MODULO INTRODUCCION

**QUE ES CLOUD?**

Es una tecnología que nos permite el acceso a una red de manera remota, según convenga y bajo demanda, a un conjunto de recursos informáticos que pueden ser rápidamente liberados con un mínimo de esfuerzo o interacción del proveedor de servicios.

Es **descentralizado** (todos los servicios que queramos llevar a producción lo podemos hacer desde cualquier parte del mundo).

**Se crean de una manera muy rápida**. Podemos crear y deployar casi inmediatamente

**Pago por uso**.

* Conceptos
  + Multidispositivos
  + Escalable
  + Seguro
  + Accesible
  + Rentable
  + Ágil
  + Eficiente
* Tecnologías
  + Que tecnologías existen
    - Almacenamiento, creación y edición de archivos por varios usuarios
    - Compartir archivos en diferentes formatos
    - Hosting web
    - Gestión de proyectos
    - CRM (Gestión de relaciones con clientes. Como Salesforce)
    - Soporte a BPM (Atención a cliente)
    - Gestión integral / ERP
    - Business proyect management
    - Email marketing / gestión de redes sociales
    - Comunicaciones
  + Que modelos existes
    - **SaaS**: Software como servicio. Contratar el software completo (caso de Gmail)
    - **PaaS**: Plataforma como servicio. Contratar infraestructura + entorno de desarrollo de software
    - **IaaS**: Infraestructura como servicio. Contratar infraestructura en la nube
  + Que nubes existen
    - Privada -> Implementar nuestro servicio en un servidor local.

Tiene una infraestructura operada de forma aislada por una sola organización, mayor control sobre los recursos, aloja de forma interna o externa.

* + - Publica -> Implementar nuestro servicio en un servidor externo (el caso de Azure, AWS y GCP).

Tiene una elevada capacidad de procesamiento y almacenamiento, **la infraestructura es de propiedad del proveedor**.

* + - Hibrida -> Tiene algunos recursos administrados de manera privada y otros de manera pública. Son entidades separadas que se mantienen unidas a través de tecnologías estandarizadas permitiendo trasmitir datos y aplicaciones entre sí. **Permite que los equipos de desarrollo y operaciones trabajen juntos en un modelo DevOps**.
    - Comunitaria -> Parecido a google drive. Podemos trabajar y usar un servicio de manera gratuita.
  + Que proveedores nos pueden proporcionar diferentes servicios ( los principales son AWS, Azure, GCP)

*AWS* (<https://aws.amazon.com/es/console/>): Tiene más de 200 servicios integrales. Ofrece servicios robustos y para Big Data. Permite ejecutar virtualmente cualquier aplicación en la nube.

* **IoT** **Core** (Internet of Things)
  + Proporciona una comunicación bidireccional segura entre dispositivos conectados
  + Permite recopilar datos de telemetría de los dispositivos
  + Almacena y analiza datos
* **S3**
  + Servicio de almacenamientos de objetos
  + Ofrece escalabilidad, disponibilidad de datos, seguridad y rendimiento
  + Permite almacenar y proteger cualquier cantidad de datos
  + Back-it, un contenedor de estos objetos con diferentes políticas de acceso
* **DynamoDB**
  + Base de datos NoSQL orientada a documentos
  + Ofrece escalabilidad de datos
  + Puede gestionar más de 10B de soluciones por día
  + Puede admitir picos de más de 20M de soluciones por segundo.
* **Lambda** 
  + Herramienta Serverless (sin servidor), AWS se encarga de darnos toda la infraestructura para el servicio que queremos alojar
  + Permite ejecutar código sin aprovisionar ni administrar servidores
  + Aplicable a casi cualquier tipo de aplicación o servicio back-end sin tener que realizar tareas de administración
* **ERM** 
  + Servicio de clúster orientado a Big Data. Procesamiento de datos.
  + Permite trabajar en un clúster de Hadoop administrado
  + Simplifica la ejecución de los marcos de trabajo de Big Data

*Azure* (<https://portal.azure.com/>): Tiene servicios más orientados a Stream Analytics, PowerBI (explotación de datos), almacenamiento, y base de datos.

*GCP* (<https://console.cloud.google.com/>): Provee recursos de computación en la nube que se utilizan para desarrollar, implementar y operar aplicaciones web. Asi como máquinas virtuales, alojamiento de páginas web, almacenamiento de archivos.

Compute, storage, machine learning, big data, y otros son los servicios que ofrece.

* Conceptos técnicos
  + Cliente-servidor
  + Modelos de costos y servicios de Cloud
  + Que son los SLA
* Calculadora de precios

Una herramienta para calcular los costos estimados por hora o por mes por utilizar una cloud. Son valores estimativos sobre la configuración que le damos y los recursos que elegimos.

**QUE ES DEVOPS?**

* De qué trata este framework

DevOps es una palabra compuesta conformada por development y operations. Es un framework de trabajo. Es una práctica de ingeniería de software que tiene como objetivo unificar el desarrollo de software (DEV) y la operación del software en la infraestructura que la contiene (OPS), aunque también en esta práctica entra el concepto de seguridad (SEC).

* Pilares
  + Estabilidad
  + Fiabilidad
  + Disponibilidad
  + Seguridad
* Principios de DevOps
  + **Flujo** -> integración continua, entrega continua, despliegue continuo
  + **Retroalimentación** -> medir el negocio, la infraestructura, el rendimiento y seguridad
  + **Mejora continua** -> incentivar la curiosidad y aprendizaje, promover experimentación e incentivar autocritica
* ¿Porque es importante entenderlo?

Generalmente, dentro de las organizaciones las áreas IT están desconectadas entre sí. Esto afecta directamente sobre la experiencia del usuario, no aporta valor y genera conflictos (integración, comunicación, etc.).

DevOps busca incorporar un puente que alinee los objetivos y estrategias del sector Dev y el sector Ops.

Con esta cultura mejoramos la calidad de software, que sea usable y se pueda mantener, y que se pueda entregar a tiempo.

* Referencias
  + <https://roadmap.sh/devops>
* Herramientas (<https://digital.ai/learn/devops-periodic-table/>)

**FUNDAMENTOS DE SISTEMAS OPERTATIVOS**

**Sistema operativo**

Networking

SSH

Shell scripting

Vim 🡪 el editor de texto (IDE)

Filesystem (Linux)

Gestiona los recursos de hardware y provee servicios a los programas de aplicación de software. Les proporciona a los programadores un conjunto abstracto de recursos simples.

Modo kernel: acceso a todo el hardware

Modo usuario: acceso a un subconjunto de instrucciones de maquina

Todo lo que es entrada y salida pasaría por el kernel.

Shell y GUI es el nivel más bajo del software y se caracteriza porque permite la ejecución de otras aplicaciones.

**KERNEL**

Kernel monolítico: un solo programa grande y compilado

Microkernel: componentes principales que se ejecutan en modo kernel y después tenemos componentes secundarios que se ejecutan en modo usuario

**DOCKER**

* Que es Docker

Es una herramienta que automatiza el despliegue de aplicaciones dentro de **contenedores** de software, proporcionando una capa de **abstracción** y **automatización** de visualización de aplicaciones en múltiples sistemas operativos.

Abstracción 🡪 permite **desplegar aplicaciones** en entornos aislados e independientes del hardware en que corren (implica que la misma imagen, receta que corro en un cloud pueda correr en cualquier cloud, en mi escritorio o en mi servidor).

Con docker engine administramos los contenedores, y docker CLI (interfaz de línea de comandos de docker) es la aplicación que usamos para controlar docker engine.

* Virtualización

La virtualización era un paradigma de tener un hardware que sirva para varias cosas (multiplexa). Plantea que por encima del Kernel apareciera un nuevo software, llamado hipervisor. Lo que permitía engañar simulando en totalidad de un equipo (una máquina virtual). Era demasiado pesado hasta que llegaron los containers.

* Cómo funciona

Docker usa containers, una unidad mínima de software del sistema operativo que hace lo mínimo y necesario para correr una aplicación. Con ello podemos virtualizar la aplicación en vez de toda la máquina.

**Imagen en docker** 🡪 es un archivo que se encuentra compuesto de diversas capas y se utiliza con el objetivo de ejecutar un código dentro de un contenedor Docker. Este archivo contiene todo el sistema inicial de ficheros en los que se va a basar el contenedor para su funcionamiento, como el entrypoint. **No se puede construir un contenedor Docker sin Imágenes**.

Porque se usa tanto

Comandos básicos (Docker CLI)

Docker en un ambiente realista

Docker registry

Persistencia en container (volúmenes)

Deploy de una aplicación

V3 00.06.17