**AWS y tu rol como Desarrollador de Software**

La Computación en la Nube marca una nueva era en la evolución del desarrollo de software; cada vez las aplicaciones informáticas dependen menos de la infraestructura donde se ejecutarán, permitiendo de esta manera a los programadores centrarse en el “core” de su negocio (desarrollar el software) y dejando a terceros la responsabilidad de proveer los recursos suficientes para que las aplicaciones puedan ejecutarse eficientemente. El Auto-escalamiento, característica proporcionada por los proveedores de plataformas Cloud, le ofrece al desarrollador la tranquilidad de saber que su aplicación continuará corriendo correctamente sin importar el número de usuarios que accedan a ella; mientras que el Pago por Consumo ofrece un mecanismo de cobro que permite a cualquier desarrollador dar vida a sus proyectos sin incurrir en altos costos de infraestructura, pudiendo cada vez competir con mayor facilidad en los mercados globales.

AWS Amazon Web Services proporciona una amplia variedad de servicios que permiten a los desarrolladores de software obtener el máximo beneficio de una plataforma en la nube que brinde soporte a sus aplicaciones. El objetivo de este documento es ofrecer a mis colegas programadores un panorama general sobre algunos de los principales servicios que pueden ser de utilidad para el desarrollo de sus proyectos con soporte en la Nube. Si quieres conocer más sobre los diferentes servicios de AWS aquí <https://youtu.be/lTyqzyk86f8>

**API Gateway**

Como desarrolladores de software requerimos de una interfaz de comunicación que permita la interacción entre la aplicación cliente y los diferentes servicios proporcionados por AWS. Esta interfaz es AWS API Gateway.

“API Gateway actúa como una "puerta principal" para que las aplicaciones accedan a datos, lógica empresarial o funcionalidad desde sus servicios backend, como cargas de trabajo que se ejecutan en Amazon Elastic Compute Cloud (Amazon EC2), código que se ejecuta en AWS Lambda, cualquier aplicación web o aplicaciones de comunicación en tiempo real” <https://docs.aws.amazon.com/es_es/apigateway/latest/developerguide/welcome.html>

A través de este servicio podemos crear nuestras API REST de manera simple y asociarlas a otros servicios para orquestar toda una serie de funcionalidades en la Nube. La figura 1 nos muestra la arquitectura de API Gateway.


                Diagrama de arquitectura de API Gateway
            

Figura 1 Arquitectura API Gateway de AWS. tomado de <https://docs.aws.amazon.com/es_es/apigateway/latest/developerguide/welcome.html>

En el siguiente link puede acceder a una serie de tutoriales oficiales de AWS donde puede realizar sus primeros pasos con el servicio API Gateway <https://docs.aws.amazon.com/es_es/apigateway/latest/developerguide/api-gateway-tutorials.html>

Bien, ahora que ya tenemos un servicio que nos proporciona una puerta de enlace para comunicar a nuestras aplicaciones clientes con la nube a través de AWS, podemos comenzar a explorar otros servicios que como desarrolladores de software pueden ser de gran utilidad.

Sin lugar a duda, el almacenamiento persistente con disponibilidad de acceso concurrente es uno de los servicios más utilizados por los desarrolladores de software. AWS nos ofrece un amplio conjunto de servicios relacionados con el almacenamiento y bases de datos. Mi objetivo es presentarte los 3 servicios que considero más relevantes.

**Amazon S3**

El nombre S3 proviene de **S**imple **S**torage **S**ervice. Amazon S3 es un servicio de almacenamiento de datos ideal para realizar operaciones con archivos como objetos y que ofrece una alta disponibilidad para su acceso a través de Internet. Con SS3 puedes almacenar imágenes, videos, documentos y cualquier otro tipo de recurso para su posterior acceso, con un nivel de seguridad extremadamente alto debido al sistema de replicación ofrecido por AWS y con un alto desempeño.

El almacenamiento de objetos en S3 se realiza en contenedores algo similar a las carpetas en un drive conocidos como Buckets. Una vez cargados nuestros archivos en S3, estos pueden ser accedidos por nuestras aplicaciones, les podemos programar eventos, permitir su descarga por pago y otras series de propiedades para lograr la funcionalidad deseada. El siguiente video te explica cómo iniciar el servicio S3 mediante la creación de un bucket y la carga de un archivo <https://youtu.be/L3dYocCSU-E>

**Amazon RDS**

Amazon **R**elational **D**atabase **S**ervice nos permite escalar una base de datos relacional MySql, SQLServer, Oracle, entre otras a la nube, proporcionando un eficiente costo por uso. Este servicio es conocido como DBaaS Database as a Service. De esta manera, como desarrolladores estaremos descargando una serie de responsabilidades relacionadas con la gestión y administración de bases de datos a nuestro proveedor de servicios cloud AWS. Entre estas responsabilidades se encuentran (1) el auto-escalamiento de recursos, 2) el manejo de backups, 3) el manejo de roles y privilegios, 4) versatilidad en el uso de diferentes gestores de bases de datos.

RDS genera una instancia en la nube que representa la base de datos. Esta instancia puede ser accedida a través de los mismos gestores de bases de datos que ya usamos para las aplicaciones standalone. Amazon RDS soporta actualmente los motores de bases de datos MySQL, MariaDB, PostgreSQL, Oracle, y Microsoft SQL Server. Las características de seguridad, disponibilidad y eficiencia hacen de RDS una muy buena opción para migrar nuestras bases de datos relacionales a la nube.

La creación de una instancia y conexión con la base de datos en RDS es ligeramente diferente para cada motor de bases de datos. En el siguiente enlace se encuentran los tutoriales oficiales de AWS para realizar este proceso de acuerdo con el respectivo motor de bases de datos. <https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_GettingStarted.html>

El siguiente enlace nos ofrece un ejemplo de la conexión de MySQL con una instancia en Amazon RDS <https://youtu.be/Ng_zi11N4_c>

**Amazon DynamoDB**

DynamoDB es una base de datos NoSQL que proporciona un excelente desempeño y velocidad además de todas las ventajas de ser 100% orientada a la nube. Al ser una base de datos NoSQL, DynamoDB posee una gran capacidad de almacenamiento y respuesta rápida a las solicitudes realizadas desde múltiples terminales conectadas a la Web. La seguridad, disponibilidad, fiabilidad y eficiencia son los principales atributos de DynamoDB. DynamoDB trabaja con tablas, ítems y atributos; una tabla es una colección de ítems donde cada item es a su vez una colección de atributos. El almacenamiento de información utiliza formato JSON. La figura 2 muestra un ejemplo de una tabla con sus ítems y atributos



Figura 2 almacenamiento en DynamoDB

Tomado de <https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/HowItWorks.CoreComponents.html>

Uno de los elementos más relevantes proporcionados por DynamoDB para los desarrolladores de software son los DynamoDB Streams. Un stream puede ser asociado a una tabla para que genere registros cuando ocurra algún evento en la tabla como la adición de un nuevo item, su actualización o eliminación. Estos streams pueden asociarse con bloques de código para realizar determinadas funciones al momento de ocurrir un evento.

El siguiente video proporciona una introducción a DynamoDB y un ejemplo básico de creación de una base de datos y una tabla <https://youtu.be/2mVR_Qgx_RU>

Después de tener claro el mecanismo de comunicación o puerta de enlace entre tu aplicación cliente y los servicios de AWS en la nube a través de API Gateway, y de conocer las principales opciones de almacenamiento en la nube basadas en archivos S3, base de datos relacionales RDS y bases de datos NoSQL DynamoDB, solo nos hace falta un elemento clave para todo desarrollador de software y es la posibilidad de ejecutar funciones en la nube. Este concepto se conoce como FaaS (Function as a Service) o funciones como servicio. A través de las FaaS, tenemos la posibilidad de producir la lógica necesaria en la nube para acceder a los servicios y proporcionar las funcionalidades requeridas sin preocuparnos por la infraestructura requerida para dicho procesamiento.

**AWS Lambda**

AWS Lambda es el servicio que nos permite proporcionarle a nuestras aplicaciones en la nube con AWS esa capacidad de cómputo para la ejecución de rutinas de código que provean la funcionalidad esperada a partir de la integración de diferentes servicios. Los desarrolladores de software tienen la capacidad de ser altamente productivos en la nube gracias a AWS Lambda, pues además de proporcionar una interfaz para la escritura de código, se tiene la ventaja de no requerir el aprendizaje de un nuevo lenguaje de programación, sino de hacerlo usando el lenguaje de programación de su preferencia.

Con AWS Lambda, el desarrollador obtiene todas las ventajas de un código que se ejecuta en la nube mientras mantiene control del mismo en todo momento. Por ejemplo, usted puede utilizar AWS Lambda para ejecutar el código en respuesta a eventos, como cambios en los datos de un bucket de Amazon S3 o una tabla de Amazon DynamoDB; para ejecutar el código en respuesta a solicitudes HTTP mediante Amazon API Gateway. En otras palabras, AWS Lambda permite al desarrollador sacar el máximo provecho de todos los servicios proporcionados por AWS a un bajo costo y alto desempeño. El siguiente es el tutorial oficial de AWS sobre el desarrollo de funciones Lambda <https://docs.aws.amazon.com/lambda/latest/dg/getting-started.html>

La figura 3 representa la integración de los tres elementos principales que como desarrolladores de software debemos conocer de AWS. La puerta de enlace (API Gateway), el almacenamiento persistente (RDS y DynamoDB) y la capa lógica que genera la integración de servicios y la funcionalidad deseada (Lambda).

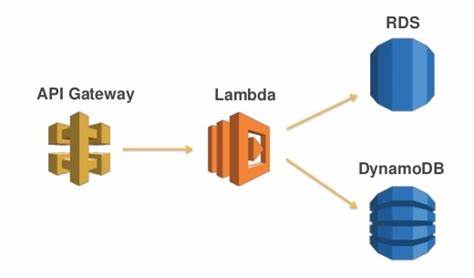


Figura 3 arquitectura de una solución integrando API Gateway, Lambda y Almacenamiento de Datos

Tomada de <https://s3.amazonaws.com/kinlane-productions/amazon/api-gateway/aws-rds-lambda-api-gateway.jpg>