**Amazon Web Services**

**En AWS el cómputo en la nube trata de los siguiente:**

* Sitios Web, una sola computadora corriendo un solo stack de programación cómo LAMP, XAMPP, entre otros.
* Respaldos y recuperación, incluso de sistemas operativos completos.
* Archivos Permanentes, también puedes guardar archivos estáticos como fotografías o documentos.
* DevOps, no solo tenemos automatización en el release de los proyectos, también cuentas con alta disponibilidad o respaldos automatizados en diversos lugares del mundo.
* Análisis Masivos
* Cómputo Serverless, en lugar de preocuparte por la cantidad de computadoras o cómo y cuándo va a escalar tu servicio puedes programar tu aplicación con microservicios mientras que AWS se encarga de darte los elementos necesarios.
* Cómputo de Alto Rendimiento, levanta tus servidores sólo cuando tu aplicación lo necesita.
* Internet of Things.
* Aplicaciones Empresariales.
* Distribución de media.
* Servicios móviles.
* Cómputo científico.
* E-commerce.
* Ambientes Híbridos.
* Blockchain.

**Introducción a la oferta de servicios de AWS y sus aplicaciones**

En la siguiente lista se encuentra la oferta de servicios con algunas de sus aplicaciones más importantes o que nos pueden servir:

**Computo:** Máquinas virtuales, serverless, infraestructura, etc.

**Storage:** Administrar archivos.

**Bases de Datos:** Opciones como Postgrade, MySQL, etc.

**Migración de Servicios:** Si ya tienes datos en una data center, te ayuda a poder migrarlos.

**Networking & Content Delivery:** Registrar dominios, administrar certificados SSL y más.

**Herramientas de Desarrollador:** Para facilitar la colaboración entre desarrolladores como xcode, commits, etc.

**Herramientas de Administración:** (CloudWatch: Permite ver que está sucediendo en tu infraestructura o en tus servidores. CloudTrail: Permite llevar una auditoria accesos y permisos y saber qué es lo que hacen).

**Media Services:** Como Elastic Transcoder que sirve si por ejemplo estás haciendo streaming de un partido podrías subir el dato que estas generando como vídeo y ellos generaran todas las versiones que necesitas para diferentes tipos de dispositivos ya sean celulares, computadoras, etc.

**Machine Learning**: Como Rekognition que puedes ir enviando imágenes y te puede reconocer que estás viendo.

**Analytics:** Cuanta RAM está usando cada visitante, cuanto CPU, que actividad está teniendo ese visitante en tu sitio.

Seguridad: Como IAM que sirve para manejar a que tienen acceso la cuenta de los colaboradores.

**GuardDuty:** Te permite hacer un recuento de los diferentes accesos de red que ha habido, entonces si hubiese un ataque a un servidor te va a permitir saber de dónde viene el ataque y te va dar opciones].

**Servicios Móviles:** Te permite generar tus propios servicios móviles de forma más sencilla.

**AR & VR:** Ya viene con el software instalado y puedes empezar a hacer identificación de algún patrón para empezar a enseñar alguna cosa en tu aplicación.

**Integración de Aplicaciones:** Como Simple Notification Service: Se puede hacer llamado a diferentes servicios, ejemplo: Enviar notificaciones a través de correo electrónico, SMS.

**Customer Engagement:** Sirve más para uso empresarial, como correo electrónico, cuentas de Amazon, etc.

**Bussiness Productivity:** Permite usar Alexa para negocios, cargar documentos al correo empresarial, etc.

**Desktop & App Streaming:** Como WorkSpaces que sirve si en algún momento necesitas compartirle a alguien alguna máquina virtual de cierta manera, que tenga un escritorio como Windows y que esa máquina tenga un navegador y que solo se pueda conectar al sitio de la empresa, escuela, etc.

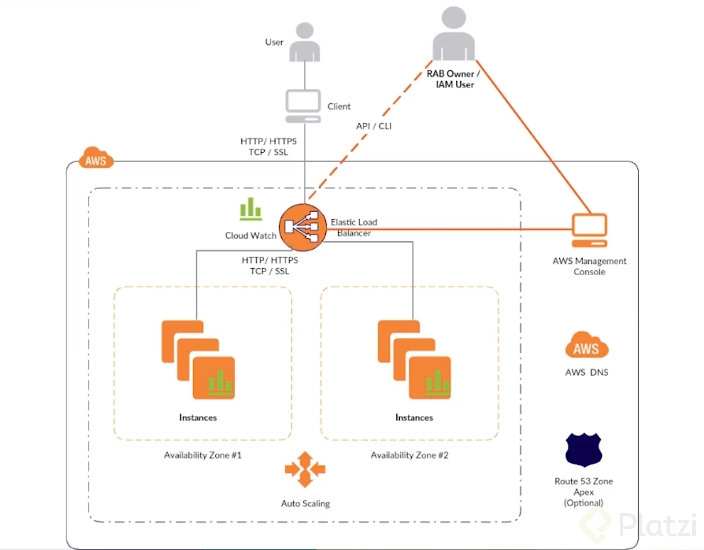
**Internet of Things:** Puede proveer de hardware y te permite crear servicios para que fácilmente puedas administrarlos, como conectarlos con datos o aplicaciones web.

**Game Development:** (Motor de juegos) Puedes generar un juego con el software que te proporcionan. Crear escenas, interacción.

**Conceptos Importantes**

**AWS Elastic Beanstalk:** es un servicio de ASW que se utiliza para implementar y escalar servicios y aplicaciones web desarrolladas en JAVA, .NET, PHP, Node.js, Python, Go, entre otros, en servidores familiares como Apache y Nginx. Además, se encargará de administrar de manera automática la implementación, el aprovisionamiento de la capacidad, el equilibrio de la carga y el escalado de nuestra aplicación.

**CloudWatch:** es un servicio de monitoreo y administración creada para desarrolladores, operadores de sistemas, entre otros. El servicio ofrece datos e información procesable para monitorear las aplicaciones, comprender cambios de rendimiento que afectan al sistema, optimizar recursos y lograr una vida unificada del estado de las operaciones.



**Que es EC2**

EC2 elimina la necesidad de invertir inicialmente en hardware, de manera que puede desarrollar e implementar aplicaciones en menos tiempo. Puede usar Amazon EC2 para lanzar tantos servidores virtuales como necesite, configurar la seguridad y las redes y administrar el almacenamiento. Amazon EC2 le permite escalar hacia arriba o hacia abajo para controlar cambios en los requisitos o picos de popularidad, con lo que se reduce la necesidad de prever el tráfico.

"EC2 son un conjunto de máquinas virtuales en línea que puedes utilizar para desarrollo, calidad o producción. Estas son algunas de sus características:

Instancias: Máquinas virtuales con diversas opciones de Sistema Operativo, vCPU, RAM, Disco Duro, etc.

Seguridad: Generación de llaves únicas para poder conectarse a tu máquina Linux o Windows de forma segura.

Espacio: Diversas opciones de espacio en disco duro, virtualmente infinito.

Redundancia: Puedes tener diversas copas de la misma máquina en diversas regiones geográficas.

Firewall:Puedes controlar de manera muy fina desde donde te puedes conectar a la máquina y por qué puertos.

Direcciones IP estáticas: Puedes optar por comprar una IP pública estática para que siempre puedas poner la última versión o máquina en esa IP.

Respaldos: Puedes respaldar toda la máquina (Ambiente, Sistema operativo, todo) cada que quieras.

Escalable: En caso necesario, puedes incrementar o decrementar los recursos de la máquina: más vCPUs, más RAM, etc.

Migración de snapshot: Puedes copiar un snapshot a otras regiones, en caso de que cualquier cosa suceda en la que estas.

**Pasos para crear una instancia EC2:**

1. Entrar al servicio EC2

2. Pulsar en Launch Instance

3. Seleccionar el sistema operativo. Si solo queremos usar las gratuitas hay que marcar el checkbox "Free tier only". Ejemplo: Amazon Linux 2 AMI (HVM), SSD Volume Type

4. Seleccionar la versión. En el ejemplo la única gratuita.

5. Clic en Next: Configure Instance Details

6. Configurar los parámetros de la instancia. En el ejemplo se deja por defecto.

7. Clic en Next: Add Storage.

8. Configurar el almacenamiento. En el ejemplo capacidad: 8GB y Volume: General Purpose SSD

9. Clic en Next: Add Tags

10. Añadimos el tag para reconocer la máquina. En el ejemplo Key = Name y EC2-Platzi-test

11. Clic en Next: Configure Security Group.

12. Crear un nuevo grupo. Pasos ejemplo:

a. Rellenar Security group name: acceso-solo-ssh

b. En Source seleccionar en la lista Anywhere.

13. Clic en Review and Launch

14. Revisar la configuración y clic en Launch

15. En el mensaje emergente tenemos que seleccionar con que llave podremos conectarnos. En el ejemplo creamos una nueva. Pasos para crear nueva llave:

a. Seleccionar Create a new key pair.

b. Rellenar el nombre que tendrá la llave.

c. Clic en Download Key Pair. Se descarga en nuestro equipo la llave generada. Si da error pueden probar a modificar el nombre de la llave. El archivo es un .pem

d. Clic en Launch Instances

**Responsabilidades a la hora de trabajar con una máquina virtual con EC2:**

**Updates:** Con las instancias, nosotros somos totalmente responsables de la actualización de OS

**Respaldos:** Nuestra instancia no se respaldará sola, tendremos que hacerlo nosotros.

**Restauración snapshot:** Podemos hacer respaldos antes de hacer grandes cambios para poder hacer rollback del Sistema en caso necesario.

**Que es Lambda y Serverless**

Lambda es un proyecto de AWS muy relacionado con el concepto de Serverless, dejar la administración de tus servidores en manos de Amazon para solo encargarte de las funciones de código que ejecutara tu aplicación.

¿Qué son?

Imagina lambda como un lugar donde puedes ejecutar funciones de tu código.

Serverless

No existe un servidor como vimos en EC2, es decir, solo está el código en lamba y AWS se encarga de ejecutarlo cuando necesites.

Lenguajes soportados

Puedes programar funciones lamba en Nodejs (JavaScript), Python, Java (8), C# (.Net Core) y Go.

Recuerda tener en cuenta los siguientes puntos:

**Memoria:** Mínima de 128MB, máxima 3000MB con incrementos de 64MB.

Límites de ejecución y espacio: Puedes correr tu aplicación hasta 300 segundos y tienes un /tmp limitado a 512MB.

**Ejecución paralela:** Esta limitada a 1000 ejecuciones concurrentes (a un mismo tiempo), no tiene límite en ejecuciones secuenciales (una detrás de otra).

Ventajas de Lambda:

**Seguridad:** Al ser una infraestructura compartida, no tienes que preocuparte de seguridad: AWS maneja todo.

**Performance:** AWS está monitoreando constantemente la ejecución de tus funciones y se encarga de que siempre tenga el mejor performance.

**Código aislado:** Tu código, aún estando en una infraestructura compartida, corre en un ambiente virtual exclusivo, aislado de las demás ejecuciones lamba.

Recuerda que AWS te regala 1 millón de peticiones lamba gratis el primer año.

**Conociendo Elastic Beanstalk**

Elastic Beanstalk es una plataforma donde en pocos pasos, obtienes un balanceador de cargas y tantas instancias EC2 como tú quieras.

Este ambiente puede escalar de manera dinámica de acuerdo con el tiempo de respuesta a los usuarios, uso de CPU, uso de RAM, etc.

Elastic Beanstalk lo que hace es el trabajo de “la nube elástica”, es decir, cuando detecta que necesita más recursos es cuando empieza el solito a escalar, está genial porque ya no solamente vamos a tener una única instancia EC2 corriendo, sino que ya tendremos la capacidad de aumentar sus capacidades, esto es algo que siempre me causaba curiosidad, quiero verlo!

Esta herramienta soporta los siguientes ambientes:

Docker Image, Go, Java SE, Java con Tomcat, .NET + Windows Server + IIS, Nodejs, PHP, Python, Ruby

**Almacenamiento S3**

AWS permite guardar archivos en su plataforma, de manera que las instancias EC2, Lambda, etc. son efímeras y pueden ser eliminadas sin ninguna preocupación.

**AWS S3:** S3 es un repositorio de archivos rápido, perfecto para uso de una aplicación a la hora de crear, manipular y almacenar datos.

**Respaldos:** S3 permite hacer respaldos en tiempo prácticamente real en otras regiones de AWS.

**Glacier:** Es un servicio de almacenamiento en la nube para archivar datos y realizar copias de seguridad a largo plazo.

Glacier es un servicio que nos ofrece AWS para el almacenamiento de todo archivos (archivos planos, imágenes, audio, etc.). El cual hace mucha referencia al tema de archivador. Cuando uno guarda información que no es de uso frecuente, lo mandas a un archivador histórico.

Ejemplos:

Facturación de más de 2 años atras

Documentos de proyectos de 2 años atras

Hojas de vida, de personas que han trabajado desde el inicio de la empresa.

Consejo: Te recomiendo que guardes información de mas de 1 año o 2 años. De uso no frecuente.

**Subiendo contenido a S3**

* Desde la consola buscar S3
* Crear bucket con nombres que sean accesibles por un sitio web
* Se suben los archivos o carpeta
* Establecer permisos para el bucket o para cada archivo. (Si se va aponer una página estática, se pone el bucket como público)

Para ello vamos a dashboard de buckets de S3 -> luego ingresamos el bucket -> entramos a permissions -> buscamos Block public access y damos Edit -> quitamos la selección a Block all public access -> save changes.

* Nuevamente entramos entramos a permissions -> buscamos Bucket policy -> damos Edit y pegamos lo siguiente



* Save changes
* Podemos tener sitios estáticos buscando en la sección de properties -> Static website hosting -> Edit -> Enable -> poner nombre del archivo de entrada (index.html) -> sabe
* Luego de que cargue se puede volver a properties -> Static website hosting y aca tendremos la dirección del sitio para acceder

**Bases de Datos – RDS**

AWS tiene un producto llamado RDS que optimiza el funcionamiento de un motor de bases de datos. Este servicio incluye mantenimiento a la base de datos, respaldos diarios, optimización dependiendo del tipo de uso, etc.

RDS tiene varias opciones de motores de bases de datos, como: Aurora PG, Aurora MySQL, MySQL, MariaDB, PostgreSQL, Oracle y Microsoft SQL Server.

Nota: Los motores de bases de datos como Oracle y Microsoft SQL Server, han sido considerados por mucho tiempo como motores empresariales o tipo Enterprise debido a que ofrecen unas prestaciones interesantes como Backups fáciles, redundancia, tolerancia a fallos, etc. AWS a través de RDS, ofrece estas prestaciones y más; por lo que ya no es necesario tener estos factores en cuenta a la hora de escoger el motor de base de datos para un nuevo proyecto.

**RDS PostgreSQL**

AWS implementa el motor de PostgreSQL (RDS PG) en una instancia optimizada para correr con la máxima eficacia.

RDS PG incluye, por omisión, tareas de optimización como vacuum, recuperación de espacio en el disco duro y planificación de queries.

También te permite hacer respaldos diarios (o incluso más seguido) de tu base de datos.

Otras ventajas de RDS PG son:

Cifrado a tu elección, tu base de datos puede estar cifrada en disco duro

Migración asistida: RDS PG tiene mecanismos que te ayudan a migrar tu información en caso de que tú ya cuentes con una base de datos con otro proveedor.

Alta disponibilidad: RDS PG te permite fácilmente configurar un ambiente de alta disponibilidad al ofrecerte diversas zonas para tu base de datos.

Recuerda que Amazon RDS provee de seguridad por omisión tan alta que no podrás conectarte a tu DB hasta que explícitamente lo permitas.