



***Alojamiento de aplicaciones Web en la nube de AWS***  
***Las mejores prácticas***

***Mayo de 2010***

***Matt Tavis***

## Resumen

El alojamiento Web ampliable y de alta disponibilidad puede ser un asunto complejo y caro. Las arquitecturas de Web ampliables tradicionales, no solamente han necesitado implementar soluciones complejas para garantizar altos niveles de fiabilidad, sino que también han requerido una predicción precisa del tráfico para ofrecer un alto nivel de servicio al cliente. Los periodos de tráfico denso y las oscilaciones tremendas en las pautas del tráfico causan una tasa de utilización baja de equipos de coste elevado, generan altos costes operativos para mantener el hardware sin uso, y además producen un uso ineficiente del capital dedicado a equipos infrautilizados. Amazon Web Services proporciona la infraestructura fiable, ampliable, segura y de altas prestaciones que requieren las aplicaciones de Web más exigentes, al tiempo que posibilita un modelo de infraestructura elástico y de escala dinámica, de forma que los costes de TTI sean acordes con las pautas de tráfico de los clientes en tiempo real.

## Audiencia

Este documento técnico se dirige a directores de TTI y arquitectos de sistemas que se planteen el uso de la nube para afrontar sus necesidades de ampliación e informática a demanda.

## Aspectos generales de los alojamientos Web tradicionales

El alojamiento Web ampliable es un problema bien conocido y esta sección del documento no presentará demasiadas sorpresas a las personas que ya conocen los modelos de alojamientos Web tradicionales. No obstante, esta sección presentará un ejemplo de arquitectura de alojamientos Web estándar que se utilizará a título comparativo cuando se plantee la forma de implementar esta arquitectura en la nube.

### Ejemplo de arquitectura de alojamientos de aplicaciones Web tradicionales.

A continuación se incluya una ilustración clásica de una arquitectura de alojamientos Web de modelo tradicional.

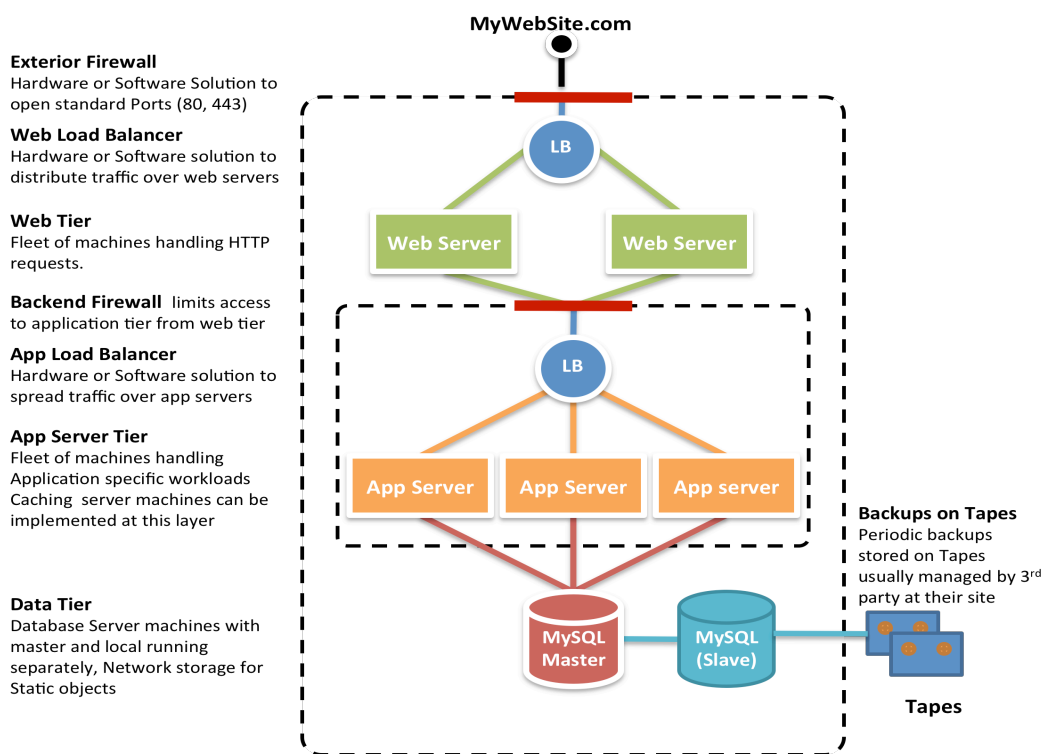


Figura 1 - Arquitectura tradicional de alojamiento Web

Esta arquitectura tradicional de alojamiento Web se construye alrededor de un modelo común de aplicaciones en tres niveles que separa la arquitectura en capas de presentación, aplicaciones y persistencia. Esta arquitectura ya se ha diseñado para ampliaciones mediante la adición de hosts adicionales en las capas de presentación, persistencia o aplicaciones e integra prestaciones, protección contra errores y funciones de disponibilidad. Las secciones siguientes describen por qué y de qué forma se puede y se debe migrar una arquitectura a la nube de Amazon Web Services.

## Alojamiento de aplicaciones Web en la nube con Amazon Web Services.

La arquitectura tradicional de alojamiento Web (figura 1) es fácil de portar a los servicios de nube que ofrecen los productos de AWS; bastan unas pocas modificaciones, pero la primera pregunta que hay que plantearse es si es una idea sensata trasladar esta aplicación a la nube:

*¿Qué valor ofrece el traslado de una solución clásica de alojamiento de aplicaciones Web a la nube de AWS?*

### Cómo puede AWS solucionar los programas más comunes del alojamiento de aplicaciones Web

Si usted es responsable de ejecutar una aplicación Web, deberá afrontar una serie de cuestiones de infraestructura y arquitectura, y ahí es donde AWS puede ofrecerle soluciones sencillas, impecables y a buen precio. A continuación se describen algunas de las numerosas ventajas de utilizar AWS, en comparación con un modelo de alojamiento tradicional:

#### Una alternativa económica a las flotas sobredimensionadas necesarias para atender los picos de demanda

En el modelo tradicional de alojamiento, es necesario disponer de servidores para atender los picos de capacidad, y los ciclos no utilizados se desperdician fuera de los periodos pico. Las aplicaciones Web alojadas en AWS pueden aprovechar la disposición de servidores adicionales a demanda, con lo que la capacidad y los costes siguen el modelo de tráfico.

Por ejemplo, el gráfico siguiente muestra una aplicación Web con un pico entre las 9 de la mañana y las 3 de la tarde, que tiene menos uso durante el resto del día. Un planteamiento de ampliación automática basado en un seguimiento de las tendencias del tráfico real y la disposición de recursos solamente cuando sea necesario tendría el resultado de una reducción de más del 50% en capacidad y costes.

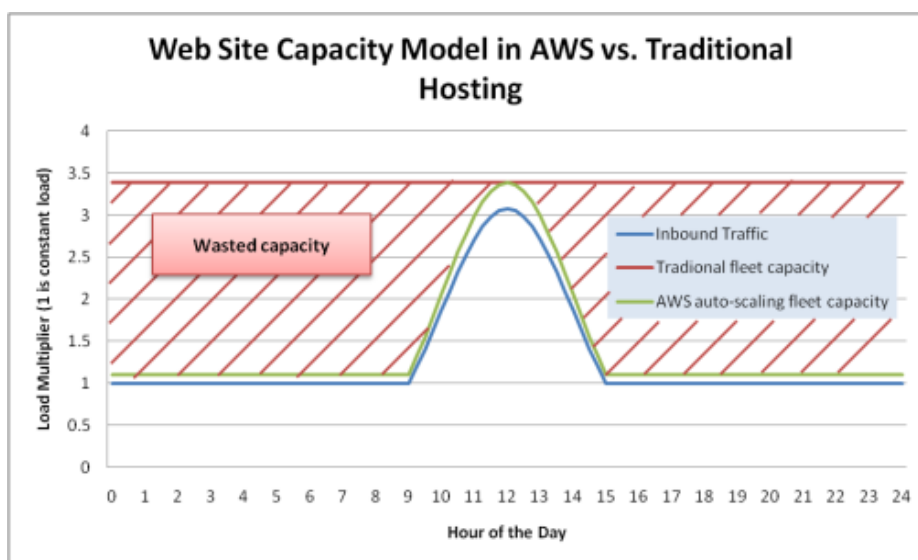


Figura 2 - Ejemplo de capacidad desperdiciada en un modelo de alojamiento clásico

## **Una solución ampliable para atender los picos de tráfico inesperados**

Una consecuencia aún más nefasta de la lentitud en la disposición de equipos asociada con el modelo tradicional de alojamiento es la incapacidad de responder a tiempo a los picos de tráfico inesperados. Existen numerosas historias de aplicaciones Web que dejan de funcionar debido a la aparición de picos de tráfico inesperados como consecuencia de aparecer en medios de comunicación de amplia difusión. La misma capacidad a demanda que ayuda a ajustar las dimensiones de las aplicaciones Web para que se adapten a los picos de tráfico también puede utilizarse para atender una carga inesperada, ya que es posible lanzar nuevos hosts y tenerlos listos en cuestión de minutos.

## **Una solución a demanda para entornos de pruebas, de carga, beta y preproducción**

Los costes de equipos necesarios para configurar un entorno de alojamiento tradicional para aplicaciones de producción Web no terminan una vez adquirida la flota de producción. Muchas veces, se crean flotas de entornos beta y de pruebas para garantizar la calidad de la aplicación Web en todas las fases del ciclo de vida de desarrollo antes de lanzarla en la flota de producción. Si bien es posible realizar varias optimizaciones para garantizar el uso máximo de este equipo de pruebas, en ocasiones sucede que estas flotas paralelas no se utilizan de forma óptima. Gran cantidad de equipo caro pasa largos periodos de tiempo sin uso. Cuando se utiliza la nube de AWS, es posible instaurar flotas de pruebas según sea necesario para garantizar que dichas flotas estén disponibles únicamente cuando sea necesario. La nube de AWS puede utilizarse además para simular tráfico de usuarios en pruebas de carga de aplicaciones candidatas a publicación. Estas flotas paralelas también pueden utilizarse como entorno intermedio en versiones de nueva producción, lo que permite cambiar desde la producción actual hasta una nueva versión de aplicación sin interrupciones, o con mínimas interrupciones, del servicio.

## Una arquitectura de nube de AWS para alojamiento Web

A continuación se muestra una vez más la arquitectura clásica de aplicaciones Web y cómo podría aprovechar la infraestructura de informática en la nube de AWS:

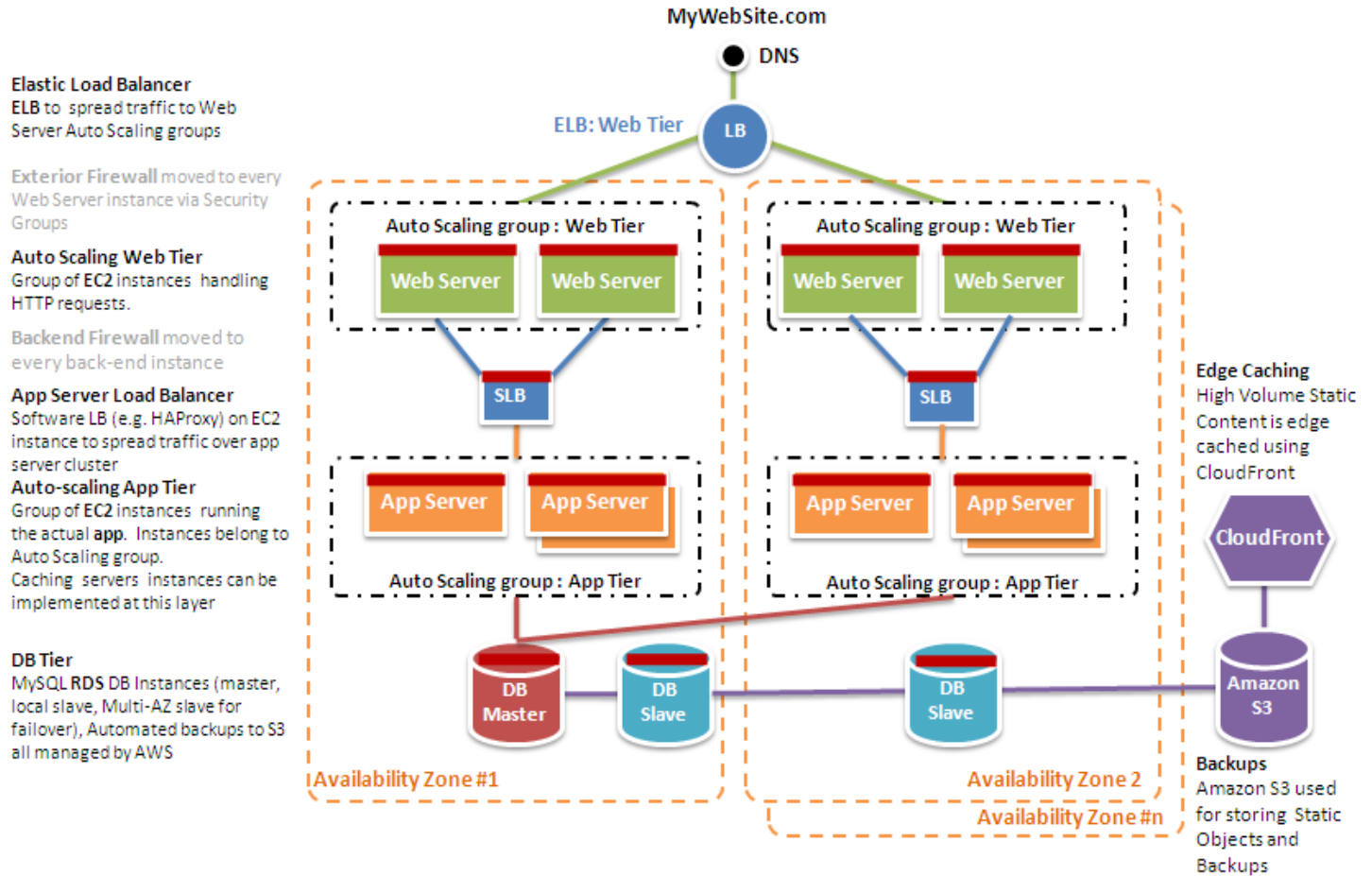


Figura 3 - Ejemplo de arquitectura de alojamiento Web en AWS

## Componentes clave de una arquitectura de alojamiento Web en AWS

Las siguientes secciones describen varios componentes clave de una arquitectura de alojamiento Web de AWS y explica en qué se diferencian de una arquitectura tradicional de alojamiento Web.

### Entrega de contenido

Todavía sigue siendo necesario el uso de memoria caché en servidores propios cuando se utiliza la infraestructura informática de la nube de Amazon Web Service. Cualquier solución que tenga instalada en su infraestructura de aplicaciones Web debería funcionar bien con la nube de AWS. No obstante, existe otra opción al utilizar AWS,

consistente en utilizar el servicio Amazon CloudFront <sup>1</sup> como caché en servidores propios para los activos de aplicaciones almacenados en el servicio de almacenamiento sencillo de Amazon <sup>2</sup>(Amazon S3). Una ventaja crucial de utilizar una solución de caché en servidores propios con Amazon EC2 es la capacidad de acelerar las prestaciones de la aplicación a la vista de sus clientes mediante un punto de presencia (POP) en la caché local, gracias al cual la carga de contenidos en secuencias o estáticos descargados (p. ej. vídeos Flash o imágenes) es mucho más rápida, y a que el contenido se sirve desde una ubicación más próxima. Otra ventaja de la memoria caché en servidores propios de CloudFront es que sigue el modelo de AWS *sin compromisos, sin mínimos y sin contratos*, lo que le permite utilizar lo que necesite, y solamente durante el tiempo que lo necesite, como servicio de caché en servidores propios.

## Gestión de DNS públicos mediante CNAME e IP elástica

Para migrar una aplicación Web a la nube de AWS es necesario realizar varios cambios en el DNS. AWS no ofrece una gestión de DNS público, de forma que para redirigir el tráfico de Internet público hasta la aplicación de la nube de AWS requeriría cambiar el DNS público para dirigirlo a un CNAME de equilibrio elástico de cargas o a una dirección IP elástica. Sin embargo, el DNS restringe el uso de CNAME a subdominios, de forma que el dominio raíz (p. ej. ejemplo.com) no puede apuntar a un CNAME equilibrador elástico de cargas. Tenga en cuenta que las direcciones IP situadas detrás de un equilibrador elástico de cargas pueden cambiar, así que en la actualidad no es posible dirigir el registro A del DNS raíz a las direcciones IP situadas detrás del CNAME equilibrador elástico de cargas. Una solución alternativa sencilla consiste en asignar direcciones IP elásticas (direcciones IP estáticas de asignación dinámica) a dos o más servidores Web de EC2 en la aplicación, y hacer que dichos servidores Web redirijan el tráfico Web al subdominio correspondiente que dirige el tráfico al CNAME equilibrador elástico de cargas (p. ej. www.ejemplo.com). La empresa de registro de nombre de dominio utilizada para adquirir su dominio público debería proporcionar un mecanismo sencillo para aplicar el CNAME equilibrador elástico de cargas al subdominio correspondiente (p. ej. www.ejemplo.com) y para configurar la lista de direcciones IP elásticas para los registros A del dominio raíz (p. ej. ejemplo.com).

## Seguridad del host

A diferencia del modelo tradicional de alojamiento Web, el filtrado de tráfico entrante a la red no debería confinarse a los propios servidores, sino a nivel de host. Amazon EC2 incluye una función denominada *grupos de seguridad*, que actúan como un cortafuegos de entrada a la red y permiten especificar los protocolos, los puertos y los rangos de direcciones IP que están autorizados para acceder a las instancias de EC2. Es posible asignar a cada instancia de EC2 uno o varios grupos de seguridad que dirijan el tráfico correspondiente a cada instancia. Es posible configurar grupos de seguridad de forma que únicamente determinadas subredes o direcciones IP tengan acceso a la instancia de EC2, o incluir en ellos referencias a otros grupos de seguridad para limitar el acceso a instancias de EC2 que formen parte de grupos específicos. Por ejemplo, en el caso de arquitectura de alojamiento Web de AWS que se muestra arriba, el grupo de seguridad de la agrupación de servidores Web podría tener acceso únicamente a los hosts mediante el protocolo TCP y a través de los puertos 80 y 443 (HTTP y HTTPS) y desde instancias del grupo de seguridad de servidor de aplicaciones en el puerto 22 (SSH) para gestión directa del host. Por otra parte, el grupo de seguridad de la agrupación de servidores de aplicaciones podría permitir el acceso desde el grupo de seguridad de servidores Web para gestionar las solicitudes de Web y desde la subred corporativa mediante el protocolo TCP a través del puerto 22 (SSH) para gestión directa de los

---

<sup>1</sup> Amazon CloudFront - <http://aws.amazon.com/cloudfront/>

<sup>2</sup> Amazon S3 – <http://aws.amazon.com/s3>

puertos. En este modelo, los ingenieros de asistencia al cliente podrían iniciar sesión directamente en los servidores de aplicaciones desde la red corporativa para luego acceder al resto de las aplicaciones desde los cuadros de los servidores de aplicaciones. El centro de seguridad de AWS describe las cuestiones de seguridad más a fondo<sup>3</sup>. Este sitio contiene boletines de seguridad, información de certificaciones y documentos técnicos de seguridad que explican las capacidades de seguridad de AWS.

¿Para qué sirve la siguiente ilustración?

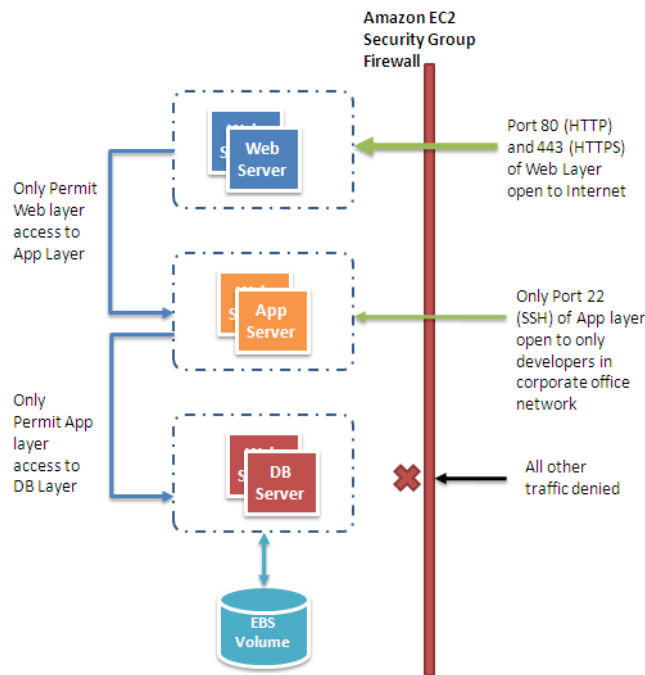


Figura 4: Grupos de seguridad en una aplicación Web

## Equilibrio de carga entre agrupaciones

Los equilibradores de carga de equipos son dispositivos de uso común en redes y en arquitectura de aplicaciones Web. AWS ofrece esta misma capacidad mediante el servicio de equilibrado elástico de cargas (Elastic Load Balancing)<sup>4</sup>, que es una solución configurable para el equilibrio de cargas que permite controlar el estado de funcionamiento de los hosts, la distribución de tráfico a instancias de EC2 en varias zonas de disponibilidad y la adición y eliminación dinámicas de hosts de EC2 en la rotación de equilibrado de carga. El equilibrado elástico de cargas también puede reducir y aumentar dinámicamente la capacidad de equilibrado de cargas para atender las demandas de aumento y reducción de tráfico, al tiempo que ofrece un punto de entrada predecible por medio de un CNAME persistente. El servicio de equilibrado elástico de cargas también permite realizar sesiones persistentes para afrontar necesidades de enrutamiento más complejas. Si la aplicación necesita capacidades más avanzadas de equilibrado de cargas, un planteamiento alternativo sería utilizar un paquete de software para equilibrado de cargas por encima de las instancias de EC2 (p. ej. Zeus,

<sup>3</sup> Centro de seguridad de AWS – <http://aws.amazon.com/security>

<sup>4</sup> Equilibrado de cargas elásticas de Amazon - <http://aws.amazon.com/elasticloadbalancing>



HAProxy, nginx) y asignar IP elásticas a dichas instancias de EC2 con equilibrado de carga a fin de minimizar los cambios en DNS.

## Búsqueda de otros hosts y servicios

Otro cambio respecto a la arquitectura tradicional de alojamiento Web es que la mayoría de los hosts tendrán direcciones IP dinámicas. Aunque cada instancia de EC2 puede tener entradas de DNS tanto públicas como privadas y puede incluirse en direcciones de Internet, las entradas de DNS y las direcciones IP se asignan dinámicamente al comenzar la instancia y no pueden asignarse manualmente. Es posible asignar direcciones IP estáticas (denominadas direcciones IP elásticas en AWS) a instancias en ejecución una vez que se hayan lanzado, pero solamente los hosts de la nube de AWS con direcciones IP elásticas tendrán puntos finales coherentes para comunicaciones en la red. Las direcciones IP elásticas deberán utilizarse en instancias y servicios que requieran puntos finales coherentes, como por ejemplo bases de datos maestras, servidores centrales de archivos y equilibradores de carga alojados en EC2. Los tipos de instancia que pueden adaptar su tamaño fácilmente, como por ejemplo los servidores Web, deberán estar disponibles para los descubran sus puntos finales dinámicos por medio de los servicios más persistentes que se describieron anteriormente. Una solución simple para esto consiste en mantener centralmente la configuración de las instancias y los puntos finales de servicios de red necesarios, a los que es posible acceder mediante puntos finales coherentes basados en direcciones IP elásticas que pueden utilizarse durante el arranque de la instancia por medio de scripts de tipo bootstrap. Dado que la mayoría de las arquitecturas de aplicaciones Web tienen un servidor de bases de datos que está siempre encendido, se trata de un repositorio común para este tipo de información de configuración. Es importante tener en cuenta la opción de usar instancias reservadas<sup>5</sup> para los servicios persistentes de la infraestructura de EC2 con objeto de reducir aún más los costes. Con este modelo, los hosts que se añadan pueden solicitar la lista de puntos finales necesarios para las comunicaciones de la base de datos dentro de una fase de bootstrap. La ubicación de la base de datos puede suministrarse como datos de usuario<sup>6</sup> que se pasan a cada instancia durante su lanzamiento. Otra alternativa consiste en utilizar el servicio SimpleDB para almacenar y conservar información de configuración, ya que es un servicio de alta disponibilidad que está disponible en un punto final bien definido.

## Caché dentro de la aplicación Web

Lo más probable es que las soluciones de caché basadas en software ya existentes en su arquitectura de aplicaciones Web también puedan usarse tal y como están en la nube de AWS. Basta con construir una instancia de EC2 con la solución de software de caché para habilitar la memoria caché en la nube de AWS. La memoria caché de Web y de la capa de aplicaciones puede activarse así, y la configuración centralizada de la base de datos puede ayudar a los servidores de Web y de aplicaciones a encontrar los servidores de caché correspondientes.

## Configuración de la base de datos, copias de seguridad y sistema a prueba de fallos

Muchas aplicaciones Web contienen algún tipo de persistencia, habitualmente en forma de base de datos. AWS ofrece dos soluciones principales para las bases de datos en la nube; alojar una base de datos relacional (RDBMS) en una instancia de Amazon EC2, o utilizar el servicio de bases de datos relacionales (RDS) de Amazon para una solución RDBMS

<sup>5</sup> Instancias reservadas - <http://aws.amazon.com/ec2/reserved-instances/>

<sup>6</sup> Datos de usuarios - <http://docs.amazonwebservices.com/AWSEC2/latest/APIReference/index.html?ApiReference-ItemType-RunInstancesType.html>

alojada. AWS admite numerosas soluciones de bases de datos en EC2, incluido MySQL, Oracle, SQLServer y DB2. El uso del servicio RDS ofrece la conectividad (p. ej., ODBC o JDBC) a la que los desarrolladores están acostumbrados, pero ofrece una administración simplificada por medio de API de servicio Web. Las tareas habituales, como la adición de almacenamiento, las copias de seguridad de datos y la migración de la base de datos a una instancia de EC2 de mayor tamaño pueden automatizarse mediante sencillas llamadas a las API. Al igual que sucede en los modelos de alojamiento tradicionales, las soluciones de bases de datos en la nube deberán tener tanto instancias maestras como instancias esclavas para permitir las copias de seguridad y el sistema a prueba de fallos. Los clientes de AWS que alojen una base de datos en EC2 han utilizado de forma satisfactoria diferentes modelos de réplicas maestro/esclavo en instancias de EC2, incluido el uso de sistemas de espejo para copias de sólo lectura y envío de registros para esclavos pasivos que están siempre listos. Muchas veces, las aplicaciones Web utilizan una copia de seguridad de base de datos y un modelo a prueba de fallos, y lo más probable es que muchos de ellos resulten fáciles de replicar en la nube de AWS. Los clientes de AWS que utilizan RDS disponen de mecanismos de copia de seguridad y sistemas a prueba de fallos contruidos mediante sencillas llamadas a API. Implementar un sistema RDBMS en varias zonas de disponibilidad que utilicen varios esclavos para protección a prueba de fallos es una opción recomendable para maximizar la disponibilidad de la base de datos. Utilizar una segunda zona de disponibilidad es muy parecido a tener un centro de datos de respaldo, ya que cada zona de disponibilidad está completamente separada de las demás zonas de la misma región con objeto de garantizar la máxima disponibilidad de la región. Los clientes de AWS que utilizan el servicio RDS pueden aprovechar la ventaja de la funcionalidad multi-AZ (múltiples zonas de disponibilidad), que implementa automáticamente una instancia esclava en espera activa en otra zona de disponibilidad diferente.

Otro aspecto que debe tenerse en cuenta al ejecutar bases de datos directamente en EC2 (es decir, que no utilizan el servicio RDS) es la disponibilidad de almacenamiento persistente y con tolerancia a fallos. A este fin, se recomienda que las bases de datos que se ejecuten en Amazon EC2 utilicen volúmenes de almacenamiento elástico en bloques de Amazon (Amazon EBS), que son afines al almacenamiento en red que utilizan las instancias de EC2 en ejecución. En las instancias de EC2 que ejecutan una base de datos, todos los datos y los registros de la base de datos deberían guardarse en volúmenes de Amazon EBS, que siguen estando disponibles aunque falle el host de la base de datos. Esto permite un supuesto sencillo a prueba de fallos en el que es posible lanzar una nueva instancia de EC2 en caso de fallo del host, y sólo hace falta conectar a la nueva instancia los volúmenes de Amazon EBS ya existentes para que la base de datos continúe la tarea desde donde se dejó. Además de lo anterior, los volúmenes de Amazon EBS ofrecen automáticamente redundancia dentro de la zona de disponibilidad, lo que mejora su disponibilidad en discos sencillos. Si las prestaciones de un único volumen de Amazon EBS no son suficientes para satisfacer las necesidades de la base de datos, es posible preparar volúmenes para mejorar las prestaciones de las operaciones de entrada/salida por segundo (IOPS) de la base de datos. Cuando se utiliza RDS, dicho servicio controla la gestión de los volúmenes de Amazon EBS.

Además de admitir bases de datos relacionales en EC2, AWS también ofrece el servicio SimpleDB, que puede ofrecer una base de datos central no relacional, ligera, de alta disponibilidad y con tolerancia a fallos que aporte consultas e indexación de datos sin el requisito de contar con un esquema fijo. SimpleDB puede ser un sustituto muy eficaz de las bases de datos en situaciones de acceso a datos que requieran una tabla de esquema de gran tamaño, muy indexada y flexible. El servicio SimpleDB se utiliza, por ejemplo, para administración de configuraciones, catálogos de productos y datos de sesiones. Además, EC2 admite el alojamiento de muchas otras tecnologías emergentes en el movimiento NoSQL, como Cassandra, CouchDB y MemcacheDB.

## Almacenamiento y datos de seguridad de datos y activos

La nube de AWS ofrece numerosas opciones de almacenamiento, acceso y copias de seguridad de los datos y los activos de su aplicación Web. El servicio de almacenamiento sencillo de Amazon (Amazon Simple Storage Service, S3) ofrece almacenamiento de objetos con alta disponibilidad y redundancia. Se trata de una excelente solución de almacenamiento para objetos que no sean muy dinámicos o que cambien pocas veces, como por ejemplo imágenes, videos y otros medios estáticos. Amazon S3 también permite el almacenamiento en caché en servidores propios y el secuenciado de dichos activos por medio del servicio CloudFront. Para sistemas de archivos adjuntos como almacenamiento, las instancias de EC2 pueden tener adjuntos volúmenes de almacenamiento en bloques elásticos de Amazon, que pueden servir de discos montables para ejecutar instancias de EC2. Amazon EBS es excelente para almacenamiento de bloques de datos que necesitan estar accesibles y requieren persistencia fuera de la instancia en ejecución, como por ejemplo particiones de bases de datos y registros de aplicaciones. Además de ser persistentes fuera de la instancia de EC2, las instantáneas de los volúmenes de EBS pueden almacenarse en Amazon S3 y utilizarse para hacer copias de seguridad de los datos de las instancias en ejecución. Las instantáneas de EBS son incrementales en lo que respecta a los datos que se incluyen en las copias de seguridad; si se toman instantáneas con más frecuencia, puede reducirse el tiempo necesario para tomarlas. Es posible crear volúmenes de Amazon EBS con un tamaño de hasta 1 TB y es posible preparar varios volúmenes de EBS para disponer de volúmenes incluso mayores, o para mejorar las prestaciones de entrada y salida. Otra función útil de las instantáneas de Amazon EBS es que pueden utilizarse como referencia para replicar datos en varios volúmenes de Amazon EBS y adjuntarlos a otras instancias en ejecución.

## Ajuste automático de la dimensiones de la flota

Una de las diferencias clave entre la arquitectura Web de AWS y el modelo de alojamiento tradicional es la capacidad de adaptar dinámicamente las dimensiones de la flota de aplicaciones Web a demanda para atender los aumentos o las reducciones de tráfico. En el modelo de alojamiento tradicional, suelen utilizarse modelos de predicción para aprovisionarse de hosts por adelantado conforme a las proyecciones de tráfico. En una arquitectura de nube de AWS, es posible realizar el aprovisionamiento de instancias sobre la marcha conforme a una serie de parámetros que activan el ajuste de las dimensiones de las flotas para que crezcan y luego vuelvan a reducirse. La función de ajuste automático de las dimensiones de Amazon puede utilizarse para crear grupos de capacidad de servidores que pueden aumentar o reducir su tamaño a demanda. El ajuste automático de las dimensiones también funciona directamente con CloudWatch en lo que respecta a los datos de métricas y al servicio de equilibrado de cargas elásticas con vistas a la adición y eliminación de hosts para la distribución de cargas. Por ejemplo, si los servidores Web comunican un uso de la CPU superior al 80% durante un periodo determinado de tiempo, sería posible implementar en un instante otro servidor Web y añadirlo automáticamente al equilibrador de cargas elásticas para su inclusión inmediata en la rotación de equilibrado de cargas. Como se muestra en el modelo de arquitectura de alojamiento Web de AWS, es posible crear varios grupos de ajuste automático de dimensiones para diferentes capas de la arquitectura con objeto de permitir que cada capa ajuste sus dimensiones de forma independiente. Por ejemplo, el grupo de ajuste automático de las dimensiones de los servidores Web podría disparar el ajuste de las dimensiones en la entrada y la salida de la red, mientras que el grupo de ajuste automático de las dimensiones de los servidores de aplicaciones podría regular el ajuste de las dimensiones en función del uso de la CPU. También es posible configurar valores mínimos y máximos para garantizar una disponibilidad las 24 horas al día, los 7 días de la semana, así como para restringir el uso dentro de un grupo. El ajuste automático de las dimensiones puede definirse para aumentar o reducir la flota en la capa en cuestión

conforme a la utilización de recursos y a las necesidades reales del tráfico. Además del servicio de ajuste automático de las dimensiones, es posible adaptar el tamaño de las flotas de EC2 directamente por medio de las API de EC2, lo que permite lanzar, terminar e inspeccionar instancias.

### **Sistemas a pruebas de fallos con AWS**

Otra ventaja de usar AWS, en comparación con el alojamiento tradicional en Web, son las zonas de disponibilidad que facilitan el acceso de los desarrolladores de aplicaciones Web a varias ubicaciones para la implementación de instancias. Las zonas de disponibilidad son ubicaciones específicas que están diseñadas para quedar aisladas de los fallos que se produzcan en otras zonas de disponibilidad y proporcionan conectividad a bajo precio y baja latencia con otras zonas de disponibilidad situadas en la misma región. Como se puede ver en la arquitectura de alojamiento Web en AWS, se recomienda diseminar hosts de EC2 por varias zonas de disponibilidad, ya que de esta forma se proporciona una solución fácil que hace que la aplicación Web sea tolerante a fallos. Será necesario actuar con precaución para asegurarse de que existen provisiones para migrar puntos de acceso individuales entre las zonas de disponibilidad en caso de fallos. Por ejemplo, se recomienda configurar una base de datos esclava en zonas de disponibilidad secundarias para asegurarse de la persistencia continua y la alta disponibilidad de los datos incluso en el improbable supuesto de que se produjera un fallo.

Si bien es a menudo necesario realizar determinados cambios en la arquitectura para migrar una aplicación Web hasta la nube de AWS, existen mejoras significativas en el ajuste de las dimensiones, la fiabilidad y la efectividad de los costes que hacen que sin duda merezca la pena utilizar la nube de AWS.

## **Aspectos clave que deben tenerse en cuenta a la hora de utilizar AWS para alojamiento Web**

A la hora de migrar a la nube de AWS, hay varias diferencias clave respecto a los modelos de alojamiento tradicional. En la sección anterior se resaltaron muchos de los ámbitos clave que es importante tener en cuenta para a la hora de implementar una aplicación Web en la nube. La siguiente sección destaca algunos de los cambios arquitectónicos clave que es necesario plantearse para traer aplicaciones a la red.

### **Ya no son necesarios los dispositivos de red físicos**

No es posible implementar dispositivos de red físicos en AWS. Por ejemplo, los cortafuegos, enrutadores y equilibradores de cargas de las aplicaciones de AWS ya no pueden residir en dispositivos físicos, sino que es necesario sustituirlos con soluciones de software. Existe una amplia gama de soluciones de calidad empresarial para estos problemas, ya se trate de equilibrar cargas (p. ej., Zeus, HAProxy, nginx, Pound, ...) o de establecer una conexión de red privada virtual VPN (p. ej., OpenVPN, OpenSwan, Vyatta, ...). No hay limitaciones respecto a qué es posible ejecutar en la nube de AWS, pero representará un cambio de la arquitectura de su aplicación si utiliza estos dispositivos en la actualidad.

## Cortafuegos en todas partes

Donde antes tenía una zona desmilitarizada (DMZ) simple y comunicaciones abiertas entre los hosts conforme a un modelo de alojamiento tradicional, AWS imponen un modelo más seguro donde todos los hosts están bloqueados. Uno de los pasos en la planificación de una implementación de AWS es el análisis del tráfico existente entre los hosts, lo que permite decidir exactamente qué puertos es necesario abrir. Es posible crear grupos de seguridad dentro de EC2 para cada tipo de host presente en la arquitectura, así como una amplia gama de modelos de seguridad simples y en niveles para habilitar el acceso mínimo entre los hosts presentes en la arquitectura.

## Varios centros de datos disponibles

Las zonas de disponibilidad de EC2 deberían concebirse como varios centros de datos. Están separados, tanto lógicamente como físicamente, y representan un modelo fácil de usar para la implementación de su aplicación entre varios centros de datos para conseguir alta disponibilidad y fiabilidad.

## Los hosts se consideran efímeros y dinámicos

Probablemente, el cambio más importante en la arquitectura de la aplicación de AWS sea que los hosts de EC2 se deben considerar efímeros y dinámicos. Cualquier aplicación que se haya construido para la nube de AWS no deberá presuponer que habrá siempre disponible un host, y deberá ser consciente de que los datos locales (es decir, los que no estén en un volumen de Amazon EBS) se perderán en caso de fallo. Además, cuando se instala un nuevo host, no se deberá asumir nada en lo que respecta a direcciones IP o ubicación dentro de una zona de disponibilidad. Por todo esto, es imprescindible un modelo de configuración más flexible y un planteamiento sólido para el proceso de bootstrap de un host, pero estas mismas técnicas son esenciales para construir y ejecutar una aplicación que pueda ajustar sus dimensiones y sea tolerante a fallos.

## Conclusiones

Hay muchos aspectos arquitectónicos y conceptuales que deben tenerse en cuenta a la hora de plantearse la migración de una aplicación Web a la nube de AWS, pero las ventajas de disponer de una infraestructura económica, con alta capacidad de ajuste de dimensiones y tolerante a fallos que crece con su negocio compensa con creces los esfuerzos de migrar a la nube de AWS.