los usuarios orquestar y administrar contenedores de manera eficiente utilizando la plataforma Kubernetes, lo que simplifica el proceso de implementación y escalado de aplicaciones en contenedores. Kubernetes fue originalmente diseñado por Google y con posterioridad liberó su código a la comunidad, por eso las tres compañías que estamos comparando utilizan Kubernetes en sus plataformas.
3. **Funciones como servicio (Function as a Service):**
   AWS: AWS Lambda es un servicio de computación sin servidor que permite a los desarrolladores ejecutar código en respuesta a eventos sin necesidad de administrar la infraestructura. Los desarrolladores solo necesitan cargar su código y AWS se encarga de la ejecución y el escalado automático según la demanda. Es el servicio más barato de cómputo en la nube a costa de no garantizar la disponibilidad inmediata, cuando los servidores se encuentran a su máxima capacidad. Este servicio no se debe utilizar para aplicaciones críticas en una organización.
   Azure: Azure Functions es el servicio de funciones sin servidor de Azure que permite a los desarrolladores ejecutar código en respuesta a eventos. Al igual que AWS Lambda, Azure Functions se encarga de la administración de la infraestructura, permitiendo a los desarrolladores centrarse en el desarrollo de sus aplicaciones.
   GCP: Google Cloud Functions es el servicio equivalente en GCP que permite a los desarrolladores escribir y ejecutar funciones sin servidor en la nube. Google Cloud Functions proporciona un entorno de ejecución sin servidor y escalado automático basado en la demanda.

Licencia Creative Commons BY-NC-SA.

En resumen, los proveedores de servicios en la nube como AWS, Azure y GCP ofrecen diferentes tipos de virtualización, incluyendo la **virtualización basada en hardware, los contenedores y las funciones sin servidor.** Cada tipo tiene sus propias características y beneficios, y es importante evaluar cuidadosamente las necesidades del proyecto para seleccionar la opción más adecuada.

3.3.- Almacenamiento en la nube

En la computación en la nube, el almacenamiento de datos es un aspecto fundamental. Los proveedores de servicios en la nube, como AWS, Azure y GCP, ofrecen diferentes tipos de almacenamiento para satisfacer las necesidades de los usuarios. A continuación, analizaremos los principales tipos de almacenamiento disponibles en cada plataforma:

1. **Almacenamiento en bloque:**
   Características: El almacenamiento en bloque se basa en la asignación de bloques de datos que se pueden acceder y manipular a nivel de bloques individuales. Proporciona un rendimiento de E/S de alto rendimiento y es ideal para aplicaciones que requieren acceso directo y rápido a los datos en un formato estructurado, como bases de datos y sistemas de archivos.
   Ejemplo: Si estás ejecutando una base de datos en la nube, como un sistema de gestión de bases de datos relacional (RDBMS), puedes utilizar el almacenamiento en bloque para alojar los archivos de datos y los registros de transacciones. Esto garantizará un acceso rápido y una alta disponibilidad de los datos, lo que es esencial para aplicaciones que requieren una alta velocidad de lectura y escritura.
   AWS: Amazon Elastic Block Store (EBS) proporciona almacenamiento en bloque persistente para instancias de Amazon EC2. Es ideal para aplicaciones que requieren un alto rendimiento y persistencia de datos. EBS permite crear y adjuntar volúmenes de almacenamiento a instancias EC2 y es compatible con características como instantáneas para la copia de seguridad y la recuperación.
   Azure: Azure Managed Disks es el servicio de almacenamiento en bloque de Azure. Permite a los usuarios crear y administrar discos virtuales persistentes para las máquinas virtuales de Azure. Los discos administrados ofrecen durabilidad, disponibilidad y redundancia de datos, lo que garantiza la integridad de la información almacenada.
   GCP: Google Compute Engine ofrece Google Persistent Disk como su servicio de almacenamiento en bloque. Es similar a EBS y Managed Disks, proporcionando almacenamiento persistente y de alto rendimiento para las máquinas virtuales de GCP. Google Persistent Disk ofrece opciones de lectura y escritura rápidas, así como características de seguridad y confiabilidad.

Licencia Creative Commons BY-NC-SA.

2. **Almacenamiento de objetos:**
   Características: El almacenamiento de objetos está diseñado para almacenar y acceder a grandes cantidades de datos no estructurados, como archivos multimedia, archivos de copia de seguridad y documentos. Los datos se organizan en objetos que contienen los datos y los metadatos asociados. El almacenamiento de objetos proporciona una escalabilidad masiva, durabilidad y capacidad de almacenamiento económica.
   Ejemplo: Si tienes una aplicación de transmisión de videos en la nube, puedes utilizar el almacenamiento de objetos para almacenar los archivos de video y los metadatos relacionados, como información sobre el título, la duración y el formato. El almacenamiento de objetos es ideal para este caso, ya que permite un acceso eficiente a los archivos de video y puede manejar una gran cantidad de datos sin problemas.
   AWS: Amazon Simple Storage Service (S3) es un servicio de almacenamiento de objetos escalable y duradero. S3 permite almacenar y recuperar grandes cantidades de datos, como archivos multimedia, archivos de copia de seguridad y archivos estáticos de sitios web. Ofrece una alta durabilidad, redundancia y capacidad de recuperación de datos.
   Azure: Azure Blob Storage es el servicio de almacenamiento de objetos de Azure. Proporciona almacenamiento de objetos altamente escalable y duradero, adecuado para almacenar y acceder a datos no estructurados. Azure Blob Storage es compatible con diferentes niveles de acceso, lo que permite optimizar los costos según la frecuencia de acceso a los datos.
   GCP: Google Cloud Storage es el servicio de almacenamiento de objetos de GCP. Permite almacenar y acceder a datos de manera segura y escalable. Google Cloud Storage ofrece opciones de clase de almacenamiento para adaptarse a diferentes requisitos de rendimiento y costo, lo que permite optimizar el almacenamiento en función de las necesidades del proyecto.

3. **Almacenamiento de archivos:**
   Características: El almacenamiento de archivos proporciona un sistema de archivos compartido en la nube, similar al sistema de archivos utilizado en los sistemas locales. Permite compartir archivos entre múltiples instancias o servicios y es adecuado para aplicaciones que requieren una colaboración y un acceso compartido a los datos.
   Ejemplo: Imagina que tienes un equipo de desarrollo trabajando en un proyecto en la nube. Puedes utilizar el almacenamiento de archivos para compartir el código fuente, los archivos de configuración y otros recursos necesarios entre los miembros del equipo. El almacenamiento de archivos facilita la colaboración y garantiza que todos los miembros del equipo tengan acceso a los mismos archivos actualizados.
   AWS: Amazon Elastic File System (EFS) ofrece almacenamiento de archivos escalable y compartido para instancias de EC2. EFS proporciona un sistema de archivos accesible desde múltiples instancias, lo que facilita el uso compartido de datos y la colaboración en entornos distribuidos.
   Azure: Azure Files permite compartir archivos en la nube de Azure. Proporciona

Licencia Creative Commons BY-NC-SA.

almacenamiento de archivos totalmente administrado y compatible con el Protocolo de Transferencia de Archivos (FTP), el Protocolo de Sistema de Archivos de Red (NFS) y el Protocolo de Bloque Extendido de Microsoft (SMB).
GCP: Google Cloud Filestore es el servicio de almacenamiento de archivos de GCP. Proporciona un sistema de archivos de red (NFS) administrado y escalable para cargas de trabajo de alto rendimiento. Google Cloud Filestore permite compartir archivos entre instancias de GCP y proporciona una experiencia similar a la del almacenamiento en red local.

Cada tipo de almacenamiento tiene sus propias ventajas y desventajas, y la elección depende de los requisitos del proyecto en términos de rendimiento, durabilidad, escalabilidad y costo.

> En resumen, los proveedores de servicios en la nube como AWS, Azure y GCP ofrecen diferentes tipos de almacenamiento, incluyendo almacenamiento en bloque, almacenamiento de objetos y almacenamiento de archivos. Cada tipo tiene sus propias características y beneficios, y es importante evaluar cuidadosamente los requisitos del proyecto para seleccionar la opción de almacenamiento más adecuada.

3.4.- Bases de datos en la nube

Las bases de datos son elementos fundamentales en la mayoría de las aplicaciones, y los proveedores de servicios en la nube, como AWS, Azure y GCP, ofrecen una variedad de opciones para satisfacer las necesidades de almacenamiento y gestión de datos de los usuarios. A continuación, analizaremos los principales tipos de bases de datos disponibles en cada plataforma:

1. **Bases de datos relacionales (SQL):**
   Características: Las bases de datos relacionales se basan en el modelo relacional, donde los datos se organizan en tablas con filas y columnas. Utilizan un lenguaje de consulta estructurado (SQL) para gestionar y manipular los datos. Proporcionan una estructura fija y definida, garantizan la integridad de los datos y admiten relaciones complejas entre las tablas.
   Ejemplo: Una tienda web que necesita almacenar información de productos, clientes y pedidos podría utilizar una base de datos relacional. Las tablas se crearían para representar cada entidad (productos, clientes, pedidos), y se establecerían relaciones entre ellas para garantizar la integridad de los datos. Esto permite realizar consultas complejas, como encontrar todos los pedidos de un cliente específico.
   AWS: Amazon Relational Database Service (RDS) ofrece bases de datos relacionales administradas, como MySQL, PostgreSQL, Oracle y SQL Server. Proporciona una gestión simplificada de las bases de datos y opciones de escalabilidad para adaptarse a las

Licencia Creative Commons BY-NC-SA.

necesidades del proyecto.
Azure: Azure SQL Database es el servicio de base de datos relacional administrada de Azure. Proporciona una base de datos SQL totalmente administrada y escalable que admite tanto aplicaciones pequeñas como empresariales. Azure SQL Database ofrece opciones de rendimiento y escalabilidad flexibles.
GCP: Google Cloud SQL es el servicio de base de datos relacional de GCP. Permite ejecutar bases de datos MySQL y PostgreSQL administradas en la nube. Google Cloud SQL ofrece alta disponibilidad, rendimiento y seguridad para aplicaciones que requieren bases de datos relacionales.

2. **Bases de datos NoSQL:**
   Características: Las bases de datos NoSQL (Not Only SQL) son diseñadas para manejar grandes volúmenes de datos no estructurados o semiestructurados. No siguen el modelo relacional y pueden almacenar datos en formatos como documentos, grafos o claves-valor. Ofrecen una alta escalabilidad, flexibilidad y rendimiento.
   Ejemplo: Un sistema de análisis de registros de servidores, que recopila y analiza grandes volúmenes de datos de registros generados por servidores, puede utilizar una base de datos NoSQL. Cada registro se almacenaría como un documento independiente, y la base de datos permitiría realizar consultas y búsquedas rápidas en función de diferentes campos, sin la necesidad de definir una estructura fija de antemano.
   AWS: Amazon DynamoDB es un servicio de base de datos NoSQL totalmente administrado y escalable en la nube. Es adecuado para aplicaciones que requieren un acceso rápido a datos estructurados y no estructurados. DynamoDB ofrece una latencia baja y una escalabilidad automática.
   Azure: Azure Cosmos DB es una base de datos NoSQL globalmente distribuida y escalable en la nube. Proporciona una latencia baja y una alta disponibilidad, y es compatible con múltiples modelos de datos, como documentos, grafos y claves-valor.
   GCP: Google Cloud Firestore es una base de datos NoSQL flexible y escalable de GCP. Permite almacenar, sincronizar y consultar datos en tiempo real. Firestore es adecuado para aplicaciones web y móviles que requieren una sincronización de datos en tiempo real y una escalabilidad horizontal.

3. **Bases de datos en memoria:**
   Características: Las bases de datos en memoria almacenan y acceden a los datos en la memoria principal en lugar de utilizar el almacenamiento en disco. Esto permite un acceso extremadamente rápido a los datos, lo que resulta en un rendimiento superior. Son ideales para aplicaciones que requieren tiempos de respuesta ultra rápidos y un acceso eficiente a los datos.
   Ejemplo: Una aplicación de transmisión en vivo, como un servicio de transmisión de música, podría utilizar una base de datos en memoria para almacenar la información sobre las canciones que se están reproduciendo actualmente y las preferencias de los usuarios. Al tener los datos en memoria, se logra una latencia baja, lo que garantiza una experiencia de transmisión fluida y sin interrupciones.

Licencia Creative Commons BY-NC-SA.

AWS: Amazon ElastiCache es un servicio de almacenamiento en caché en memoria compatible con Redis y Memcached. Proporciona un rendimiento de alta velocidad y una capacidad de escalado automático para aplicaciones que requieren un acceso rápido a datos en memoria.
Azure: Azure Cache for Redis es un servicio de caché en memoria basado en Redis. Permite almacenar datos en memoria y acelerar las aplicaciones con una latencia baja y un alto rendimiento.
GCP: Google Cloud Memorystore es un servicio de almacenamiento en memoria basado en Redis. Proporciona un almacenamiento de caché en memoria totalmente administrado y escalable para mejorar el rendimiento de las aplicaciones.

> En resumen, los proveedores de servicios en la nube ofrecen una variedad de tipos de bases de datos, incluyendo bases de datos relacionales (SQL), bases de datos NoSQL y bases de datos en memoria. Cada tipo tiene sus propias características y beneficios, y es importante considerar las necesidades del proyecto al seleccionar la opción de base de datos adecuada.

Licencia Creative Commons BY-NC-SA.

4.- Administración de sistemas en la nube

La administración de sistemas en la nube es una parte fundamental de la computación en la nube. **Implica la configuración, supervisión y mantenimiento de los recursos y servicios** en la nube para garantizar un rendimiento óptimo, seguridad y disponibilidad.

La configuración de recursos implica la creación y personalización de instancias virtuales, redes, almacenamiento y otros componentes en la nube. Los proveedores de servicios en la nube ofrecen interfaces intuitivas y herramientas para configurar y adaptar estos recursos según las necesidades del proyecto.
Algunas tareas comunes de configuración incluyen **la creación de máquinas virtuales, asignación de recursos de red, configuración de cortafuegos y establecimiento de políticas de acceso**.

El mantenimiento regular y las actualizaciones son esenciales para garantizar la seguridad y la estabilidad de los sistemas en la nube. Los administradores deben aplicar parches de seguridad, actualizaciones de software y realizar tareas de mantenimiento programadas.
La automatización de tareas de mantenimiento, como la programación de actualizaciones y copias de seguridad, es fundamental para garantizar un entorno confiable y minimizar el tiempo de inactividad.

4.1.- Seguridad en la nube

La seguridad en la nube es una preocupación fundamental para proteger los datos y los recursos en los entornos de computación en la nube. A continuación, exploraremos algunas de las principales alternativas y enfoques de seguridad en la nube:

- **Acceso seguro:**
  Los proveedores de servicios en la nube ofrecen mecanismos para garantizar un acceso seguro a los recursos. Esto incluye autenticación multifactor (MFA), políticas de contraseñas fuertes y control de acceso basado en roles (RBAC). Estas medidas ayudan a proteger los datos y evitan accesos no autorizados.
  Aunque el acceso seguro es una práctica estándar, los usuarios deben ser conscientes de

Licencia Creative Commons BY-NC-SA.

la importancia de mantener sus credenciales seguras y evitar el uso de contraseñas débiles o compartidas. La pérdida o el compromiso de las credenciales pueden abrir brechas de seguridad.

- **Encriptación de datos:**
  La encriptación de datos es una medida esencial para proteger la confidencialidad de la información almacenada y transmitida en la nube. Los proveedores de servicios en la nube ofrecen opciones de encriptación para proteger los datos en reposo y en tránsito. Esto garantiza que incluso si los datos son interceptados o comprometidos, serán incomprensibles sin la clave de encriptación.
  Desventajas: La encriptación puede aumentar la complejidad y el costo de implementación. Además, es fundamental mantener un control estricto sobre las claves de encriptación para evitar su pérdida o mal uso.

- **Respaldo y recuperación de datos:**
  Los servicios en la nube ofrecen opciones de respaldo y recuperación de datos para proteger contra la pérdida o corrupción de información. Estas opciones permiten realizar copias de seguridad periódicas y restaurar los datos en caso de desastres o fallos del sistema.
  Si bien los proveedores de servicios en la nube se encargan del respaldo de datos, es responsabilidad del usuario configurar y gestionar adecuadamente las políticas de respaldo. Una configuración inadecuada puede llevar a una pérdida de datos irreparable.

- **Protección contra amenazas:**
  Los proveedores de servicios en la nube implementan medidas de seguridad avanzadas, como firewalls, sistemas de detección y prevención de intrusiones (IDS/IPS) y protección contra malware. Estas soluciones ayudan a mitigar amenazas y proteger los recursos en la nube.
  Aunque los proveedores de servicios en la nube ofrecen protección contra amenazas, los usuarios también deben tomar medidas adicionales, como mantener sus sistemas actualizados, implementar soluciones de seguridad en sus aplicaciones y mantenerse informados sobre las últimas vulnerabilidades y ataques.

Ten en cuenta que la seguridad en la nube es un tema complejo y en constante evolución, por lo que la comparativa siguiente se basa en características generales y puede variar con el tiempo:

1. **Seguridad en Amazon Web Services:**
   Ventajas: Amplia gama de servicios de seguridad, como AWS Identity and Access Management (IAM), que permite gestionar el acceso y los permisos de los usuarios. Ofrece herramientas avanzadas de seguridad, como AWS CloudTrail para el registro de auditoría y AWS Shield para la protección contra ataques DDoS.
   Cumple con numerosos estándares de seguridad y privacidad, como ISO 27001, HIPAA y

Licencia Creative Commons BY-NC-SA.

GDPR.
Desventajas: Puede ser un poco complejo de configurar y administrar para usuarios sin experiencia previa en la nube.
Algunas características de seguridad avanzadas pueden tener un costo adicional.

2. **Seguridad en Microsoft Azure:**
   Ventajas: Integra de manera nativa con los servicios y herramientas de Microsoft, lo que facilita la administración y el cumplimiento de políticas de seguridad.
   Ofrece Azure Active Directory para la gestión de identidades y acceso.
   Proporciona opciones de cifrado y cumplimiento de normativas como GDPR.
   Desventajas: Al igual que AWS, puede requerir un tiempo de aprendizaje para comprender completamente todas las opciones de seguridad disponibles.
   Algunos servicios de seguridad avanzados pueden tener un costo adicional.
3. **Seguridad en Google Cloud Platform:**
   Ventajas: Enfoque en la seguridad por diseño, con una infraestructura confiable y certificaciones de cumplimiento, como ISO 27001 y SOC 2.
   Proporciona herramientas como Cloud Identity and Access Management (IAM) y Cloud Security Scanner.
   Ofrece opciones avanzadas de seguridad, como el cifrado de datos en reposo y en tránsito.
   Desventajas: La interfaz de usuario y la documentación pueden no ser tan intuitivas para los principiantes en la nube.
   Algunos servicios de seguridad pueden requerir conocimientos técnicos más avanzados.

> Es importante destacar que **la seguridad en la nube es una responsabilidad compartida entre el proveedor de servicios en la nube y el cliente**. Los usuarios deben estar concienciados de las mejores prácticas de seguridad y tomar medidas adicionales para proteger sus datos y recursos en la nube.

4.2.- Redes y entrega de contenido

En el campo de la computación en la nube, la administración de las redes para la comunicación entre diferentes instancias (máquinas virtuales y contenedores) y con los usuarios, es un campo de vital importancia. A continuación, definiremos las principales conceptos para las redes y entrega de contenidos:

Licencia Creative Commons BY-NC-SA.

- **Aislamiento y segmentación:** La computación en la nube permite crear redes virtuales aisladas para cada máquina virtual, lo que proporciona un entorno seguro y aislado para las aplicaciones y los datos.

- **Conectividad y escalabilidad:** Las máquinas virtuales pueden comunicarse entre sí a través de redes virtuales, lo que facilita la configuración de aplicaciones distribuidas y la escalabilidad horizontal mediante la adición de más instancias de máquinas virtuales.

- **Enrutamiento y balanceo de carga:** La nube ofrece funcionalidades para enrutar el tráfico de red entre las máquinas virtuales y distribuir la carga de trabajo de manera equilibrada, lo que mejora el rendimiento y la disponibilidad de las aplicaciones.

- **Microservicios y arquitectura modular:** Los contenedores permiten desplegar aplicaciones en unidades más pequeñas y modulares, lo que facilita la construcción de arquitecturas basadas en microservicios. Cada contenedor puede tener su propia configuración de red, lo que proporciona un aislamiento y una gestión más eficiente de los recursos.

- **Orquestación y escalabilidad:** Las plataformas de orquestación de contenedores, como Kubernetes, permiten administrar y escalar automáticamente los contenedores en función de la demanda de tráfico y recursos. Esto facilita la gestión de aplicaciones complejas y la adaptación rápida a cambios en la carga de trabajo.

- **Servicios de red definidos por software (SDN):** Los contenedores se pueden conectar a través de redes definidas por software, lo que proporciona flexibilidad y control sobre la topología de red. Esto permite la implementación de políticas de seguridad y enrutamiento personalizadas.

- **Firewall:** En el contexto de la computación en la nube, un Firewall en la nube es un servicio que se encarga de monitorear y controlar el tráfico de red que ingresa y sale de una infraestructura en la nube. Sus funciones principales son: filtrado de tráfico, segmentación de red, detección y prevención de intrusiones.

- **DNS (Domain Name System):** El DNS es un sistema fundamental que se utiliza para traducir nombres de dominio (por ejemplo, www.ejemplo.com) en direcciones IP numéricas que las computadoras pueden entender. En el contexto de la computación en la nube, los servicios de DNS en la nube ofrecen:
  - **Resolución de nombres de dominio:** Los servicios de DNS en la nube permiten asociar nombres de dominio con direcciones IP de los recursos en la nube, como máquinas virtuales o servicios web. Esto facilita el acceso a estos recursos utilizando nombres de dominio amigables en lugar de recordar direcciones IP numéricas. Gestión de registros DNS: Los servicios de DNS en la nube proporcionan interfaces y

Licencia Creative Commons BY-NC-SA.

herramientas para administrar los registros DNS, como la creación, modificación o eliminación de registros DNS. Esto permite una fácil gestión y actualización de los nombres de dominio y las direcciones IP asociadas.
- **Redireccionamiento y balanceo de carga:** Los servicios de DNS en la nube también pueden redirigir el tráfico de manera inteligente entre diferentes recursos o ubicaciones en función de las políticas de balanceo de carga. Esto permite distribuir eficientemente la carga de trabajo y mejorar el rendimiento de las aplicaciones en la nube.

Se ha realizado una comparativa de las principales alternativas en el mercado de la computación en la nube en términos de redes y entrega de contenido:

1. **AWS (Amazon Web Services):**
   Amazon Virtual Private Cloud (VPC): Permite crear una red virtual aislada en la nube, donde puedes definir subredes, reglas de firewall y configurar la conectividad con tu red local.
   Amazon CloudFront: Es un servicio de distribución de contenido (CDN) que ayuda a acelerar la entrega de contenido estático y dinámico a nivel mundial, reduciendo la latencia y mejorando la experiencia del usuario.
2. **Azure (Microsoft Azure):**
   Virtual Network (VNet): Permite crear una red virtual en Azure, donde puedes definir subredes, grupos de seguridad de red y establecer conexiones VPN para conectar con tu red local.
   Azure Content Delivery Network (CDN): Proporciona una red de distribución de contenido global para acelerar la entrega de contenido estático y dinámico, ofreciendo una experiencia de usuario más rápida.
3. **GCP (Google Cloud Platform):**
   Virtual Private Cloud (VPC): Ofrece redes virtuales aisladas para crear una infraestructura de red personalizada, permitiendo la definición de subredes, firewalls y conectividad con redes locales mediante VPN.
   Cloud CDN: Es un servicio de distribución de contenido que utiliza la infraestructura de Google para entregar contenido de manera rápida y segura, reduciendo la latencia y optimizando el rendimiento.

En términos generales, todas las plataformas ofrecen funcionalidades similares para crear redes virtuales y optimizar la entrega de contenido. Sin embargo, hay diferencias en las características y enfoques específicos de cada proveedor.

Licencia Creative Commons BY-NC-SA.

En resumen, tanto las redes de máquinas virtuales como las redes de contenedores en la computación en la nube ofrecen ventajas en términos de aislamiento, escalabilidad y gestión eficiente de recursos. La elección entre máquinas virtuales y contenedores dependerá de las necesidades y características específicas de cada aplicación o proyecto.

4.3.- Monitoreo y escalado automático en la nube

El monitoreo en la computación en la nube es una práctica crucial para garantizar el rendimiento, la disponibilidad y la seguridad de los recursos y servicios en la nube. Consiste en la recopilación, el análisis y la presentación de datos relacionados con el estado y el funcionamiento de los componentes de la infraestructura en la nube.

La supervisión es crucial para mantener un entorno en la nube seguro y optimizado. Los administradores de sistemas deben monitorear el rendimiento de los recursos, analizar métricas y registros, y recibir alertas en caso de problemas.

Herramientas como paneles de control, monitoreo en tiempo real y análisis de registros facilitan la supervisión de los sistemas en la nube. Los administradores pueden verificar la utilización de recursos, detectar cuellos de botella y realizar ajustes para mejorar el rendimiento. A continuación, se explicarán algunos aspectos importantes del monitoreo en la computación en la nube:

- **Supervisión de recursos:** El monitoreo en la nube implica el seguimiento de los recursos en tiempo real, como instancias de máquinas virtuales, contenedores, bases de datos y servicios. Esto implica recopilar datos sobre el uso de recursos, como la CPU, la memoria, el almacenamiento y el ancho de banda, para evaluar su rendimiento y detectar posibles cuellos de botella o problemas.
- **Recopilación de registros (logs) y eventos:** Además de supervisar los recursos, el monitoreo en la nube también implica la recopilación y el análisis de registros y eventos. Los registros son logs detallados de las actividades que ocurren en los sistemas, mientras que los eventos son notificaciones sobre cambios importantes o sucesos relevantes. La recopilación y el análisis de estos registros y eventos pueden ayudar a identificar problemas, analizar el comportamiento de las aplicaciones y respaldar la investigación de incidentes de seguridad.
- **Alertas y notificaciones:** El monitoreo en la nube permite configurar alertas y notificaciones para recibir advertencias en tiempo real sobre problemas o eventos

Licencia Creative Commons BY-NC-SA.

anormales. Estas alertas pueden enviarse por correo electrónico, mensajes de texto u otros medios, y ayudan a los administradores a responder rápidamente a situaciones críticas y tomar medidas correctivas antes de que afecten la disponibilidad o el rendimiento de los servicios en la nube.

- **Análisis y visualización de datos:** Una parte esencial del monitoreo en la nube es el análisis y la visualización de los datos recopilados. Esto implica utilizar herramientas y técnicas para analizar tendencias, identificar patrones y extraer información significativa de los datos de monitoreo. La visualización de datos en forma de gráficos, tablas o paneles facilita la comprensión y la toma de decisiones basadas en el monitoreo.
- **Automatización y escalado:** En la computación en la nube, el monitoreo se puede integrar con la automatización y el escalado automático de recursos. Esto significa que, en función de los datos de monitoreo y las reglas predefinidas, se pueden tomar acciones automáticas, como agregar o eliminar instancias de máquinas virtuales, ajustar la capacidad de almacenamiento o aplicar políticas de escalado para mantener el rendimiento y la disponibilidad óptimos.

La escalabilidad y la elasticidad son características clave de la computación en la nube. Los administradores de sistemas deben comprender cómo escalar los recursos de manera efectiva para satisfacer las demandas cambiantes de la carga de trabajo.
Mediante el monitoreo y la planificación, los administradores pueden ajustar la capacidad de los recursos en función de la demanda. Esto implica agregar o reducir instancias virtuales, ajustar la capacidad de almacenamiento y configurar la distribución de carga para garantizar un rendimiento óptimo.

Tipos de escalado en el Cloud Computing:

- **Escalado horizontal (o "scale-out"):**
  El escalado horizontal implica agregar más instancias o nodos a un sistema distribuido para distribuir la carga de trabajo de manera más equitativa. En lugar de aumentar la capacidad de una única instancia, se agregan más instancias idénticas y se distribuye la carga entre ellas. Esto permite manejar una mayor cantidad de solicitudes y aumentar la capacidad del sistema.
  Por ejemplo, si una aplicación en la nube está experimentando un aumento en la demanda, en lugar de aumentar los recursos de una única instancia, se pueden agregar más instancias virtuales que ejecuten la misma aplicación. De esta manera, se distribuye la carga de trabajo entre las instancias adicionales, lo que permite aumentar la capacidad y mantener un rendimiento óptimo.
  El escalado horizontal es altamente flexible y permite una mejor resistencia a fallos, ya que si una instancia falla, las demás pueden continuar atendiendo las solicitudes. Además, es posible agregar o eliminar instancias según la demanda, lo que permite un uso eficiente de los recursos.
- **Escalado vertical (o "scale-up"):**

Licencia Creative Commons BY-NC-SA.

El escalado vertical implica aumentar la capacidad de una instancia o nodo existente en términos de recursos como CPU, memoria RAM o capacidad de almacenamiento. En lugar de agregar más instancias, se mejoran las características de una única instancia para manejar una mayor carga de trabajo.
Por ejemplo, si una aplicación en la nube está experimentando un aumento en el procesamiento intensivo de CPU, se puede escalar verticalmente agregando más núcleos de CPU a la instancia existente. Esto permite que la instancia maneje más tareas simultáneamente y aumente su capacidad de rendimiento.
El escalado vertical puede ser limitado por las capacidades físicas de la instancia y puede requerir tiempo y recursos adicionales para llevar a cabo la mejora. Además, si la instancia falla, puede haber un impacto significativo en la disponibilidad del sistema hasta que se resuelva el problema.

El escalado automático en la computación en la nube es una funcionalidad clave que permite ajustar automáticamente la capacidad de los recursos de acuerdo con la demanda en tiempo real. Consiste en agregar o eliminar recursos de manera dinámica en función de métricas predefinidas, como la carga de trabajo, el rendimiento o la utilización de recursos. A continuación, se explicará algunos aspectos importantes del escalado automático en la computación en la nube:

- **Métricas de escalado:** Para realizar el escalado automático de manera efectiva, se definen métricas de escalado, como la utilización de la CPU, la memoria o el tráfico de red. Estas métricas se monitorean constantemente y se comparan con umbrales predefinidos. Cuando se supera un umbral superior, se inicia el proceso de escalado hacia arriba, y cuando se cae por debajo de un umbral inferior, se inicia el proceso de escalado hacia abajo.
- **Políticas de escalado:** Las políticas de escalado definen cómo se lleva a cabo el proceso de escalado automático. Estas políticas determinan, por ejemplo, el número de instancias a agregar o eliminar, el intervalo de tiempo entre los ajustes, la estrategia de distribución de carga, entre otros aspectos. Las políticas de escalado pueden ser configuradas según las necesidades específicas de la aplicación o el servicio en la nube.
- **Integración con el monitoreo:** El escalado automático está estrechamente relacionado con el monitoreo en la nube. Las métricas utilizadas para el escalado se obtienen a partir de datos de monitoreo en tiempo real, como la utilización de recursos, el rendimiento de las aplicaciones o la carga de trabajo. El monitoreo constante permite tomar decisiones de escalado basadas en datos actualizados y evitar situaciones de sobrecarga o subutilización de recursos.
- **Optimización de costos y rendimiento:** El escalado automático permite optimizar tanto los costos como el rendimiento de los recursos en la nube. Al agregar instancias cuando la demanda aumenta y eliminarlas cuando disminuye, se evita el desperdicio de recursos y se optimiza el consumo. Además, el escalado automático permite mantener un rendimiento óptimo incluso en situaciones de alta carga, asegurando una buena

Licencia Creative Commons BY-NC-SA.

experiencia para los usuarios finales.

> En resumen, el escalado automático es una funcionalidad esencial en la computación en la nube, ya que permite adaptar la capacidad de los recursos de forma dinámica y eficiente, asegurando un equilibrio entre el rendimiento, la disponibilidad y los costos. Proporciona la flexibilidad necesaria para enfrentar cambios en la demanda y garantizar que los recursos se ajusten de manera adecuada a las necesidades de las aplicaciones y servicios en la nube. Este proceso de escalado no se podría llevar a cabo sin el monitoreo en tiempo real del estado de las máquinas virtuales, contenedores y demás recursos en la nube.

Licencia Creative Commons BY-NC-SA.

5.- Orquestación de contenedores para DevOps

La orquestación de contenedores es una práctica esencial en DevOps que permite administrar eficientemente aplicaciones y asegurar su despliegue, escalado y funcionamiento adecuado.

1. **¿Qué es la orquestación de contenedores?:** La orquestación de contenedores se refiere a la administración automatizada de contenedores en un entorno de desarrollo, pruebas y/o producción. Implica la creación, el despliegue, la escalabilidad y la monitorización de contenedores en una aplicación, todo ello de forma coordinada.

2. **Herramientas de orquestación de contenedores más relevantes son:**
   Docker Swarm: Es una herramienta de orquestación nativa de Docker que permite administrar un clúster de contenedores. Proporciona funcionalidades para el escalado automático, la distribución de carga y la recuperación ante fallos.
   Kubernetes: Es una plataforma de código abierto ampliamente adoptada para la orquestación de contenedores. Permite administrar y coordinar automáticamente la implementación y el escalamiento de contenedores en clústeres.
   Apache Mesos: Es una plataforma de orquestación de contenedores que proporciona capacidades avanzadas de administración y programación de recursos. Permite ejecutar varios frameworks de aplicaciones, como Marathon y Chronos, para gestionar contenedores.

3. **Ventajas de la orquestación de contenedores:**
   Escalabilidad: Las herramientas de orquestación permiten escalar automáticamente los contenedores en función de la demanda, lo que asegura que la aplicación pueda manejar un aumento en el tráfico sin problemas.
   Alta disponibilidad: La orquestación de contenedores garantiza que la aplicación se mantenga en funcionamiento incluso en caso de fallos en los contenedores individuales.
   Gestión simplificada: Las herramientas de orquestación proporcionan una interfaz unificada para administrar y monitorear todos los contenedores en un clúster, lo que simplifica las tareas de gestión y solución de problemas.

Licencia Creative Commons BY-NC-SA.

5.1.- ¿Qué es DevOps?

En el mundo de la computación en la nube, **DevOps es una metodología que combina** las prácticas de **desarrollo** de software (**Dev**) y las **operaciones de TI** (**Ops**) para lograr una colaboración estrecha y una entrega continua de aplicaciones de alta calidad. El objetivo principal de DevOps es agilizar el ciclo de vida del desarrollo de software, mejorando la eficiencia, la calidad y la confiabilidad de los productos y servicios ofrecidos.

Algunas de las principales alternativas en el mercado para implementar DevOps son:

- **AWS CodeStar:** Es un servicio de AWS que permite a los equipos de desarrollo comenzar rápidamente con proyectos de software, proporcionando plantillas predefinidas, herramientas de colaboración y una integración sencilla con otras herramientas de AWS.
- **Azure DevOps:** Es una plataforma integral de Microsoft que proporciona un conjunto completo de herramientas para la colaboración, la entrega continua, la administración de proyectos y el monitoreo en el desarrollo de software. Incluye características como control de versiones, seguimiento de problemas, compilación y despliegue automatizado.
- **Google Cloud Build:** Es una herramienta de Google Cloud que permite la creación, prueba y despliegue de aplicaciones de manera rápida y confiable. Proporciona integración con otras herramientas de Google Cloud, como Kubernetes Engine, para orquestar el despliegue de contenedores.

DevOps y las metodologías ágiles de desarrollo, como Scrum y Kanban, están estrechamente relacionados y se complementan entre sí para acelerar la entrega de software y mejorar la calidad del producto final.

Las metodologías ágiles se centran en la colaboración, la flexibilidad y la entrega incremental de funcionalidades. Estas metodologías promueven equipos multidisciplinarios que trabajan en ciclos cortos de desarrollo, llamados sprints, donde se priorizan y entregan constantemente las características más valiosas para el cliente. El objetivo principal es obtener retroalimentación temprana y frecuente para iterar y mejorar continuamente el producto.

DevOps se alinea perfectamente con las metodologías ágiles al abordar la colaboración y la entrega continua de software. DevOps enfatiza la integración y la colaboración entre los equipos de desarrollo y operaciones, eliminando las barreras entre ellos. En lugar de tener departamentos separados donde los equipos de desarrollo crean software y luego los equipos de operaciones lo despliegan y mantienen, **DevOps fomenta un enfoque de equipo único y colaborativo**.

El despliegue continuo es una práctica clave en DevOps y es una extensión natural de las metodologías ágiles. Consiste en automatizar y agilizar el proceso de entrega de software, lo que permite lanzamientos frecuentes y confiables de nuevas funcionalidades y correcciones de errores. En lugar de realizar grandes despliegues

Licencia Creative Commons BY-NC-SA.

infrecuentes, el despliegue continuo permite realizar cambios pequeños y frecuentes, lo que minimiza los riesgos y los tiempos de inactividad.

> En resumen, DevOps es una metodología que promueve la colaboración y la entrega continua de software. **DevOps fomenta un enfoque de equipo único y colaborativo entre los desarrolladores y los técnicos de sistemas.** Las alternativas en el mercado, como AWS CodeStar, Azure DevOps y Google Cloud Build, brindan herramientas y servicios para implementar eficazmente DevOps en entornos de computación en la nube.

5.2.- Para qué necesitamos la orquestación de contenedores

La orquestación de contenedores es una tecnología que nos permite gestionar y coordinar eficientemente un conjunto de contenedores. Proporciona un conjunto de herramientas y servicios que facilitan la administración, escalabilidad y alta disponibilidad de las aplicaciones basadas en contenedores.

Algunos de los beneficios y usos principales de la orquestación de contenedores son:

- **Gestión y coordinación:** La orquestación permite gestionar de manera centralizada y coordinada todos los contenedores que forman parte de una aplicación. Esto incluye tareas como el despliegue, la configuración, la supervisión y la gestión de la comunicación entre los contenedores.
- **Escalabilidad automática:** Con la orquestación, podemos escalar automáticamente nuestros contenedores según la demanda de la aplicación. Esto significa que podemos aumentar o disminuir el número de contenedores en función del tráfico o la carga de trabajo, asegurando así un rendimiento óptimo.
- **Alta disponibilidad:** La orquestación garantiza la alta disponibilidad de nuestras aplicaciones al distribuir los contenedores en varios nodos o servidores. Si un nodo falla, los contenedores son automáticamente reubicados en otros nodos disponibles, evitando así la interrupción del servicio.
- **Balanceo de carga:** La orquestación de contenedores también ofrece funcionalidades de balanceo de carga. Distribuye el tráfico entrante de manera equilibrada entre los contenedores, asegurando una distribución uniforme de la carga y evitando que un contenedor se sobrecargue.

Licencia Creative Commons BY-NC-SA.

- **Gestión de la configuración:** La orquestación facilita la gestión de la configuración de los contenedores. Permite definir y mantener consistentes las variables de entorno, las configuraciones de red y otros parámetros relevantes para el correcto funcionamiento de la aplicación.
- **Despliegue declarativo:** La orquestación se basa en un enfoque declarativo, lo que significa que especificamos el estado deseado de nuestra aplicación en lugar de detallar los pasos específicos para llegar a ese estado. Esto simplifica el despliegue y la gestión de la aplicación, ya que la orquestación se encarga de implementar el estado declarado de manera automática.

> En resumen, la orquestación de contenedores es esencial para gestionar y coordinar eficientemente los contenedores en un entorno de computación en la nube. Proporciona características como la gestión y coordinación centralizada, la escalabilidad automática, la alta disponibilidad, el balanceo de carga, la gestión de la configuración y el despliegue declarativo. Estas características son fundamentales para asegurar un despliegue eficiente, flexible y confiable de las aplicaciones basadas en contenedores.

5.3.- Control de versiones para el trabajo en equipo y el despliegue continuo de aplicaciones

El control de versiones es una parte fundamental del trabajo en equipo y el desarrollo de aplicaciones. **Permite gestionar los cambios realizados en el código fuente de manera colaborativa, mantener un historial de modificaciones** y facilitar la integración de nuevas funcionalidades. En el contexto de la computación en la nube, el control de versiones cobra aún más importancia, ya que es necesario coordinar el desarrollo y despliegue de contenedores en un entorno distribuido.

Una herramienta popular para el control de versiones en la nube es GitLab. Permite almacenar y gestionar el código fuente de manera centralizada, brindando un entorno colaborativo para el trabajo en equipo.

Algunas de las características clave de GitLab son:

- **Repositorios Git:** GitLab utiliza el sistema de control de versiones distribuido Git, lo que permite un seguimiento preciso de los cambios realizados en el código y la colaboración

Licencia Creative Commons BY-NC-SA.

entre desarrolladores.

- **Ramas y fusiones:** GitLab permite crear ramas para desarrollar nuevas funcionalidades o solucionar problemas sin afectar la rama principal. Posteriormente, se pueden fusionar las ramas para incorporar los cambios al código base.
- **Integración continua y despliegue continuo:** GitLab ofrece herramientas para implementar procesos de integración continua y despliegue continuo (**CI/CD**), lo que permite automatizar las pruebas y el despliegue de los contenedores a medida que se van realizando cambios en el código.

Por ejemplo, supongamos que tenemos un equipo de desarrollo que trabaja en una aplicación web basada en contenedores. Utilizando GitLab, cada desarrollador puede crear su propia rama para trabajar en nuevas características o solucionar errores. Una vez que los cambios se consideren listos, se fusionan en la rama principal (o también llamada master).

> **GitLab** también puede integrarse con herramientas de automatización, como Docker, para construir y desplegar automáticamente los contenedores actualizados en un entorno de producción. Esto permite un despliegue continuo de la aplicación, asegurando que las últimas versiones del código estén disponibles para los usuarios finales de manera eficiente y confiable.

Licencia Creative Commons BY-NC-SA.

6.- Licencia y autoría de este material

Materiales desarrollados inicialmente por **Daniel Cano Verdú (2024)** profesor de FP de la Junta de Andalucía y actualizados por el profesorado de la Junta de Andalucía bajo licencia **Creative Commons BY-NC-SA**.

Antes de cualquier uso leer detenidamente el siguiente Aviso legal