Documento técnico de AWS

Información general acerca de AWS Lambda

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Información general acerca de AWS Lambda: Documento técnico de AWS

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon, de ninguna manera que pueda causar confusión entre los clientes y de ninguna manera que menosprecie o desacredite a Amazon. Todas las demás marcas comerciales que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

Resumen i
Resumen 1
Introducción 2
Acerca de AWS Lambda 3
Beneficios de Lambda 3
No se administran servidores. 4
Escalado continuo 4
Medición en milisegundos 4
Aumenta la innovación 4
Modernice sus aplicaciones 4
Ecosistema completo 4
Coste de ejecución de aplicaciones basadas en Lambda 5
El modelo de responsabilidad compartida 6
Funciones de Lambda 7
Modos de invocación de Lambda 8
Ejecuciones de Lambda 10
Entornos de ejecución de Lambda 10
Rol de ejecución 12
MicroVM y procesos de trabajo de Lambda 12
Tecnologías de aislamiento de Lambda 14
Almacenamiento y estado 15
Mantenimiento en tiempo de ejecución en Lambda 16
Supervisión y auditoría de funciones de Lambda 18
Amazon CloudWatch 18
Amazon CloudTrail 18
AWS X-Ray 18
AWS Config 19
Diseño y funcionamiento de las funciones de Lambda 20
Lambda y cumplimiento 21
Fuentes de eventos de Lambda 22
Conclusión 24
Colaboradores 25
Documentación adicional 26
Revisiones del documento 27

Avisos 28

Información general acerca de AWS Lambda

Fecha de publicación: 12 de febrero de 2021 (Revisiones del documento)

Resumen

En este documento técnico, se analiza en detalle el servicio de AWS Lambda a través de un enfoque de seguridad. Se proporciona una imagen completa del servicio, que resulta útil para los nuevos usuarios y profundiza en los conocimientos de Lambda para los usuarios actuales.

El público al que se dirige este documento técnico son directores de seguridad de la información (CISO), ingenieros de seguridad de la información, arquitectos empresariales, equipos de cumplimiento y cualquier otra persona interesada en comprender los fundamentos de AWS Lambda.

Introducción

En la actualidad, cada vez hay más cargas de trabajo que utilizan AWS Lambda para conseguir escalabilidad, rendimiento y rentabilidad, sin tener que administrar la infraestructura subyacente. Estas cargas de trabajo se escalan a miles de solicitudes simultáneas por segundo. Lambda es uno de los muchos servicios importantes que ofrece AWS en la actualidad. Cientos de miles de clientes de Amazon Web Services (AWS) utilizan Lambda para atender billones de solicitudes al mes.

Lambda es adecuada para aplicaciones esenciales en muchos sectores. Una amplia variedad de clientes, desde medios de comunicación y entretenimiento hasta servicios financieros y otras industrias reguladas, utilizan Lambda. Estos clientes reducen el tiempo de comercialización, optimizan los costes y mejoran la agilidad al centrarse en lo que mejor saben hacer: administrar su negocio.

El modelo de entorno en tiempo de ejecuciónadministrado permite a Lambda administrar gran parte de los detalles de la implementación de la ejecución de cargas de trabajo sin servidor. Este modelo reduce aún más la superficie de ataque y simplifica la seguridad en la nube. Este documento técnico presenta los fundamentos de ese modelo, junto con las prácticas recomendadas para los desarrolladores, los analistas de seguridad, los equipos de seguridad y cumplimiento y otras partes interesadas.

Acerca de AWS Lambda

AWS Lambda es un servicio de computación basado en eventos y sin servidor que amplía otros servicios de AWS con lógica personalizada o crea otros servicios de backend que funcionan con escala, rendimiento y seguridad. Lambda puede ejecutar código automáticamente en respuesta a varios eventos, como solicitudes HTTP a través de Amazon API Gateway, modificaciones de los objetos en buckets de Amazon S3, actualizaciones de tablas en Amazon DynamoDB y transiciones de estado en AWS Step Functions. También puede ejecutar código directamente desde cualquier aplicación web o móvil. Lambda ejecuta el código en una infraestructura de computación de alta disponibilidad y realiza todas las tareas de administración de la plataforma subyacente, incluido el mantenimiento del servidor y del sistema operativo, el aprovisionamiento de capacidad y el escalado automático, así como la supervisión del código y las funciones de registro.

Con Lambda, puede cargar el código y configurar cuándo se invoca; Lambda se encarga de todo lo demás que es necesario para ejecutar el código con alta disponibilidad. Lambda se integra con muchos otros servicios de AWS y permite crear aplicaciones sin servidor o servicios de backend, que van desde tareas de automatización sencillas que se desencadenan periódicamente hasta aplicaciones de microservicios completas.

Lambda también se puede configurar para acceder a los recursos de la nube virtual privada de Amazon y, por extensión, a los recursos locales.

Puede encapsular Lambda fácilmente con una posición de seguridad sólida con AWS Identity and Access Management (IAM) y otras técnicas que se analizan en este documento técnico para mantener un alto nivel de seguridad y auditoría, y para satisfacer sus necesidades de cumplimiento.

Temas

- Beneficios de Lambda
- Coste de ejecución de aplicaciones basadas en Lambda

Beneficios de Lambda

Los clientes que desean dar rienda suelta a la creatividad y la velocidad de sus organizaciones de desarrollo, pero sin comprometer la capacidad de su equipo de TI de proporcionarles una infraestructura escalable, rentable y administrable, descubren que AWS Lambda les permite cambiar la complejidad operativa por agilidad y mejores precios, sin comprometer la escala o la fiabilidad.

Lambda tiene muchos beneficios, entre los que cabe destacar los siguientes:

No se administran servidores.

Lambda ejecuta el código en una infraestructura de alta disponibilidad y tolerante a errores que se extiende por varias zonas de disponibilidad (AZ) en una sola región, implementa código sin problemas y proporciona todas las tareas de administración, mantenimiento y aplicación de revisiones de la infraestructura. Lambda también ofrece funciones de registro y supervisión integradas, incluida la integración con Amazon CloudWatch, CloudWatch Logs y AWS CloudTrail.

Escalado continuo

Lambda administra con precisión el escalado de las funciones (o aplicaciones) ejecutando en paralelo código desencadenado por eventos y procesando cada evento de forma individual.

Medición en milisegundos

Con AWS Lambda, se le cobra por cada milisegundo (ms) de ejecución del código y por la cantidad de veces que se activa el código. Paga por un rendimiento uniforme o por la duración de la ejecución, no por unidad de servidor.

Aumenta la innovación

Lambda libera sus recursos de programación al hacerse cargo de la administración de la infraestructura, lo que les permite centrarse más en la innovación y el desarrollo de la lógica empresarial.

Modernice sus aplicaciones

Lambda le permite utilizar funciones con modelos de machine learning previamente entrenados para inyectar inteligencia artificial con facilidad en las aplicaciones. Una única solicitud de interfaz de programa de aplicación (API) puede clasificar imágenes, analizar vídeos, convertir voz en texto, realizar procesamiento de lenguaje natural y mucho más.

Ecosistema completo

Lambda ayuda a los desarrolladores con AWS Serverless Application Repository a descubrir, implementar y publicar aplicaciones sin servidor y con AWS Serverless Application Model a crear aplicaciones sin servidor e integraciones con varios entornos de desarrollo (IDE) como AWS Cloud9,

AWS Toolkit for Visual Studio, AWS Tools for Visual Studio Team Services y muchos otros. Lambda se integra con servicios de AWS adicionales para proporcionarle un ecosistema completo para crear aplicaciones sin servidor.

Coste de ejecución de aplicaciones basadas en Lambda

Lambda ofrece un modelo de precios granular y de pago por uso. Con este modelo, se le cobra en función del número de invocaciones de funciones y su duración (es decir, el tiempo que tarda en ejecutarse el código). Además de este modelo de precios flexible, Lambda también ofrece 1 millón de solicitudes gratuitas perpetuas al mes, lo que permite a muchos clientes automatizar sus procesos sin ningún coste.

El modelo de responsabilidad compartida

Los asuntos relacionados con la seguridad y la conformidad son una responsabilidad compartida entre AWS y el cliente. Este modelo de responsabilidad compartida puede aliviar su carga operativa, ya que AWS opera, administra y controla los componentes del sistema operativo host y la capa de virtualización y los sistemas de seguridad física de las instalaciones en las que operan los servicios.

En AWS Lambda, AWS administra la infraestructura y los servicios básicos subyacentes, el sistema operativo y la plataforma de aplicaciones. Usted es responsable de la seguridad de su código y de la administración de Identity and Access Management (IAM) para el servicio Lambda y en su función.

La figura 1 muestra el modelo de responsabilidad compartida que se aplica a los componentes comunes y distintos de AWS Lambda. Las responsabilidades de AWS aparecen debajo de la línea de puntos de color naranja y las responsabilidades de los clientes aparecen por encima de la línea de puntos de color azul.

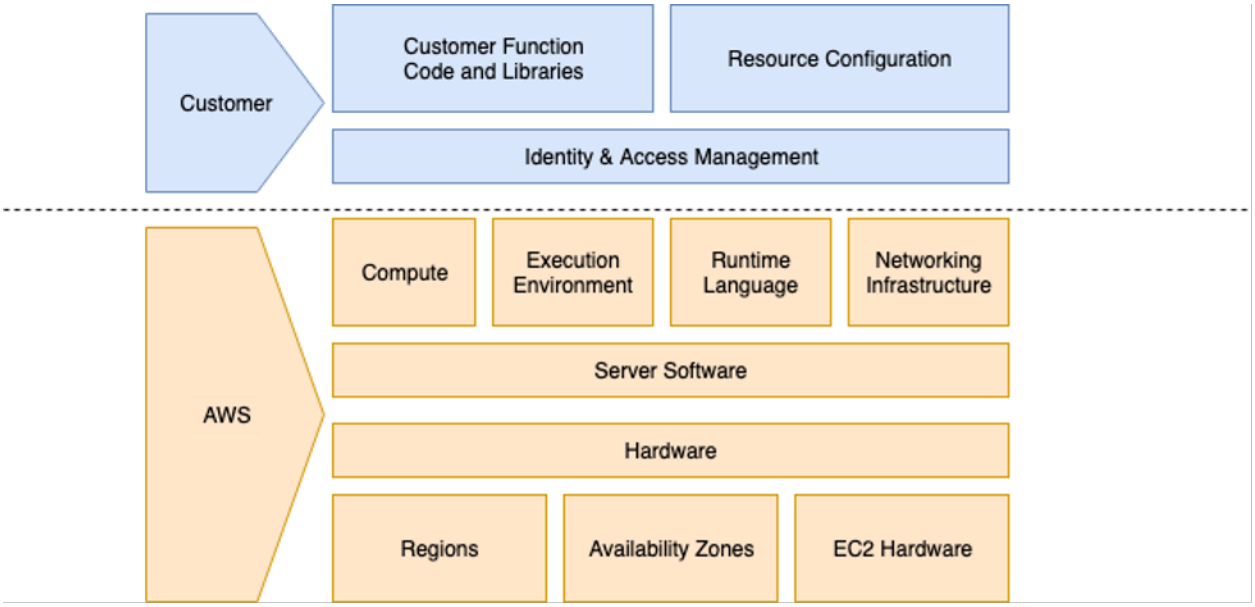

Figura 1: Modelo de responsabilidad compartida de AWS Lambda

Funciones y capas de Lambda

Con Lambda, puede ejecutar código prácticamente sin administrar la infraestructura subyacente. Usted solo es responsable del código que proporciona a Lambda y de la configuración para que Lambda ejecute ese código en su nombre. En la actualidad, Lambda admite dos tipos de recursos de código: funciones y capas.

Una función es un recurso que se puede invocar para ejecutar el código en Lambda. Las funciones pueden incluir un recurso común o compartido llamado capas. Las capas se pueden utilizar para compartir código o datos comunes en diferentes funciones o cuentas de AWS. Usted es responsable de la administración de todo el código que contienen sus funciones o capas. Cuando Lambda recibe la función o el código de capa de un cliente, protege el acceso a este cifrándolo en reposo con AWS Key Management Service (AWS KMS) y en tránsito mediante TLS 1.2+.

Puede administrar el acceso a sus funciones y capas mediante políticas de AWS Lambda o mediante permisos basados en recursos. Para obtener una lista completa de las características de IAM que se admiten en IAM, consulte Servicios de AWS que funcionan con IAM.

También puede controlar todo el ciclo de vida de sus funciones y capas a través de las API del plano de control de Lambda. Por ejemplo, puede eliminar su función llamando a `DeleteFunction` o revocar los permisos de otra cuenta llamando a `RemovePermission`.

Modos de invocación de Lambda

La API Invoke se puede llamar en dos modos: el modo de evento y el modo solicitud-respuesta.

- El modo de evento pone en cola la carga para realizar una invocación asíncrona.
- El modo de solicitud-respuesta invoca sincrónicamente la función con la carga proporcionada y devuelve una respuesta de inmediato.

En ambos casos, la ejecución de la función siempre se lleva a cabo en un entorno de ejecución de Lambda, pero la carga toma rutas diferentes. Para obtener más información, consulte “Entornos de ejecución de Lambda” en este documento.

También puede utilizar otros servicios de AWS que realicen invocaciones en su nombre. El modo de invocación que se utilice depende del servicio de AWS que use y de la forma en que esté configurado. Para obtener información adicional sobre cómo se integran otros servicios de AWS con Lambda, consulte Uso de AWS Lambda con otros servicios.

Cuando Lambda recibe una invocación de solicitud-respuesta, se pasa directamente al servicio de invocación. Si el servicio de invocación no está disponible, los autores de la llamada pueden poner en cola temporalmente la carga del lado del cliente para volver a intentar la invocación un número determinado de veces. Si el servicio de invocación recibe la carga, el servicio intenta identificar un entorno de ejecución disponible para la solicitud y pasa la carga a ese entorno para completar la invocación. Si no existe ningún entorno de ejecución o no hay ninguno que sea apropiado, se crea uno dinámicamente en respuesta a la solicitud. Mientras están en tránsito, las cargas de invocación enviadas al servicio de invocación están protegidas con TLS 1.2+. El tráfico dentro del servicio Lambda (desde el equilibrador de carga hacia abajo) pasa a través de una nube virtual privada (VPC) interna aislada, que es propiedad del servicio Lambda, dentro de la región de AWS a la que se envió la solicitud.

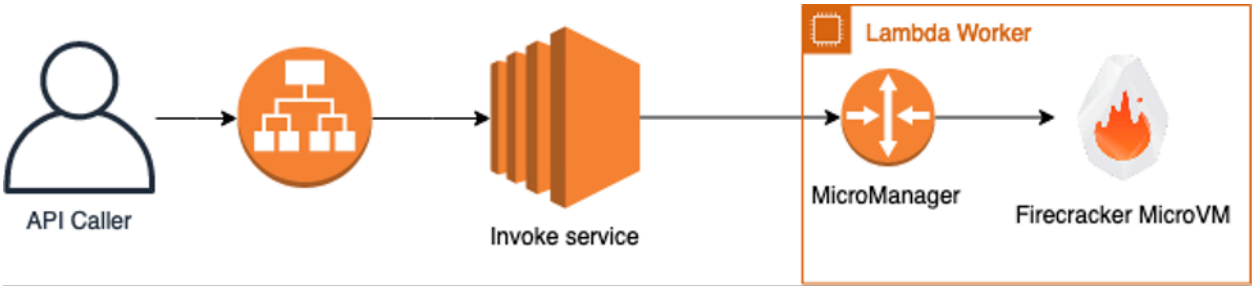

Figura 2: Modelo de invocación de solicitud-respuesta de AWS Lambda

Las cargas del modo de invocación de eventos siempre se ponen en cola para procesarlas antes de la invocación. Todas las cargas se ponen en cola para procesarlas en una cola de Amazon Simple Queue Service (Amazon SQS). Los eventos en cola siempre se protegen en tránsito con TLS 1.2+, pero actualmente no se cifran en reposo. Las colas de Amazon SQS que utiliza Lambda las administra el servicio Lambda y no son visibles para los clientes. Los eventos en cola se pueden almacenar en una cola compartida, pero se pueden migrar o asignar a colas dedicadas en función de una serie de factores que los clientes no pueden controlar directamente (por ejemplo, la tasa de invocación, el tamaño de los eventos, etc.).

La flota de sondeadores de Lambda recupera los eventos en cola en lotes. La flota de sondeadores es un grupo de instancias de EC2 cuyo propósito es procesar invocaciones de eventos en cola que aún no se han procesado. Cuando la flota de sondeadores recupera un evento en cola que tiene que procesar, lo hace pasándoselo al servicio de invocación tal y como lo haría un cliente en una invocación en modo de solicitud-respuesta.

Si no se puede realizar la invocación, la flota de sondeadores almacena temporalmente el evento en memoria en el host hasta que pueda realizar correctamente la ejecución o hasta que se haya superado el número de intentos de ejecución. Nunca se escriben datos de carga en el disco en la propia flota de sondeadores. La flota de sondeadores se puede asignar a todos los clientes de AWS, lo que permite conseguir el tiempo de invocación mínimo. Para obtener más información sobre qué servicios pueden utilizar el modo de invocación de eventos, consulte Uso de AWS Lambda con otros servicios.

Ejecuciones de Lambda

Cuando Lambda ejecuta una función en su nombre, administra tanto el aprovisionamiento como la configuración de los sistemas subyacentes necesarios para ejecutar el código. Esto permite a los desarrolladores centrarse en la lógica empresarial y escribir código, no en administrar y gestionar los sistemas subyacentes.

El servicio Lambda se divide en el plano de control y el plano de datos. Cada plano tiene un propósito distinto en el servicio. El plano de control proporciona las API de administración (por ejemplo, `CreateFunction`,`UpdateFunctionCode`,`PublishLayerVersion`, etc.) y administra las integraciones con todos los servicios de AWS. Las comunicaciones con el plano de control de Lambda están protegidas en tránsito por TLS. Todos los datos de los clientes almacenados en el plano de control de Lambda se cifran en reposo mediante el uso de AWS KMS, que está diseñado para protegerlos de la divulgación o manipulación no autorizadas.

El plano de datos es la API Invoke de Lambda, que desencadena la invocación de funciones de Lambda. Cuando se invoca una función de Lambda, el plano de datos asigna un entorno de ejecución en un proceso de trabajo AWS Lambda (o para abreviar, un proceso de trabajo, que es un tipo de instancia de Amazon EC2) para esa versión de la función, o elige un entorno de ejecución existente que ya se ha configurado para esa versión de la función, que luego usa para completar la invocación. Para obtener más información, consulte la sección “MicroVM y procesos de trabajo de AWS Lambda” de este documento.

Entornos de ejecución de Lambda

El servicio de invocación de Lambda enruta cada invocación a un entorno de ejecución en un proceso de trabajo que puede atender la solicitud. Salvo a través del plano de datos, los clientes y otros usuarios no pueden iniciar directamente las comunicaciones de red entrantes/de entrada con un entorno de ejecución. Esto ayuda a garantizar que las comunicaciones con su entorno de ejecución se autentiquen y autoricen.

Los entornos de ejecución están reservados para una versión de la función específica y no se pueden reutilizar en otras versiones de la función, funciones o cuentas de AWS. Esto significa que una sola función que tenga dos versiones diferentes daría como resultado al menos dos entornos de ejecución únicos.

Cada entorno de ejecución solo se puede usar para una invocación simultánea a la vez, y se pueden reutilizar en múltiples invocaciones de la misma versión de la función por motivos de rendimiento.

Dependiendo de varios factores (por ejemplo, velocidad de invocación, configuración de funciones, etc.), pueden existir uno o más entornos de ejecución para la versión de una función determinada. Con este enfoque, Lambda puede proporcionar aislamiento a nivel de versión de las funciones para sus clientes.

Lambda no aísla las invocaciones en el entorno de ejecución de la versión de una función. Lo que esto significa es que una invocación puede dejar un estado que podría afectar a la siguiente invocación (por ejemplo, archivos escritos en /tmp o datos en memoria). Si desea asegurarse de que una invocación no afecte a otra, Lambda recomienda crear funciones distintas adicionales. Por ejemplo, podría crear funciones distintas para operaciones de análisis complejas que son más propensas a errores y reutilizar funciones que no realizan operaciones en las que la seguridad es importante. Lambda no limita actualmente la cantidad de funciones que pueden crear los clientes. Para obtener más información sobre los límites, consulte la página Cuotas de Lambda.

Lambda supervisa y administra continuamente los entornos de ejecución, que pueden crearse o destruirse por diversos motivos, entre los que se incluyen, entre otros:

- Llega una nueva invocación y no existe un entorno de ejecución adecuado.
- Se produce una implementación de software de proceso de trabajo o tiempo de ejecución interna.
- Se publica una nueva configuración de simultaneidad aprovisionada.
- El tiempo de concesión en el entorno de ejecución, o el proceso de trabajo, se está acabando o ha superado la vida útil máxima.
- Otros procesos de reequilibrio de la carga de trabajo interna.

Los clientes pueden administrar el número de entornos de ejecución aprovisionados previamente que existen para una versión de función configurando la simultaneidad aprovisionada en su configuración de funciones. De esta manera, Lambda creará, administrará y garantizará que siempre exista la cantidad configurada de entornos de ejecución. Esto garantiza que los clientes tengan un mayor control sobre el rendimiento inicial de sus aplicaciones sin servidor a cualquier escala.

Excepto con una configuración de simultaneidad aprovisionada, los clientes no pueden controlar de manera determinista la cantidad de entornos de ejecución que Lambda crea o administra en respuesta a las invocaciones.

Rol de ejecución

Cada función de Lambda también debe configurarse con un rol de ejecución, que es un rol de IAM que asume el servicio Lambda al realizar operaciones del plano de control y el plano de datos relacionadas con la función. El servicio Lambda asume esta función para obtener credenciales de seguridad temporales que luego están disponibles como variables de entorno durante la invocación de una función. Por motivos de rendimiento, el servicio Lambda almacena en caché estas credenciales y puede reutilizarlas en diferentes entornos de ejecución que usan el mismo rol de ejecución.

Para garantizar el cumplimiento del principio de privilegios mínimos, Lambda recomienda que cada función tenga su propio rol único y que se configure con el conjunto mínimo de permisos que requiere.

El servicio Lambda también puede asumir el rol de ejecución para realizar determinadas operaciones del plano de control, como las relacionadas con la creación y configuración de interfaces de red elásticas (ENI) para funciones de VPC, el envío de registros a Amazon CloudWatch Application Insights, el envío de rastros a AWS X-Ray u otras operaciones no relacionadas con la invocación. Los clientes siempre pueden revisar y auditar estos casos de uso en los registros de auditoría que se encuentran en AWS CloudTrail.

Para obtener más información sobre este tema, consulte la página de documentación del rol de ejecución de AWS Lambda.

MicroVM y procesos de trabajo de Lambda

Lambda crea sus entornos de ejecución en una flota de instancias de Amazon EC2 denominadas procesos de trabajo AWS Lambda. Los procesos de trabajo son instancias Nitro EC2bare metal completas que Lambda lanza y administra en una cuenta de AWS aislada e independiente que no es visible para los clientes. Los procesos de trabajo tienen una o más micromáquinas virtuales (MVM) virtualizadas por hardware creadas por Firecracker. Firecracker es un monitor de máquina virtual (VMM) de código abierto que utiliza la máquina virtual basada en el kernel (KVM) de Linux para crear y administrar MVM. Está diseñado específicamente para crear y administrar contenedores multiinquilino seguros y servicios basados en funciones que proporcionan modelos operativos sin servidor. Para obtener más información sobre el modelo de seguridad de Firecracker, consulte el sitio web del proyecto Firecracker.

Como parte del modelo de responsabilidad compartida, Lambda es responsable de mantener la configuración de seguridad, los controles y el nivel de revisiones de los procesos de trabajo. El equipo de Lambda utiliza Amazon Inspector para detectar posibles problemas de seguridad conocidos, así como otros mecanismos de notificación de problemas de seguridad personalizados y listas de divulgación previa, de modo que los clientes no tengan que administrar la posición de seguridad subyacente de su entorno de ejecución.

Figura 3: Modelo de aislamiento para procesos de trabajo AWS Lambda

La concesión de los procesos de trabajo tiene una duración máxima de 14 horas. Cuando un proceso de trabajo está llegando al final de su tiempo máximo de concesión, no se le enrutan más invocaciones, las MVM se terminan correctamente y la instancia del proceso de trabajo subyacente se termina. Lambda supervisa y alerta continuamente sobre las actividades del ciclo de vida útil de su flota.

Todas las comunicaciones del plano de datos a los procesos de trabajo se cifran mediante el estándar de cifrado avanzado con Galois/Counter Mode (AES-GCM). Excepto con las operaciones del plano de datos, los clientes no pueden interactuar directamente con un proceso de trabajo, ya que está alojado en una Amazon VPC aislada en red administrada por Lambda en las cuentas de servicio de Lambda.

Cuando un proceso de trabajo tiene que crear un nuevo entorno de ejecución, se le da autorización durante un tiempo limitado para acceder a los artefactos de funciones del cliente. Estos artefactos están optimizados específicamente para el entorno de ejecución y los procesos de trabajo de Lambda. El código de función que se carga con el formato ZIP se optimiza una vez y, a continuación, se almacena en un formato cifrado con una clave administrada por AWS y AES-GCM.

También se optimizan las funciones que se cargan en Lambda con el formato de imagen de contenedor. La imagen del contenedor se descarga primero de su fuente original, se optimiza en fragmentos distintos y, a continuación, se almacena como fragmentos cifrados mediante un método de cifrado convergente autenticado que utiliza una combinación de AES-CTR, AES-GCM y SHA-256 MAC. El método de cifrado convergente permite a Lambda deduplicar de forma segura fragmentos cifrados. Todas las claves necesarias para descifrar los datos del cliente están protegidas mediante una AWS KMS clave maestra del cliente (CMK) administrada por el cliente. El uso de CMK por parte del servicio Lambda está disponible para los clientes en los registros de AWS CloudTrail para fines de seguimiento y auditoría.

Tecnologías de aislamiento de Lambda

Lambda utiliza una gran variedad de tecnologías de aislamiento de código abierto y patentadas para proteger los procesos de trabajo y los entornos de ejecución. Cada entorno de ejecución contiene una copia especializada de los siguientes elementos:

- El código de la versión de la función en particular
- Cualquier capa de AWS Lambda seleccionada para la versión de la función
- El tiempo de ejecución de la función elegido (por ejemplo, Java 11, NodeJS 12, Python 3.8, etc.) o el tiempo de ejecución personalizado de la función
- Un directorio /tmp en el que se puede escribir
- Un espacio de usuario de Linux mínimo basado en Amazon Linux 2

Los entornos de ejecución se aíslan unos de otros mediante varias tecnologías similares a contenedores que están integradas en el kernel de Linux, junto con tecnologías de aislamiento patentadas de AWS. Estas tecnologías incluyen:

- cgroups: se utiliza para restringir el acceso de la función a la CPU y a la memoria.
- espacios de nombres: cada entorno de ejecución se ejecuta en un espacio de nombres dedicado. Para ello, tenemos ID de procesos de grupo únicos, ID de usuario, interfaces de red y otros recursos administrados por el kernel de Linux.
- seccomp-bpf: para limitar las llamadas al sistema (syscalls) que se pueden usar desde el entorno de ejecución.
- iptables y tablas de enrutamiento: para evitar las comunicaciones de red de entrada y aislar las conexiones de red entre las MVM.
- chroot: proporciona acceso con ámbito al sistema de archivos subyacente.
- Configuración de Firecracker: se utiliza para limitar el rendimiento de los dispositivos de bloques y los dispositivos de red.
- Funciones de seguridad de Firecracker: para obtener más información sobre el diseño de seguridad actual de Firecracker, consulte el documento de diseño más reciente de Firecracker.

Junto con las tecnologías de aislamiento patentadas de AWS, estos mecanismos proporcionan un fuerte aislamiento entre los entornos de ejecución.

Almacenamiento y estado

Los entornos de ejecución nunca se reutilizan entre diferentes versiones de funciones o clientes, pero un solo entorno se puede reutilizar entre invocaciones de la misma versión de la función. Esto significa que los datos y el estado pueden persistir entre las invocaciones. Los datos o el estado pueden persistir durante horas antes de que se destruyan como parte de la administración del ciclo de vida normal del entorno de ejecución. Por motivos de rendimiento, las funciones pueden aprovechar este comportamiento para mejorar la eficiencia al mantener y reutilizar cachés locales o conexiones de larga duración entre invocaciones. Dentro de un entorno de ejecución, estas múltiples invocaciones se gestionan mediante un único proceso, por lo que cualquier estado que abarque todo el proceso (como un estado estático en Java) se puede reutilizar en las futuras invocaciones, si la invocación se produce en un entorno de ejecución reutilizado.

Cada entorno de ejecución de Lambda también incluye un sistema de archivos de escritura, disponible en `/tmp`. No es posible acceder a este almacenamiento ni compartirlo en todos los entornos de ejecución. Al igual que con el estado del proceso, los archivos que se escriben en `/tmp` se conservan durante toda la vida del entorno de ejecución. Esto permite amortizar las costosas operaciones de transferencia, como la descarga de modelos de machine learning (ML), en múltiples invocaciones. Las funciones que no desean conservar datos entre invocaciones no deben escribir en /tmp ni eliminar sus archivos de /tmp entre invocaciones. El directorio `/tmp` está respaldado por un almacén de instancias de Amazon EC2 y se cifra en reposo.

Los clientes que deseen conservar los datos en el sistema de archivos fuera del entorno de ejecución deben considerar la posibilidad de utilizar la integración de Lambda con Amazon Elastic File System (Amazon EFS). Para obtener más información, consulte Uso de Amazon EFS con AWS Lambda.

Si los clientes no desean conservar los datos o el estado entre invocaciones, Lambda recomienda que no utilicen el contexto de ejecución ni el entorno de ejecución para almacenar datos o estados. Si los clientes quieren evitar activamente la filtración de datos o estados entre invocaciones, Lambda recomienda que creen funciones distintas para cada estado. Lambda no recomienda que los clientes usen o almacenen un estado sensible a la seguridad en el entorno de ejecución, ya que puede mutar entre las invocaciones. En su lugar, recomendamos que se vuelva a calcular el estado en cada invocación.

Mantenimiento en tiempo de ejecución en Lambda

Lambda admite estos tiempos de ejecución mediante la búsqueda e implementación continua de actualizaciones y revisiones de seguridad compatibles, y mediante otras actividades de mantenimiento en tiempo de ejecución. Esto permite a los clientes centrarse solo en el mantenimiento y la seguridad de cualquier código que se incluya en su función y capa. El equipo de Lambda utiliza Amazon Inspector para detectar problemas de seguridad conocidos, así como otros mecanismos de notificación de problemas de seguridad personalizados y listas de divulgación previa para garantizar que nuestros lenguajes de tiempo de ejecución y entorno de ejecución siempre tengan las revisiones más recientes. Si se identifican nuevas revisiones o actualizaciones, Lambda prueba e implementa las actualizaciones en tiempo de ejecución sin que los clientes tengan que intervenir. Para obtener más información sobre el programa de cumplimiento de Lambda, consulte la sección “Lambda y cumplimiento” de este documento.

Por lo general, no se requiere ninguna acción para obtener las revisiones más recientes para los tiempos de ejecución de Lambda compatibles, pero a veces es posible que sea necesario hacer algo para probar las revisiones antes de implementarlas (por ejemplo, cuando hay revisiones de tiempo de ejecución incompatibles conocidas). Si los clientes deben realizar alguna acción, Lambda se pondrá en contacto con ellos a través de Personal Health Dashboard, a través del correo electrónico de la cuenta de AWS o por otros medios para informarles de las medidas específicas que deben tomarse.

Los clientes pueden usar otros lenguajes de programación en Lambda implementando un tiempo de ejecución personalizado. En el caso de que tengan tiempos de ejecución personalizados, el mantenimiento del tiempo de ejecución pasa a ser responsabilidad del cliente, lo que incluye asegurarse de que tenga las revisiones de seguridad más recientes. Para obtener más información, consulte Tiempos de ejecución personalizados de AWS Lambda en la Guía para desarrolladores de AWS Lambda.

Cuando los encargados del mantenimiento de un lenguaje en tiempo de ejecución principal anuncian el fin de vida (EOL) de su lenguaje, Lambda deja de admitir la versión del lenguaje del tiempo de ejecución. Cuando las versiones del tiempo de ejecución se marcan como obsoletas en Lambda, Lambda deja de permitir la creación de nuevas funciones y actualizaciones de funciones existentes que se crearon en el tiempo de ejecución obsoleto. Para avisar a los clientes de los tiempos de ejecución que se van a eliminar por estar obsoletos, Lambda les envía notificaciones sobre la próxima fecha de obsolescencia y les explica lo que va a ocurrir. Lambda no proporciona actualizaciones de seguridad, asistencia técnica ni correcciones para tiempos de ejecución

obsoletos, y se reserva el derecho de deshabilitar en cualquier momento las invocaciones de las funciones que están configuradas para ejecutarse en un motor de tiempo de ejecución obsoleto. Si los clientes quieren seguir ejecutando versiones de tiempos de ejecución obsoletas o incompatibles, pueden crear su propio tiempo de ejecución de AWS Lambda personalizado. Para obtener información sobre cuándo los tiempos de ejecución pasan a estar obsoletos, consulte la Política de compatibilidad de tiempos de ejecución de AWS Lambda.

Supervisión y auditoría de funciones de Lambda

Puede supervisar y auditar las funciones de Lambda con muchos servicios y métodos de AWS, incluidos los siguientes servicios.

Amazon CloudWatch

AWS Lambda supervisa automáticamente las funciones de Lambda en su nombre. A través de Amazon CloudWatch, ofrece métricas, como la cantidad de solicitudes, la duración de la ejecución por solicitud y la cantidad de solicitudes que provocan un error. Estas métricas se exponen a nivel de función, que luego puede aprovechar para configurar alarmas de CloudWatch. Para obtener una lista de las métricas expuestas por Lambda, consulte Métricas de AWS Lambda.

Amazon CloudTrail

Con AWS CloudTrail, puede implementar la gobernanza, el cumplimiento, la auditoría operativa y la auditoría de riesgos de toda su cuenta de AWS, incluido Lambda. CloudTrail le permite registrar, supervisar de forma continua y retener la actividad de la cuenta relacionada con acciones en su infraestructura de AWS, lo que proporciona un historial completo de eventos de las acciones que se realizan a través de Consola de administración de AWS, los SDK de AWS, las herramientas de línea de comandos y otros servicios de AWS. Con CloudTrail, tiene la opción de cifrar los archivos de registro con AWS KMS y también utilizar la validación de la integridad de los archivos de registro de CloudTrail para una aserción positiva.

AWS X-Ray

Con AWS X-Ray, puede analizar y depurar aplicaciones distribuidas y de producción basadas en Lambda, lo que le permite conocer el rendimiento de su aplicación y sus servicios subyacentes, para que pueda identificar y solucionar la causa raíz de los problemas de rendimiento y errores. La vista integral de X-Ray de las solicitudes mientras se desplazan por su aplicación muestra un mapa de los componentes subyacentes de la aplicación, para que pueda analizar las aplicaciones durante el desarrollo y la producción.

AWS Config

Con AWS Config, puede realizar un seguimiento de los cambios de configuración en las funciones de Lambda (incluidas las funciones eliminadas), los entornos en tiempo de ejecución, las etiquetas, el nombre del controlador, el tamaño del código, la asignación de memoria, la configuración de tiempo de espera y la configuración de simultaneidad, junto con el rol de ejecución de Lambda de IAM, la subred y las asociaciones de grupos de seguridad. De esta manera, tiene una visión holística del ciclo de vida de la función de Lambda y le permite mostrar esos datos para los posibles requisitos de auditoría y cumplimiento.

Diseño y funcionamiento de las funciones de Lambda

En esta sección, se analizan la arquitectura y las operaciones de Lambda. Para obtener información sobre las prácticas recomendadas estándar de las aplicaciones sin servidor, consulte el documento técnico Enfoque de aplicaciones sin servidor del AWS Well-Architected Framework, en el que se definen y examinan los pilares de AWS Well Architected Framework en un contexto sin servidor.

- El pilar de la excelencia operativa: la capacidad de ejecutar y supervisar sistemas para ofrecer valor empresarial y mejorar continuamente los procedimientos y los procesos de apoyo.
- El pilar de la seguridad: la capacidad de proteger la información, los sistemas y los recursos, mientras que se ofrece valor empresarial mediante evaluaciones de riesgos y estrategias de mitigación.
- El pilar de la fiabilidad: la capacidad de un sistema para recuperarse de las interrupciones de la infraestructura o del servicio, adquirir dinámicamente recursos de computación para satisfacer la demanda y mitigar interrupciones como las configuraciones incorrectas o los problemas transitorios de red.
- El pilar de la eficiencia del rendimiento: se centra en el uso eficiente de los recursos de computación para satisfacer los requisitos y el mantenimiento de dicha eficiencia a medida que la demanda cambia y las tecnologías evolucionan.
- El pilar de la optimización de costes: el proceso continuo de refinamiento y mejora para garantizar que se logren los resultados empresariales y, al mismo tiempo, minimizar los costes a medida que la demanda cambia y las tecnologías evolucionan.

El documento técnico Enfoque de aplicaciones sin servidor del AWS Well-Architected Framework incluye temas como el registro de métricas y alarmas, las limitaciones y los límites, la asignación de permisos a las funciones de Lambda y la puesta a disposición de la información confidencial para las funciones de Lambda.

Lambda y cumplimiento

Como explicamos en la sección “Modelo de responsabilidad compartida”, usted es responsable de determinar qué régimen de cumplimiento se aplica a sus datos. Una vez que haya determinado las necesidades de su plan de cumplimiento, puede utilizar las diversas características de Lambda para esos controles. Puede ponerse en contacto con expertos de AWS (por ejemplo, arquitectos de soluciones, expertos en dominios, administradores de cuentas técnicas y otros recursos humanos) para obtener ayuda. Sin embargo, AWS no puede asesorar a los clientes sobre si (o qué) planes de cumplimiento se aplican a un caso de uso concreto.

A partir de noviembre de 2020, Lambda cubre los informes SOC 1, SOC 2 y SOC 3, que son informes de examen independientes de terceros que demuestran cómo AWS logra los controles y objetivos clave de cumplimiento. Para obtener una lista actualizada de información de conformidad, consulte la página Servicios de AWS en el ámbito del programa de conformidad.

Debido a la naturaleza confidencial de algunos informes de cumplimiento, estos no se pueden hacerse públicos. Para acceder a ellos, puede iniciar sesión en Consola de administración de AWS y utilizar AWS Artifact, un portal de autoservicio gratuito para obtener acceso bajo demanda a los informes de cumplimiento de AWS.

Fuentes de eventos de Lambda

Lambda se integra con más de 140 servicios de AWS mediante la integración directa y el bus de eventos de Amazon EventBridge. Las fuentes de eventos de Lambda más utilizadas son:

- Amazon API Gateway
- Amazon CloudWatch Events
- Amazon CloudWatch Logs
- Amazon DynamoDB Streams
- Amazon EventBridge
- Amazon Kinesis Data Streams
- Amazon S3
- Amazon SNS
- Amazon SQS
- AWS Step Functions

Con estas fuentes de eventos, puede:

- Utilizar AWS Identity and Access Management para administrar el acceso al servicio y los recursos de forma segura.
- Cifrar sus datos en reposo.* Todos los servicios cifran los datos en tránsito.
- Acceder desde su Amazon Virtual Private Cloud mediante puntos de conexión de VPC (que utilizan AWS PrivateLink).
- Utilizar Amazon CloudWatch Application Insights para recopilar, generar informes y generar alarmas sobre las métricas.
- Utilizar AWS CloudTrail para registrar, supervisar de forma continua y retener la actividad de la cuenta relacionada con acciones en su infraestructura de AWS, para proporcionar un historial de eventos completo de las acciones realizadas a través de los Consola de administración de AWSSDK de AWS, las herramientas de la línea de comandos y otros servicios de AWS.

*En el momento de publicar este documento, el cifrado de los datos en reposo no estaba disponible para Amazon EventBridge. Siga visitando las páginas de inicio del servicio para conocer las novedades sobre estas capacidades.

Conclusión

AWS Lambda ofrece un conjunto de herramientas muy útil para crear aplicaciones seguras y escalables. Muchas de las prácticas recomendadas sobre seguridad y cumplimiento en Lambda son las mismas que en todos los servicios de AWS, pero algunas son específicas de Lambda. En este documento técnico, se describen los beneficios de Lambda, su idoneidad para las aplicaciones y el entorno en tiempo de ejecución administrado por Lambda. También incluye información sobre la supervisión y la auditoría, además de las prácticas recomendadas de seguridad y cumplimiento. Cuando piense en su próxima implementación, tenga en cuenta lo que ha aprendido sobre AWS Lambda y cómo podría mejorar su próxima solución de carga de trabajo.

Colaboradores

Entre los colaboradores de este documento, están las siguientes personas:

- Mayank Thakkar, arquitecto de soluciones globales para ciencias biológicas
- Marc Brooker, ingeniero jefe sénior (sin servidor)
- Osman Surkatty, ingeniero sénior de seguridad (sin servidor)

Documentación adicional

Para obtener información adicional, consulte la siguiente documentación:

- Modelo de responsabilidad compartida, que explica cómo se concibe en AWS la seguridad en general.
- Prácticas recomendadas de seguridad en AWS, que incluye recomendaciones para el servicio AWS Identity and Access Management (IAM).
- Enfoque de aplicaciones sin servidor del AWS Well-Architected Framework, que trata sobre AWS Well-Architected Framework e identifica los elementos clave para garantizar que sus cargas de trabajo se diseñen de acuerdo con las prácticas recomendadas.
- Introducción a la seguridad de AWS, que proporciona una amplia introducción al planteamiento de la seguridad en AWS.
- Riesgo y conformidad de AWS, que incluye información general sobre el cumplimiento en AWS.

Revisiones del documento

Para recibir notificaciones sobre las actualizaciones de este documento técnico, suscríbase a la fuente RSS.

update-history-change	update-history-description	update-history-date
Actualizado	Actualizaciones importantes	15 de febrero de 2021
Publicación inicial	Primera publicación del documento técnico	3 de enero de 2019

Avisos

Los clientes son responsables de realizar sus propias evaluaciones de la información contenida en este documento. Este documento: (a) solo tiene fines informativos, (b) representa las prácticas y las ofertas de productos vigentes de AWS, que están sujetas a cambios sin previo aviso, y (c) no crea ningún compromiso ni garantía de AWS y sus empresas afiliadas, proveedores o concesionarios de licencias. Los productos o servicios de AWS se proporcionan “tal cual”, sin garantías, representaciones ni condiciones de ningún tipo, ya sean explícitas o implícitas. Las responsabilidades y obligaciones de AWS en relación con sus clientes se rigen por los acuerdos de AWS, y este documento no modifica ni forma parte de ningún acuerdo entre AWS y sus clientes.

© 2021 Amazon Web Services, Inc. o sus empresas afiliadas. Todos los derechos reservados.