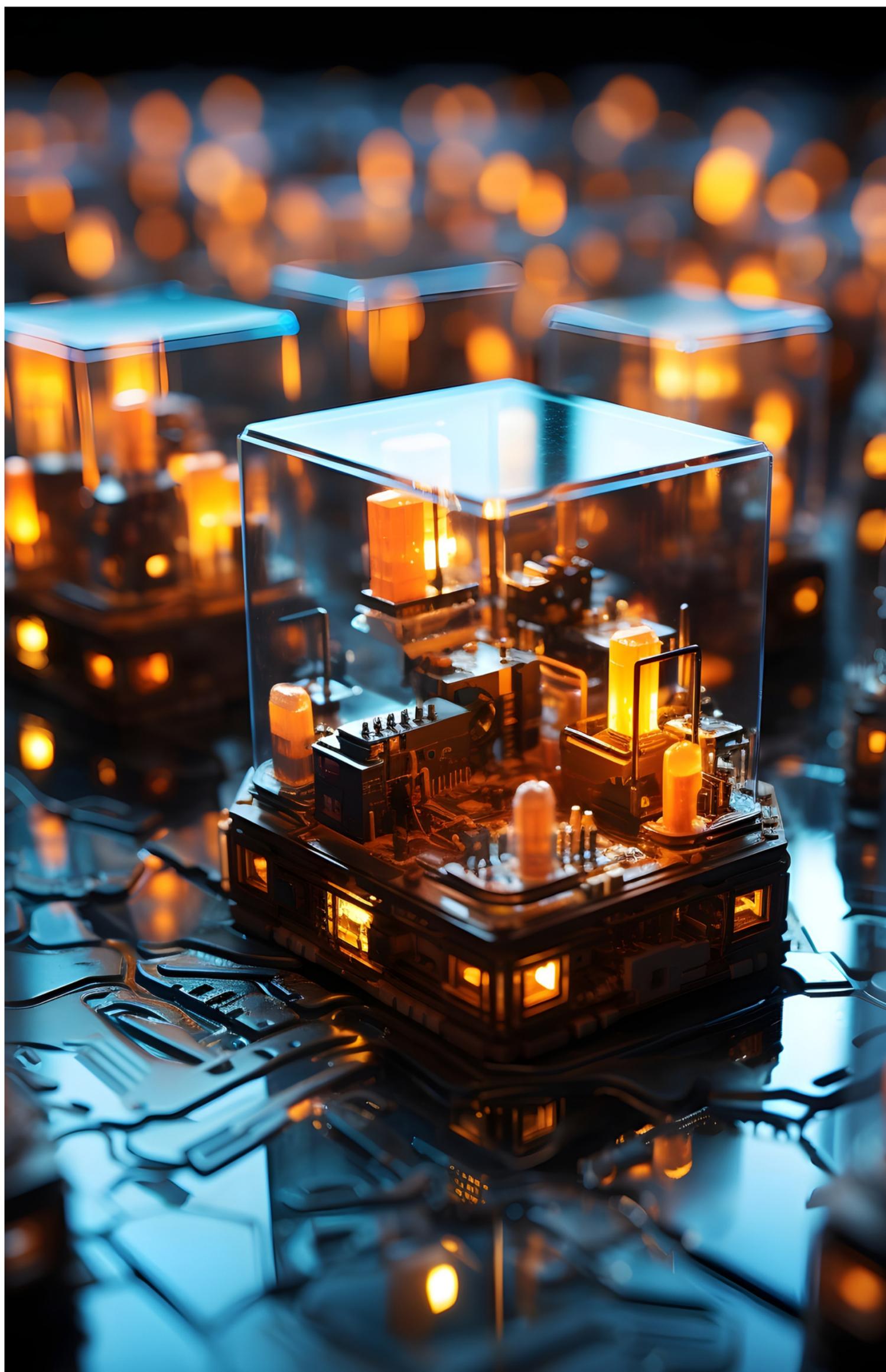
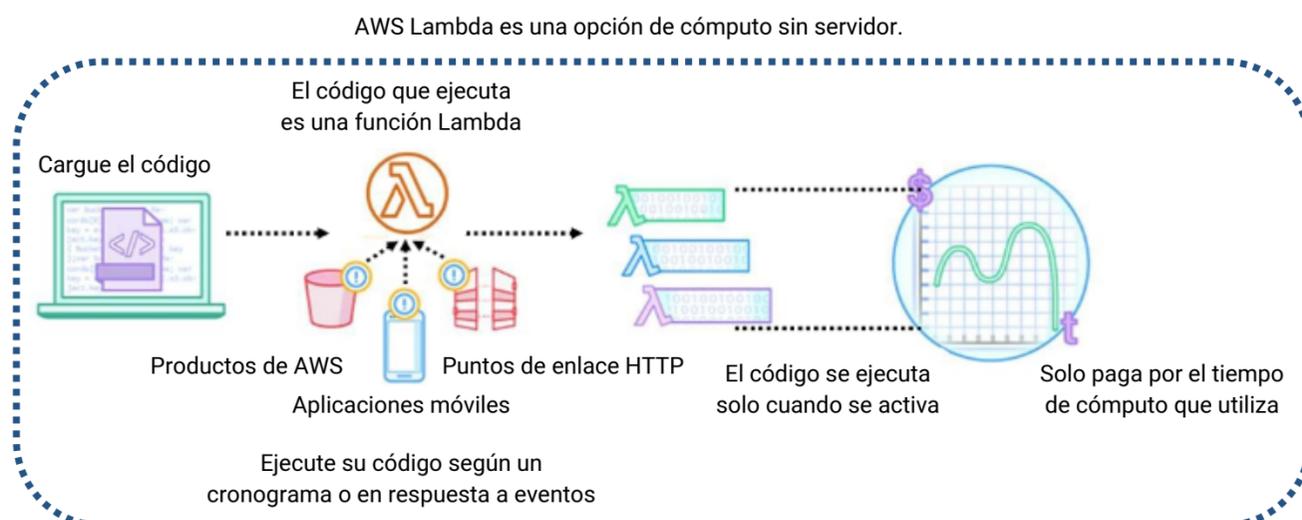


Lección 5

Introducción a AWS Lambda e Introducción a AWS Elastic Beanstalk



AWS Lambda: ejecutar código sin servidores



Como se vio en las lecciones anteriores de esta unidad, AWS ofrece muchas opciones de cómputo. Por ejemplo, Amazon EC2 proporciona máquinas virtuales. Como otro ejemplo, Amazon ECS y Amazon EKS son servicios de cómputo basados en contenedores.

Sin embargo, existe otro enfoque del cómputo, que no requiere aprovisionar ni administrar servidores. Este tercer enfoque se conoce a menudo como cómputo sin servidor.

AWS Lambda es un servicio de cómputo sin servidor que se basa en eventos. Lambda le permite ejecutar código sin necesidad de aprovisionar ni administrar servidores.

Crea una función Lambda, que es el recurso de AWS que contiene el código que carga. Luego, configura la función Lambda para que se active, ya sea de forma programada o en respuesta a un evento. El código se ejecuta solo cuando se activa.

Solo pagará por el tiempo de cómputo que consuma: no se incurre en cargos cuando el código no se está en ejecución.

1010

Beneficios de Lambda



- Es compatible con varios lenguajes de programación
- Administración automatizada por completo
- Tolerancia a fallas integrada
- Admite la coordinación de múltiples funciones
- Precios de pago por uso

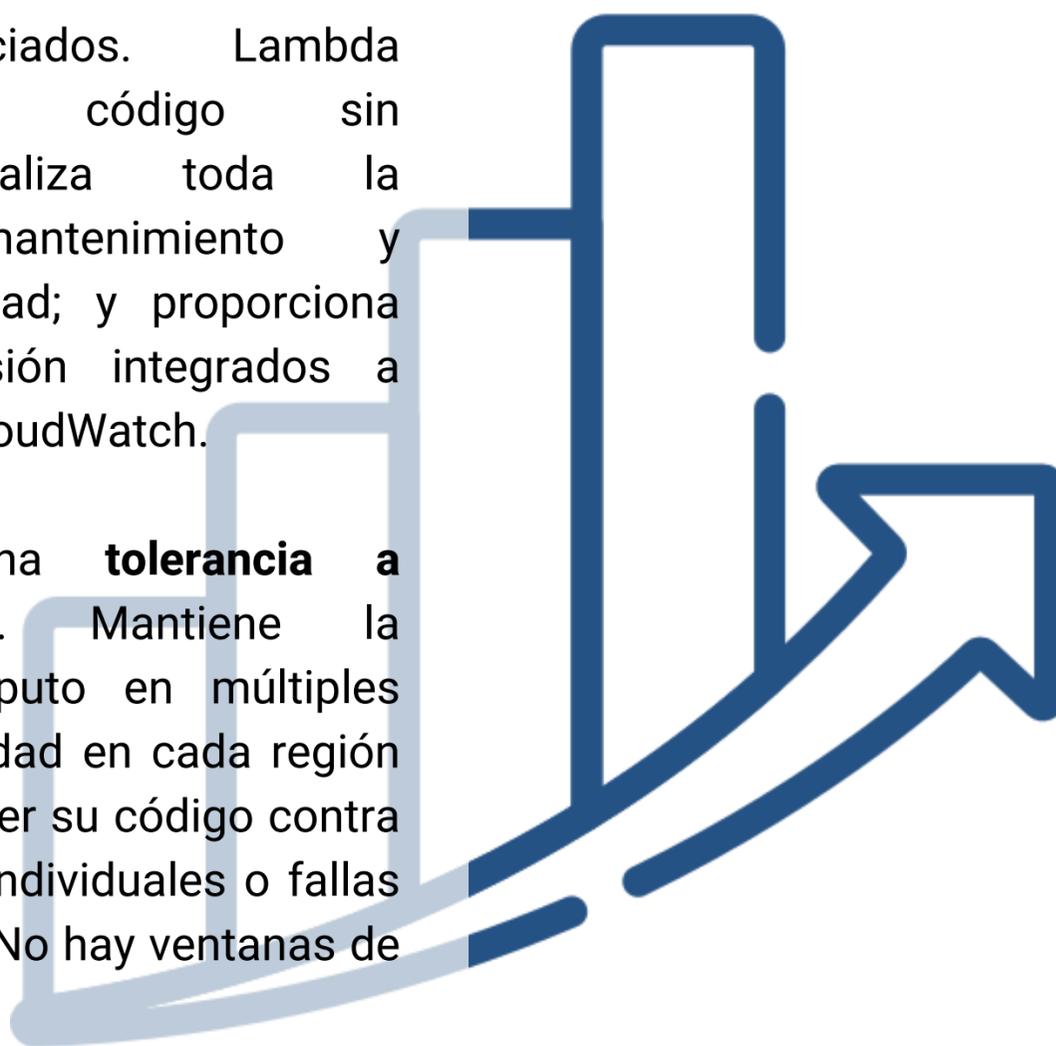
AWS Lambda

Lambda automatiza completamente la administración. Administra toda la infraestructura para ejecutar su código en una infraestructura altamente disponible y tolerante a fallas, lo que le permite concentrarse en crear servicios backend diferenciados. Lambda implementa su código sin interrupciones; realiza toda la administración, mantenimiento y parches de seguridad; y proporciona registro y supervisión integrados a través de Amazon CloudWatch.

Lambda proporciona **tolerancia a errores integrada**. Mantiene la capacidad de cómputo en múltiples zonas de disponibilidad en cada región para ayudar a proteger su código contra fallas de máquinas individuales o fallas del centro de datos. No hay ventanas de mantenimiento ni tiempos de inactividad programados.

Puede coordinar varias funciones de Lambda para tareas complejas o de larga duración mediante la creación de flujos de trabajo con AWS Step Functions. Utilice funciones de pasos para definir flujos de trabajo. Estos flujos de trabajo desencadenan una colección de funciones Lambda mediante el uso de pasos secuenciales, paralelos, de bifurcación y de manejo de errores. Con Step Functions y Lambda, puede crear procesos con estado y de larga duración para aplicaciones y backends.

Con AWS Lambda, no es necesario aprender idiomas, herramientas ni marcos de trabajo nuevos. Lambda admite múltiples lenguajes de programación, entre ellos Java, Go, PowerShell, Node.js, C#, Python y Ruby. Su código puede utilizar cualquier biblioteca, ya sea nativa o de terceros.

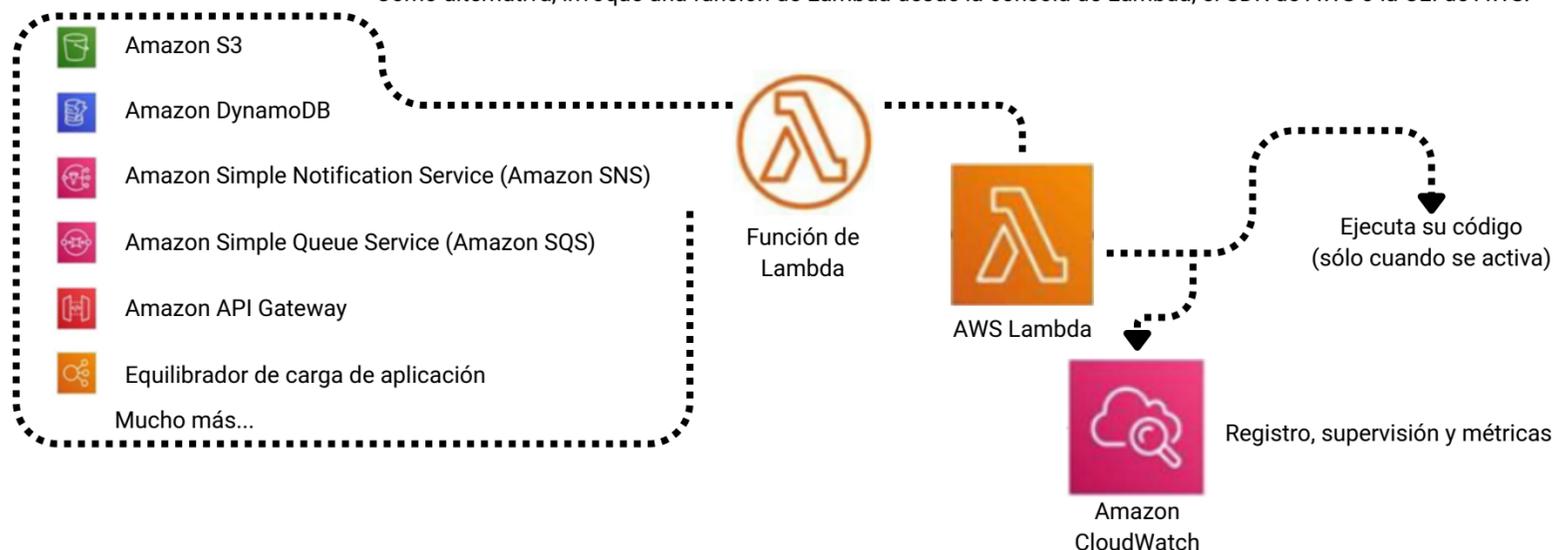


Con Lambda, se paga solo por las solicitudes atendidas y el tiempo de cómputo necesario para ejecutar su código. La facturación se mide en incrementos de 100 milisegundos, lo que la hace rentable y fácil de escalar automáticamente desde unas pocas solicitudes diarias hasta miles de solicitudes por segundo.

Fuentes de eventos de AWS Lambda

Configure otros AWS como orígenes de eventos para invocar su función como se muestra aquí.

Como alternativa, invoque una función de Lambda desde la consola de Lambda, el SDK de AWS o la CLI de AWS.



Una fuente de eventos es un servicio de AWS o una aplicación creada por el desarrollador que desencadena una función de AWS Lambda para que se ejecute.

Algunos servicios publican eventos en Lambda invocando la función Lambda directamente. Estos servicios que invocan funciones Lambda de forma asincrónica incluyen, entre otros, Amazon S3, Amazon Simple Notification Service (Amazon SNS) y Amazon CloudWatch Events.

Puede invocar funciones de Lambda directamente con la consola de Lambda, la API de Lambda, el kit de desarrollo de software (SDK) de AWS, la CLI de AWS y los conjuntos de herramientas de AWS. El enfoque de invocación directa puede resultar útil, por ejemplo, cuando está desarrollando una aplicación móvil y desea que la aplicación llame a funciones Lambda.

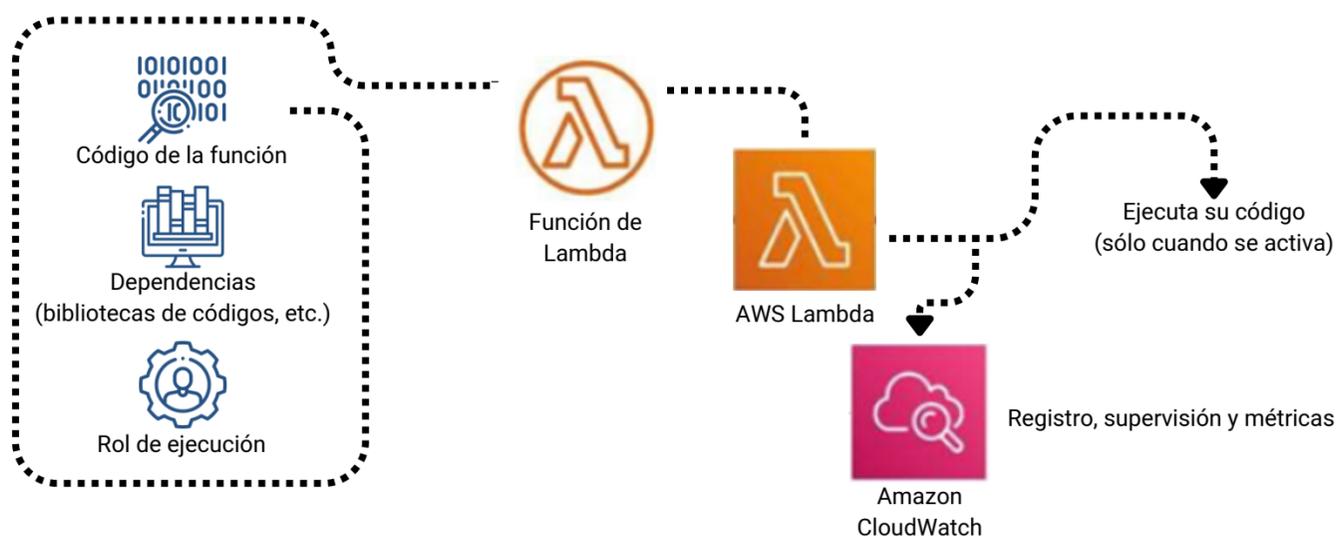
Lambda también puede sondear recursos en otros servicios que no publican eventos en Lambda. Por ejemplo, Lambda puede extraer registros de una cola de Amazon Simple Queue Service (Amazon SQS) y ejecutar una función Lambda para cada mensaje obtenido. Lambda puede leer eventos de Amazon DynamoDB.

Algunos servicios, como Elastic Load Balancing (Application Load Balancer) y Amazon API Gateway, pueden **invocar su función Lambda directamente**.

Para obtener más detalles sobre todos los servicios compatibles, consulte la documentación sobre el uso de Lambda con otros servicios en <https://docs.aws.amazon.com/lambda/latest/dg/lambda-services.html>.

AWS Lambda supervisa automáticamente las funciones de Lambda a través de Amazon CloudWatch. Para ayudarlo a solucionar los errores de una función, Lambda registra todas las solicitudes gestionadas por la función. Lambda también almacena automáticamente los registros generados por el código a través de Amazon CloudWatch Logs.

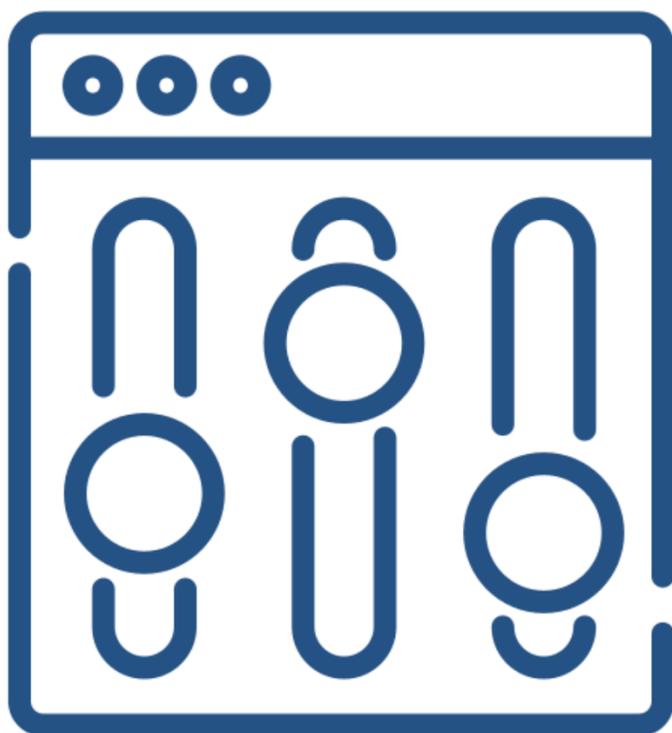
Configuración de la función de AWS Lambda



Recuerde que una función Lambda es el código personalizado que escribe para procesar eventos y que Lambda ejecuta la función Lambda en su nombre.

Al utilizar la Consola de administración de AWS para crear una función Lambda, primero otórguele un nombre a la función. Luego, lo que usted especifica:

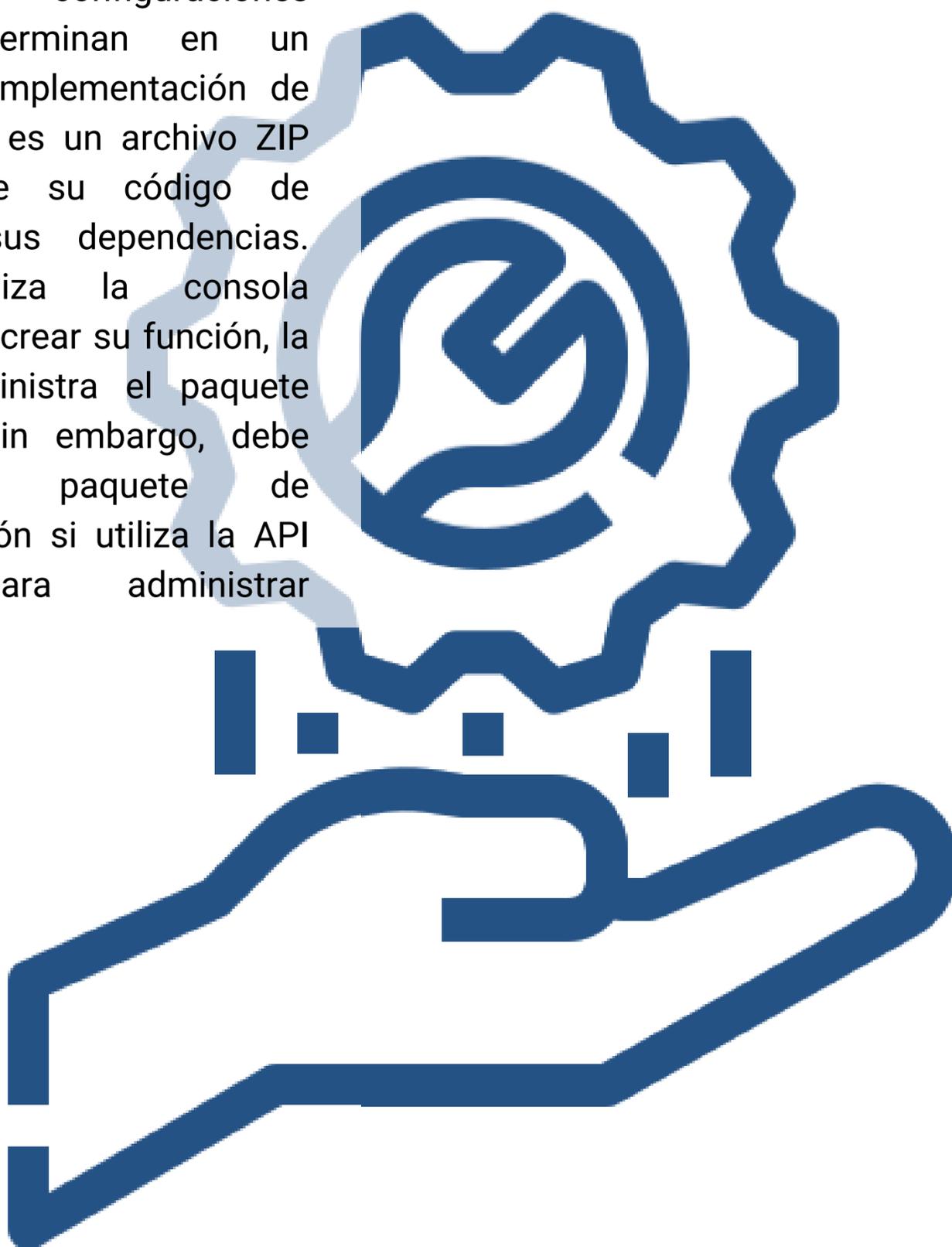
- El entorno en tiempo de ejecución que utilizará la función (por ejemplo, una versión de Python o Node.js)
- Un rol de ejecución (para otorgar permiso de IAM a la función para que pueda interactuar con otros AWS según sea necesario)



Lo siguiente, después de hacer clic en Crear función, configure la función. Las configuraciones incluyen:

- Agregue un desencadenador (especifique uno de las fuentes de eventos disponibles en la diapositiva anterior)
- Agregue su código de función (utilice el editor de código proporcionado o cargue un archivo que contenga su código)
- Especifique la memoria en MB para asignar a su función (128 MB a 10 240 MB)
- De manera opcional, especifique las variables de entorno, la descripción, el tiempo de espera, la nube privada virtual (VPC) específica para ejecutar la función, las etiquetas que le gustaría usar y otras configuraciones. Para obtener más información, consulte Configuración de funciones en la consola de AWS Lambda <https://docs.aws.amazon.com/lambda/latest/dg/configuration-console.html> en la documentación de AWS.

Todas las configuraciones anteriores terminan en un paquete de implementación de Lambda, que es un archivo ZIP que contiene su código de función y sus dependencias. Cuando utiliza la consola Lambda para crear su función, la consola administra el paquete por usted. Sin embargo, debe crear un paquete de implementación si utiliza la API Lambda para administrar funciones.



Ejemplo de función de Lambda basada en programación: Iniciar y detener instancias EC2

Ejemplo de detener las instancias



Ejemplo de iniciar las instancias



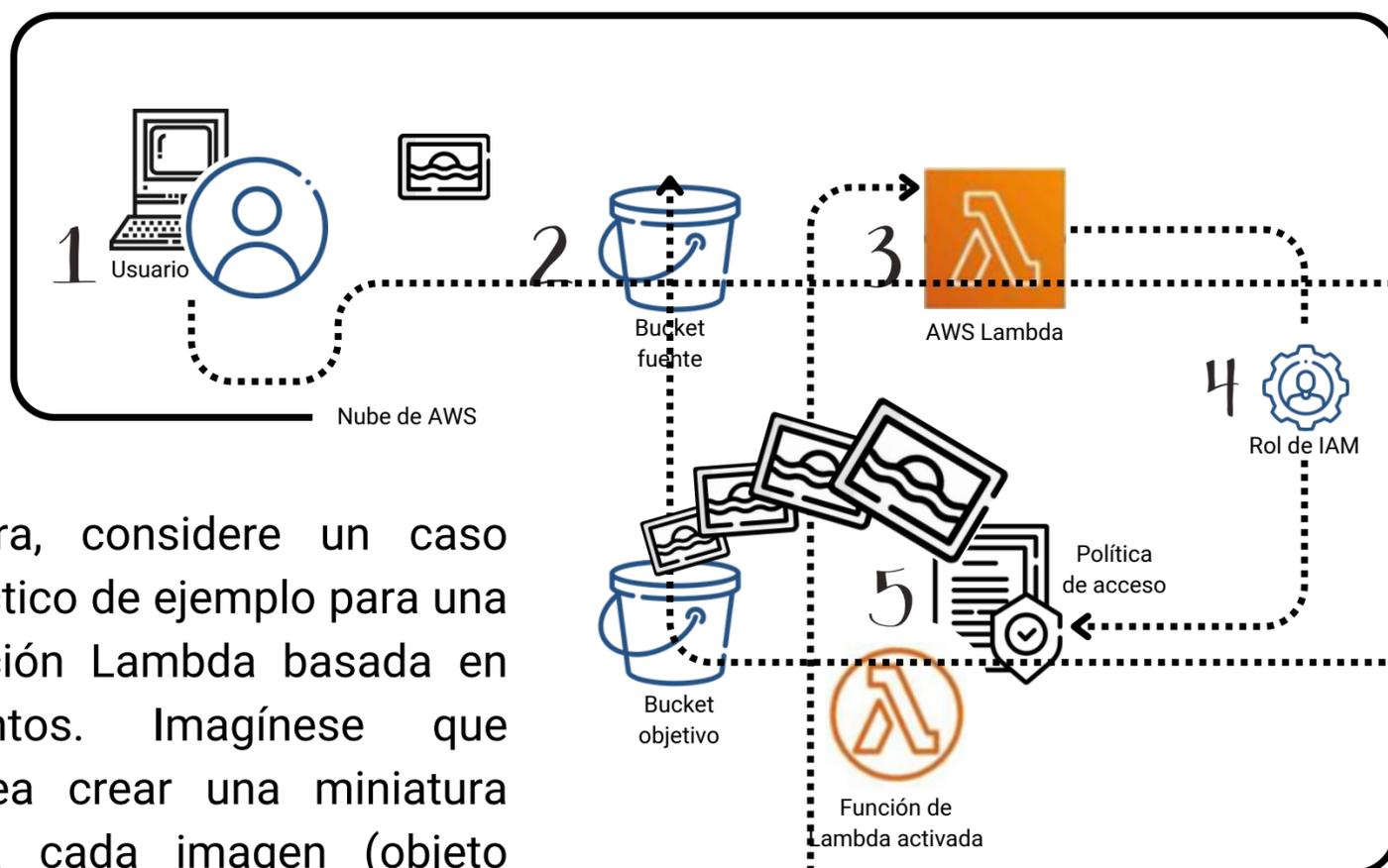
Considere un caso práctico de ejemplo para una función Lambda basada en programación. Supongamos que se encuentra en una situación en la que desea reducir el uso de Amazon EC2. Usted decide que desea detener las instancias a una hora predefinida (por ejemplo, por la noche, cuando nadie accede a ellas) y luego desea reiniciar las instancias por la mañana (antes de iniciar la jornada laboral).

En esta situación, puede configurar **AWS Lambda** y Amazon CloudWatch Events para realizar estas acciones automáticamente.

El ejemplo muestra lo que sucede en cada paso:

1. Está programado un evento de CloudWatch para ejecutar una función Lambda para detener sus instancias de EC2 (por ejemplo) a las 22:00 GMT.
2. La función de Lambda se activa y se ejecuta con el rol de IAM que le otorga permisos para detener las instancias de EC2.
3. Las instancias de Amazon EC2 entran en estado detenido.
4. Más tarde, a las (por ejemplo) 05:00 a. m. UTC, se programa un evento de CloudWatch Events para ejecutar una función de Lambda para iniciar las instancias de EC2.
5. La función de Lambda se activa y se ejecuta con el rol de IAM que le otorga permisos para iniciar las instancias de EC2.
6. Las instancias de Amazon EC2 entran en estado en ejecución.

Ejemplo de función de Lambda basada en eventos: Crear imágenes en miniatura



Ahora, considere un caso práctico de ejemplo para una función Lambda basada en eventos. Imagínese que desea crear una miniatura para cada imagen (objeto .jpg o .png) que se carga en un bucket de S3.

Para crear una solución, puede crear una función Lambda que Amazon S3 invoca cuando se cargan objetos. Luego, la función Lambda lee el objeto de imagen del bucket de origen y crea una imagen en miniatura en un bucket de destino. Funciona de esta manera:

1. El usuario carga un objeto en el bucket de origen de Amazon S3 (evento object-created).
2. Amazon S3 detecta el evento object-created.
3. Amazon S3 publica el evento object-created en Lambda mediante la invocación de la función de Lambda y el pase de los datos del evento.
4. Lambda ejecuta la función Lambda asumiendo el rol de ejecución que especificó cuando creó la función Lambda.
5. A partir de los datos del evento que recibe, la función de Lambda conoce el nombre del bucket de origen y el nombre de la clave del objeto. La función de Lambda lee el objeto y crea una miniatura con las bibliotecas de gráficos y la guarda en el bucket de destino.

Cuotas de AWS Lambda

Límites suaves por región:

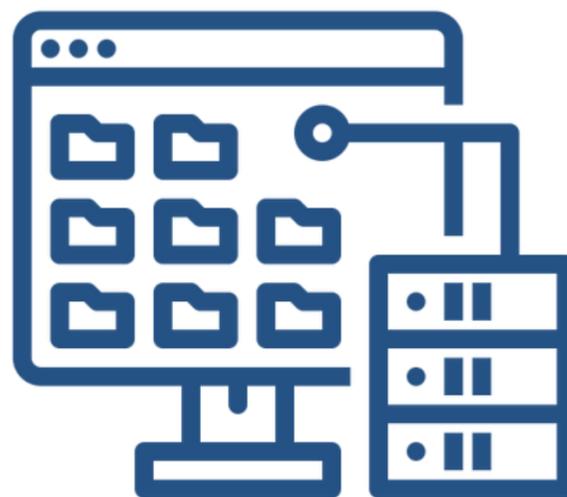
- Ejecuciones simultáneas = 1.000
- Almacenamiento de funciones y capas = 75 GB

Límites estrictos para funciones individuales:

- Asignación máxima de memoria de función = 10 240 MB
- Tiempo de espera de la función = 15 minutos
- Tamaño del paquete de implementación = 250 MB descomprimidos, incluidas las capas
- Tamaño del paquete del código de imagen del contenedor = 10 GB

También existen límites adicionales. Los detalles se encuentran en la documentación de cuotas de AWS Lambda en

<https://docs.aws.amazon.com/lambda/latest/dg/gettingstarted-limits.html>.



AWS Lambda tiene algunas cuotas que debe conocer al crear e implementar funciones Lambda.

AWS Lambda limita la cantidad de recursos de cómputo y almacenamiento que puede usar para ejecutar y almacenar funciones. Por ejemplo, en el momento de redactar este documento, el máximo de asignación de memoria para una sola función de Lambda es de 10,240 GB. También tiene límites de 1.000 ejecuciones simultáneas en una Región. Las funciones Lambda se pueden configurar para que se ejecuten hasta 15 minutos por ejecución. Puede establecer el tiempo de espera en cualquier valor entre 1 segundo y 15 minutos. Si solucionará problemas de una implementación de Lambda, tenga en cuenta estos límites.

Existen límites en el tamaño del paquete de implementación de una función (250 MB). Una capa es un archivo ZIP que contiene bibliotecas, un tiempo de ejecución personalizado u otras dependencias. Con las capas, puede utilizar las bibliotecas en la función sin necesidad de incluirlas en el paquete de implementación. El uso de capas puede ayudar a evitar alcanzar el límite de tamaño del paquete de implementación. Las capas también son una buena forma de compartir código y datos entre funciones Lambda.

Para cargas de trabajo más grandes que dependen de dependencias considerables, como machine learning o cargas de trabajo con uso intensivo de datos, puede implementar su función Lambda en una imagen de contenedor de hasta 10 GB de tamaño.

Los límites son suaves o duros. Los límites suaves de una cuenta se pueden suavizar potencialmente mediante el envío de un ticket de soporte y proporcionando una justificación de la solicitud. Los límites duros no se pueden aumentar.

Los detalles se encuentran en la documentación de cuotas de AWS Lambda en <https://docs.aws.amazon.com/lambda/latest/dg/gettingstarted-limits.html>.

Entre los aprendizajes clave de esta lección de la unidad, se incluyen los siguientes:

- El cómputo sin servidor le permite crear y ejecutar aplicaciones y servicios sin aprovisionar ni administrar servidores.
- AWS Lambda es un servicio de cómputo sin servidor que proporciona tolerancia a errores integrada y escalado automático.
- Una fuente de eventos es un servicio de AWS o una aplicación creada por el desarrollador que desencadena una función de Lambda para que se ejecute.
- El máximo de asignación de memoria para una sola función de Lambda es de 10,240 GB.
- El máximo de tiempo de ejecución para una función de Lambda es de 15 minutos.

AWS Elastic Beanstalk



Una forma sencilla de poner aplicaciones web en funcionamiento
Un servicio administrado que maneja automáticamente:

- Administración de aprovisionamiento y configuración
- implementación
- Balanceo de carga
- Escalado automático
- monitoreo de la salud
- Análisis y depuración
- Registros

No se aplican cargos adicionales por utilizar Elastic Beanstalk

- Solo paga por los servicios subyacentes que se utilizan

AWS Elastic Beanstalk

Mantenga el control. Una vez desarrollada toda la plataforma, lo único que tiene que hacer es cargar el código. Elija su tipo de instancia, su base de datos, configure y ajuste el escalado automático, actualice la aplicación, acceda a los archivos de registro del servidor y habilite HTTP en el equilibrador de carga.

Tiene que cargar el código y Elastic Beanstalk administrará de manera automática la implementación, desde el aprovisionamiento de la capacidad, el equilibrio de carga y el escalado automático hasta la monitorización del estado de la aplicación. Además, tiene el control absoluto de los recursos de AWS que alimentan la aplicación y puede acceder a los recursos subyacentes en cualquier momento.

No se aplican cargos adicionales por utilizar AWS Elastic Beanstalk. Paga por los recursos de AWS (por ejemplo, instancias de EC2 o buckets de S3) que crea para almacenar y ejecutar su aplicación. Solo paga por lo que usa, cuando lo usa. No hay tarifas mínimas ni compromisos iniciales.

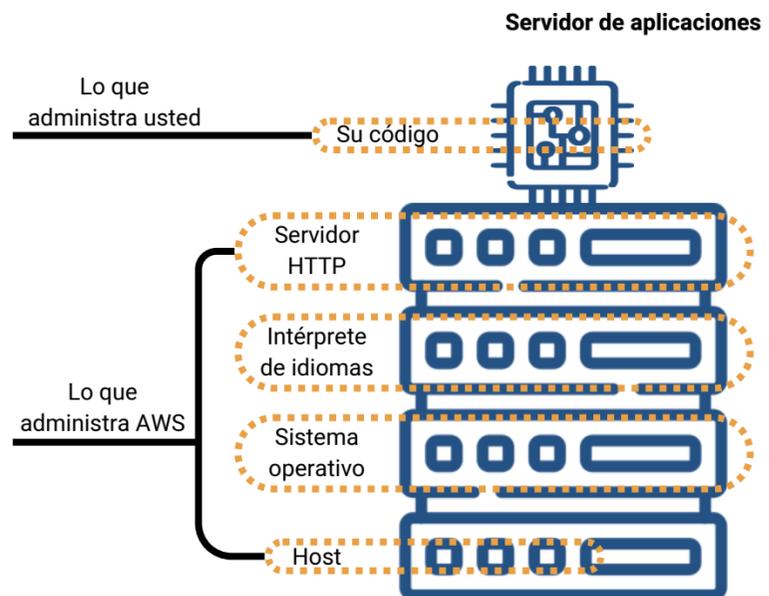
AWS Elastic Beanstalk es otra opción de servicio de cómputo. Elastic Beanstalk es una plataforma como servicio (PaaS) que facilita la rápida implementación, el escalado y la administración de las aplicaciones web y servicios.

Implementación de AWS Elastic Beanstalk

Admite aplicaciones web escritas para plataformas comunes, Java, .NET, PHP, Node.js, Python, Ruby, Go y Docker

Cargue su código

- Elastic Beanstalk maneja automáticamente la implementación
- Se desarrollan en servidores tales como Apache, NGINX, Passenger, Puma y Microsoft Internet Information Services (IIS).
 - Information Services (IIS).



AWS Elastic Beanstalk le permite implementar su código a través de la Consola de administración de AWS, la interfaz de línea de comandos de AWS (AWS CLI), Visual Studio y Eclipse. Elastic Beanstalk ofrece todos los servicios de aplicaciones que necesita para la aplicación. Solo tiene que crear es tu código. Elastic Beanstalk está diseñado para hacer que la implementación de su aplicación sea un proceso rápido y sencillo.

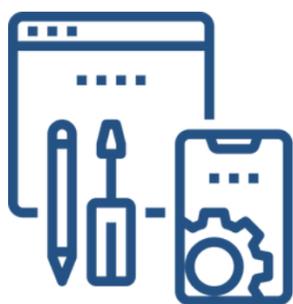
Elastic Beanstalk admite una gran variedad de plataformas. Las plataformas compatibles incluyen Docker, Go, Java, .NET, Node.js, PHP, Python y Ruby.

AWS Elastic Beanstalk implementa su código en **Apache Tomcat** para aplicaciones Java; **Servidor HTTP Apache** para aplicaciones PHP y Python; NGINX o Apache HTTP Server para aplicaciones Node.js; **Passenger** o **Puma** para aplicaciones Ruby; y **Microsoft Internet Information Services (IIS)** para aplicaciones .NET, Java SE, Docker y Go.

Beneficios de AWS Elastic Beanstalk



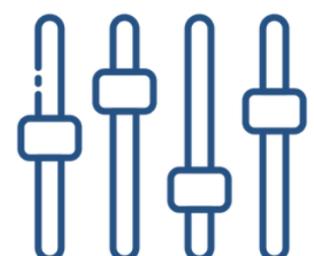
Rápido y sencillo de empezar a usar



Productividad para los desarrolladores



Difícil de optimizar



Control completo de recursos

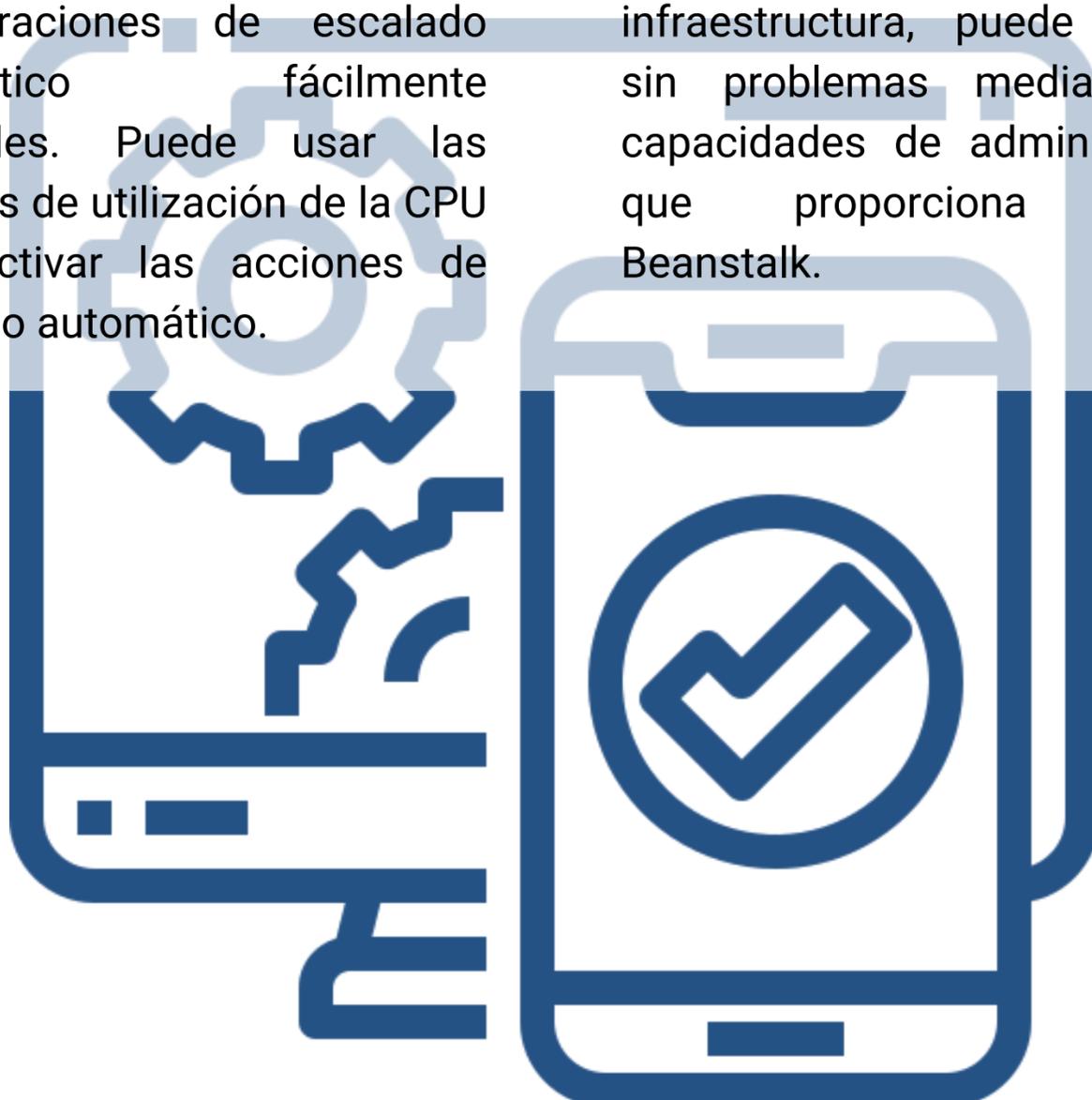
Elastic Beanstalk es rápido y sencillo de empezar a utilizar. Puede utilizar la consola de administración de AWS, un repositorio Git o un entorno de desarrollo integrado (IDE) tales como Apache Eclipse o Microsoft Visual Studio para cargar la aplicación. Elastic Beanstalk gestiona de forma automática los detalles de implementación de aprovisionamiento de capacidad, balanceo de carga, escalado automático y monitoreo del estado de las aplicaciones.

Puede mejorar su **productividad del desarrollador** al enfocarse en escribir código en lugar de administrar y configurar servidores, bases de datos, balanceadores de carga, firewalls y redes. AWS actualiza la plataforma subyacente que ejecuta su aplicación con parches y actualizaciones.

Elastic Beanstalk es difícil de superar. Con Elastic Beanstalk, la aplicación puede manejar las cargas máximas de trabajo o tráfico y usted puede minimizar los costos al mismo tiempo. Escala de forma automática la aplicación de forma ascendente o descendente, en función de las necesidades de la aplicación, mediante el uso de configuraciones de escalado automático fácilmente ajustables. Puede usar las métricas de utilización de la CPU para activar las acciones de escalado automático.

Tiene la libertad de seleccionar los recursos de AWS (tales como el tipo de instancia de Amazon EC2) que sean óptimos para su aplicación.

Elastic Beanstalk le permite mantener el control total de los recursos de AWS que alimentan a su aplicación. Si decide que desea hacerse cargo de algunos (o todos) los elementos de su infraestructura, puede hacerlo sin problemas mediante las capacidades de administración que proporciona Elastic Beanstalk.



Entre los aprendizajes clave de esta lección de la unidad, se incluyen los siguientes:

- AWS Elastic Beanstalk mejora la productividad de los desarrolladores.
 - Simplifica el proceso de implementación de su aplicación.
 - Reduce la complejidad de administración.
- Elastic Beanstalk es compatible con Java, .NET, PHP, Node.js, Python, Ruby, Go y Docker.
- No se aplican cargos adicionales por utilizar Elastic Beanstalk. Pague solo por los recursos AWS que utilice.

Ahora es el momento de revisar la unidad y concluir con una evaluación de conocimientos y una discusión sobre una pregunta del examen de certificación de práctica.



En resumen, en esta unidad 3 aprendieron a hacer lo siguiente:

- Proporcionar información general de los diferentes servicios de cómputo de AWS en la nube.
- Mostrar cómo se usa Amazon Elastic Compute Cloud (Amazon EC2)
- Identificar la funcionalidad en la consola de Amazon EC2
- Realizar funciones básicas en Amazon EC2 para construir un entorno informático virtual
- Identificar los elementos de optimización de costos de Amazon EC2
- Mostrar cuándo usar AWS Elastic Beanstalk
- Mostrar cuándo usar AWS Lambda
- Identificar cómo ejecutar aplicaciones en contenedores en un clúster de servidores administrados