



Desarrollo y pruebas en Amazon Web Services

Noviembre de 2012
Carlos Conde, Attila Narin

(Consulte <<http://aws.amazon.com/whitepapers/>> para obtener la versión más reciente de este documento)

Contenido

Contenido.....	2
Resumen	3
Introducción.....	3
Fase de desarrollo	4
Repositorio de código fuente.....	4
Herramientas de administración de proyectos	5
Entornos de desarrollo bajo demanda.....	7
Detener o terminar instancias Amazon EC2	8
Integración con API de AWS y mejoras del IDE.....	9
Fase de compilación.....	9
Versiones de desarrollo	9
Compilaciones bajo demanda.....	10
Almacenamiento y distribución de la salida de la compilación	11
Fase de pruebas	12
Automatizar los entornos de prueba	12
Aprovisionamiento de instancias.....	12
Aprovisionar bases de datos	13
Aprovisionar entornos completos.....	13
Prueba de carga	14
Pruebas de carga de red	14
Prueba de carga para AWS.....	15
Optimización de costos con instancias de spot	15
Pruebas de aceptación de usuario	16
Pruebas en paralelo	16
Pruebas de tolerancia a errores.....	17
Administración de recursos	17
Asignación de costos y varias cuentas de AWS.....	18
Conclusión.....	18
Documentación adicional	19

Resumen

En este documento técnico, se describe cómo Amazon Web Services (AWS) aporta valor en las distintas fases del ciclo de desarrollo de software, con especial énfasis en el desarrollo y las pruebas. Para la fase de desarrollo, se muestra cómo usar AWS para administrar el control de versiones; se describen herramientas de administración de proyectos, el proceso de compilación y entornos hospedados en AWS; y, por último, se muestran prácticas recomendadas. Para la fase de pruebas, se describe cómo administrar entornos de prueba y ejecutar distintos tipos de pruebas, como pruebas de carga, pruebas de aceptación, pruebas de tolerancia a errores, etc. AWS ofrece ventajas únicas en cada uno de estos escenarios y fases, permitiéndole elegir y seleccionar las más adecuadas para su proyecto de desarrollo de software. Este documento va dirigido a jefes de proyecto, desarrolladores, evaluadores, arquitectos de sistemas, o bien cualquiera que participe en actividades de producción de software.

Introducción

Las organizaciones desarrollan software por distintos motivos, desde necesidades empresariales básicas (la organización es un proveedor de software) hasta la personalización o la integración de software. Las organizaciones también crean distintos tipos de software: aplicaciones web, aplicaciones independientes, agentes automatizados, etc. En todos esos casos, se pide a los equipos de desarrollo que produzcan software de alta calidad y con la mayor rapidez posible para reducir el plazo de comercialización o el plazo de producción.

En este documento, “desarrollo y pruebas” hace referencia a las distintas herramientas y procedimientos que se aplican en la producción de software. Independientemente del tipo de software que se desarrolle, para tener éxito es esencial contar con un conjunto adecuado de procedimientos de desarrollo y pruebas. Sin embargo, para producir aplicaciones, no solo se necesitan ingenieros de software, sino también recursos de TI, que están sujetos a limitaciones de tiempo, dinero y experiencia.

El ciclo de vida del software suele estar formado por los siguientes elementos principales:



En este documento técnico, se describen aspectos de las fases de desarrollo, compilación y pruebas. Para cada una de estas fases, se necesitan distintos tipos de infraestructura de TI. Aquí es donde AWS ofrece distintas ventajas a los equipos de desarrollo de software. AWS ofrece acceso bajo demanda a una amplia variedad de servicios de infraestructura en la nube, y solo se cobran los recursos usados. AWS permite eliminar la necesidad de comprar costoso hardware, así como las complicaciones administrativas derivadas de ser propietario y realizar el mantenimiento. Ser propietario de hardware y de una infraestructura de TI suele implicar unos gastos de capital durante un período de 3 a 5 años, donde la mayoría de los equipos de desarrollo y pruebas necesitan computarse o almacenarse en horas, días, semanas o meses. La diferencia en escalas de tiempo puede causar fricción debido a la dificultad de las operaciones de TI para satisfacer las solicitudes simultáneas de los equipos de proyecto, incluso aunque estén limitados por un conjunto fijo de recursos. Como resultado, los equipos de proyecto invierten mucho más tiempo en justificar, obtener y mantener recursos. Este tiempo se podría dedicar a centrarse en el trabajo principal.

Al aprovisionar únicamente los recursos necesarios durante las fases de desarrollo o ejecución de pruebas, su empresa puede obtener ahorros importantes en comparación con la inversión inicial en hardware tradicional.¹ Con el nivel adecuado de detalle, puede asignar recursos basándose en las necesidades y el presupuesto de cada proyecto. Además de esos beneficios económicos, AWS también ofrece importantes ventajas operativas, como la capacidad de configurar una infraestructura de desarrollo y pruebas en cuestión de minutos, en lugar de semanas o meses, y ampliar o reducir la capacidad para proporcionar los recursos de TI necesarios, solo cuando sean necesarios.

En este documento, se describen algunas de las prácticas recomendadas relacionadas con el desarrollo y las pruebas en AWS. Por ejemplo, en la fase de desarrollo, se explica cómo configurar de forma segura y a largo plazo herramientas y procesos como el control de versiones, entornos de colaboración y procesos de compilación automatizada; en la fase de pruebas, se explica cómo configurar los entornos de prueba de forma automatizada y cómo ejecutar distintos tipos de prueba, como pruebas en paralelo, pruebas de carga, pruebas de esfuerzo, pruebas de resistencia, etc.

Fase de desarrollo

Independientemente del tamaño del equipo, el tipo de software que se desarrolle o la duración del proyecto, las herramientas de desarrollo son esenciales para racionalizar el proceso, coordinar las iniciativas y centralizar la producción. Al igual que cualquier sistema de TI, las herramientas de desarrollo necesitan tareas de administración y mantenimiento adecuadas. Usar esas herramientas en AWS no solo libera a su equipo de desarrollo de las tareas de mantenimiento de sistemas de bajo nivel (como la configuración de red, la configuración de hardware, etc.), sino que también facilita la ejecución de tareas más complejas. En las secciones siguientes, se describe cómo operar los componentes principales de las herramientas de desarrollo en AWS.

Repositorio de código fuente

El repositorio de código fuente es una herramienta clave para los equipos de desarrollo. Por lo tanto, necesita estar disponible y, además, los datos que contiene (archivos de código fuente con control de versiones) necesitan almacenarse a largo plazo, con políticas de backup adecuadas. Para garantizar estas dos características (disponibilidad y durabilidad), es necesario invertir en recursos, experiencia y tiempo, que no suelen ser una de las competencias básicas de un equipo de desarrollo de software.

¹ O bien, completar campañas de prueba.

Para crear un repositorio de código fuente en AWS, es necesario crear una instancia de [Amazon Elastic Compute Cloud](#)² (Amazon EC2) e instalar de forma remota el software de control de versiones en esa instancia. En cuestión de minutos, los desarrolladores podrán crear instancias Amazon EC2, que son máquinas virtuales sobre las que tendrá control total. Hay disponible una amplia variedad de sistemas operativos y distribuciones como Amazon Machine Images (AMI). Una AMI es una plantilla que contiene una configuración de software (sistema operativo, servidor de aplicaciones y aplicaciones) que se puede ejecutar en Amazon EC2. Después de instalar y configurar correctamente la instancia del repositorio de código fuente, le recomendamos que cree una AMI para esta configuración con el fin de poder recrear rápidamente esa instancia sin tener que reinstalar y volver a configurar el software de control de versiones.

Puede almacenar los datos del repositorio en una ubicación separada del sistema host para simplificar las operaciones de migración o mantenimiento. [Amazon Elastic Block Store](#)³ ([Amazon EBS](#)) proporciona volúmenes de almacenamiento fuera de la instancia que persisten independientemente de la duración de esta. Después de crear un volumen, puede conectarlo a una instancia Amazon EC2 en ejecución. Como tal, se aprovisiona y conecta un volumen de Amazon EBS a la instancia para almacenar los datos del repositorio de control de versiones.

Obtendrá durabilidad al crear instantáneas de momento específico del volumen EBS que contenga los datos del repositorio. Las instantáneas de EBS se almacenan en [Amazon Simple Storage Service](#)⁴ (Amazon S3), un almacén de datos de larga duración y escalable. Los objetos en Amazon S3 se almacenan de forma redundante en varios dispositivos de diversas instalaciones dentro de una región de Amazon S3. A continuación, puede usar estas instantáneas como el punto de inicio para nuevos volúmenes de Amazon EBS, y puede proteger sus datos para garantizar una duración a largo plazo. Al usar instantáneas de EBS, puede administrar fácilmente los backups del repositorio de código fuente. En caso de error, puede recrear el volumen de datos del repositorio a partir de las instantáneas que se almacenaron a largo plazo en Amazon S3 y recrear la instancia del repositorio de código fuente a partir de una AMI.

Una dirección IP elástica ofrece un punto de enlace estático a una instancia Amazon EC2, y también puede usarse en combinación con la configuración DNS (por ejemplo, detrás de un registro CNAME de DNS). Esto permitirá a los equipos obtener acceso a los servicios hospedados (como el repositorio de código fuente) de una forma coherente, incluso si se modifica la infraestructura subyacente (por ejemplo, al ampliar o reducir su capacidad, o cuando se retira una instancia de sustitución).

A medida que crece el repositorio de código fuente y se necesite más capacidad de almacenamiento, hay disponibles dos soluciones: aprovisionar y conectar volúmenes de Amazon EBS adicionales a la instancia del repositorio, o bien aprovisionar un nuevo volumen de Amazon EBS de mayor tamaño (hasta 1 TB) basado en una instantánea reciente de los datos del repositorio existentes. A continuación, el nuevo volumen sustituirá al volumen existente, que podrá eliminar. En ambos casos, los nuevos volúmenes de Amazon EBS ofrecen una capacidad adicional que se puede crear bajo demanda en cuestión de minutos.

Herramientas de administración de proyectos

Además del repositorio de código fuente, con frecuencia los equipos usan herramientas adicionales (por ejemplo, de seguimiento de problemas, seguimiento de proyectos, análisis de calidad del código, colaboración, uso compartido de contenido, etc.) En su mayoría, esas herramientas se proporcionan como aplicaciones web. Al igual que cualquier otra aplicación web clásica, se necesita un servidor para ejecutarlas y, con frecuencia, también una base de datos relacional. Ambos componentes se pueden instalar en instancias Amazon EC2 siguiendo procedimientos de implementación similares a los usados en los entornos clásicos, donde la base de datos usa volúmenes de Amazon EBS para el almacenamiento de datos.

² Consulte: <http://aws.amazon.com/ec2/>

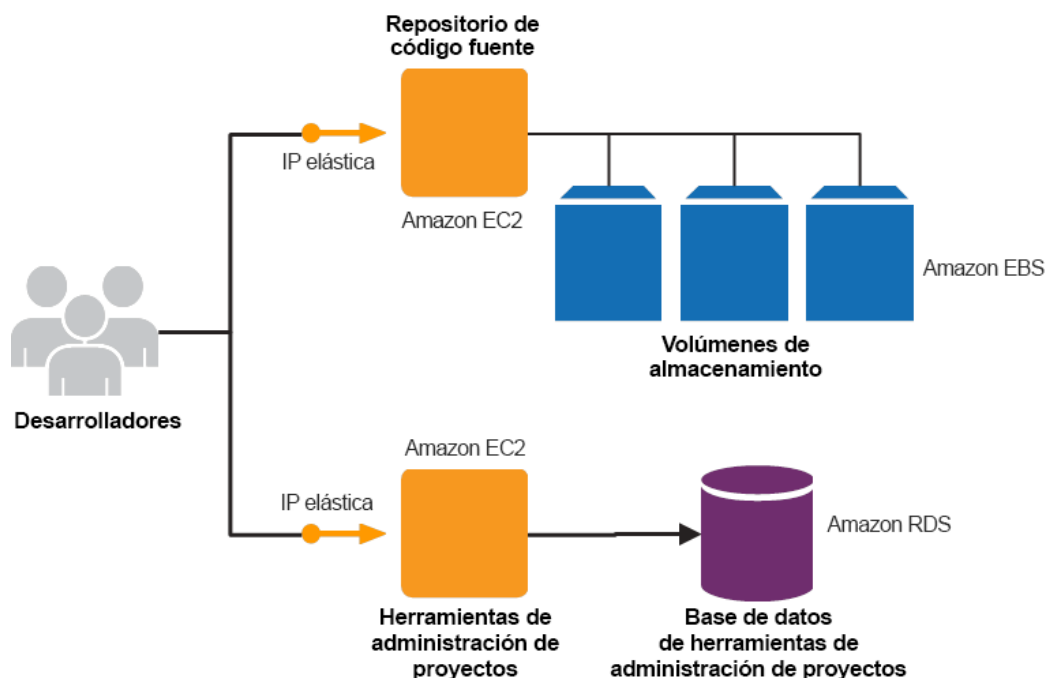
³ Consulte: <http://aws.amazon.com/ebs/>

⁴ Consulte: <http://aws.amazon.com/s3/>

Las herramientas de administración de proyectos tienen las mismas necesidades que los repositorios de código fuente: necesitan estar disponibles y los datos deben almacenarse a largo plazo. Aunque se puede mitigar la pérdida de informes de análisis de código al recrearlos basándose en la versión del repositorio deseada, perder información de seguimiento de problemas o de proyectos puede tener consecuencias más graves. Puede dar solución a los problemas de disponibilidad del servicio de una aplicación web de administración de proyectos si usa AMI para crear instancias Amazon EC2 de sustitución en caso de que se produzcan errores, igual que lo haría con los repositorios de código fuente. Para obtener una durabilidad adecuada para una base de datos, se necesita más esfuerzo y atención: incluso aunque se usen volúmenes de Amazon EBS, es necesario crear instantáneas en un sistema de archivos inmovilizado para garantizar una coherencia. Además, para restaurar una base de datos, puede que sea necesario realizar operaciones adicionales, además de restaurar un volumen a partir de una instantánea y conectarlo a una instancia Amazon EC2.

Para facilitar esta parte, Amazon Relational Database Service (Amazon RDS) ofrece una forma sencilla de configurar, usar y escalar una base de datos relacional en AWS. Ofrece una capacidad redimensionable y rentable al administrar las tareas de administración de bases de datos laboriosas, ya que libera al equipo del proyecto de esta responsabilidad. Las instancias de bases de datos de Amazon RDS se pueden aprovisionar en cuestión de minutos. De manera opcional, Amazon RDS también garantizará que el software de base de datos relacional esté actualizado con las revisiones más recientes. La característica de backups automáticos de Amazon RDS permite la recuperación a un momento dado de instancias de bases de datos, por lo que se puede restaurar una instancia de base de datos a cualquier momento específico del período de retención de backups.

A medida que el equipo de desarrollo crece o se añaden más herramientas a la instancia de administración de proyectos, puede que necesite más capacidad, tanto para la instancia de la aplicación web como para la instancia de la base de datos. En AWS, escalar instancias verticalmente es una operación muy sencilla. Solo tiene que crear un servidor de aplicaciones web desde la AMI en un tipo de instancia Amazon EC2 con más capacidad y reemplazar el servidor anterior. Las instancias de bases de datos de Amazon RDS se pueden escalar en términos de recursos de memoria y computación con solo unos clics en la consola de administración de AWS.



Nota: para realizar una implementación incluso más rápida y sencilla, hay disponible una amplia variedad de herramientas de administración de proyectos en [AWS Marketplace](https://aws.amazon.com/marketplace/b/2649274011/)⁵ o como [Amazon Machine Images](https://aws.amazon.com/amis)⁶.

⁵ Consulte: <https://aws.amazon.com/marketplace/b/2649274011/>

⁶ Consulte: <https://aws.amazon.com/amis>

Entornos de desarrollo bajo demanda

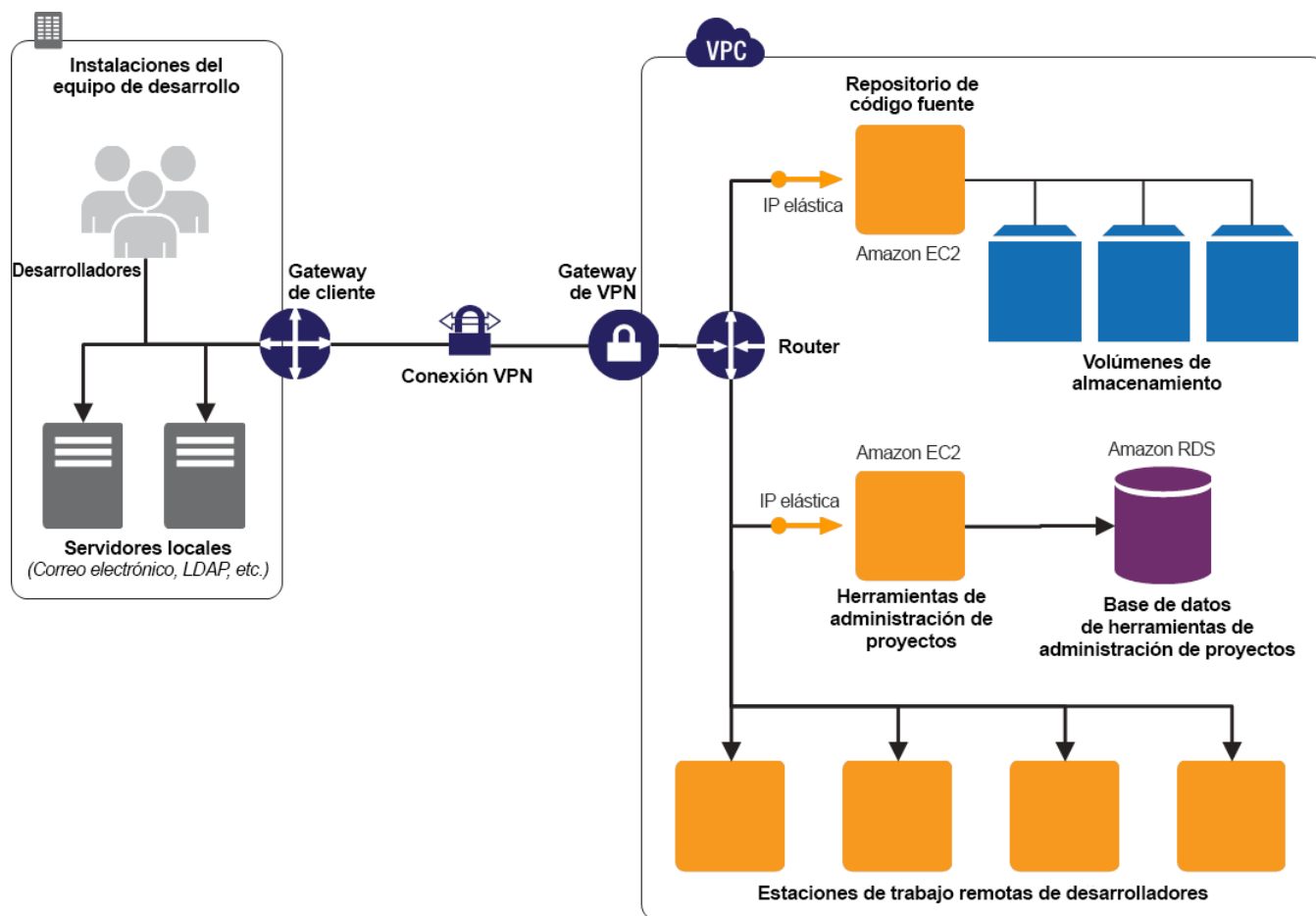
Los desarrolladores usan principalmente equipos de escritorio o portátiles locales para ejecutar los entornos de desarrollo. Aquí es donde suele instalarse el IDE, se realizan las pruebas de unidades, se comprueba el código fuente, etc. Sin embargo, también hay algunos casos en los que resulta útil hospedar en AWS los entornos de desarrollo bajo demanda.

En algunos proyectos de desarrollo, puede que se usen conjuntos de herramientas especializadas donde sería complicado o costoso en términos de recursos instalarlos y mantenerlos en equipos locales, especialmente si esas herramientas no se usan con frecuencia. En esos casos, puede preparar y configurar entornos de desarrollo con las herramientas necesarias (herramientas de desarrollo, control de código fuente, conjuntos de pruebas de unidades, IDE, etc.) y, a continuación, agruparlos como AMI. Cuando se necesite un entorno, puede iniciar fácilmente el entorno adecuado y ponerlo en funcionamiento con un tiempo y esfuerzo mínimos. Después, cuando ya no necesite el entorno, puede apagarlo para liberar recursos. Esto también puede resultarle útil si necesita cambiar de contexto al tener un trabajo en curso y haber extraído código del repositorio. En lugar de administrar ramas o trabajar con inserciones parciales en el repositorio, puede poner en marcha un nuevo entorno temporal.

En AWS, tiene acceso a una [amplia variedad de tipos de instancia](#), algunas con configuraciones de hardware muy específicas. Si desarrolla específicamente para una configuración concreta, puede que le resulte útil tener un entorno de desarrollo en la misma plataforma donde se ejecutará el sistema en producción.

El concepto de escritorios hospedados no está limitado a los entornos de desarrollo, sino que también puede aplicarse a otros roles o funciones. Para entornos de trabajo más complejos, AWS CloudFormation permite configurar fácilmente conjuntos de recursos de AWS. Este tema se describe con más detalle en la sección “Pruebas” a continuación, en el contexto de la configuración de entornos de prueba. En muchos casos, dichos entornos se configuran en [Amazon Virtual Private Cloud](#) (Amazon VPC), que le permite ampliar la red privada local a la nube. Después, puede aprovisionar los entornos de desarrollo como si estuvieran en la red local, aunque se ejecutan en AWS. Esto puede resultarle útil si esos entornos necesitan recursos locales (como LDAP).

En el diagrama siguiente, se muestra una implementación donde se ejecutan entornos de desarrollo en instancias Amazon EC2 en una Amazon VPC. Se usa una conexión VPN segura para obtener acceso de forma remota a esas instancias desde una red empresarial.



Detener o terminar instancias Amazon EC2

Cuando no se usen los entornos de desarrollo (por ejemplo, durante las horas en que no se trabaja o cuando un proyecto específico está en suspensión), puede apagarlos fácilmente para ahorrar recursos y costos. Existen dos posibilidades: puede detener las instancias, que equivale aproximadamente a hibernar el sistema operativo; o bien puede terminarlas, que sería algo muy parecido a descartar el sistema operativo.

Al detener una instancia (posible para AMI hospedadas en Amazon EBS), se liberan los recursos de computación y ya no se aplicarán más cargos por hora para la instancia. El volumen de Amazon EBS guardará su estado y, la próxima vez que inicie la instancia, esta contendrá los datos de trabajo que había en el momento en que la detuvo (lo que equivaldría a volver a abrir la tapa de un equipo portátil). (Nota: los datos almacenados en unidades efímeras no estarán disponibles después de una secuencia de parada/inicio).

Al terminar una instancia, se eliminará automáticamente el dispositivo raíz, así como cualquier otro dispositivo conectado durante el inicio de la instancia (excepto si la marca "DeleteOnTermination" de un volumen se establece en "falso"); por lo tanto, los datos podrían perderse si no hay disponible un backup o una instantánea para los volúmenes eliminados. Las instancias terminadas dejan de existir y, si es necesario, tendrá que recrearlas a partir de una AMI. Normalmente, se termina la instancia de un entorno de desarrollo si se confirmó todo el trabajo, o bien si el entorno específico ya no se usará.

Integración con API de AWS y mejoras del IDE

Con AWS, ahora puede desarrollar código en una infraestructura de TI y controlarlo, independientemente de si la plataforma objetivo del proyecto es AWS o si el proyecto tiene el objetivo de organizar recursos en AWS. En esos casos, puede usar los distintos AWS SDK para integrar fácilmente sus aplicaciones con las API de AWS, lo que facilitará el desarrollo de código directamente en una interfaz de un servicio web y la obtención de datos de autenticación, reintentos, control de errores, etc. Las herramientas del AWS SDK están disponibles para varios lenguajes: [Java](#)⁷, [.Net](#)⁸, [PHP](#)⁹, [Ruby](#)¹⁰ y, para [plataformas móviles](#)¹¹, Android y iOS.

AWS también ofrece [herramientas](#)¹², como AWS Toolkit for Visual Studio y AWS Toolkit for Eclipse, que le permiten interactuar fácilmente con AWS desde sus IDE.

Fase de compilación

El proceso de crear una aplicación está compuesto por diversos pasos, como la compilación, la generación de recursos y la creación de paquetes. Para aplicaciones de gran tamaño, en cada paso se usan varias dependencias, como la creación de bibliotecas internas, el uso de aplicaciones auxiliares, la generación de recursos en distintos formatos, la elaboración de la documentación, etc. Es posible que, en algunos proyectos, incluso sea necesario crear las entregas para distintas arquitecturas de CPU, plataformas o sistemas operativos. En definitiva, todo el proceso de compilación puede tardar varias horas en completarse, lo que afecta directamente a la agilidad del equipo de desarrollo de software. Este impacto es mucho más patente en equipos que adoptan métodos como la integración continua¹³, donde todas las confirmaciones en el repositorio de código fuente disparan una compilación automatizada, seguida de conjuntos de pruebas.

Versiones de desarrollo

Para mitigar este problema, los equipos que trabajan en proyectos con tiempos de compilación prolongados suelen adoptar el método de “versión de desarrollo” (o compilación neutra¹⁴), o bien dividir el proyecto en subproyectos más pequeños (o incluso una combinación de ambos). Para crear versiones de desarrollo, se necesita una máquina de compilación que compruebe el código fuente más reciente del repositorio y que compile las entregas del proyecto por la noche. Puede que los equipos de desarrollo no puedan compilar tantas versiones como les gustaría, y la compilación debe completarse a tiempo para que las pruebas comiencen al día siguiente. Dividir un proyecto en partes más pequeñas y fáciles de administrar podría ser una solución si cada subproyecto pudiera compilarse con mayor rapidez de manera independiente. Sin embargo, un paso de integración donde se combinen todos los subproyectos sigue siendo con frecuencia necesario para que el equipo supervise el proyecto en general y garantice que las distintas partes sigan funcionando correctamente de forma conjunta.

⁷ Consulte: <http://aws.amazon.com/sdkforjava/>

⁸ Consulte: <http://aws.amazon.com/sdkfornet/>

⁹ Consulte: <http://aws.amazon.com/sdkforphp/>

¹⁰ Consulte: <http://aws.amazon.com/sdkforruby/>

¹¹ Consulte: <http://aws.amazon.com/mobile/>

¹² Consulte: <http://aws.amazon.com/developertools/>

¹³ Consulte: http://es.wikipedia.org/wiki/Integraci3n_continua

¹⁴ Consulte: http://en.wikipedia.org/wiki/Nightly_build

Compilaciones bajo demanda

Una solución más práctica es usar más potencia computacional para el proceso de compilación. En entornos tradicionales, donde el servidor de compilación se ejecuta en hardware adquirido por la organización, esta opción puede que no sea viable debido a limitaciones económicas o retrasos de aprovisionamiento. Un servidor de compilación que se ejecute en una instancia Amazon EC2 se puede escalar verticalmente en cuestión de minutos, como ya se explicó en las secciones anteriores, por lo que se reduce el tiempo de compilación al proporcionar más capacidad de memoria o CPU cuando sea necesario.

En el caso de equipos con varias compilaciones que se disparan en un mismo día, es posible que una única instancia Amazon EC2 no sea capaz de producir las compilaciones con suficiente rapidez. Una solución podría ser aprovechar el carácter de pago por uso bajo demanda de Amazon EC2 para usar varias instancias de compilación (nodos de trabajador). Cada vez que el equipo de desarrollo solicita una nueva compilación o se dispara una nueva confirmación en el repositorio de código fuente, el proceso de compilación se distribuye a la flota de nodos de trabajador. La distribución de tareas a los nodos de trabajador se puede realizar con una cola que contenga todas las compilaciones que sea necesario procesar. Cuando los nodos de trabajador estén libres, seleccionarán la próxima compilación que sea necesario procesar. Para implementar este sistema, [Amazon Simple Queue Service](#)¹⁵ (Amazon SQS) ofrece un servicio de cola hospedada, de confianza y altamente escalable. Amazon SQS permite fácilmente crear un flujo de trabajo de compilación automatizada, que funciona estrechamente con Amazon EC2 y otros AWS Infrastructure Services. En esta configuración, los desarrolladores confirmarían el código en el repositorio de código fuente que, a su vez, enviaría un mensaje de compilación a la cola de Amazon SQS. Los nodos de trabajador sondean esta cola. Un nodo de trabajador extraerá un mensaje y ejecutará la compilación de forma local según los parámetros del mensaje (por ejemplo, la rama o la versión del código fuente que se usará).

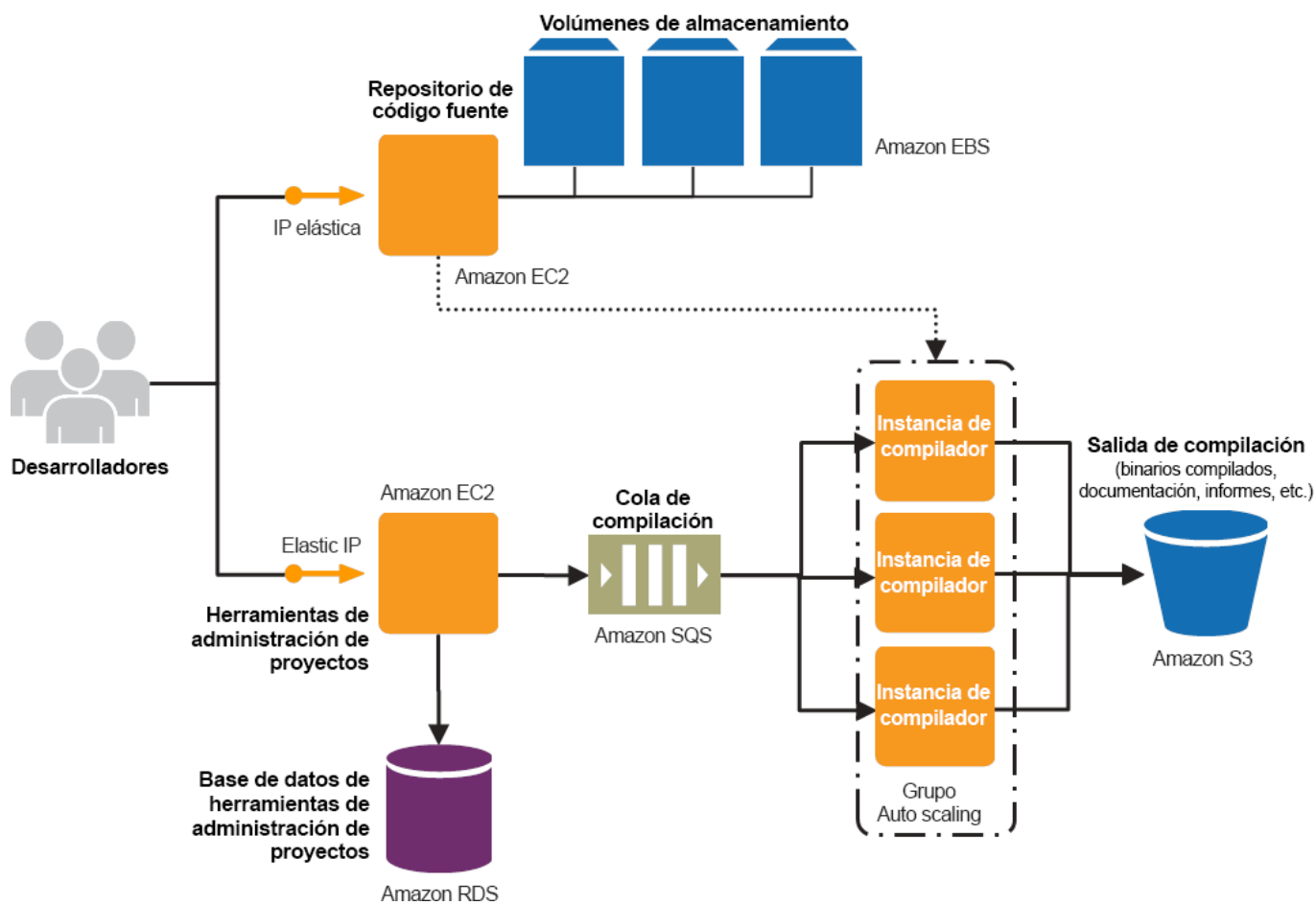
Puede mejorar aún más esta configuración si ajusta de forma dinámica el grupo de nodos de trabajador que usan la cola. [Auto Scaling](#)¹⁶ es un servicio que permite escalar fácilmente el número o los nodos de trabajador automáticamente según condiciones predefinidas. Con Auto Scaling, se puede aumentar fácilmente la capacidad de nodos de trabajador durante picos de demanda para mantener una generación de compilación rápida que se reduzca automáticamente durante los períodos de menor demanda para minimizar los costos. Puede definir las condiciones de escalado con [Amazon CloudWatch](#)¹⁷, un servicio de monitoreo para recursos en la nube de AWS. Por ejemplo, Amazon CloudWatch puede monitorear el número de mensajes en la cola de compilación y notificar a Auto Scaling si se necesita más o menos capacidad, según el número de mensajes en la cola. En el diagrama siguiente, se resume este escenario¹⁸:

¹⁵ Consulte: <http://aws.amazon.com/sqs/>

¹⁶ Consulte: <http://aws.amazon.com/autoscaling/>

¹⁷ Consulte: <http://aws.amazon.com/cloudwatch/>

¹⁸ En la arquitectura de referencia, "Procesamiento por lotes", se describe una configuración más genérica. Consulte: http://d36cz9buwru1tt.cloudfront.net/architecturecenter/AWS_ac_ra_batch_03.pdf



Almacenamiento y distribución de la salida de la compilación

Cada vez que genere una compilación, necesita almacenar los resultados en algún lugar. Para este fin, Amazon S3 es un servicio adecuado. Inicialmente, la cantidad de datos que se almacenará para un proyecto específico puede que sea pequeña, pero crecerá con el tiempo a medida que genere más compilaciones. En este caso, las características de capacidad y pago por uso de Amazon S3 son especialmente atractivas. Cuando ya no necesite la salida de la compilación, puede eliminarla, o bien puede usar las políticas de retención de datos de Amazon S3 para hacerlo.

Para distribuir la salida de la compilación (por ejemplo, para implementarla en entornos de prueba, almacenamiento provisional o producción, o bien para descargarlo en clientes), AWS ofrece varias opciones. Puede distribuir los paquetes de salida de compilación directamente desde Amazon S3, tanto públicamente como mediante la configuración de políticas de buckets o ACL para restringir la distribución. Otra opción es usar [Amazon CloudFront](http://aws.amazon.com/cloudfront/)¹⁹, un servicio web para la entrega de contenido, que permite distribuir fácilmente paquetes a los usuarios finales con baja latencia y elevadas velocidades de transferencia de datos, lo que mejora la experiencia del usuario final. Esto puede resultar útil, por ejemplo, si un gran número de clientes descarga actualizaciones o paquetes de instalación. Amazon CloudFront ofrece varias opciones (por ejemplo, para autorizar o restringir el acceso), aunque una explicación completa está fuera del alcance de este documento.

¹⁹ Consulte: <http://aws.amazon.com/cloudfront/>

Fase de pruebas

Las pruebas son una parte crítica del desarrollo de software. Garantizan la calidad del software, pero, aún más importante, permiten identificar problemas de forma anticipada en la fase de desarrollo, lo que reduce el costo de corregirlos posteriormente durante el proyecto. Existen distintos tipos de prueba: pruebas de unidades, pruebas de desempeño, pruebas de aceptación de usuarios, pruebas de integración, etc., y para todas es necesario ejecutar recursos de TI. Por lo tanto, los equipos de prueba se enfrentan a las mismas dificultades que los equipos de desarrollo: necesitan tener suficientes recursos de TI, pero solo durante el período limitado de la ejecución de pruebas. Además, los entornos de prueba cambian frecuentemente y son distintos en cada proyecto; como resultado, puede que se necesite una infraestructura de TI distinta o capacidad variable.

Las propuestas de valor de pago por uso y bajo demanda de AWS se adaptan bien a esas limitaciones. AWS permite a sus equipos de prueba eliminar tanto la necesidad de hardware costoso como la complicación administrativa derivada de tenerlo en propiedad y mantenerlo en funcionamiento. AWS también ofrece ventajas operativas importantes para los evaluadores: los entornos de prueba se pueden configurar en minutos, en lugar de semanas o meses, y se pueden ejecutar pruebas con una amplia variedad de recursos (incluidos distintos tipos de instancia).

Automatizar los entornos de prueba

Hay un gran número de herramientas de software y marcos disponibles para automatizar el proceso de ejecución de pruebas; pero, para ejecutar esas pruebas, es necesario disponer de una infraestructura adecuada. Esto implica aprovisionar recursos de infraestructura e inicializarlos con un conjunto de datos de ejemplo, implementar el software que se probará, organizar ejecución de pruebas y recopilar los resultados. La complicación aquí no es solo disponer de recursos suficientes para implementar toda la aplicación con todos los distintos servidores o servicios que pueda necesitar, sino también poder inicializar el entorno de prueba con el software adecuado y los datos correctos de forma repetida. Los entornos de prueba deben ser idénticos entre ejecuciones de pruebas; de lo contrario, resultará más difícil comparar los resultados.

Otra ventaja importante de ejecutar pruebas en AWS es la capacidad de automatizarlas de distintas formas. Puede operar AWS mediante programación con las API de AWS, o bien con las herramientas de la interfaz de línea de comandos (CLI). Las tareas que necesitan la intervención humana en los entornos clásicos (asignar un nuevo servidor, asignar y conectar almacenamiento, asignar una base de datos, etc.) se pueden automatizar por completo en AWS. Los evaluadores pueden diseñar conjuntos de pruebas en AWS para automatizar una prueba, incluso hasta el funcionamiento de los componentes (que, tradicionalmente, son dispositivos de hardware estáticos). La automatización mejora la eficiencia de los equipos de prueba al eliminar el esfuerzo de crear e inicializar entornos de prueba, y consigue eliminar las probabilidades de errores al limitar la intervención humana durante la creación de esos entornos. Un entorno de pruebas automatizadas se puede vincular al proceso de compilación siguiendo principios de integración continua.²⁰ Cada vez que se genera correctamente una compilación, se puede aprovisionar un entorno de prueba y ejecutar pruebas automatizadas en esta.

En las secciones siguientes, se describe cómo aprovisionar automáticamente instancias Amazon EC2, bases de datos y entornos completos.

Aprovisionamiento de instancias

Puede aprovisionar fácilmente instancias Amazon EC2 desde AMI²¹. Una AMI encapsula el sistema operativo y cualquier otro software o archivos de configuración, que estarán preinstalados en la instancia. Al iniciar la instancia, todas las aplicaciones ya estarán cargadas desde la AMI y preparadas para su ejecución. Para obtener

²⁰ Consulte: http://es.wikipedia.org/wiki/Integraci3n_continua

²¹ Consulte: <http://docs.amazonwebservices.com/AWSEC2/latest/UserGuide/launching-an-instance.html>

información sobre cómo crear AMI, consulte la [documentación de Amazon EC2](#)²². Las dificultades que presentan las implementaciones basadas en AMI es que, cada vez que necesite actualizar el software, tendrá que crear una nueva AMI. Aunque el proceso de crear una AMI (y eliminar la anterior) se puede automatizar por completo, esto obliga a definir una estrategia de administración y mantenimiento de varias versiones de AMI.

Un método alternativo es incluir solo los componentes en la AMI que no cambian con frecuencia (sistema operativo, plataforma de lenguaje y bibliotecas de bajo nivel, servidor de aplicaciones, etc.). Los componentes más volátiles, como la aplicación en desarrollo, se obtienen e implementan en la instancia en el tiempo de ejecución. Para obtener más información sobre cómo crear instancias con un proceso arranque automático, consulte el documento “[Arranque de aplicaciones en AWS](#)”²³.

Aprovisionar bases de datos

Las bases de datos de prueba se pueden implementar de manera eficiente como instancias de bases de datos de Amazon RDS. Los equipos de prueba pueden crear fácilmente una instancia de una base de datos completamente operativa y cargar un conjunto de datos de prueba desde una instantánea. Para crear este conjunto de datos de prueba, primero se aprovisiona una instancia Amazon RDS. Después de insertar el conjunto de datos, cree una instantánea de la instancia²⁴. A partir de ese momento, cada vez que necesite una base de datos de prueba para un entorno de prueba, podrá crear fácilmente uno como una instancia Amazon RDS a partir de esa instantánea inicial²⁵. Cada instancia Amazon RDS iniciada desde la misma instantánea contendrá el mismo conjunto de datos, lo que permite garantizar la coherencia de las pruebas.

Aprovisionar entornos completos

Aunque puede crear entornos de prueba complejos que contengan varias instancias con las API de AWS, las herramientas de línea de comandos o la consola de administración de AWS, [AWS CloudFormation](#)²⁶ permite crear con mayor facilidad una colección de recursos de AWS relacionados y aprovisionarlos de forma predecible y ordenada. AWS CloudFormation usa plantillas para crear y eliminar una colección de recursos de forma conjunta como una sola unidad (una pila). Un entorno de prueba completo que se ejecute en AWS se puede describir en una plantilla, que es un archivo de texto en formato JSON. Como las plantillas no son más que archivos de texto, puede editarlas y administrarlas en el mismo repositorio de código fuente que use para su proyecto de desarrollo de software. De esta forma, en la plantilla se replicará el estado del proyecto, y los entornos de prueba que coincidan con versiones de código fuente anteriores se podrán aprovisionar fácilmente. Esto es especialmente útil al tratar con errores de regresión: en solo unos pasos, puede aprovisionar todo el entorno de prueba, lo que permite a desarrolladores y evaluadores simular errores de código detectados en versiones anteriores del software. Las plantillas de AWS CloudFormation también admiten parámetros que se pueden usar para especificar que se cargue una versión del software concreta, los tamaños de la instancia Amazon EC2 para el entorno de prueba, el conjunto de datos que se usará para las bases de datos, etc.

Para obtener más información sobre cómo crear y automatizar implementaciones en AWS con AWS CloudFormation, consulte: <http://aws.amazon.com/cloudformation/aws-cloudformation-articles-and-tutorials/>

²² Consulte: <http://docs.amazonwebservices.com/AWSEC2/latest/UserGuide/creating-an-ami.html>

²³ Consulte: <https://s3.amazonaws.com/cloudformation-examples/BoostrappingApplicationsWithAWSCloudFormation.pdf>

²⁴ Consulte: http://docs.amazonwebservices.com/AmazonRDS/latest/UserGuide/USER_CreateSnapshot.html

²⁵ Consulte: http://docs.amazonwebservices.com/AmazonRDS/latest/UserGuide/USER_RestoreFromSnapshot.html

²⁶ Consulte: <http://aws.amazon.com/cloudformation/>

Prueba de carga

Las pruebas de funcionamiento que se realizan en entornos controlados son herramientas útiles para garantizar la calidad del software, pero ofrecen poca información sobre cuál será el desempeño de una aplicación o una implementación completa con una carga elevada. Por ejemplo, algunos sitios web se crean específicamente para prestar un servicio durante un tiempo limitado: ventas de entradas para eventos deportivos, ventas especiales (Black Friday), lanzamientos de edición limitada, etc. Es necesario desarrollar y diseñar cuidadosamente estos sitios web para que su desempeño sea eficiente durante períodos de uso máximo.

En algunos casos, en los requisitos del proyecto se indican claramente las métricas de desempeño mínimo que deben cumplirse en condiciones de carga elevada (por ejemplo, los resultados de búsqueda tienen que devolverse en menos de 100 ms para un máximo de 10 000 solicitudes simultáneas), y se ejecutan pruebas de carga para garantizar que el sistema admita la carga dentro de esos límites. En otros casos, no es posible (o no resulta práctico) especificar la carga que debe mantener un sistema. En esos casos, las pruebas de carga se realizan para medir el comportamiento en condiciones de carga elevada. El objetivo es aumentar de forma progresiva la carga de un sistema con el fin de determinar el punto en que se degrada el desempeño hasta que el sistema deja de funcionar.

Las pruebas de carga simulan entradas de carga elevada que evalúan el desempeño y el esfuerzo de un sistema. Según el proyecto, las entradas pueden ser una gran cantidad de solicitudes entrantes simultáneas, un conjunto de datos enorme que procesar, etc. Una de las principales dificultades en la prueba de carga es ser capaz de generar grandes cantidades de entradas para llevar el sistema probado a sus límites. Normalmente, para esto se necesitan grandes cantidades de recursos de TI para implementar el sistema que se probará, así como para generar la entrada de prueba, que necesita una mayor infraestructura. Como las pruebas de carga no suelen ejecutarse durante más de un par de horas, el modelo de pago por uso de AWS resulta ideal para este caso de uso.

También puede automatizar pruebas de carga con las técnicas descritas en la sección anterior, lo que permite a los evaluadores ejecutarlas con mayor frecuencia para asegurarse de que todos los cambios importantes en el proyecto no afecten de forma negativa a la eficiencia y el desempeño del sistema. Por el contrario, al iniciar pruebas de carga automatizadas, resulta fácil comprobar si un nuevo algoritmo, una capa de almacenamiento en caché o un diseño de arquitectura son más eficientes y benefician al proyecto.

Nota: para realizar una configuración rápida y sencilla, también hay disponibles soluciones y herramientas de pruebas en AWS Marketplace:

<https://aws.amazon.com/marketplace/b/2649283011/>

Pruebas de carga de red

Para realizar una prueba de carga de red de una aplicación o servicio, es necesario enviar un gran número de solicitudes al sistema probado. Existen numerosas soluciones de software disponibles para simular escenarios de solicitudes, pero puede que sea necesario usar varias instancias Amazon EC2 para generar tráfico suficiente. Las instancias Amazon EC2 están disponibles bajo demanda y se cambian cada hora, por lo que resultan adecuadas para escenarios de pruebas de carga de red. Es importante tener en cuenta las características de los diferentes tipos de instancia: como norma general, los tipos de instancia de mayor tamaño proporcionan una mayor capacidad de red de E/S, el recurso principal usado durante las pruebas de carga de red.

Con AWS, los equipos de prueba también pueden realizar pruebas de carga de red en aplicaciones que se ejecuten fuera de AWS. Disponer de agentes de prueba de carga dispersos en distintas regiones de AWS permite realizar pruebas desde distintas geografías (por ejemplo, para comprender mejor la experiencia del usuario final). En ese escenario, tiene sentido recopilar la información de logs de las instancias que simulan la carga. Estos logs contienen información importante, como los tiempos de respuesta del sistema probado. Al ejecutar los agentes de carga desde diferentes regiones, el tiempo de respuesta de las aplicaciones probadas se puede medir desde distintas geografías. Esto le permite comprender la experiencia del usuario mundialmente. Como puede terminar las



instancias Amazon EC2 de pruebas de carga justo después de realizar la prueba, deberá transferir los datos de logs de registro a Amazon S3 para su almacenamiento y posterior análisis.

Prueba de carga para AWS

Realizar una prueba de carga de una aplicación que se ejecute en AWS es útil para asegurarse de que se implementen correctamente las características de elasticidad. Probar la carga de red de un sistema es importante para asegurarse de que las configuraciones de los front-end web, Auto Scaling y Elastic Load Balancing²⁷ sean correctas. Auto Scaling ofrece un gran número de parámetros²⁸ y puede usar varias condiciones definidas con Amazon CloudWatch para ampliar o reducir el número de instancias de front-end. Estos parámetros y condiciones afectan a la rapidez con la que un grupo de Auto Scaling añadirá o quitará instancias. El tiempo posterior al aprovisionamiento de una instancia de Amazon EC2 también podría afectar a la capacidad de una aplicación de escalarse lo suficientemente rápido. Después de inicializar el sistema operativo que se ejecuta en las instancias Amazon EC2, se inicializarán servicios adicionales (como servidores web, servidores de aplicaciones, memorias caché, servicios de middleware, etc.). El tiempo de inicialización de estos servicios afecta al retraso de escalado vertical, especialmente cuando se necesita recuperar paquetes de software adicionales de un repositorio. Las pruebas de carga ofrecerán métricas útiles sobre la rapidez con que puede añadirse capacidad a un sistema específico.

Auto Scaling no solo se usa para sistemas front-end. También se puede usar para grupos internos de escalado de instancias, como consumidores que realicen sondeos en una cola de Amazon SQS²⁹, o bien empleados y responsables de la toma de decisiones que participen en un flujo de trabajo de Amazon Simple Workflow Service (Amazon SWF)³⁰. En ambos casos, la prueba de carga del sistema puede permitirle garantizar que implementó y configuró correctamente los grupos de Auto Scaling u otras técnicas de escalado automatizadas, con el fin de que la aplicación final sea lo más rentable y escalable posible.

Optimización de costos con instancias de spot

Para realizar las pruebas de carga, es posible que se necesiten varias instancias, especialmente cuando se ejecuten sistemas diseñados para admitir cargas muy elevadas. Aunque puede aprovisionar instancias Amazon EC2 bajo demanda y descartarlas cuando se complete la prueba (y solo pagar por una hora), existe una forma incluso más rentable de realizar estas pruebas con instancias de spot de Amazon EC2. Las [instancias de spot](#)³¹ permiten a los clientes pujar por la capacidad no usada de Amazon EC2. Las instancias se cobran según el precio de spot, establecido por Amazon EC2. Este precio fluctúa periódicamente en función de la oferta y la demanda de capacidad de instancias de spot. Para usar instancias de spot, necesita realizar una solicitud de instancia de spot, donde especificará el tipo de instancia, la zona de disponibilidad deseada³², el número de instancias de spot que se ejecutarán y el precio máximo que desee pagar por hora y por instancia. Puede consultar el historial de precios de spot de los últimos 90 días mediante la API de Amazon EC2 y la consola de administración de AWS³³. Si la puja de precio máximo supera el precio de spot actual, se completará la solicitud y se iniciarán y ejecutarán las instancias hasta que se terminen, o bien hasta que el precio de spot supere al precio máximo (lo que ocurra antes).

²⁷ Consulte el artículo técnico “Prácticas recomendadas al evaluar Elastic Load Balancing”, que está disponible en <http://aws.amazon.com/articles/1636185810492479>.

²⁸ Consulte la documentación de Auto Scaling:

<http://docs.amazonwebservices.com/AutoScaling/latest/DeveloperGuide/Welcome.html>

²⁹ Consulte el artículo de blog: <http://aws.typepad.com/aws/2011/07/additional-cloudwatch-metrics-for-amazon-sqs-and-amazon-sns.html>

³⁰ Consulte el artículo de blog: <http://aws.typepad.com/aws/amazon-simple-workflow-service/>

³¹ Consulte: <http://aws.amazon.com/ec2/spot-instances/>

³² Una zona de disponibilidad es una ubicación distinta dentro de una región diseñada para estar aislada de los errores que se produzcan en otras zonas de disponibilidad, y ofrece una conectividad de red de latencia baja y económica para otras zonas de disponibilidad en la misma región.

³³ El precio de spot actual está disponible aquí: <http://aws.amazon.com/ec2/spot-instances/#6>

Para obtener más información sobre las instancias de spot, consulte: <http://aws.amazon.com/ec2/spot-instances/>. Aquí encontrará un caso práctico sobre cómo usar instancias de spot para la prueba de carga: <http://aws.amazon.com/solutions/case-studies/browsermob/>.

Pruebas de aceptación de usuario

El objetivo de las pruebas de aceptación de usuario es presentar la versión actual a un equipo de pruebas que represente la base de usuarios final para determinar si se cumplen los requisitos y las especificaciones del proyecto. Si los usuarios pueden probar el software con antelación, podrán identificar problemas conceptuales que se introdujeron durante la fase de análisis, así como aclarar áreas poco definidas en los requisitos del proyecto. Al probar el software con más frecuencia, los usuarios pueden identificar con antelación errores de implementación funcionales o ideas equivocadas en el flujo de la aplicación o en la interfaz de usuario, lo que reduce el costo y el impacto de corregirlos. Los defectos detectados por las pruebas de aceptación de usuario pueden ser muy difíciles de detectar mediante otros métodos. Si realiza pruebas de aceptación con más frecuencia, mejor será para el proyecto, ya que los usuarios finales ofrecen comentarios útiles a los equipos de desarrollo a medida que evolucionan los requisitos.

Sin embargo, al igual que con cualquier otro procedimiento de pruebas, para realizar las pruebas de aceptación se necesitan recursos para ejecutar el entorno donde se implementará la aplicación. Como se describe en las secciones anteriores, AWS ofrece capacidad bajo demanda según sea necesario y de una forma rentable, lo que también es muy adecuado para las pruebas de aceptación. Al usar algunas de las técnicas descritas anteriormente, AWS permite automatizar por completo el proceso de aprovisionar nuevos entornos de prueba y eliminar los entornos que ya no sean necesarios. Los entornos de prueba solo se pueden proporcionar para períodos específicos, o bien de forma continua para la versión del código fuente más reciente (o para cada versión principal).

Al implementar el entorno de prueba de aceptación en Amazon VPC, los usuarios internos pueden obtener acceso de forma transparente a la aplicación que se probará. Además, como una aplicación también se puede integrar con otros servicios de producción dentro de la empresa (como LDAP, servidores de correo electrónico, etc.), se ofrece un entorno de prueba a los usuarios finales que será incluso más próximo al entorno de producción final y real.

Pruebas en paralelo

Las pruebas en paralelo son un método usado para comparar un sistema de control con un sistema de prueba. El objetivo es evaluar si los cambios aplicados en el sistema de prueba mejoran una métrica deseada, en comparación con el sistema de control. Puede usar esta técnica para optimizar el desempeño de sistemas complejos, donde una multitud de distintos parámetros podrían afectar a la eficiencia global. Conocer qué parámetro tendrá el efecto deseado no siempre resulta obvio, especialmente cuando se usan varios componentes de forma conjunta, y esto afecta al desempeño del resto de los componentes. También puede usar esta técnica al introducir cambios importantes en un proyecto, como nuevos algoritmos, memorias caché, distintos motores de bases de datos o software de terceros. En esos casos, el objetivo es asegurarse de que los cambios afecten de forma positiva al desempeño global del sistema.

Después de implementar los sistemas de control y de prueba, envíe la misma entrada a ambos con técnicas de cargas de prueba o entradas de prueba sencillas. Por último, recopile las métricas de desempeño y los logs de ambos sistemas, y compárelos para determinar si los cambios que introdujo en el sistema de prueba ofrecen una mejora en comparación con el sistema de control.

Al aprovisionar entornos de prueba completos bajo demanda, puede ejecutar de manera eficiente pruebas en paralelo. Aunque puede realizar pruebas en paralelo sin un aprovisionamiento automático del entorno, al usar las

técnicas de automatización descritas anteriormente, resulta más fácil ejecutar esas pruebas cuando sea necesario, ya que podrá aprovechar el modelo de pago por uso de AWS. En cambio, con el hardware tradicional, es posible que no pueda ejecutar fácilmente y de forma simultánea varios entornos de prueba para distintos proyectos.

Las pruebas en paralelo también resultan útiles desde un punto de vista de optimización de costos: al comparar los dos entornos en distintas cuentas de AWS, podrá obtener fácilmente índices de costo y desempeño para comparar los dos entornos. Al probar de forma continua el desempeño de costos de los cambios de arquitectura, puede optimizar las arquitecturas para la eficiencia.

Pruebas de tolerancia a errores

Si AWS es el entorno de producción objetivo para la aplicación que desarrolló, existen procedimientos de prueba específicos que proporcionan información sobre cuál será el comportamiento del sistema ante casos no previstos, como errores de componentes. AWS ofrece un gran número de opciones para crear sistemas con tolerancia a errores. Algunos servicios ya cuentan por definición con tolerancia a errores (por ejemplo, Amazon S3, Amazon DynamoDB, Amazon SimpleDB, Amazon SQS, Amazon Route 53, Amazon CloudFront, etc.). Otros servicios (como Amazon EC2, Amazon EBS y Amazon RDS) ofrecen características que permiten diseñar sistemas con tolerancia a errores y alta disponibilidad. Por ejemplo, Amazon RDS ofrece la opción Zona de disponibilidad múltiple³⁴, que mejora la disponibilidad de las bases de datos al aprovisionar y administrar automáticamente una réplica en otra zona de disponibilidad. Para obtener más información sobre cómo crear arquitecturas con tolerancia a errores que se ejecuten en AWS, consulte el documento técnico “[Crear aplicaciones con tolerancia errores en AWS](#)”³⁵ y los recursos disponibles en el [Centro de arquitectura de AWS](#)³⁶.

Muchos clientes de AWS ejecutan aplicaciones críticas en AWS y necesitan asegurarse de que su arquitectura cuente con tolerancia a errores. Como resultado, un procedimiento importante para todos los sistemas es probar su capacidad de tolerancia a errores. Aunque un escenario de prueba evalúa el desempeño del sistema (al usar técnicas similares para las pruebas de carga), algunos componentes se desactivan de forma intencionada para comprobar si el sistema puede recuperarse ante dicho error simulado. Puede usar la consola de administración de AWS o la interfaz de línea de comandos para interactuar con el entorno de prueba. Por ejemplo, puede terminar instancias Amazon EC2 y, por lo tanto, probar si un grupo de Auto Scaling funciona según lo esperado y se aprovisiona automáticamente una instancia de sustitución. También puede automatizar este tipo de prueba. Le recomendamos que use herramientas automatizadas que, por ejemplo, terminen de forma ocasional y aleatoria instancias Amazon EC2.

Administración de recursos

Con AWS, sus equipos de desarrollo y pruebas pueden disponer de sus propios recursos, escalados según sus necesidades. El aprovisionamiento de entornos complejos o plataformas compuestas de varias instancias se puede realizar fácilmente con pilas de AWS CloudFormation, o mediante otras de las técnicas de automatización descritas. En grandes organizaciones formadas por varios equipos, es recomendable crear un rol interno o servicio responsable de centralizar y administrar los recursos de TI que se ejecuten en AWS. Este rol suele estar compuesto por:

- La promoción de los procedimientos internos de desarrollo y pruebas descritos aquí.
- El desarrollo y mantenimiento de AMI de plantilla y pilas de AWS CloudFormation de plantilla con las diferentes herramientas y plataformas usadas en la organización.

³⁴ Consulte: <http://docs.amazonwebservices.com/AmazonRDS/latest/UserGuide/Concepts.DBInstance.html#Concepts.MultiAZ>

³⁵ Disponible en: http://d36cz9buwru1tt.cloudfront.net/AWS_Building_Fault_Tolerant_Applications.pdf

³⁶ Consulte: <http://aws.amazon.com/architecture/>

- La recopilación de solicitudes de recursos de equipos de proyecto y el aprovisionamiento de recursos en AWS según las políticas de su organización, como la configuración de red (por ejemplo, Amazon VPC) y configuraciones de seguridad (por ejemplo, grupos de seguridad y credenciales de IAM).
- Monitorear el uso de recursos y los cargos con Amazon CloudWatch, y asignar estos a presupuestos de equipos.

Aunque puede usar la consola de administración de AWS para completar las tareas anteriores, le recomendamos que desarrolle su propio aprovisionamiento interno y portal de administración para conseguir una integración más estrecha con los procesos internos. Para hacerlo, puede usar uno de los AWS SDK, que permiten obtener acceso mediante programación a recursos que se ejecutan en AWS.

Asignación de costos y varias cuentas de AWS

A algunos clientes les resulta útil crear cuentas específicas para actividades de desarrollo y pruebas. Esto puede ser importante cuando el entorno de producción también se ejecuta en AWS y necesita equipos y responsabilidades separados. Las cuentas separadas se aíslan entre sí de forma predeterminada, por lo que, por ejemplo, los usuarios de desarrollo y pruebas no interfieren con los recursos de producción. Para facilitar la colaboración, AWS ofrece distintas características que permiten el uso compartido de recursos en distintas cuentas (por ejemplo, objetos de Amazon S3, AMI e instantáneas de Amazon EBS).

AWS ofrece varias opciones para desglosar y asignar el costo de las distintas actividades y fases del ciclo de desarrollo y pruebas. Una opción es usar cuentas distintas (por ejemplo, para desarrollo, pruebas, almacenamiento provisional y producción), donde cada cuenta tendrá su propia factura. También puede consolidar varias cuentas (por ejemplo, para simplificar los pagos). Otra opción es usar el Informe de asignación de costos, que le permite organizar y realizar un seguimiento de sus costos de AWS con el etiquetado de recursos. En el contexto de desarrollo y pruebas, las etiquetas pueden representar las distintas fases o equipos del ciclo de desarrollo, aunque puede elegir libremente las dimensiones que le resulten más útiles.

Conclusión

Los procedimientos de desarrollo y pruebas exigen el uso de recursos específicos en momentos concretos del ciclo de desarrollo. En los entornos tradicionales, puede que sus recursos no estén disponibles, o bien no lo estén en el período de tiempo necesario. Cuando esos recursos están disponibles, ofrecen una capacidad específica que puede resultar insuficiente (especialmente en actividades variables como pruebas) o puede desperdiciarse (pero pagando por la capacidad) si no se usan los recursos³⁷.

Amazon Web Services ofrece una alternativa rentable a las infraestructuras tradicionales de desarrollo y pruebas. En lugar de esperar semanas o incluso meses para el hardware, puede aprovisionar al instante los recursos necesarios, escalar al instante a medida que crecen las cargas de trabajo y liberar recursos cuando ya no sean necesarios. Independientemente de si los entornos de desarrollo y pruebas están formados por un par de instancias o por cientos de ellas, si se necesitan para unas pocas horas de forma ininterrumpida, solo pagará por lo que use. AWS es una plataforma que no está vinculada a sistemas operativos ni lenguajes de programación específicos, y le permite elegir la plataforma de desarrollo o el modelo de programación que se use en su negocio. Gracias a esta flexibilidad, puede centrarse en su proyecto, no en el funcionamiento y el mantenimiento de su infraestructura.

AWS también permite otras posibilidades que eran difíciles de alcanzar con el hardware tradicional. Puede automatizar por completo los recursos en AWS para aprovisionar y retirar entornos sin la intervención humana. Puede iniciar entornos de desarrollo bajo demanda; activar compilaciones cuando sea necesario, sin limitaciones de

³⁷ Consulte: <http://aws.amazon.com/economics/>

disponibilidad de recursos; aprovisionar recursos de prueba; y organizar automáticamente campañas o ejecuciones de pruebas completas.

AWS le permite experimentar y procesar iteraciones con una infraestructura que se puede modificar rápidamente. Los equipos de proyecto pueden usar una capacidad económica para realizar cualquier tipo de pruebas o experimentar con nuevas ideas, sin gastos iniciales ni compromisos a largo plazo, por lo que AWS es una plataforma ideal para procedimientos de desarrollo y pruebas.

Documentación adicional

- Documento técnico: “Listas de comprobación de funcionamiento de AWS”
http://media.amazonwebservices.com/AWS_Operational_Checklists.pdf
- Documento técnico: “Cómo funcionan los precios de AWS”
http://media.amazonwebservices.com/AWS_Pricing_Overview.pdf
- Centro de arquitectura de AWS <http://aws.amazon.com/architecture>
- Documentos técnicos de AWS <http://aws.amazon.com/whitepapers/>