Creación de aplicaciones tolerantes a errores en AWS

Octubre de 2011

Jeff Barr, Attila Narin y Jinesh Varia

Contenido

Introducción

El software se ha convertido en un aspecto fundamental de la vida cotidiana en casi todo el mundo. Independientemente de donde estamos, interactuamos con software, ya sea a través de nuestro teléfono móvil, para retirar dinero de un cajero automático o incluso para parar frente a la luz de un semáforo.

Dado que el software se ha convertido en una parte integral de nuestra vida diaria, queda mucho trabajo por hacer para garantizar su funcionamiento continuo y su disponibilidad.

En términos generales, esta área de estudio se conoce como *tolerancia a errores*, la capacidad de un sistema de seguir funcionando aunque fallen algunos de los componentes utilizados para crearlo.

Si bien es cierto que los sistemas esenciales deben estar disponibles en todo momento, también esperamos que un rango más amplio de software esté siempre disponible para nosotros. Por ejemplo, es posible que deseemos visitar un sitio de E-Commerce para adquirir un producto. Ya sean las 9 a. m. de un lunes por la mañana o las 3 a. m. en vacaciones, esperamos que el sitio esté disponible y listo para aceptar nuestra compra. El costo de no satisfacer estas expectativas puede paralizar a muchas empresas. Incluso con suposiciones muy conservadoras, se estima que un ocupado sitio de E-Commerce podría perder miles de dólares cada minuto que no está disponible. Esto es solo un ejemplo de por qué las empresas y las organizaciones se esfuerzan en desarrollar sistemas de software que sobrevivan errores.

Amazon Web Services (AWS) proporciona una plataforma ideal para crear sistemas de software tolerantes a errores. Sin embargo, este atributo no es único de nuestra plataforma. Teniendo en cuenta recursos y tiempo suficientes, se puede crear un sistema de software tolerante a errores en casi cualquier plataforma. La plataforma de AWS es única ya que le permite crear sistemas tolerantes a errores que funcionan con una intervención humana mínima e inversiones financieras iniciales.

Los errores no deberían tener TANTA atención

Cuando un servidor se bloquea o un disco duro se queda sin espacio en un entorno de centro de datos en las instalaciones, se notifica inmediatamente a los administradores, ya que se trata de eventos notables que requieren, como mínimo, su atención, sino su intervención. El estado ideal en un entorno de centro de datos tradicional en las instalaciones tiende a ser uno en el que las notificaciones se entregan de forma fiable al personal del administrador que esté listo para actuar con el fin de resolver el problema. Muchas organizaciones pueden llegar a este estado de nirvana de TI; sin embargo, esto normalmente requiere vasta experiencia, inversión financiera inicial y recursos humanos significativos.

Este no es el caso cuando se utiliza la plataforma proporcionada por Amazon Web Services. Idealmente, los errores que se produzcan en las aplicaciones creadas en nuestra plataforma pueden ser abordados automáticamente por el propio sistema y, como resultado, son bastante poco interesantes.

Amazon Web Services le ofrece acceso a una gran cantidad de infraestructura de TI, informática, de almacenamiento y de comunicaciones, que puede asignar automáticamente (o casi automáticamente) a la cuenta para casi cualquier tipo de error. Solo se le cobrará por los recursos que realmente utiliza, por lo que no debe realizar ninguna inversión financiera inicial.

Amazon Machine Images

Amazon Elastic Compute Cloud (Amazon EC2) es un servicio web dentro de Amazon Web Services que proporciona recursos informáticos (literalmente, instancias de servidor) que se utilizan para crear y alojar sistemas de software. Amazon EC2 es un punto de entrada natural a Amazon Web Services para el desarrollo de su aplicación. Puede crear un sistema fiable y tolerante a errores con varias instancias EC2, mientras utiliza las herramientas y los servicios auxiliares como Auto Scaling y Elastic Load Balancing.

En la superficie, las instancias de Amazon EC2 son muy similares a los servidores de hardware tradicional. Las instancias de Amazon EC2 utilizan sistemas operativos familiares, como Linux, Windows u OpenSolaris. Por tanto, pueden alojar casi cualquier tipo de software que se ejecute en estos sistemas operativos. Además, las instancias de Amazon EC2 tienen direcciones IP para poder usar los métodos habituales de interacción con una máquina remota (por ejemplo, SSH o RDP).

La plantilla que se utiliza para definir las instancias de su servicio se denomina una imagen de máquina de Amazon (AMI). Esta plantilla básicamente contiene una configuración de software (es decir, sistema operativo, servidor de aplicaciones y aplicaciones) y se aplica a un tipo de *instancia*.[1]

Los tipos de instancia en Amazon EC2 son básicamente arquetipos de hardware: usted elige un tipo de instancia que coincida con la cantidad de memoria (es decir, RAM) y de potencia de computo (es decir, el número de CPU) que necesita para su aplicación.

Una única AMI se puede utilizar para crear los recursos del servidor de diferentes tipos de instancias; esta relación se ilustra a continuación.

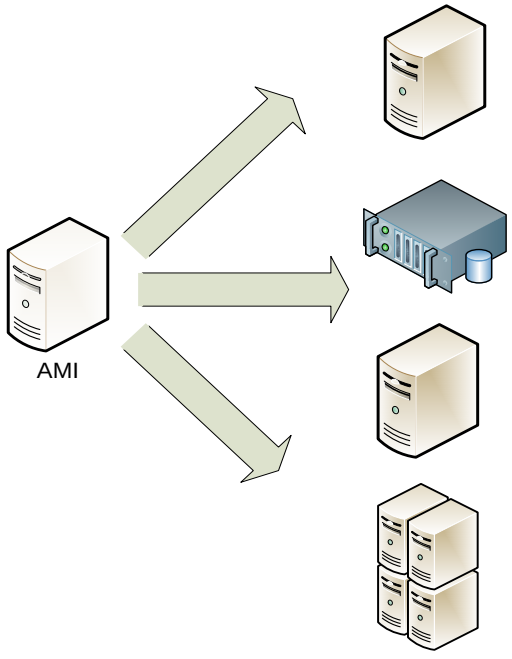

Figura 1: Amazon Machine Image

[1] Tipos de instancia: http://aws.amazon.com/ec2/instance-types/

Amazon publica muchas AMI que contienen configuraciones de software comunes. Además, varios miembros de la comunidad de desarrolladores de AWS también han publicado sus propias AMI personalizadas. Todas estas AMI se encuentran en la página de recursos de Amazon Machine Image,[2] en el sitio web de AWS.

Sin embargo, el primer paso hacia la creación de aplicaciones tolerantes en AWS es crear una biblioteca de sus propias AMI. La aplicación debe estar compuesta por al menos una AMI que haya creado. Iniciar la aplicación, entonces, es simplemente cuestión de lanzar la AMI.

Por ejemplo, si la aplicación es un servicio o un sitio web, la AMI debe estar configurada con un servidor web (por ejemplo, Apache o Microsoft Internet Information Server), el contenido estático asociado y el código para todas las páginas dinámicas. De forma alternativa, puede configurar la AMI para instalar todos los componentes necesarios para el software y el contenido mediante la ejecución de un script de arranque tan pronto como se lanza la instancia. Como resultado, después de lanzar la AMI, el servidor web se iniciará y su aplicación comenzará a aceptar solicitudes.

Una vez que haya creado una AMI, la sustitución de una instancia es muy sencilla: puede literalmente lanzar una instancia de sustitución que utilice la misma AMI como su plantilla.

Esto se puede hacer a través de una invocación a API, a través de las herramientas de línea de comandos para las que pueden crearse scripts o de la consola de administración de AWS, tal como se muestra a continuación. Más adelante en este documento, presentamos el servicio Auto Scaling, lo que puede sustituir automáticamente instancias fallidas o degradadas con nuevas.

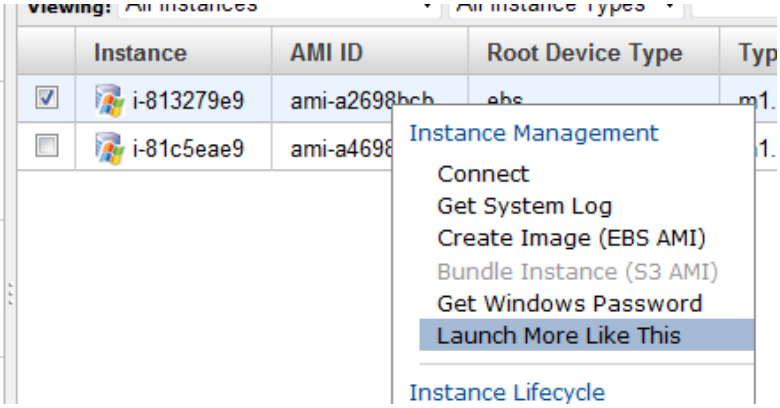

Figura 2: Lanzamiento de una instancia de Amazon EC2

Este es solo el primer paso hacia la tolerancia a errores. En este punto, puede recuperarse rápidamente de errores; si una de ellas sufre un error, o si no se comporta tal como desea, puede simplemente lanzar otra en función de la misma plantilla. Para minimizar el tiempo de inactividad, es posible que incluso siempre se esté ejecutando una instancia libre, lista para asumir el control en caso de error. Esto se puede hacer de forma eficaz mediante las *direcciones IP elásticas*. Puede realizar una conmutación por error a una instancia de reemplazo o una instancia de ejecución de repuesto; para ello, reasigne su dirección IP elástica a la nueva instancia. Las direcciones IP elásticas se describen con más detalle más adelante en el documento.

[2] Página de recursos de Amazon Machine Image - http://developer.amazonwebservices.com/connect/kbcategory.jspa?categoryID=171

La posibilidad de lanzar rápidamente las instancias de sustitución en función de una AMI que usted defina es un primer paso esencial hacia la tolerancia a errores. El siguiente paso consiste en almacenar datos persistentes a los que estas instancias de servidor tienen acceso.

Elastic Block Store

Amazon Elastic Block Store (Amazon EBS) proporciona volúmenes de almacenamiento de nivel de bloque diseñados para usarse con instancias de Amazon EC2. Los volúmenes de Amazon EBS son almacenamiento fuera de la instancia que persiste independientemente de la duración de esta.

Los volúmenes de Amazon EBS son esencialmente discos duros que se pueden adjuntar a una instancia de Amazon EC2 en ejecución. Amazon EBS es especialmente adecuado para aplicaciones que requieren una base de datos, un sistema de archivos o acceso al almacenamiento de nivel de bloque sin procesar. Los volúmenes de EBS almacenan datos de forma redundante, lo que los hace más duraderos que un disco duro típico. La tasa de errores anual (AFR) para un volumen de EBS es de 0,1% y 0,5%, en comparación con el 4% para un disco duro.

Amazon EBS y Amazon EC2 se suelen utilizar conjuntamente a la hora de crear una aplicación con tolerancia a errores en la plataforma de AWS. Todos los datos que necesiten persistir debe ser almacenados en los volúmenes de Amazon EBS, no en el denominado “almacenamiento efímero” asociado a cada instancia de Amazon EC2. Si la instancia de Amazon EC2 falla y debe ser reemplazada, el volumen de Amazon EBS se puede asociar a la nueva instancia de Amazon EC2. Dado que esta nueva instancia se considera un duplicado de la original, no debería haber pérdida de datos o funcionalidad.

Los volúmenes de Amazon EBS son muy fiables, pero para mitigar aún más la posibilidad de que se produzca un error, se pueden crear copias de seguridad de estos volúmenes de datos utilizando una función denominada *instantáneas*. Una estrategia de copia de seguridad robusta incluirá un intervalo (tiempo entre copias de seguridad, generalmente a diario, pero quizás más frecuentemente para ciertas aplicaciones), un período de retención (dependiendo de la aplicación y los requisitos comerciales para la reversión) y un plan de recuperación. Las instantáneas se almacenan para alta durabilidad en Amazon S3.

Se pueden utilizar para crear nuevos volúmenes de Amazon EBS, una réplica exacta del volumen original en el momento en que se tomó la instantánea. Debido a que las copias de seguridad representan el estado en el disco de la aplicación, se debe tener cuidado de eliminar los datos en memoria en el disco antes de iniciar una captura de pantalla.

Estas operaciones de Amazon EBS se pueden realizar a través de la API o desde AWS Management Console, como se ilustra a continuación.

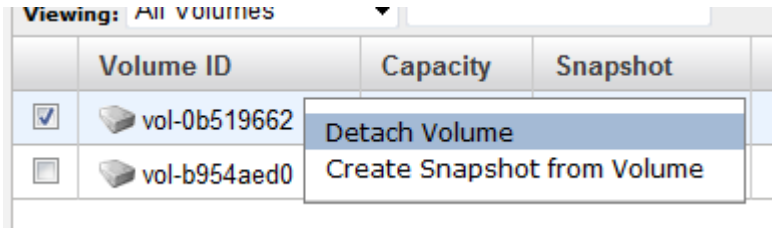

Figura 3: Amazon EBS

Direcciones IP elásticas

Las direcciones IP elásticas son públicas y se pueden asignar (enrutar) a cualquier instancia EC2 dentro de una región EC2 en particular. Las direcciones están asociadas con una cuenta de AWS, no a una instancia específica o la vida útil de una instancia y están diseñadas para ayudar en la construcción de aplicaciones tolerantes a errores. Una dirección IP elástica puede separarse de una instancia fallida y luego asignarse a una instancia de reemplazo en un período de tiempo muy corto. Al igual que con los volúmenes de Amazon EBS (y para el resto de los recursos de EC2, en este caso), todas las operaciones en las direcciones IP elásticas pueden realizarse mediante la programación a través de la API o manualmente desde la consola de administración de AWS:

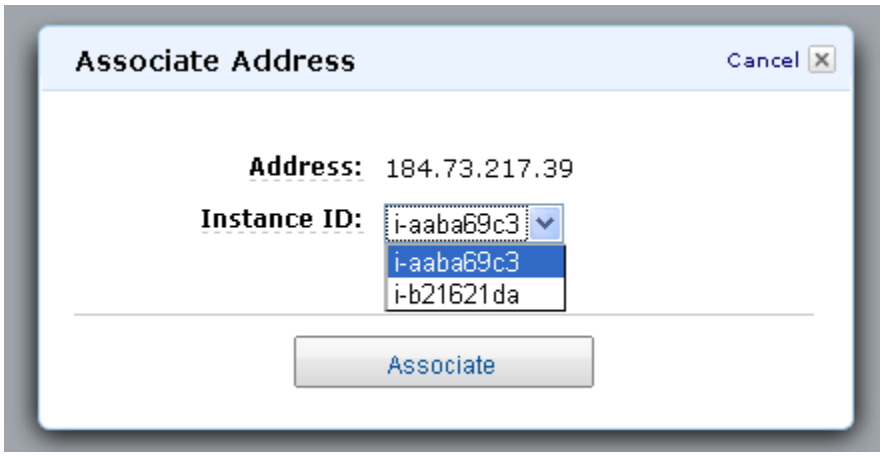

Figura 4: direcciones IP elásticas

Los errores pueden ser útiles

> *"No soy un verdadero programador. Junto cosas hasta que funcionan, y después sigo. Los programadores reales dirán 'sí, funciona, pero estás perdiendo memoria en todas partes. Quizá deberíamos solucionar el problema'. Voy reiniciar Apache cada 10 solicitudes".*
>
> *Rasmus Lerdorf (creador de PHP)*

Aunque a menudo no se admite fácilmente, la realidad es que la mayoría de los sistemas de software se degradarán con el tiempo. Esto se debe, en parte, a alguna de las siguientes razones:

1. El software pierde memoria o recursos. Esto incluye el software que usted ha escrito, así como el software del que depende (por ejemplo, marcos de trabajo de aplicaciones, sistemas operativos y controladores de dispositivos).
2. Los sistemas de archivos se fragmentan con el paso del tiempo y repercuten en el desempeño.
3. Los dispositivos de hardware (particularmente el almacenamiento) se degradarán físicamente con el tiempo.

La ingeniería de software disciplinada puede mitigar algunos de estos problemas pero, en última instancia, incluso el sistema de software más sofisticado depende de una serie de componentes que están fuera de su control (por ejemplo, sistema operativo, firmware y hardware). Con el tiempo, alguna combinación de hardware, software del sistema y su software provocará un error y interrumpirá la disponibilidad de la aplicación.

En un entorno de TI tradicional, el hardware se puede mantener regularmente, pero existen límites prácticos y financieros a la forma agresiva de hacerlo. Sin embargo, con Amazon EC2, puede terminar y volver a crear los recursos que necesita a su voluntad.

Una aplicación que aprovecha la plataforma de AWS puede *actualizarse* periódicamente con nuevas instancias de servidor. De este modo, se garantiza que cualquier degradación posible no afecte negativamente a su sistema en su conjunto. En cierto modo, se utiliza lo que se considera un error (por ejemplo, un servidor) como una función de terminación, lo que fuerza la función de actualización de este recurso.

Al utilizar este enfoque, una aplicación AWS se define con mayor precisión como el servicio que proporciona a sus clientes, en lugar de la(s) instancia(s) del servidor de las que está compuesta. Con esta mentalidad, las instancias del servidor se vuelven inmateriales e incluso desechables.

Auto Scaling

El concepto de aprovisionar y escalar automáticamente los recursos informáticos es un aspecto crucial de cualquier aplicación bien diseñada y tolerante a fallas que se ejecute en la plataforma Amazon Web Services. Auto Scaling[3] es una opción muy potente que usted puede aplicar fácilmente a la aplicación.

Auto Scaling le permite escalar automáticamente la capacidad de Amazon EC2. Puede definir reglas que determinan cuando más (o menos) necesarias son las instancias de servidor, como por ejemplo:

1. Cuando el número de instancias de servidor en funcionamiento está por encima (o por debajo) de un cierto número de instancias del servidor de lanzamiento (o terminación).
2. Cuando la utilización de los recursos (es decir, CPU, red o disco) de la flota de instancias del servidor está por encima (o por debajo) de un cierto umbral de instancias del servidor de lanzamiento (o terminación). Estas métricas se recogerán desde el servicio de Amazon CloudWatch, que monitorea las instancias de Amazon EC2.

Auto Scaling le permite terminar instancias de servidor según sus necesidades, sabiendo que las instancias de sustitución se lanzarán de forma automática. Auto Scaling también le permite añadir varias instancias en respuesta a un aumento de la carga; y, cuando dichas instancias ya no son necesarias, se eliminan automáticamente.

Estas reglas le permiten implementar muy fácilmente una serie de patrones de redundancia tradicional.

Por ejemplo, “N+1 redundancia[4]” es una estrategia muy popular para garantizar que un recurso (por ejemplo, una base de datos) esté disponible en todo momento. “N+1” dictamina que debería haber recursos N+1 operativos, cuando los recursos son suficientes para controlar la carga anticipada.

Este enfoque es ideal para Auto Scaling. Para implementar N+1 con Auto Scaling, solo tiene que definir una regla que debe ser siempre al menos 2 instancias de una determinada AMI disponible. Cuando se utiliza junto con Elastic Load Balancing, cada instancia manejaría una fracción de la carga entrante, con suficiente espacio libre (capacidad no utilizada) en cada una para manejar toda la carga si es necesario. Si una instancia falla, Auto Scaling iniciará de inmediato un reemplazo, ya que se retrasa el umbral mínimo de 2 instancias. Auto Scaling se asegurará siempre de que dispone de 2 instancias de servidor en buen estado.

[3] Auto Scaling es aplicable en varios escenarios; este documento examinará cómo hacerlo específicamente para lograr tolerancia a fallas.
[4] http://en.wikipedia.org/wiki/N%2B1_redundancy

Dado que Auto Scaling detectará automáticamente las fallas y lanzará instancias de reemplazo, si una instancia no se comporta como se esperaba (p. ej., se está ejecutando con bajo rendimiento), simplemente puede finalizar esa instancia y se lanzará una nueva.

Al usar Auto Scaling, puede (y debe) voltear regularmente sus instancias para asegurarse de que cualquier fuga o degradación no afecte su aplicación; literalmente puede establecer fechas de caducidad en las instancias de su servidor para garantizar que permanezcan “frescas”.

Con un enfoque en “N+1”, también puede hacer que el servidor adicional acepte solicitudes: este le permite a su aplicación una perfecta transición en caso de que el servidor principal produzca un error. La característica de Elastic Load Balancing en Amazon EC2 es una forma ideal para equilibrar la carga entre los servidores.

Elastic Load Balancing

Elastic Load Balancing es un producto de AWS que distribuye el tráfico entrante a la aplicación a través de varias instancias de Amazon EC2. Cuando se utiliza Elastic Load Balancing, recibirá un nombre de host de DNS: cualquier solicitud enviada a este nombre de host se delega a un grupo de instancias de Amazon EC2.

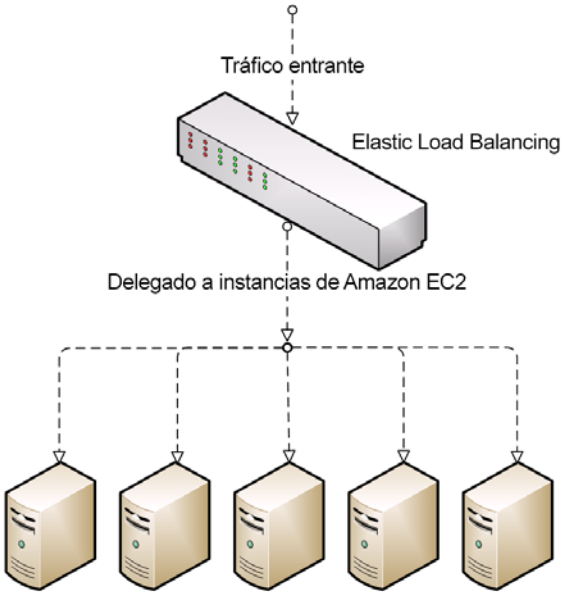

Figura 5: Elastic Load Balancing

Elastic Load Balancing detecta instancias en mal estado dentro de su grupo de instancias de Amazon EC2 y las redirecciona automáticamente en buen estado, hasta que se restauren las instancias en mal estado.

Auto Scaling y Elastic Load Balancing son una combinación ideal: Elastic Load Balancing le brinda un único nombre DNS para el direccionamiento y Auto Scaling asegura siempre que el número correcto de instancias amables de Amazon EC2 acepte solicitudes.

Regiones y zonas de disponibilidad

Otro elemento clave para lograr una mayor tolerancia a errores es la distribución geográfica de la aplicación. Si un único centro de Amazon Web Services falla por cualquier motivo, puede proteger su aplicación; para ello, ejecútelo simultáneamente en un centro de datos distante geográficamente.

Amazon Web Services está disponible en *regiones* geográficas. Cuando utiliza AWS, puede especificar la región en la que se almacenarán sus datos, las instancias en ejecución, las colas iniciadas y las bases de datos creadas. Para la mayoría de los AWS Infrastructure Services, incluidos Amazon EC2, hay cinco Regiones: EE.UU. Este (Norte de Virginia), EE.UU. Oeste (Norte de California), UE (Irlanda), Asia Pacífico (Singapur) y Asia Pacífico (Japón). Amazon S3 tiene una estructura de regiones ligeramente diferente: EE. UU. Estándar, que abarca centros de datos en los Estados Unidos, EE.UU. Oeste (Norte de California), UE (Irlanda), Asia Pacífico (Singapur) y Asia Pacífico (Japón).

Cada región se divide en *Zonas de disponibilidad (AZ)*. Las zonas de disponibilidad son regiones diferentes que están diseñadas para estar aisladas de errores que se produzcan en otras zonas de disponibilidad, y que proporcionan conectividad de red económica de baja latencia a otras zonas de disponibilidad de la misma región. Al iniciar instancias en Zonas de disponibilidad separadas, puede proteger sus aplicaciones de un error (por improbable que sea) que afecte a toda una zona.

Las regiones constan de una o más zonas de disponibilidad, están geográficamente dispersas y se encuentran en áreas geográficas o países separados. El acuerdo de nivel de servicios de Amazon EC2 ofrece una disponibilidad de 99,95% para cada región de Amazon EC2.

Compilación de arquitecturas Multi-AZ para lograr alta disponibilidad

Puede alcanzar alta disponibilidad mediante la implementación de la aplicación que abarca varias zonas de disponibilidad. Las instancias redundantes para cada nivel (por ejemplo, web, aplicación y base de datos) de una aplicación podrían ubicarse en distintas zonas de disponibilidad, creando así una solución de múltiples sitios. El objetivo deseado es tener una copia independiente de cada pila de aplicaciones en dos o más zonas de disponibilidad.

Para conseguir aún más tolerancia a errores con menos intervención manual, puede utilizar Elastic Load Balancing. Obtiene tolerancia a fallas mejorada cuando ubica sus instancias de cómputo detrás de Elastic Load Balancer, ya que puede equilibrar automáticamente el tráfico entre varias instancias y múltiples zonas de disponibilidad y garantizar que solo las instancias de Amazon EC2 en buen estado reciban tráfico. Puede configurar un Elastic Load Balancer para equilibrar el tráfico entrante de las aplicaciones entre las instancias de Amazon EC2 en una única zona de disponibilidad o en varias de ellas. Elastic Load Balancing puede detectar el estado de las instancias de Amazon EC2. Cuando detecta las instancias de Amazon EC2 en mal estado, deja de enrutar el tráfico a dichas instancias en mal estado. En lugar de ello, reparte la carga entre las instancias restantes en buen estado. Si todas sus instancias de Amazon EC2 en una zona de disponibilidad particular no son saludables, pero ha configurado instancias en múltiples zonas de disponibilidad, Elastic Load Balancing enrutará el tráfico a sus instancias sanas de Amazon EC2 en esas zonas. Se reanudará el balanceo de carga original a las instancias de Amazon EC2 cuando vuelvan a tener un estado correcto.

Esta solución de múltiples sitios está altamente disponible y, por diseño, cubrirá los fallos de componentes individuales o incluso la Zona de disponibilidad.

La siguiente figura muestra el un sistema altamente disponible en AWS, que abarca dos zonas de disponibilidad (AZ).

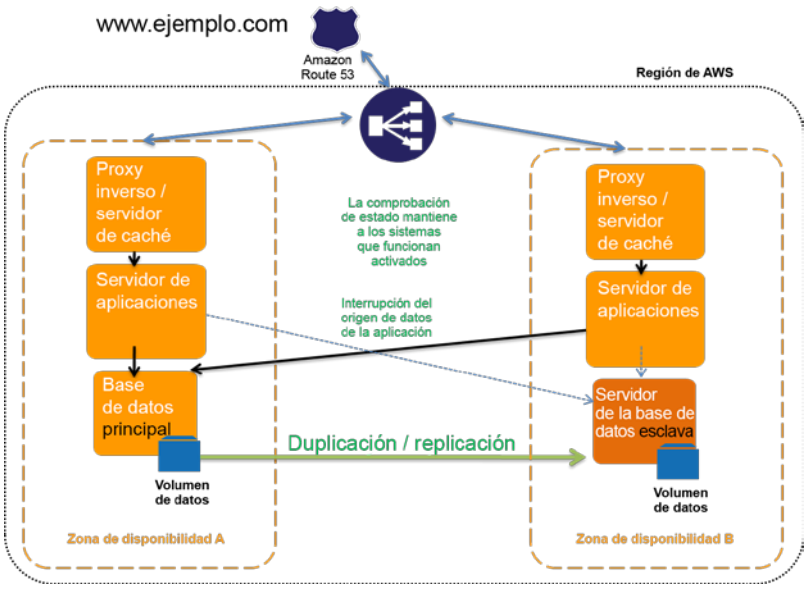

Figura 6: Aproveche Elastic Load Balancer y múltiples zonas de disponibilidad

Las direcciones IP elásticas desempeñan un papel clave en el diseño de una aplicación tolerante a errores que abarcan varias zonas de disponibilidad. El mecanismo de conmutación por error puede redirigir fácilmente la dirección IP (y, por lo tanto, el tráfico entrante) lejos de una instancia o zona fallida a una instancia de reemplazo.

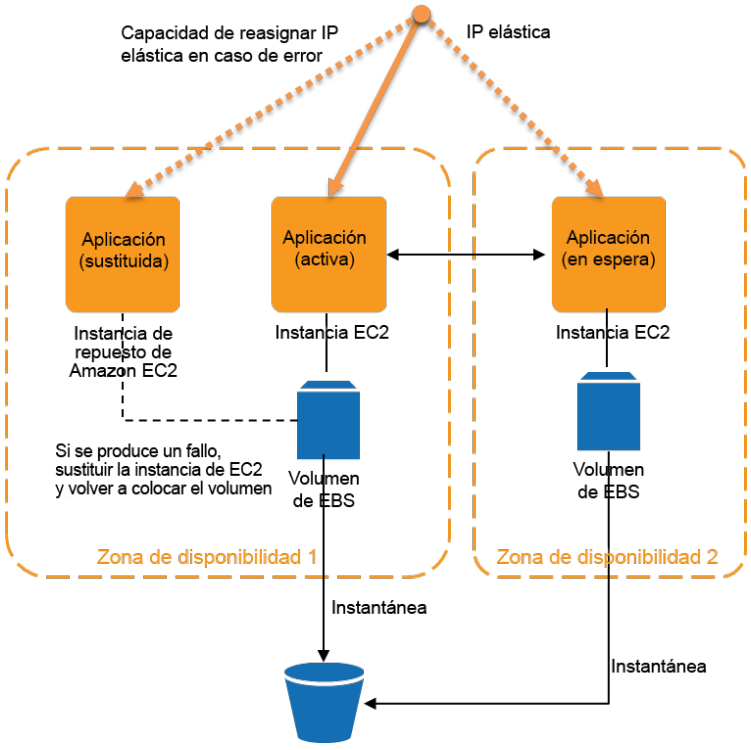

Figura 7: Aproveche Elastic IP y múltiples zonas de disponibilidad

Auto Scaling puede trabajar en varias zonas de disponibilidad en una región de AWS, lo que facilita aumentar o reducir automáticamente la capacidad. Las ofertas de bases de datos de AWS, como SimpleDB y Amazon Relational Database

Service (Amazon RDS) pueden ayudar a reducir el costo y la complejidad de operar un sistema de múltiples sitios. Consulte la sección Bloques de construcción tolerantes a fallas para obtener más información.

Instancias reservadas

Todas las técnicas examinadas hasta ahora han confiado en el supuesto de que podrá adquirir instancias de Amazon EC2 siempre que sea necesario.

Amazon Web Services tiene recursos masivos de hardware a su disposición, pero al igual que cualquier proveedor de informática en la nube, esos recursos son limitados. La mejor manera para los usuarios de maximizar su acceso a estos recursos es reservar una parte de la capacidad informática que necesitan. Esto se puede hacer a través de una función denominada instancias reservadas.

Con las instancias reservadas, usted reserva literalmente la capacidad informática en la nube de Amazon Web Services. Esto le permite beneficiarse de un precio inferior, pero lo que es más importante en el contexto de la tolerancia a errores, maximiza las posibilidades de obtener la capacidad informática que necesita.

Bloques de compilación tolerantes a errores

Amazon EC2 y sus características relacionadas proporcionan una plataforma económica potente para implementar y crear sus aplicaciones. Sin embargo, solo son un aspecto de Amazon Web Services en su conjunto.

Amazon Web Services ofrece una serie de productos que pueden incorporarse en el desarrollo de su aplicación. Estos servicios web tolerantes a errores de manera implícita; por lo tanto, con su uso, aumentará la tolerancia a errores de sus propias aplicaciones.

Amazon Simple Queue Service

Amazon Simple Queue Service (Amazon SQS) es un sistema de mensajería distribuido de alta fiabilidad que se puede utilizar como la columna vertebral de su aplicación con tolerancia a errores.

Los mensajes se almacenan en colas que usted crea: cada cola se define como una URL, por lo que puede acceder a ella cualquier servidor que tenga acceso a Internet, sujeto a la Lista de control de acceso (ACL) de la cola. Puede usar Amazon SQS para ayudarlo a asegurarse de que su cola esté siempre disponible; todos los mensajes que envíe a una cola se conservarán durante un máximo de cuatro días (o hasta que su aplicación los lea y elimine).

A continuación, se ilustra una arquitectura de sistema canónica mediante Amazon SQS.

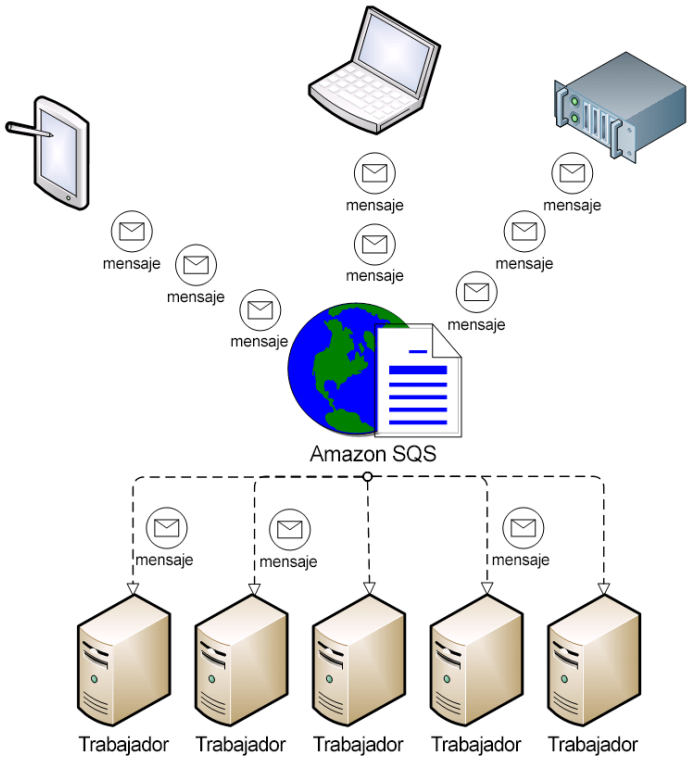

Figura 8: Arquitectura del sistema de Amazon SQS

En este ejemplo, una cola de Amazon SQS se utiliza para aceptar solicitudes. Un número de instancias de Amazon EC2 sondean constantemente esa cola para las solicitudes. Cuando se recibe una solicitud, una de estas instancias de Amazon EC2 la recoge y la procesa. Cuando dicha instancia termina de procesar la solicitud, va al sondeo.

Si la cantidad de mensajes en una cola comienza a crecer o si el tiempo promedio para procesar un mensaje es demasiado alto, puede escalar hacia arriba simplemente agregando más trabajadores en instancias adicionales de Amazon EC2.

Es común incorporar Auto Scaling para administrar estas instancias de Amazon EC2 y garantizar el suministro adecuado de las instancias EC2 que ejecuten los trabajadores que consumen mensajes de la cola. Incluso en un caso extremo donde todos los procesos laborales han fallado, Amazon SQS simplemente almacena los mensajes que recibe. Los mensajes se almacenan durante un máximo de cuatro días, por lo que dispone de tiempo suficiente para lanzar la sustitución de las instancias de Amazon EC2.

Una vez que un mensaje se ha extraído de una cola de SQS, pasa a ser invisible para otros consumidores por un intervalo de tiempo configurable conocido como *tiempo de espera de visibilidad*. Una vez que el consumidor haya procesado el mensaje, debe eliminar el mensaje de la cola. Si el intervalo de tiempo especificado por el tiempo de espera de visibilidad ha pasado, pero el mensaje no se elimina, se visibiliza una vez más en la cola y otro consumidor podrá extraerlo y procesarlo. Este modelo de dos fases garantiza que no se pierda ningún elemento de la cola si la aplicación consumidora falla mientras se procesa un mensaje.

Amazon Simple Storage Service

Amazon Simple Storage Service (Amazon S3) es un servicio web simple que proporciona almacenamiento de datos con tolerancia a errores de larga duración. Amazon Web Services es responsable de mantener la disponibilidad y tolerancia a errores, sino que simplemente se paga el espacio de almacenamiento que utilice.

En segundo plano, Amazon S3 almacena los objetos de forma redundante en varios dispositivos de diversas instalaciones dentro de una región de Amazon S3, por lo que, incluso en el caso de que se produzca un error en un centro de datos de Amazon Web Services, también tendrá que tener acceso a sus datos.

Amazon S3 es ideal para cualquier tipo de requisitos de almacenamiento de datos de objeto que su aplicación pueda tener. A Amazon S3 se accede por URL como Amazon SQS, por lo que cualquier recurso informático que tenga acceso a Internet puede usarlo.

La función de control de versiones de Amazon S3 le permite conservar versiones anteriores de los objetos almacenados en S3 y también protege contra las eliminaciones accidentales iniciadas por una aplicación que funciona mal. El control de versiones se puede habilitar para cualquiera de los buckets de S3.

Al utilizar Amazon S3, puede delegar la responsabilidad de un aspecto crítico de la tolerancia a fallas, el almacenamiento de datos, en Amazon Web Services.

Amazon SimpleDB

Amazon SimpleDB es una solución tolerante a errores y duradera para el almacenamiento de datos estructurados. Con Amazon SimpleDB, puede decorar sus datos con atributos y consultarlos en función de los valores de esos atributos. En muchos casos, Amazon SimpleDB se puede utilizar para aumentar o incluso reemplazar el uso de bases de datos relacionales tradicionales como MySQL o Microsoft SQL Server.

Amazon SimpleDB está altamente disponible para su uso, como Amazon S3 y otros servicios. Mediante el uso de Amazon SimpleDB, puede beneficiarse de un servicio escalable diseñado para alta disponibilidad y tolerancia a errores. Los datos almacenados en Amazon SimpleDB se guardan de forma redundante sin puntos únicos de errores.

Amazon Relational Database Service

Amazon Relational Database Service (Amazon RDS) es un servicio web que facilita la ejecución de bases de datos relacionales en la nube. En el contexto de la creación de aplicaciones tolerantes a errores y de alta disponibilidad, Amazon RDS ofrece varias características para mejorar la fiabilidad de las bases de datos críticas.

Las copias de seguridad automatizadas de su base de datos permiten la recuperación puntual para su instancia de base de datos. Amazon RDS realizará un backup de su base de datos y sus registros de transacciones y los almacenará durante un periodo de retención especificado por el usuario. Esta característica está habilitado de forma predeterminada.

Al igual que las instantáneas de Amazon EBS, con Amazon RDS puede iniciar instantáneas de su instancia de base de datos. Amazon RDS almacena estos backups completos de la base de datos hasta que los elimine expresamente. A continuación, puede crear una nueva instancia de base de datos a partir de una instantánea de base de datos, donde desee. Esto puede ayudarlo a recuperarse de fallas de nivel superior, como la modificación involuntaria de datos, ya sea por error del operador o por errores en la aplicación.

Amazon RDS también admite una función de implementación *Multi-AZ*. Si está habilitada, se aprovisiona una réplica en espera sincrónica de la base de datos en una zona de disponibilidad diferente. Las actualizaciones de su instancia de base de datos se replican sincronizadamente en zonas de disponibilidad en el modo de espera para mantener ambas

bases de datos sincronizadas. En el caso de un escenario de conmutación por error, el modo en espera se promueve para que sea el principal y se encargará de las operaciones de su base de datos. Ejecutar su instancia de base de datos como una implementación Multi-AZ protege sus datos en el improbable caso de una falla en el componente de instancia de base de datos o una interrupción en el estado del servicio en una zona de disponibilidad.

Conclusión

Amazon EC2 es un punto de entrada natural para el desarrollo de su aplicación; sus instancias de servidor son conceptualmente muy similares a los servidores tradicionales; esto reduce en gran medida la curva de aprendizaje para desarrollar aplicaciones para la nube. Sin embargo, el uso de las instancias de servidor Amazon EC2 de la misma manera que las instancias de servidores de hardware tradicionales es solo un punto de partida; al hacerlo, no se mejorará en gran medida la tolerancia a fallas, el rendimiento o incluso el costo total.

Los beneficios completos de la plataforma Amazon Web Services se realizan cuando incorpora más características de Amazon EC2, así como otros productos de Amazon Web Services.

Para crear aplicaciones tolerantes a errores en Amazon EC2, es importante seguir las mejores prácticas, como la posibilidad de encargar rápidamente instancias de reemplazo, utilizar Amazon EBS para el almacenamiento persistente y aprovechar las múltiples zonas de disponibilidad y las direcciones IP elásticas.

El uso de Auto Scaling le permite reducir en gran medida la cantidad de tiempo y los recursos que necesita para supervisar sus servidores: si se produce un error, se lanzará automáticamente un reemplazo para usted. Diagnosticar un servidor no saludable puede ser tan simple como terminarlo y dejar que Auto Scaling lance uno nuevo para usted.

Elastic Load Balancing le permite publicar un único punto final conocido para su aplicación. El flujo y reflujo de las instancias de Amazon EC2 que se inician, fallan, se terminan y se reinician se ocultarán a sus usuarios.

Amazon SQS, Amazon S3 y Amazon SimpleDB son componentes básicos de nivel superior que puede incorporar en su aplicación. Estos servicios son excelentes ejemplos de cómo lograr tolerancia a errores y, a su vez, aumentarla en su aplicación. Con Amazon RDS, tiene fácil acceso a las funciones que permiten implementaciones de bases de datos tolerantes a errores, incluidas copias de seguridad automáticas, instantáneas y despliegues Multi-AZ.

Sobre todo, el modelo de precios de Amazon Web Services le ofrece la opción de experimentar; no hay una inversión inicial, solo tiene que pagar por lo que utilice. Si un aspecto determinado de la plataforma de Amazon Web Services resulta no adecuado para su aplicación, sus inversiones se completan en el momento en que deje de utilizarlo.

La potencia, sofisticación y transparencia económica que ofrece Amazon Web Services le proporcionará una plataforma incomparable para crear su software tolerante a errores.

Documentación adicional

1. **Prácticas recomendadas para utilizar IP elásticas y zonas de disponibilidad** - http://support.rightscale.com/09-Clouds/AWS/02_Amazon_EC2/Designing_Failover_Architectures_on_EC2/00-Best_Practices_for_using_Elastic_IPs_%28EIP%29_and_Availability_Zones

2. **Configuración de sitio tolerante a errores con zonas de disponibilidad de Amazon** - http://blog.rightscale.com/2008/03/26/setting-up-a-fault-tolerant-site-using-amazons-availability-zones/

3. **Scalr** - https://scalr.net/login.php

4. **Creación de un centro de datos virtual con Scalr y Amazon Web Services** - http://scottmartin.net/2009/07/11/creating-a-virtual-datacenter-with-scalr-and-amazon-web-services/

5. **Elastic Load Balancing de Amazon** - http://aws.amazon.com/elasticloadbalancing/

6. **Auto Scaling Service** - http://aws.amazon.com/autoscaling

7. **Tipos de instancia** - http://aws.amazon.com/ec2/instance-types/

8. **Elastic Block Store** - http://aws.amazon.com/ebs/

9. **Recursos de Amazon Machine Images** - http://developer.amazonwebservices.com/connect/kbcategory.jspa?categoryID=171

10. **Amazon Relational Database Service** - http://aws.amazon.com/rds/