Facundo Paez
facundo.paez@incluit.com

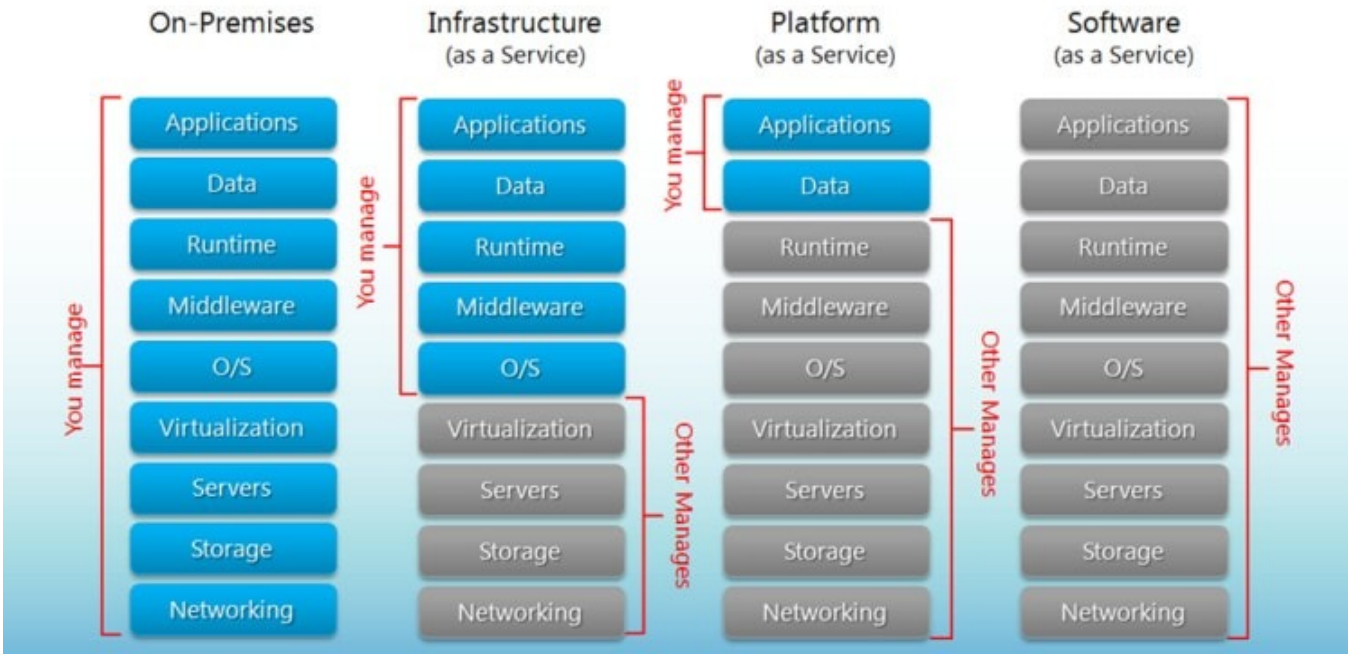
índice

Capítulo 1

Cloud Computing

De una manera simple, la computación en la nube (cloud computing) es una tecnología que permite acceso remoto a softwares, almacenamiento de archivos y procesamiento de datos por medio de Internet, siendo así, una alternativa a la ejecución en una computadora personal o servidor local. En el modelo de nube, no hay necesidad de instalar aplicaciones localmente en computadoras.
La computación en la nube ofrece a los individuos y a las empresas la capacidad de un pool de recursos de computación con buen mantenimiento, seguro, de fácil acceso y bajo demanda.

Modelos de la computación en la nube



Infraestructura como servicio (IaaS): el proveedor de servicios en la nube otorga a su cliente la capacidad para aprovecharse del procesamiento, almacenamiento, redes y otros recursos de computación fundamentales en base a los cuales pueda desplegar el software de su elección, incluyendo aplicaciones y sistemas operativos. Si bien la empresa consumidora no tiene control sobre cuestiones relacionadas con la infraestructura en la nube, en algunos casos sí podría ceder algunos derechos de control limitado sobre componentes de red seleccionados como, por ejemplo, algunos relativos a la seguridad.

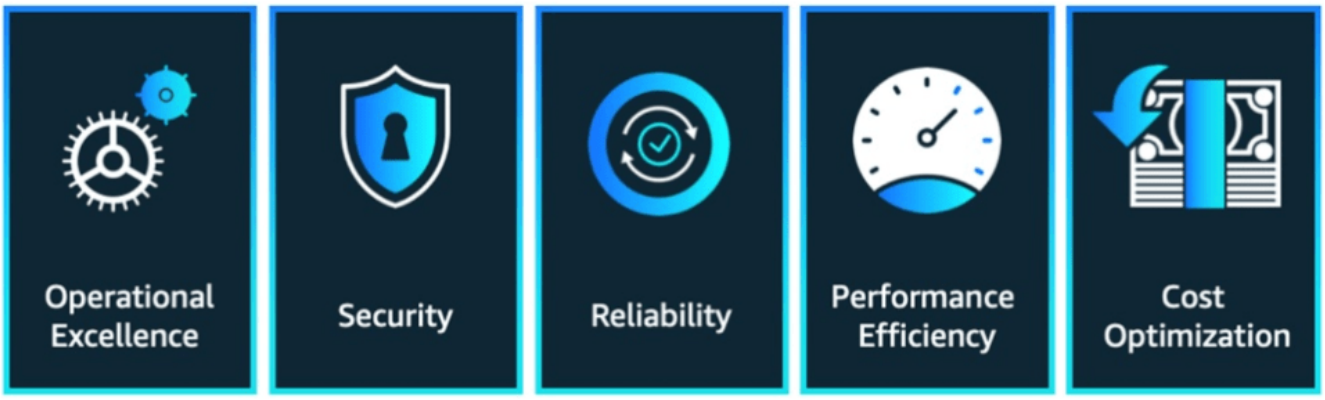
Plataforma como servicio (PaaS): en este caso, la empresa puede desplegar sus propias aplicaciones en la infraestructura de nube elegida. De nuevo, quien administra la infraestructura subyacente en el cloud es el proveedor. PaaS permite evitar el coste y la complejidad de comprar y administrar licencias de software, la infraestructura de aplicaciones subyacente y el middleware o las herramientas de desarrollo y otros recursos; garantizando la escalabilidad, ya que el cliente adquiere los recursos que necesita de su proveedor según lo dicten sus necesidades.

Software como servicio (SaaS): a través de este servicio la empresa usuaria puede utilizar las aplicaciones del proveedor, ejecutadas en su infraestructura de la nube, sin poder ejercer capacidades de gestión ni control. Si bien ofrece la ventaja de un acceso prácticamente ilimitado y desde cualquier tipo de dispositivo cliente; presenta el inconveniente de que la última palabra sobre cuestiones relacionadas con la propia red, los servidores, los sistemas operativos, el almacenamiento o incluso las capacidades de las aplicaciones individuales la tenga el proveedor de servicios en la nube.

Marco de buena arquitectura en AWS

El marco de buena arquitectura es una síntesis de más de una década de experiencia en la creación de aplicaciones escalables en la nube.

Los cinco pilares



Seguridad

El pilar se seguridad se centra en cómo asegurar la infraestructura en la nube. Los asuntos relacionados con la seguridad y la conformidad son una responsabilidad compartida entre AWS y el cliente. En este modelo de responsabilidad compartida, AWS es responsable por la seguridad de la nube. Esto incluye la infraestructura física, el software y las capacidades de red de los servicios en la nube de AWS. El responsable de la seguridad en la nube es el usuario. Esto incluye la configuración de servicios específicos en la nube, el software de la aplicación y la administración de los datos confidenciales.

Cuando pensamos en la seguridad en términos de zero trust, significa que tenemos que aplicar medidas de seguridad en todos los niveles de nuestro sistema. A continuación, se presentan tres conceptos importantes para asegurar los sistemas con zero trust en la nube:

Identity and Access Management (IAM)
Seguridad de la red
Cifrado de datos

Eficacia del rendimiento

El pilar de eficacia del rendimiento se centra en cómo puede ejecutar los servicios de manera eficiente y escalable en la nube. Mientras la nube le brinda los medios para gestionar cualquier cantidad de tráfico, requiere que elija y configure los servicios con la escala en mente.

Fiabilidad
El pilar de fiabilidad se centra en cómo puede crear servicios que son resistentes a las interrupciones de infraestructura y servicio. Al igual que con la eficacia del rendimiento, mientras la nube le proporciona los medios para crear servicios resistentes que pueden soportar la interrupción, requiere que diseñe los servicios con la fiabilidad en mente.

Excelencia operativa
El pilar de excelencia operativa se centra en cómo puede mejorar de manera continua su habilidad para ejecutar sistemas, crear mejores procedimientos y obtener información. Cuando piensa en operaciones como automatización, debe centrar la atención en las áreas que actualmente requieren la mayor parte del trabajo manual y que podrían conllevar el mayor nivel de error. También es fundamental implementar un proceso mediante el que se pueda supervisar, analizar y mejorar el trabajo operativo.

Nos centraremos en los siguientes dos conceptos de excelencia operativa:

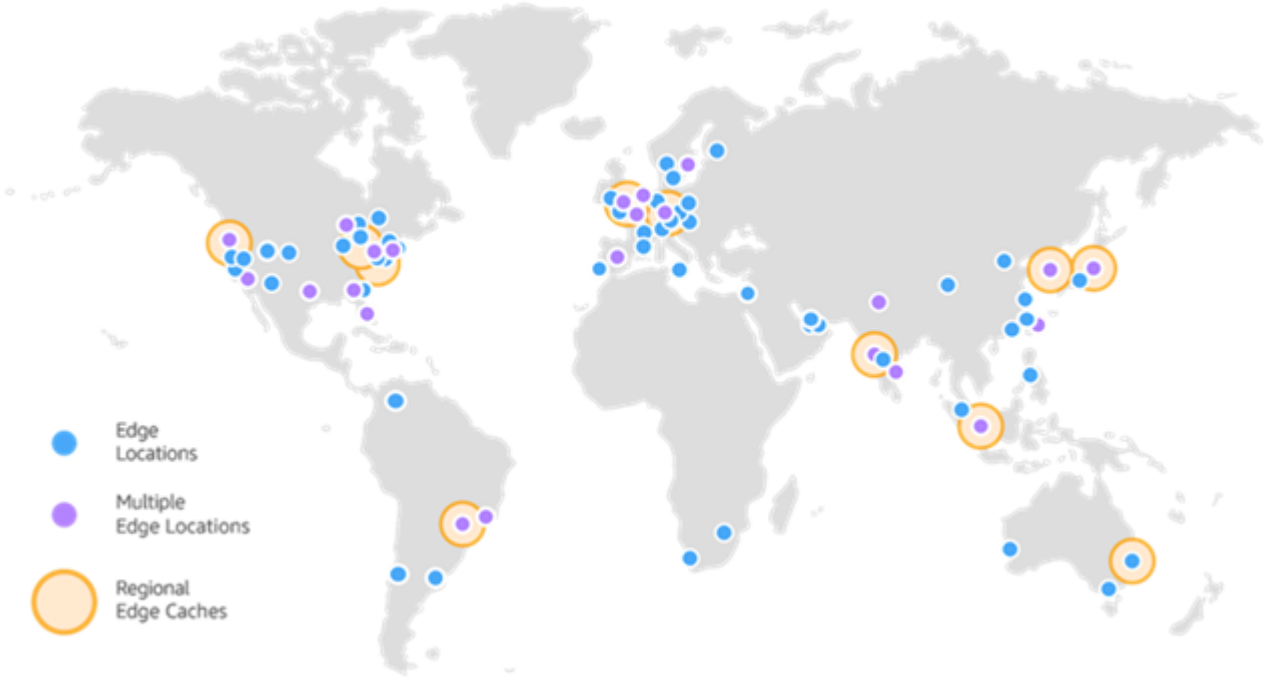
Infraestructura como código
Observabilidad

Optimización de costos
El pilar de optimización de costos ayuda a lograr resultados empresariales mientras se minimizan los costos.
El traslado de un modelo de inversión de capital a uno de gasto operativo cambia totalmente el enfoque del costo de la infraestructura. En lugar de grandes pagos iniciales por costos fijos, piense en pequeños gastos variables continuos.

El modelo de pago por uso introduce los siguientes cambios al proceso de optimización de costos:

Pago por uso
Optimización de costos del ciclo de vida

Regiones y zonas de disponibilidad



Regiones
AWS tiene el concepto de una región, que es una ubicación física en todo el mundo donde agrupamos los centros de datos. Llamamos a cada grupo de centros de datos lógicos “zona de disponibilidad”. Cada región de AWS consta de varias zonas de disponibilidad aisladas y separadas físicamente dentro de un área geográfica.

Zonas de disponibilidad
Una zona de disponibilidad (AZ) es uno o más centros de datos discretos con alimentación, redes y conectividad redundantes en una región de AWS. Las zonas de disponibilidad permiten que los clientes operen bases de datos y aplicaciones de producción con un nivel de disponibilidad, tolerancia a errores y escalabilidad mayor que el que ofrecería un centro de datos único. Todas las zonas de disponibilidad en una región de AWS están interconectadas con redes de alto ancho de banda y baja latencia.

Edge location
Las ubicaciones de borde son centros de datos de AWS diseñados para brindar servicios con la latencia más baja posible
Un subconjunto de servicios para los que la latencia es realmente importante utiliza ubicaciones de borde, que incluyen:

- CloudFront, que utiliza ubicaciones de borde para almacenar en caché copias del contenido que sirve, de modo que el contenido esté más cerca de los usuarios y se les pueda entregar más rápido.

- Route 53, que sirve respuestas de DNS desde ubicaciones de borde, de modo que las consultas de DNS que se originan cerca puedan resolverse más rápido (y, al contrario de lo que podría pensar, también es la principal base de datos de Amazon).
- Web Application Firewall y AWS Shield, que filtran el tráfico en ubicaciones de borde para detener el tráfico no deseado lo antes posible.

Capítulo 2

AWS Identity and Access Management

AWS Identity and Access Management (IAM) le permite controlar de forma segura el acceso de sus usuarios a servicios y recursos de AWS. Con IAM, puede crear y administrar usuarios y grupos de AWS, así como utilizar permisos para conceder o denegar el acceso de estos a los recursos de AWS. IAM le permite hacer lo siguiente:

- Administrar usuarios de IAM y su acceso: puede crear usuarios en IAM, asignarles credenciales de seguridad individuales (claves de acceso, contraseñas y dispositivos de autenticación multifactor) o solicitar credenciales de seguridad temporales para proporcionar a los usuarios acceso a los servicios y recursos de AWS. Es posible administrar los permisos para controlar qué operaciones puede realizar cada usuario.
- Administrar roles de IAM y sus permisos: puede crear roles en IAM y administrar permisos para controlar qué operaciones puede llevar a cabo la entidad o el servicio de AWS que asume el rol. También puede definir a qué entidad se le permite asumir la función.
- Administrar usuarios federados y sus permisos: puede habilitar las identidades federadas a fin de permitir que las identidades ya existentes (usuarios, grupos y roles) de su empresa puedan acceder a la consola de administración de AWS, llamar a las API de AWS y acceder a los recursos sin necesidad de crear un usuario de IAM para cada identidad.

Prácticas recomendadas de seguridad en IAM

Proteger las claves de acceso de usuario raíz de Cuenta de AWS

- Recomendamos encarecidamente que no utilice el usuario raíz en sus tareas cotidianas, ni siquiera en las tareas administrativas. En su lugar, utilice las credenciales de usuario raíz solo para crear un usuario administrador de IAM. A continuación, guarde las credenciales del usuario raíz en un lugar seguro y utilícelas tan solo para algunas tareas de administración de cuentas y servicios. Para las tareas cotidianas, no utilice el usuario administrador de IAM. En su lugar, utilice roles para delegar permisos.
- Si ya tiene una clave de acceso para su usuario raíz de Cuenta de AWS, elimínela. Si debe conservarla, cambie la clave de acceso de forma periódica. Para eliminar o cambiar sus claves de acceso , diríjase a la página Mis credenciales de seguridad en la AWS Management Console e inicie sesión con la dirección de correo electrónico y contraseña de su cuenta. Puede administrar sus claves de acceso en la sección Access keys (Claves de acceso).
- Nunca comparta las claves de acceso ni la contraseña de su usuario raíz Cuenta de AWS con otras personas.
- Utilice una contraseña segura para proteger el acceso de nivel de cuenta a la AWS Management Console.
- Active la autenticación multifactor (MFA) de AWS en su cuenta de usuario raíz de Cuenta de AWS.

Utilizar roles para delegar permisos

Puede asumir un rol de IAM mediante operaciones de AWS Security Token Service o puede cambiar a un rol en la AWS Management Console para recibir una sesión de rol de credenciales temporales. Esto es más seguro que usar su contraseña a largo plazo o las credenciales de clave de acceso. Una sesión tiene una duración limitada, lo que reduce el riesgo si sus credenciales se ven comprometidas.

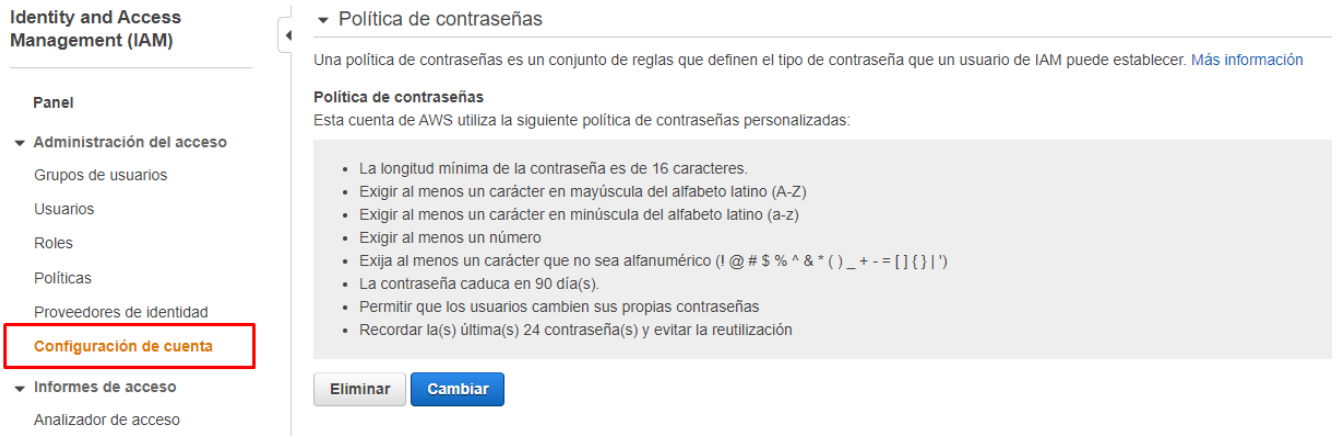
Conceder privilegios mínimos

Al crear políticas de IAM, siga los consejos de seguridad estándar de concesión de *privilegios mínimos* o garantizando solo los permisos necesarios para realizar una tarea. Determine las tareas que tienen que realizar los usuarios (y las funciones) y elabore políticas que les permitan realizar *solo* esas tareas.

Comience con un conjunto mínimo de permisos y conceda permisos adicionales según sea necesario. Por lo general, es más seguro que comenzar con permisos que son demasiado tolerantes e intentar hacerlos más estrictos más adelante.

Configurar una política de contraseña segura para los usuarios

Si permite a los usuarios cambiar sus propias contraseñas, cree una política de contraseñas personalizada que les exija crear contraseñas seguras y cambiarlas periódicamente.



Habilitar MFA

Para mayor seguridad, le recomendamos exigir la autenticación multifactor (MFA) para todos los usuarios de su cuenta. Con MFA, los usuarios tienen un dispositivo que genera una respuesta a un reto de autenticación. Tanto las credenciales del usuario como la respuesta generada por el dispositivo son necesarios para llevar a cabo el proceso de inicio de sesión. Si la contraseña o las claves de acceso de un usuario se ven comprometidas, los recursos de su cuenta siguen estando seguros debido al requisito de autenticación adicional.

Amazon Simple Storage Service

Amazon Simple Storage Service (Amazon S3) es un servicio de almacenamiento de objetos que ofrece escalabilidad, disponibilidad de datos, seguridad y rendimiento.

Clases de almacenamiento
Amazon S3 ofrece varios tipos de almacenamiento diseñados para distintos casos de uso. Por ejemplo, puede almacenar datos de producción críticos en S3 Standard para obtener acceso frecuente, ahorrar costos al almacenar datos a los que se accede con poca frecuencia en S3 Standard-IA o S3 One Zone-IA, y archivar datos con los costos más bajos en S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval y S3 Glacier Deep Archive.

Consistencia sólida
Amazon S3 proporciona una sólida coherencia de lectura tras escritura para las operaciones PUT y DELETE de objetos del bucket de Amazon S3 en todas las Regiones de AWS.
Las actualizaciones en una sola clave son atómicas. Por ejemplo, si aplica PUT a una clave existente de un hilo y realiza una operación GET en la misma clave desde un segundo hilo simultáneamente, obtendrá los datos antiguos o los datos nuevos, pero nunca datos parciales o dañados.

Buckets
Un bucket es un contenedor para objetos almacenados en Amazon S3. Puede almacenar cualquier cantidad de objetos en un bucket y puede tener hasta 100 buckets en su cuenta.

Objetos

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

Control de versiones de S3
Puede usar el control de versiones de S3 para conservar diversas variantes de un objeto en el mismo bucket. Puede utilizar el control de versiones de S3 para conservar, recuperar y restaurar todas las versiones de los objetos almacenados.

Regiones
Puede elegir la Región de AWS geográfica donde Amazon S3 almacenará los buckets que usted cree. Puede elegir una región para optimizar la latencia, minimizar los costos o cumplir con requisitos legales.

¿Qué nivel de durabilidad ofrece Amazon S3?

Amazon S3 Standard, S3 Standard-IA, S3 Intelligent-Tiering, S3 One Zone-IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval y S3 Glacier Deep Archive están todos diseñados para ofrecer una durabilidad de datos del 99,999999999 % (11 nueves) de los objetos durante un periodo de un año. Este nivel de durabilidad corresponde a una pérdida anual media prevista del 0,000000001 % de los objetos. Por ejemplo, si almacena 10 000 objetos con Amazon S3, podría esperar incurrir en una pérdida promedio de un objeto cada 10 000 000 años.

Amazon Virtual Private Cloud (VPC)

Amazon Virtual Private Cloud (VPC) es un servicio que permite lanzar recursos de AWS en una red virtual aislada de forma lógica que usted defina. Puede controlar todos los aspectos del entorno de red virtual, como la selección de su propio rango de direcciones IP, la creación de subredes y la configuración de tablas de enrutamiento y gateways de red.

Ofrece las siguientes características que permiten aumentar y monitorear la seguridad de la VPC:

Registros de flujo
Puede monitorear los registros de flujo de la VPC entregados a Amazon Simple Storage Service (Amazon S3) o a Amazon CloudWatch para obtener una visibilidad operativa de las

dependencias de red y los patrones de tráfico, detectar anomalías y evitar filtraciones de datos y solucionar problemas de configuración y conectividad de la red. Los metadatos enriquecidos de los registros de flujo ayudan a obtener más información acerca de quién inició las conexiones TCP y la fuente y el destino a nivel de paquete para el tráfico que fluye a través de capas intermedias, como la gateway de NAT. También puede archivar los registros de flujo para ayudar a cumplir ciertos requisitos de cumplimiento.

Direccionamiento IP
Las direcciones IP habilitan a los recursos de la VPC para que se comuniquen entre sí y con recursos a través de Internet. Amazon VPC admite los protocolos de direccionamiento IPv4 e IPv6. Amazon también le ofrece múltiples opciones para asignar direcciones IP públicas a sus instancias. Puede utilizar las direcciones IPv4 públicas proporcionadas por Amazon, las direcciones IPv4 elásticas o una dirección IP de los CIDR IPv6 proporcionados por Amazon.

Direccionamiento de entrada
Con esta característica, puede dirigir todo el tráfico entrante y saliente que fluye desde o hacia una puerta de enlace de Internet o una puerta de enlace privada virtual a una instancia específica de Amazon EC2 que tiene una interfaz de red elástica.

Network Access Analyzer
Network Access Analyzer ayuda a verificar que la red de AWS se ajusta a la seguridad de su red y a sus requisitos de cumplimiento. Network Access Analyzer permite especificar la seguridad de la red y los requisitos de cumplimiento e identificar el acceso a la red involuntario que no cumple con los requisitos indicados. Puede utilizar Network Access Analyzer para comprender el acceso de la red a los recursos, lo que ayuda a identificar las mejoras en la posición de seguridad de la nube y demostrar cumplimiento con facilidad.

Lista de control de acceso de red
Una lista de control de acceso de red (ACL de red) es una capa de seguridad opcional para la VPC que actúa como un firewall que controla el tráfico entrante y saliente de una o más subredes. Puede configurar las ACL de red con reglas similares a las de los grupos de seguridad.

Reachability Analyzer
Esta herramienta de análisis de configuración estática lo habilita a analizar y depurar la accesibilidad de red entre dos recursos en la VPC. Después de especificar los recursos fuente y de destino, Reachability Analyzer produce detalles salto por salto de la ruta virtual entre ellos cuando son accesibles e identifica el componente que genera un bloqueo cuando no son accesibles.

Grupos de seguridad

Cree grupos de seguridad para que actúen como un firewall para las instancias de Amazon EC2 asociadas y así controlar el tráfico de entrada y de salida a nivel de instancia. Al lanzar una instancia, puede asociarla a uno o más grupos de seguridad. Si no especifica un grupo, la instancia se asocia automáticamente al grupo predeterminado de la VPC. Cada instancia dentro de la VPC puede pertenecer a un conjunto diferente de grupos.

Capítulo 3

Amazon Elastic Compute Cloud (Amazon EC2)

Amazon Elastic Compute Cloud (Amazon EC2) proporciona capacidad de computación escalable en la nube de Amazon Web Services (AWS). El uso de Amazon 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 los cambios en los requisitos o los picos de popularidad, con lo que se reduce la necesidad de prever el tráfico.

Almacenamiento en EC2.

Elastic Block Store (Amazon EBS)

Proporciona volúmenes de almacenamiento de nivel de bloque para su uso con instancias de EC2. Los volúmenes de EBS se comportan como dispositivos de bloques sin formatear. Amazon EBS se recomienda cuando los datos deben estar accesibles rápidamente y se necesita una persistencia a largo plazo.
Amazon EBS ofrece los siguientes tipos de volúmenes: SSD de uso general, SSD de IOPS provisionadas, HDD con rendimiento optimizado y HDD en frío.

Almacén de instancias Amazon EC2

El *almacén de instancias* ofrece un almacenamiento de nivel de bloques temporal para la instancia. Este almacenamiento se encuentra en discos que están conectados físicamente al equipo host. El almacén de instancias es perfecto para el almacenamiento temporal de información que cambia con frecuencia, como los búferes, las cachés, los datos de pruebas y otro contenido temporal, o de datos que se replican en una flota de instancias, como un grupo de servidores web con balance de carga.

Amazon EFS con Amazon EC2

Amazon EFS proporciona almacenamiento de archivos escalable para su uso con Amazon EC2. Puede usar un sistema de archivos de EFS como un origen de datos común para aplicaciones y cargas de trabajo que se ejecutan en varias instancias.

Imágenes de máquina de Amazon (AMI)

Una Amazon Machine Image (AMI) proporciona la información necesaria para lanzar una instancia. Debe especificar una AMI al lanzar una instancia. Cuando necesite varias instancias con la misma configuración, puede lanzarlas desde una misma AMI. Cuando necesite instancias con distintas configuraciones, puede usar distintas AMI para lanzarlas.

Una AMI incluye lo siguiente:

- Una o más instantáneas de Amazon Elastic Block Store (Amazon EBS) o, para las AMI con respaldo en el almacenamiento de la instancia, una plantilla para el volumen raíz de la instancia (por ejemplo, un sistema operativo, un servidor de aplicaciones y aplicaciones).
- Permisos de lanzamiento que controlan qué cuentas de AWS pueden utilizar la AMI para lanzar instancias.
- Un mapeo de dispositivos de bloques que especifica los volúmenes que se van a adjuntar a la instancia cuando se lance

Tipos de instancias

Cuando se lanza una instancia, el tipo de instancia que especifique determinará el hardware del equipo host utilizado para la instancia. Cada tipo de instancia ofrece distintas características de computación, memoria y almacenamiento, y se agrupa en una familia de instancias en función de dichas características. Seleccione un tipo de instancia en función de los requisitos de la aplicación o del software que tenga previsto ejecutar en la instancia.

https://aws.amazon.com/es/ec2/instance-types/

Opciones de compra de instancias

- **Instancias bajo demanda:** pague, por segundo, solo las instancias que lance.

- **Savings Plans:** reduzca los costos de Amazon EC2 comprometiéndose a una cantidad de uso constante, en USD por hora, durante un periodo de 1 o 3 años.
- **Instancias reservadas:** reduzca sus costos de Amazon EC2 comprometiéndose a tener una configuración de instancia coherente, incluido el tipo de instancia y la región, por un periodo de 1 o 3 años.
- **Instancias de spot:** solicite instancias EC2 no utilizadas, que pueden reducir sus costos de Amazon EC2 considerablemente.
- **Dedicated Hosts (Hosts dedicados):** pague por un host físico dedicado exclusivamente a ejecutar sus instancias y utilice sus propias licencias de software por socket, por núcleo o por VM para reducir costos.
- **Instancias dedicadas:** pague por hora las instancias que se ejecutan en hardware de usuario único.
- **Reservas de capacidad:** reserve capacidad para las instancias EC2 en una zona de disponibilidad específica de cualquier duración.

Seguridad

Grupos de seguridad de Amazon EC2

Un *grupo de seguridad* funciona como un firewall virtual para las instancias EC2 para controlar el tráfico entrante y saliente. Las reglas de entrada controlan el tráfico entrante a la instancia y las reglas de salida controlan el tráfico saliente desde la instancia. Al lanzar una instancia puede especificar uno o varios grupos de seguridad. Si no especifica un grupo de seguridad, Amazon EC2 utiliza el grupo de seguridad predeterminado. Puede añadir reglas a cada grupo de seguridad que permitan el tráfico a o desde sus instancias asociadas. Puede modificar las reglas de un grupo de seguridad en cualquier momento.

Pares de claves de Amazon EC2

Un par de claves, que consta de una clave pública y una clave privada, es un conjunto de credenciales de seguridad que se utiliza para demostrar su identidad cuando se conecta a una instancia de Amazon EC2. Amazon EC2 almacena la clave pública en su instancia y usted almacena la clave privada. Para las instancias de Linux, la clave privada le permite usar SSH para conectarse de forma segura a su instancia. Cualquier persona que posea su clave privada puede conectarse a sus instancias, por lo que es importante que almacene su clave privada en un lugar seguro.

AWS Lambda

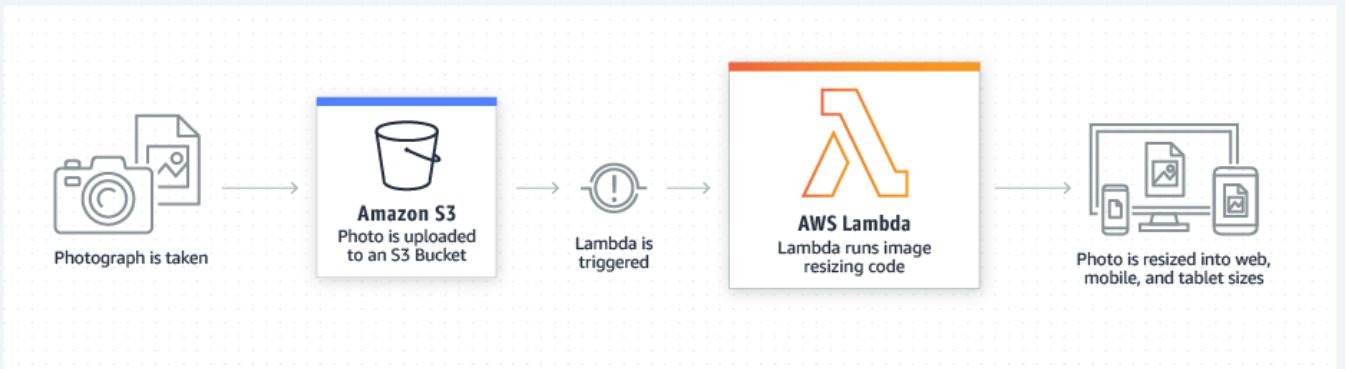
AWS Lambda es un servicio informático sin servidor y basado en eventos que le permite ejecutar código para prácticamente cualquier tipo de aplicación o servicio backend sin necesidad de aprovisionar o administrar servidores. Puede activar Lambda desde más de 200 servicios de AWS y aplicaciones de software como servicio (SaaS), y solo paga por lo que utiliza.

Puede invocar sus funciones de Lambda utilizando la API de Lambda, o Lambda puede ejecutar las funciones en respuesta a eventos de otros servicios de AWS. Por ejemplo, puede utilizar Lambda para:

- Crear desencadenadores de procesamiento de datos para servicios de AWS como Amazon Simple Storage Service (Amazon S3) y Amazon DynamoDB.
- Procesar datos de streaming almacenados en Amazon Kinesis.
- Crear su propio backend que funcione a escala de AWS, rendimiento y seguridad.

ejemplos

Procesamientos de archivos.



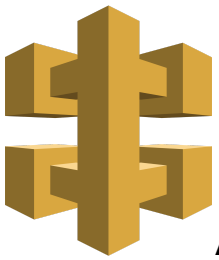
Aplicaciones web

Características.

Lambda ejecuta el código en una infraestructura informática de alta disponibilidad y lleva a cabo toda la administración de los recursos informáticos. Entre otras cosas, se encarga del mantenimiento del servidor y del sistema operativo, del aprovisionamiento de capacidad y del escalado automático, de la implementación de código y de parches de seguridad, y del monitoreo y el registro del código. Lo único que tiene que hacer es proporcionar el código.

Escalado automático

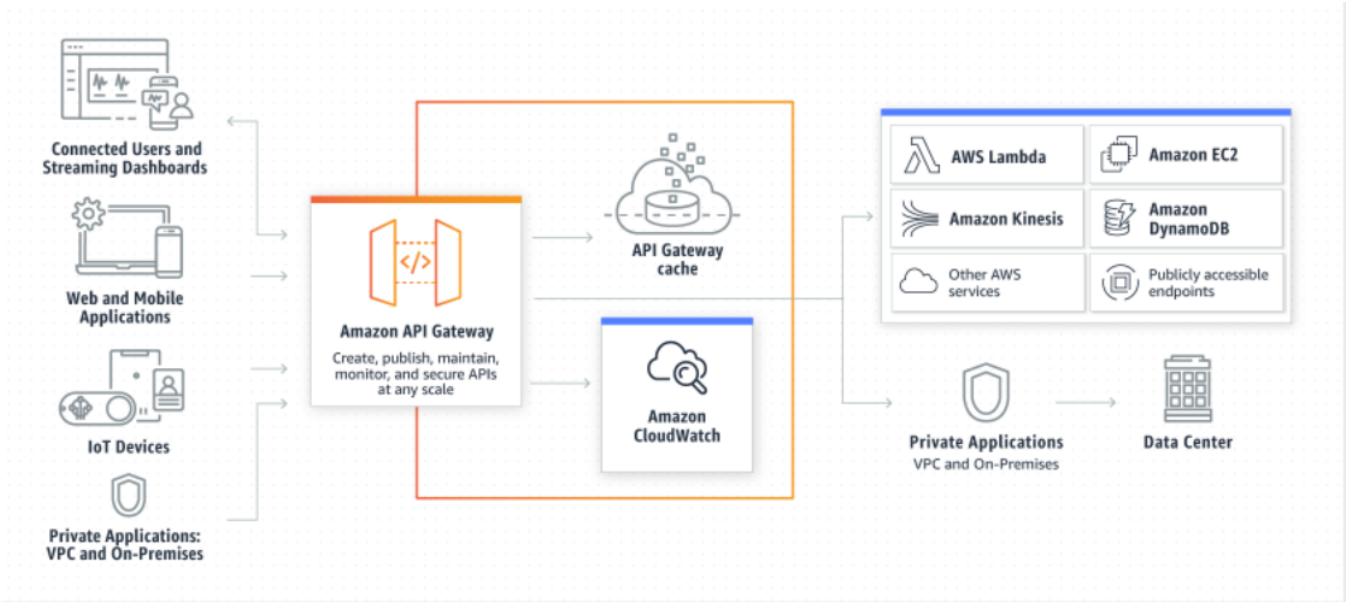
AWS Lambda invoca su código solo cuando es necesario y escala de forma automática para soportar la tasa de solicitudes entrantes sin ninguna configuración manual. No hay ningún límite en cuanto al número de solicitudes que el código puede administrar. AWS Lambda, por lo general, comienza a ejecutar el código al cabo de unos milisegundos después de un evento. Dado que Lambda escala de forma automática, el rendimiento se mantiene siempre alto a medida que aumenta la frecuencia de los eventos. Dado que el código no tiene estado, Lambda puede iniciar tantas instancias como sean necesarias sin implementaciones largas ni retrasos en la configuración.



Amazon API Gateway

Amazon API Gateway es un servicio completamente administrado que facilita a los desarrolladores la creación, la publicación, el mantenimiento, el monitoreo y la protección de API a cualquier escala. Las API actúan como la "puerta de entrada" para que las aplicaciones accedan a los datos, la lógica empresarial o la funcionalidad de sus servicios de backend. Con API Gateway, puede crear API RESTfull y API WebSocket que permiten aplicaciones de comunicación bidireccional en tiempo real. API Gateway admite cargas de trabajo en contenedores y sin servidor, así como aplicaciones web.

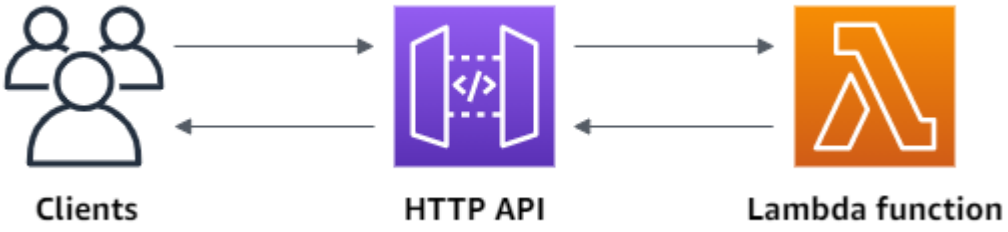
Funcionamiento:



Parte de la infraestructura sin servidor de AWS

Junto con AWS Lambda, API Gateway es la parte de la infraestructura sin servidor de AWS orientada a la aplicación.

Para que una aplicación llame a los servicios de AWS disponibles públicamente, puede utilizar Lambda para interactuar con los servicios necesarios y exponer las funciones de Lambda a través de los métodos de API de API Gateway. AWS Lambda ejecuta el código en una infraestructura informática de alta disponibilidad. Realiza todos los procesos de ejecución y administración que necesitan los recursos informáticos. Para habilitar las aplicaciones sin servidor, API Gateway es compatible con las integraciones de proxy optimizadas con puntos de enlace de AWS Lambda y HTTP.



AWS Database(RDS, Aurora, DynamoDB)

AWS RDS

Amazon Relational Database Service (Amazon RDS) es un servicio web que facilita la configuración, la operación y la escala de una base de datos relacional en Nube de AWS. Proporciona una capacidad rentable y de tamaño ajustable para una base de datos relacional estándar y se ocupa de las tareas de administración de bases de datos comunes.

Amazon RDS y Amazon EC2

Amazon RDS es un servicio de base de datos administrada. Es responsable de la mayoría de las tareas de administración. Al eliminar las tediosas tareas manuales, Amazon RDS le permite centrarse en su aplicación y en sus usuarios. Recomendamos Amazon RDS sobre Amazon EC2 como opción predeterminada para la mayoría de las implementaciones de bases de datos.

En la siguiente tabla podrá encontrar una comparación de los modelos de administración de Amazon EC2 y Amazon RDS.

Característica	Administración de Amazon EC2	Administración de Amazon RDS
Optimización de aplicaciones	Cliente	Cliente
Escalado	Cliente	AWS
Alta disponibilidad	Cliente	AWS
Copias de seguridad de bases de datos	Cliente	AWS
Revisiones de software de base de datos	Cliente	AWS
Instalación de software de base de datos	Cliente	AWS
Revisiones de sistema operativo	Cliente	AWS
Instalación del sistema operativo	Cliente	AWS
Mantenimiento de servidores	AWS	AWS
Ciclo de vida del hardware	AWS	AWS
Alimentación, red y refrigeración	AWS	AWS

Instancias de base de datos

Una instancia de base de datos es un entorno de base de datos aislado en la AWS nube. El componente básico de Amazon RDS es la instancia de base de datos.

Su instancia de base de datos puede contener una o más bases de datos creadas por el usuario. Puede acceder a su instancia de base de datos utilizando las mismas herramientas y aplicaciones que utiliza con una instancia de base de datos independiente. Puede crear y modificar una instancia de base de datos mediante la AWS Command Line Interface, la API de Amazon RDS o la AWS Management Console.

Motores de base de datos
Un motor de base de datos es el software de base de datos relacional específico que se ejecuta en la instancia de base de datos. Amazon RDS admite actualmente los siguientes motores:

MySQL
MariaDB
PostgreSQL
Oracle
Microsoft SQL Server

AWS Aurora

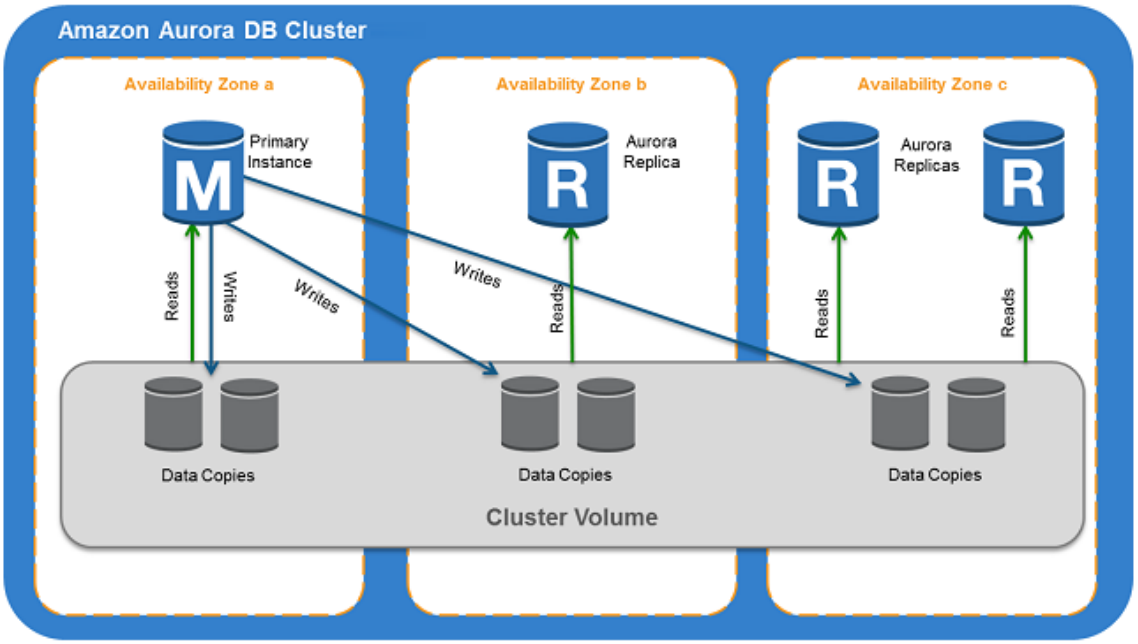
Amazon Aurora (Aurora) es un motor de base de datos relacional completamente administrado compatible con MySQL y PostgreSQL.

Clústeres de base de datos de Amazon Aurora
Un clúster de bases de datos de Amazon Aurora se compone de una o varias instancias de base de datos y de un volumen de clúster que administra los datos de esas instancias de base de datos. Un volumen de clúster de Aurora es un volumen de almacenamiento de base de datos virtual que abarca varias zonas de disponibilidad, de modo que una de esas zonas tiene una copia de los datos del clúster de bases de datos. Un clúster de bases de datos Aurora se compone de dos tipos de instancias de base de datos:

Instancia de base de datos principal: admite operaciones de lectura y escritura y realiza todas las modificaciones de los datos en el volumen de clúster. Cada clúster de bases de datos Aurora tiene una instancia de base de datos principal.

Réplica de Aurora: se conecta con el mismo volumen de almacenamiento que la instancia de base de datos principal y solo admite operaciones de lectura. Cada clúster de bases de datos Aurora puede tener hasta 15 réplicas de Aurora, además de la instancia de base de datos principal. Mantenga una alta disponibilidad localizando réplicas de Aurora en distintas zonas de disponibilidad. Aurora cambiará

automáticamente a una réplica de Aurora en caso de que la instancia de base de datos principal deje de estar disponible. Puede especificar la prioridad de conmutación por error para réplicas de Aurora. Las réplicas de Aurora también pueden descargar las cargas de trabajo de lectura desde la instancia de base de datos principal.



AWS DynamoDB

Amazon DynamoDB es un servicio de base de datos NoSQL totalmente administrado que ofrece un rendimiento rápido y predecible, así como una perfecta escalabilidad.
DynamoDB le permite delegar las cargas administrativas que supone tener que utilizar y escalar bases de datos distribuidas, para que no tenga que preocuparse del aprovisionamiento, la instalación ni la configuración del hardware, ni tampoco de las tareas de replicación, aplicación de parches de software o escalado de clústeres. DynamoDB también ofrece el cifrado en reposo, que elimina la carga y la complejidad operativa que conlleva la protección de información confidencial.

Alta disponibilidad y durabilidad
DynamoDB distribuye automáticamente los datos y el tráfico de las tablas entre un número suficiente de servidores para satisfacer sus requisitos de almacenamiento y rendimiento, al mismo tiempo que mantiene un desempeño uniforme y rápido. Todos los datos se almacenan en discos de estado sólido (SSD) y se replican automáticamente en varias zonas de disponibilidad de una región de AWS.

Consistencia de lectura

Lecturas consistentes finales

Al leer datos de una tabla de DynamoDB, la respuesta podría no reflejar los resultados de una operación de escritura reciente. La respuesta podría incluir algunos datos anticuados. Si repite la solicitud de lectura tras un breve intervalo de tiempo, la respuesta debería devolver los datos más recientes.

Lecturas de consistencia alta

Cuando se solicita una lectura de consistencia alta, DynamoDB devuelve una respuesta con los datos más actualizados, de tal forma que refleja las actualizaciones de todas las operaciones de escritura anteriores que se han llevado a cabo correctamente. No obstante, esta coherencia conlleva algunas desventajas:

- Una lectura de consistencia alta podría no estar disponible si se produce un retraso o una interrupción en la red. En este caso, DynamoDB podría devolver un error de servidor (HTTP 500).
- Las lecturas con coherencia alta pueden tener mayor latencia que las lecturas con coherencia final.
- Las lecturas altamente consistentes no se admiten para los índices secundarios globales.
- Las lecturas con coherencia alta utilizan mayor capacidad de rendimiento que las lecturas con coherencia final.

Capítulo 4

AWS Simple Notification Service

Amazon Simple Notification Service (Amazon SNS) es un servicio web que coordina y gestiona la entrega o el envío de mensajes a los puntos de enlace o clientes suscritos.

En Amazon SNS existen dos tipos de clientes, editores y suscriptores, también conocidos como productores y consumidores.

Los publicadores se comunican de forma asíncrona con los suscriptores generando y enviando un mensaje a un tema, que es un punto de acceso lógico y un canal de comunicación. Los suscriptores (servidores web, direcciones de correo electrónico, colas de Amazon SQS, funciones de Lambda) consumen o reciben el mensaje o la notificación por medio de uno de los protocolos admitidos (Amazon SQS, HTTP/S, correo electrónico, SMS,AWS Lambda) cuando se hayan suscrito al tema.

Amazon SNS le permite enviar notificaciones directamente a sus clientes. Amazon SNS admite mensajería de texto en más de 200 países, notificaciones push móviles en dispositivos Amazon, Android, Apple, Baidu y Microsoft, y notificaciones de correo electrónico. Amazon SNS proporciona redundancia en múltiples proveedores de SMS y le permite enviar notificaciones push móviles con una API única para todas las plataformas móviles.

Amazon CloudWatch

Amazon CloudWatch monitorea los recursos y las aplicaciones de Amazon Web Services (AWS) que ejecuta en AWS en tiempo real. Puede utilizar CloudWatch para recopilar y hacer un seguimiento de métricas, que son las variables que puede medir en los recursos y aplicaciones.
Puede crear alarmas que vigilen métricas y enviar notificaciones o realizar cambios automáticamente en los recursos que está monitoreando cuando se infringe un umbral. Por ejemplo, puede monitorear el uso de la CPU y las lecturas y escrituras de disco de las instancias de Amazon EC2 y, a continuación, utilizar esos datos para determinar si se deben lanzar instancias adicionales para gestionar el aumento de la carga. También puede utilizar estos datos para parar las instancias infrautilizadas a fin de ahorrar dinero.

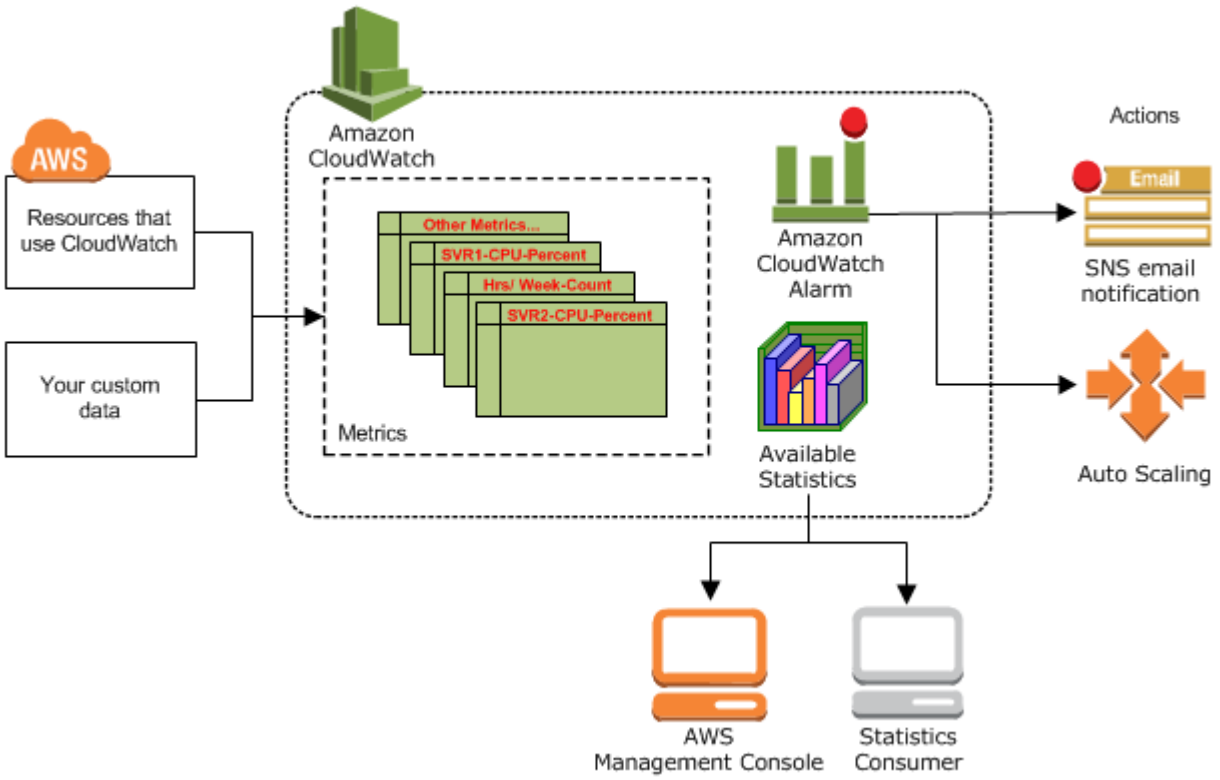
Los siguientes servicios se utilizan junto con Amazon CloudWatch:

- Amazon Simple Notification Service (Amazon SNS) coordina y administra la entrega o el envío de mensajes a los puntos de enlace o clientes suscritos. Amazon SNS se utiliza con CloudWatch para enviar mensajes cuando se alcanza un umbral de alarma.

- Amazon EC2 Auto Scaling le permite lanzar o terminar instancias de Amazon EC2 automáticamente de acuerdo con las políticas que el usuario define, las verificaciones de estado y las programaciones. Puede utilizar una alarma de CloudWatch con Amazon EC2 Auto Scaling para regular las instancias EC2 en función de la demanda.

- AWS Identity and Access Management (IAM) es un servicio web que ayuda a controlar de forma segura el acceso de los usuarios a los recursos de AWS. Utilice IAM para controlar quién puede usar los recursos de AWS (autenticación), así como cuáles de ellos pueden usar y de qué manera pueden hacerlo (autorización).
-

Amazon CloudWatch es básicamente un repositorio de métricas. Un servicio de AWS, como Amazon EC2, coloca las métricas en el repositorio, lo que logrará que recupere las estadísticas en función de dichas métricas. Si coloca sus propias métricas personalizadas en el repositorio, puede recuperar estadísticas sobre estas métricas también.


AWS
Resources that use CloudWatch
Your custom data
Amazon CloudWatch
Other Metrics...
SVR1-CPU-Percent
Hrs/ Week-Count
SVR2-CPU-Percent
Metrics
Amazon CloudWatch Alarm
Available Statistics
Actions
Email
SNS email notification
Auto Scaling
AWS Management Console
Statistics Consumer

Capítulo 5

CloudFormation

AWS CloudFormation es un servicio que le ayuda a modelar y configurar sus recursos de AWS, por lo que podrá dedicar menos tiempo a la administración de dichos recursos y más tiempo a centrarse en las aplicaciones que se ejecutan en AWS. Puede crear una plantilla que describa todos los recursos de AWS que desea (como instancias de Amazon EC2 o instancias de base de datos de Amazon RDS) y CloudFormation se encargará del aprovisionamiento y la configuración de dichos recursos.
https://docs.aws.amazon.com/es_es/AWSCloudFormation/latest/UserGuide/Welcome.html

```
AWSTemplateFormatVersion: "2010-09-09"

Description: Resources Regional
Parameters:
  env:
    Type: "String"
    Description: Environment name (dev/qa/prod)
    AllowedValues:
      - dev
      - qa
      - prod
  apiGatewayName:
    Type: String
    Default: "test-api"

Resources:
  ApiGatewayRestApi:
    Type: AWS::ApiGateway::RestApi
    Properties:
      ApiKeySourceType: HEADER
      Description: An API Gateway for Lambda APIs
      EndpointConfiguration:
        Types:
          - EDGE
      Name: !Sub ${apiGatewayName}-${env}

  #Path CL
  ResourceCl:
    Type: AWS::ApiGateway::Resource
    Properties:
```

```
      RestApiId: !Ref ApiGatewayRestApi
      ParentId: !GetAtt ApiGatewayRestApi.RootResourceId
      PathPart: "test"

  ProxyResourceBidireccionalidadCl:
    Type: "AWS::ApiGateway::Resource"
    Properties:
      RestApiId: !Ref ApiGatewayRestApi
      ParentId: !Ref ResourceCl
      PathPart: !Ref apiResourceBidireccionalidadCl

  ProxyResourceMethodBidireccionalidadCl:
    Type: "AWS::ApiGateway::Method"
    Properties:
      RestApiId: !Ref ApiGatewayRestApi
      ResourceId: !Ref ProxyResourceBidireccionalidadCl
      HttpMethod: GET
      ApiKeyRequired: false
      AuthorizationType: COGNITO_USER_POOLS
      AuthorizationScopes:
        - otc/read_operation
      AuthorizerId: !Ref ApiGatewayAuthorizer
      OperationName: !Ref apiResourceBidireccionalidadCl
      Integration:
        Type: AWS_PROXY
        IntegrationHttpMethod: POST
        Uri: !Sub
          -
"arn:aws:apigateway:${AWS::Region}:lambda:path/2015-03-31/functions/${ArnLambda
}/invocations"
          - ArnLambda: !ImportValue ArnLambdaAPIVistas

  #APIGW Deploy
  ApiGatewayLogGroup:
    Type: AWS::Logs::LogGroup
    Properties:
      LogGroupName: !Sub /aws/apigateway/${apiGatewayName}-${env}
      RetentionInDays: 30

  ApiGatewayStage:
```

```
    Type: AWS::ApiGateway::Stage
    Properties:
      DeploymentId: !Ref ApiGatewayDeployment
      Description: Lambda API Stage v1
      RestApiId: !Ref ApiGatewayRestApi
      StageName: !Ref env
      MethodSettings:
        - ResourcePath: /
          HttpMethod: GET
          MetricsEnabled: "true"
          DataTraceEnabled: "false"
        - ResourcePath: /stack
          HttpMethod: GET
          MetricsEnabled: "true"
          DataTraceEnabled: "false"
          ThrottlingBurstLimit: "999"
        - ResourcePath: /stack
          HttpMethod: GET
          MetricsEnabled: "true"
          DataTraceEnabled: "false"
          ThrottlingBurstLimit: "555"
      AccessLogSetting:
        DestinationArn: !GetAtt ApiGatewayLogGroup.Arn
        Format: '{ "requestId": "$context.requestId", "ip":
"$context.identity.sourceIp", "caller": "$context.identity.caller", "user":
"$context.identity.user", "requestTime": "$context.requestTime", "httpMethod":
"$context.httpMethod", "resourcePath": "$context.resourcePath", "status":
"$context.status", "protocol": "$context.protocol", "responseLength":
"$context.responseLength", "X-Ray": "$context.xrayTraceId", "path":
"$context.path", "authorizerLatency": "$context.authorizer.latency",
"authenticateLatency": "$context.authenticate.latency", "responseLatency":
"$context.responseLatency", "requestTime": "$context.requestTime" }'

  ApiGatewayDeployment:
    Type: AWS::ApiGateway::Deployment
    DependsOn:
      #CL
      - ProxyResourceMethodBidireccionalidadCl
    Properties:
      Description: Lambda API Deployment v2
```

```
      RestApiId: !Ref ApiGatewayRestApi
```

código github action y aws cli para deployar el cloudformation

```
- name: Deploy API
  id: api
  shell: bash
  if: contains(github.event.pull_request.title, '-api') ||
contains(github.event.pull_request.title, '-all')
  run: |
    TIMESTAMP=$(date +'%Y%m%d%H%M%S')
    sed -ie "s/ApiGatewayDeployment/ApiGatewayDeployment$TIMESTAMP/g"
templates/otc-apigw-waf.yaml
    aws cloudformation deploy --template-file templates/otc-apigw-waf.yaml \
    --region $REGION \
    --stack-name otc-api-$ENVIRONMENT  \
    --capabilities CAPABILITY_NAMED_IAM \
    --no-fail-on-empty-changeset \
    --parameter-overrides env=$ENVIRONMENT

- name: Show failed events for apigw-waf stack
  if: ${{ failure() }}
  shell: bash
  run: |
    aws cloudformation describe-stack-events --stack-name otc-api-$ENVIRONMENT
```