



Documento técnico de AWS

# Prácticas recomendadas para WordPress en AWS



---

# Prácticas recomendadas para WordPress en AWS: Documento técnico de AWS

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

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

---

# Table of Contents

Resumen .....	1
Resumen .....	1
Introducción .....	2
Implementación sencilla .....	3
Consideraciones .....	3
Enfoques disponibles .....	3
Amazon Lightsail .....	4
Selección de un plan de precios de Amazon Lightsail .....	4
Instalación de WordPress .....	5
Recuperación ante errores .....	5
Mejora del rendimiento y la rentabilidad .....	7
Aceleración de la entrega de contenido .....	7
Descarga de contenido estático .....	8
Contenido dinámico .....	8
Almacenamiento en caché de base de datos .....	10
Almacenamiento en caché de código de bytes .....	10
Implementación elástica .....	12
Arquitectura de referencia .....	12
Escalado de la nivel web .....	14
Nivel web sin estado .....	16
Almacenamiento compartido (Amazon S3 y Amazon EFS) .....	16
Nivel de datos (Amazon Aurora y Amazon ElastiCache) .....	17
Quick Start WordPress high availability by Bitnami on AWS .....	18
Conclusión .....	21
Colaboradores .....	22
Revisiones del documento .....	23
Apéndice A: Configuración de CloudFront .....	24
Orígenes y comportamientos .....	24
Creación de la distribución de CloudFront .....	24
Apéndice B: Instalación y configuración de complementos .....	28
Complemento de AWS para WordPress .....	28
Instalación y configuración del complemento .....	28
Amazon CloudFront y AWS Certificate Manager .....	30
Traduzca el contenido y haga que se lea en voz alta .....	30

---

Transmisión de podcasts mediante Amazon Pollycast .....	33
Lectura de contenido a través de dispositivos Amazon Alexa .....	33
Configuración de contenido estático .....	34
Creación de usuarios de IAM .....	35
Creación de buckets de Amazon S3 .....	35
Creación de origen estático .....	36
Apéndice C: Copia de seguridad y recuperación .....	38
Apéndice D: Implementación de nuevos complementos y temas .....	40
Avisos .....	41

# Prácticas recomendadas para WordPress en AWS

Fecha de publicación: 19 de octubre de 2021 ([Revisiones del documento](#))

## Resumen

Este documento técnico proporciona a los administradores de sistemas directrices concretas para empezar a trabajar con WordPress en Amazon Web Services (AWS) y mejorar tanto la rentabilidad de la implementación como la experiencia del usuario final. En el documento, también se describe una arquitectura de referencia que satisface los requisitos habituales de escalabilidad y alta disponibilidad.

# Introducción

WordPress es un sistema de gestión de contenido (CMS) y una herramienta de blogs de código abierto basada en PHP y MySQL que permite crear desde blogs personales a sitios web con un alto nivel de tráfico.

La primera versión de WordPress se lanzó en 2003 y no se creó teniendo en cuenta infraestructuras elásticas, modernas, escalables y basadas en la nube. Gracias al trabajo de la comunidad de WordPress y al lanzamiento de varios módulos de esta herramienta, las capacidades de la solución de CMS están en constante expansión. En la actualidad, es posible crear una arquitectura de WordPress que aproveche muchos de los beneficios de la nube de AWS.

# Implementación sencilla

Para sitios web o blogs de poco tráfico sin requisitos estrictos de alta disponibilidad, una implementación sencilla puede ser más que suficiente. Esta implementación no es la arquitectura más resistente o escalable, pero es la forma más rápida y económica de poner en marcha su sitio web.

## Temas

- [Consideraciones](#)
- [Enfoques disponibles](#)
- [Amazon Lightsail](#)

## Consideraciones

Este análisis comienza con una implementación de un único servidor web. Puede haber ocasiones en la que no sea suficiente, por ejemplo:

- La máquina virtual en la que se implementa su sitio web de WordPress es un punto único de error. Un problema con esta instancia provoca una pérdida de servicio en su sitio web.
- Los recursos de escalado para mejorar el rendimiento solo se pueden lograr con el “escalado vertical”, es decir, aumentando el tamaño de la máquina virtual que ejecuta su sitio web de WordPress.

## Enfoques disponibles

AWS tiene una serie de diferentes opciones para aprovisionar las máquinas virtuales. Hay tres formas principales de alojar su sitio web de WordPress en AWS:

- Amazon Lightsail
- Amazon Elastic Compute Cloud (Amazon EC2)
- AWS Marketplace

[Amazon Lightsail](#) es un servicio que le permite lanzar rápidamente un servidor privado virtual (una instancia de Lightsail) para alojar un sitio web de WordPress. Lightsail es la forma más sencilla de

empezar si no necesita tipos de instancias enormemente configurables o acceso a características de red avanzadas.

[Amazon EC2](#) es un servicio web que proporciona capacidad de computación de tamaño variable para que pueda lanzar un servidor virtual en cuestión de minutos. Amazon EC2 ofrece más opciones de configuración y administración que Lightsail, algo deseable en arquitecturas más avanzadas. Tiene acceso administrativo a sus instancias de EC2 y puede instalar cualquier paquete de software que elija, incluido WordPress.

[AWS Marketplace](#) es una tienda en línea donde puede buscar, comprar e implementar rápidamente software que se ejecute en AWS. Puede usar la implementación de 1-Click para lanzar imágenes de WordPress preconfiguradas directamente en Amazon EC2 en su propia cuenta de AWS en unos pocos minutos. Hay una serie de proveedores de Marketplace que ofrecen instancias de WordPress listas para ejecutarse.

Este documento técnico se ocupa de la opción de Lightsail como la implementación recomendada para un sitio web de WordPress de servidor único.

## Amazon Lightsail

Lightsail es la manera más sencilla de comenzar con AWS para desarrolladores, pequeños negocios, estudiantes y otros usuarios que necesitan una solución de servidor privado virtual (VPS) sencilla.

El servicio simplifica muchos de los elementos más complejos de la administración de infraestructuras. Es, por tanto, un punto de partida ideal si tiene menos experiencia con infraestructuras o si debe prestar la máxima atención a ejecutar su sitio web y un producto simplificado es suficiente para sus necesidades.

Con Amazon Lightsail, puede elegir sistemas operativos Windows o Linux/Unix y aplicaciones web populares, como WordPress, e implementarlas con tan solo un clic a partir de plantillas preconfiguradas.

A medida que aumentan sus necesidades, puede dejar atrás las limitaciones iniciales y conectarse a bases de datos, almacenamiento de objetos, almacenamiento en caché y servicios de distribución de contenido adicionales de AWS.

## Selección de un plan de precios de Amazon Lightsail

Un [plan de Lightsail](#) define el coste mensual de los recursos de Lightsail que utiliza para alojar su sitio web de WordPress. Hay varios planes que se adaptan a una amplia variedad de casos de uso,

con niveles variables de recursos de CPU, memoria, almacenamiento en unidades de estado sólido (SSD) y transferencia de datos. Si el sitio web es complejo, es posible que necesite una instancia más grande con más recursos. Para ello, puede migrar su servidor a un plan más grande [mediante la consola web](#) o como se describe en la [documentación de la CLI de Amazon Lightsail](#).

## Instalación de WordPress

Lightsail ofrece plantillas para las aplicaciones más usadas como WordPress. Esta plantilla es un magnífico punto de partida para ejecutar su propio sitio web de WordPress, ya que tiene preinstalado la mayoría del software que necesita. Puede instalar software adicional o personalizar la configuración de software mediante la terminal en el navegador o su propio cliente SSH, o mediante la interfaz web de administración de WordPress.

Amazon Lightsail tiene una alianza con el producto Pro Sites de GoDaddy para ayudar a los clientes de WordPress a administrar con facilidad sus instancias de manera gratuita. Los servidores virtuales de WordPress en Lightsail están preconfigurados y optimizados para obtener un alto nivel de rendimiento y seguridad, lo que facilita y agiliza la puesta en funcionamiento de su sitio de WordPress. A los clientes que tienen varias instancias con WordPress les resulta complicado y tedioso actualizar, mantener y administrar todos sus sitios. Con esta integración, puede administrar fácilmente sus múltiples instancias de WordPress en cuestión de minutos con solo unos pocos clics.

Para obtener más información sobre la administración de WordPress en Lightsail, consulte [Introducción al uso de WordPress desde la instancia de Amazon Lightsail](#). Una vez que haya terminado de personalizar su sitio web de WordPress, le recomendamos que realice una instantánea de su instancia.

Una [instantánea](#) es una forma de crear una imagen de copia de seguridad de su instancia de Lightsail. La instantánea que hemos creado es una copia del disco de sistema y también almacena la configuración de la máquina original (memoria, CPU, tamaño de disco y tasa de transferencia de datos). Las instantáneas se pueden usar para volver a una configuración correcta conocida después de una mala implementación o actualización.

Esta instantánea le permite recuperar el servidor si es necesario, pero también lanzar nuevas instancias con las mismas personalizaciones.

## Recuperación ante errores

Un servidor web único es un punto único de error, de modo que debe asegurarse de haber realizado copias de seguridad de los datos de su sitio web. El mecanismo de la instantánea antes descrito

también se puede emplear con este fin. Para recuperarse de un error, puede restaurar una instancia nueva a partir de la instantánea más reciente. Para reducir la cantidad de datos que se podrían perder durante una restauración, la instantánea debe ser lo más reciente posible.

Para minimizar la posible pérdida de datos, asegúrese de tomar instantáneas periódicamente. Puede programar instantáneas automáticas de sus instancias de Linux/Unix de Lightsail. Para ver los pasos, consulte [Habilitar o deshabilitar instantáneas automáticas para instancias o discos en Amazon Lightsail](#).

AWS recomienda usar una IP estática, una dirección IP fija y pública exclusiva para su cuenta de Lightsail. Si tiene que reemplazar la instancia por otra, puede reasignar la IP estática a la nueva instancia. De esta forma, no tendrá que volver a configurar los sistemas externos (como los registros DNS) para que apunten a una nueva dirección IP cada vez que desee reemplazar la instancia.

# Mejora del rendimiento y la rentabilidad

Con el tiempo, puede necesitar algo más que la implementación de un solo servidor. En este caso, tendrá que considerar opciones para mejorar el rendimiento del sitio web. Antes de migrar a una implementación escalable de varios servidores (como analizaremos más adelante en este documento), hay una serie de pasos para mejorar el rendimiento y la rentabilidad que puede aplicar. Se trata de prácticas recomendadas que debe seguir, de todos modos, incluso si no va a pasarse a una arquitectura de varios servidores.

Las siguientes secciones presentan una serie de opciones que pueden mejorar aspectos del desempeño y la escalabilidad de su sitio web de WordPress. Algunas se aplican a la implementación de un servidor único, mientras que otras aprovechan la escalabilidad de varios servidores. Muchas de estas modificaciones exigen el uso de uno o más complementos de WordPress. Aunque hay diversas opciones disponibles, [W3 Total Cache](#) es una opción popular que combina muchas de estas modificaciones en un solo complemento.

## Temas

- [Aceleración de la entrega de contenido](#)
- [Almacenamiento en caché de base de datos](#)
- [Almacenamiento en caché de código de bytes](#)

## Aceleración de la entrega de contenido

Un sitio web de WordPress debe entregar una combinación de contenido estático y dinámico. El contenido estático incluye imágenes, archivos JavaScript u hojas de estilos. El contenido dinámico incluye cualquier elemento generado en el servidor mediante código PHP de WordPress, por ejemplo, elementos de su sitio que se generan desde la base de datos o se personalizan para cada espectador.

Un aspecto importante de la experiencia del usuario final es la latencia de red implicada al entregar el contenido anterior a usuarios de todo el mundo. La aceleración de la entrega del contenido anterior mejorará la experiencia del usuario final, en especial de los usuarios repartidos por todo el mundo. Esto se puede lograr con una red de entrega de contenido (CDN), como Amazon CloudFront.

[Amazon CloudFront](#) es un servicio web que ofrece un método fácil y rentable de distribuir contenido con baja latencia y altas velocidades de transferencia de datos a través de varias ubicaciones de

borde en todo el mundo. Las solicitudes de los espectadores se enrutan automáticamente a una [ubicación de borde](#) de CloudFront adecuada para reducir la latencia. Si el contenido se puede almacenar en caché (durante unos segundos, minutos o incluso días) y ya está almacenado en una ubicación de borde determinada, CloudFront lo entrega de inmediato. Si el contenido no se puede guardar en caché, ha caducado o no está actualmente en esa ubicación de borde, CloudFront recupera contenido de una o más fuentes de confianza, conocidas como origen (en este caso, la instancia de Lightsail) en la configuración de CloudFront. Esta recuperación tiene lugar en conexiones de red optimizadas, que contribuyen a acelerar la entrega de contenido en su sitio web. Además de mejorar la experiencia del usuario final, el modelo que hemos examinado también reduce la carga de sus servidores de origen y tiene el potencial de ahorrar costes de manera significativa.

## Descarga de contenido estático

Esto incluye archivos de imágenes, CSS y JavaScript, bien los que son parte de sus temas de WordPress o bien los archivos multimedia cargados por los administradores de contenido. Todos esos archivos se pueden almacenar en Amazon Simple Storage Service (Amazon S3) con un complemento como W3 Total Cache y presentarse a los usuarios de una manera escalable y altamente disponible. [Amazon S3](#) ofrece una infraestructura de almacenamiento de datos altamente escalable, fiable y de baja latencia a bajo coste, a la que se puede acceder a través de las API de REST. Amazon S3 almacena los objetos de forma redundante, no solo en varios dispositivos, sino también en varias instalaciones de una región de AWS, lo que proporciona niveles de durabilidad excepcionalmente altos.

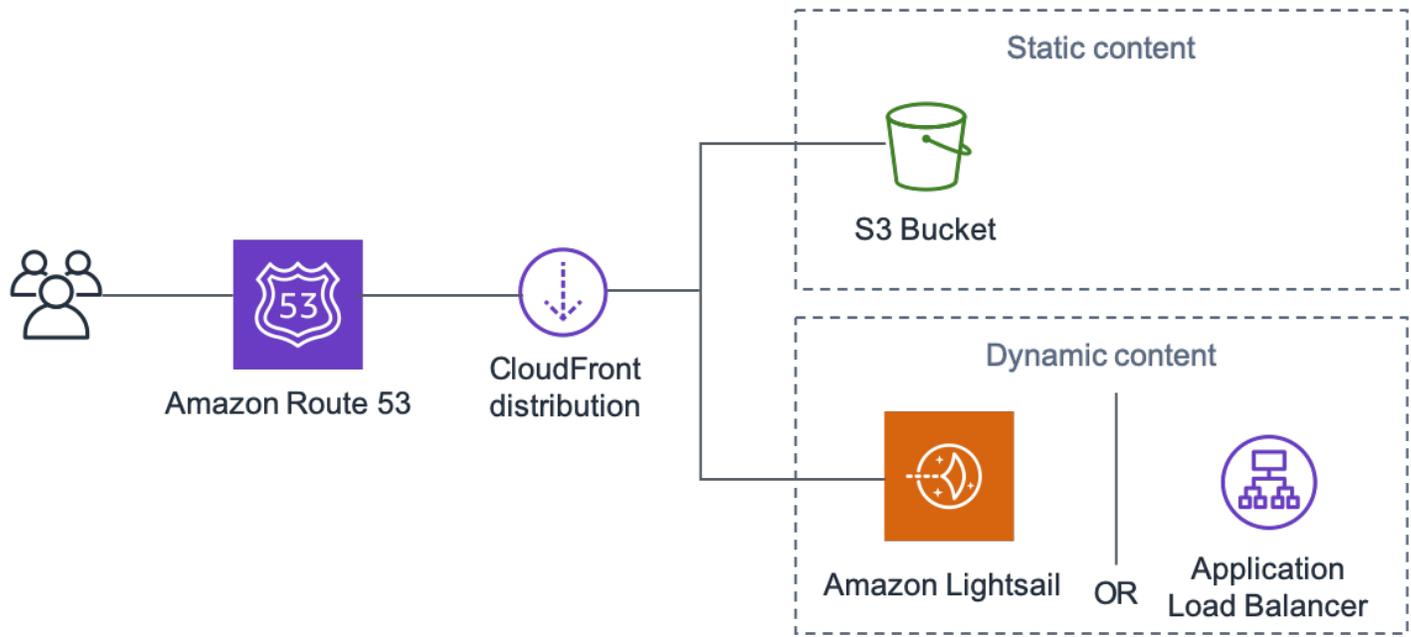
Esto tiene el efecto positivo de descargar esta carga de trabajo de su instancia de Lightsail y le permite centrarse en la generación de contenido dinámico. De esta manera, se reduce la carga en el servidor y es un paso importante para crear una arquitectura sin estado (lo que es un requisito previo para implementar el escalado automático).

Luego puede configurar Amazon S3 como origen para CloudFront para mejorar la entrega de esos activos estáticos a los usuarios de todo el mundo. Aunque WordPress no está integrado en Amazon S3 y CloudFront "de serie", hay una variedad de complementos y asistencia para estos servicios (por ejemplo, W3 Total Cache).

## Contenido dinámico

El contenido dinámico incluye la salida del lado del servidor de scripts PHP de WordPress. También se puede entregar contenido dinámico mediante CloudFront si configura el sitio web de WordPress como origen. Como el contenido dinámico incluye contenido personalizado, tendrá que configurar

CloudFront para que envíe determinadas cookies HTTP y encabezados HTTP como parte de una solicitud al servidor de origen personalizado. CloudFront usa los valores de cookie enviados como parte de la clave que identifica un objeto único en su caché. Para garantizar la máxima eficiencia del almacenamiento en caché, debe configurar CloudFront para que envíe solo las cookies HTTP y encabezados HTTP que realmente varían el contenido (no las cookies que solo se usan en el lado cliente o por aplicaciones de terceros, por ejemplo, análisis web).



## Entrega de todo el sitio web mediante Amazon CloudFront

La figura anterior incluye dos orígenes: uno para el contenido estático y otro para el contenido dinámico. Para obtener información sobre la implementación, consulte el [Apéndice A: Configuración de CloudFront](#) y el [Apéndice B: Instalación y configuración de complementos](#).

CloudFront usa encabezados de control de caché estándar para identificar si debe guardar en caché (y durante cuánto tiempo) las respuestas HTTP específicas. Los navegadores web usan los mismos encabezados de control de caché para decidir cuándo y durante cuánto tiempo se almacena localmente en caché el contenido para ofrecer la experiencia de usuario final más óptima (por ejemplo, un archivo `.css` que ya se ha descargado no se volverá a descargar cada vez que un visitante vea la página). Puede configurar encabezados de control de caché en el nivel del servidor web (por ejemplo, a través de archivos `.htaccess` o modificaciones del archivo `httpd.conf`) o instalar un complemento de WordPress (por ejemplo, W3 Total Cache) para determinar cómo se configuran esos encabezados para el contenido estático y dinámico.

## Almacenamiento en caché de base de datos

El almacenamiento en caché de la base de datos puede reducir notablemente la latencia y aumentar la producción de cargas de trabajo de aplicaciones sobre todo de lectura como WordPress. El rendimiento de la aplicación mejora al almacenar partes de datos a las que se accede a menudo en la memoria para el acceso de baja latencia (por ejemplo los resultados de bases de datos con muchas E/S). Cuando un gran porcentaje de las consultas se entregan desde el caché, el número de consultas que tienen que llegar a la base de datos es reducido, lo que resulta en un menor costo asociado con la ejecución de la base de datos.

Aunque WordPress tiene capacidades de caché limitadas "de serie", hay una gran variedad de complementos que admiten la integración con [Memcached](#), un sistema de almacenamiento en caché de objetos de memoria ampliamente usado. El complemento W3 Total Cache es un buen ejemplo.

En la situación más sencilla, instala Memcached en su servidor web y captura el resultado como una nueva instantánea. En este caso, usted es responsable de las tareas administrativas asociadas con ejecutar una caché.

Otra opción es aprovechar un servicio administrado como [Amazon ElastiCache](#) y evitar esa carga operativa. ElastiCache facilita la implementación, el funcionamiento y el escalado de almacenamiento en memoria caché distribuida en la nube. Encontrará información acerca de cómo conectarse a sus nodos de clúster de ElastiCache en la [documentación de Amazon ElastiCache](#).

Si aún usa Lightsail y desea acceder a un clúster de ElastiCache en su cuenta de AWS de manera privada, puede hacerlo mediante emparejamiento de VPC. Para obtener instrucciones sobre cómo habilitar el emparejamiento de VPC, consulte [Set up Amazon VPC peering to work with AWS resources outside of Amazon Lightsail](#).

## Almacenamiento en caché de código de bytes

Cada vez que se ejecuta un script PHP, este se analiza y compila. Al usar un almacenamiento en caché de código de bytes de PHP, la salida de la compilación de PHP se almacena en RAM de modo que no es necesario compilar el mismo script una y otra vez. Esto reduce los gastos relacionados con la ejecución de scripts PHP, lo que resulta en un mejor desempeño y menos requisitos de CPU.

El almacenamiento en caché de código de bytes se puede instalar en cualquier instancia de Lightsail que aloja WordPress y puede reducir enormemente su carga. Para PHP 5.5 y versiones posteriores, AWS recomienda el uso de [OPcache](#), una extensión empaquetada con esa versión de PHP.

Tenga en cuenta que OPcache se habilita de forma predeterminada en la plantilla de Lightsail de Bitnami WordPress de modo que no es necesario realizar ninguna otra acción.

# Implementación elástica

Hay muchas situaciones donde una implementación de servidor único podría no ser suficiente para su sitio web. En estas situaciones, necesita una arquitectura escalable de varios servidores.

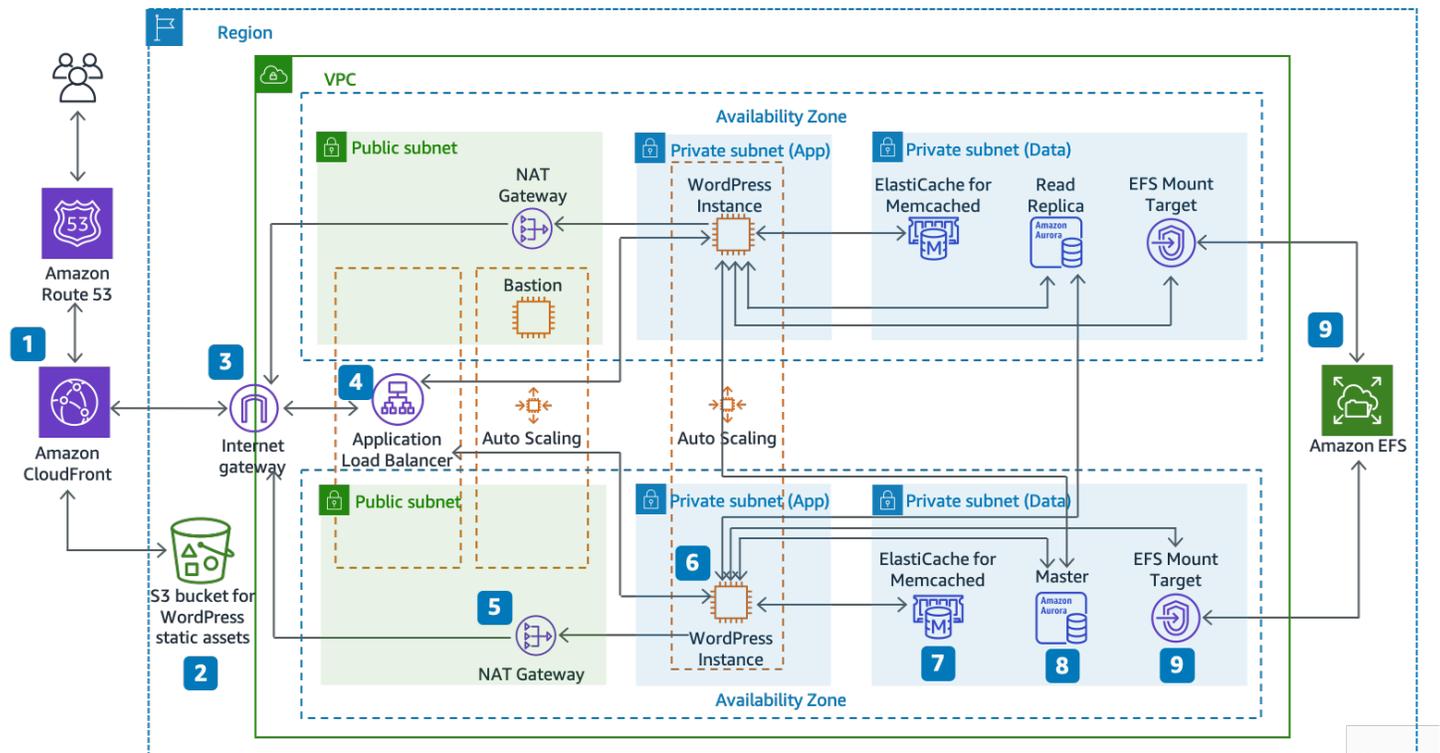
## Temas

- [Arquitectura de referencia](#)
- [Escalado de la nivel web](#)
- [Nivel web sin estado](#)
- [Quick Start WordPress high availability by Bitnami on AWS](#)

## Arquitectura de referencia

En [Hosting WordPress on AWS reference architecture](#), disponible en GitHub, se describen las prácticas recomendadas para implementar WordPress en AWS y se incluye un conjunto de plantillas de AWS CloudFormation para que pueda empezar a trabajar rápidamente. La siguiente arquitectura está basada en esa arquitectura de referencia. En el resto de esta sección, se revisarán los motivos subyacentes a las elecciones de arquitectura.

La AMI basada en GitHub se cambió de Amazon Linux1 a Amazon Linux2 en julio de 2021. Sin embargo, las plantillas de implementación de S3 aún no se han cambiado. Se recomienda usar plantillas en GitHub si hay algún problema para implementar la arquitectura de referencia con plantillas en S3.



## Arquitectura de referencia para alojar WordPress en AWS

### Componentes de arquitectura

La arquitectura de referencia ilustra la implementación recomendada completa para un sitio web de WordPress en AWS.

- Comienza con un almacenamiento en caché de borde en Amazon CloudFront (1) para almacenar en caché contenido cercano a los usuarios finales para acelerar la entrega.
- CloudFront toma contenido estático de un bucket de S3 (2) y contenido dinámico de un Application Load Balancer (4) delante de las instancias web.
- Las instancias web se ejecutan en un grupo de Auto Scaling de instancias de Amazon EC2(6).
- Un clúster de ElastiCache (7) almacena en caché datos consultados frecuentemente para acelerar las respuestas.

Una instancia de MySQL de Amazon Aurora (8) aloja la base de datos de WordPress.

- Las instancias de EC2 de WordPress acceden a datos de WordPress compartidos en un sistema de archivos Amazon EFS mediante un destino de montaje de EFS (9) en cada zona de disponibilidad.

- Una puerta de enlace de Internet (3) permite la comunicación entre los recursos de su VPC e Internet.
- Las puertas de enlace NAT (5) en cada zona de disponibilidad permiten que las instancias de EC2 en subredes privadas (aplicaciones y datos) accedan a Internet.

Dentro de Amazon VPC, hay dos tipos de subredes: pública (subred pública) y privada (subred de aplicación y subred de datos). Los recursos implementados en las subredes públicas recibirán una dirección IP pública y serán visibles públicamente en Internet. Aquí se implementa el Application Load Balancer (4) y un host bastión para la administración. Los recursos implementados en las subredes privadas recibirán solo una dirección de IP privada y, por tanto, no serán visibles públicamente en Internet, lo que mejora la seguridad de esos recursos. Las instancias de servidor web de WordPress (6), las instancias de clúster de ElastiCache (7), las instancias de base de datos MySQL de Aurora (8) y los destinos de montaje de EFS (9) se implementan todos ellos en subredes privadas.

En el resto de esta sección, se tratan cada uno de estos aspectos en más detalle.

## Escalado de la nivel web

Para convertir su arquitectura de un solo servidor en una arquitectura escalable y de varios servidores, debe usar cinco componentes clave:

- Instancias de Amazon EC2
- Imágenes de Amazon Machine (AMI)
- Equilibradores de carga
- Escalado automático
- Comprobaciones de estado

AWS ofrece una amplia variedad de tipos de instancias de EC2 de modo que puede elegir la mejor configuración de servidor en cuanto a rendimiento y coste. En términos generales, el tipo de instancia optimizado para computación (por ejemplo, C4) puede ser una buena opción para un servidor web de WordPress. Puede implementar sus instancias en varias zonas de disponibilidad dentro de una región de AWS para aumentar la fiabilidad de la arquitectura general.

Como tiene control completo de su instancia de EC2, puede iniciar sesión con acceso raíz para instalar y configurar todos los componentes del software requeridos para ejecutar el sitio web de

WordPress. Una vez hecho esto, puede guardar esa configuración como una nueva AMI, que puede utilizar para lanzar nuevas instancias con todas las personalizaciones que ha realizado.

Para distribuir las solicitudes del usuario final a varios nodos de servidor web, necesita una solución de balanceador de carga. AWS proporciona esta capacidad a través de [Elastic Load Balancing](#), un servicio de alta disponibilidad que distribuye el tráfico entre varias instancias de EC2. Como su sitio web entregará contenido a los usuarios a través de HTTP o HTTPS, recomendamos que use Application Load Balancer, un equilibrador de carga de capa de aplicación con enrutamiento de contenido y la capacidad de ejecutar varios sitios web de WordPress en distintos dominios de ser necesario.

Elastic Load Balancing admite la distribución de numerosas solicitudes en varias zonas de disponibilidad dentro de una región de AWS. También puede configurar una comprobación de estado de modo que Application Load Balancer detenga automáticamente el envío de tráfico a instancias individuales que han producido errores (por ejemplo, debido a un problema de hardware o bloqueo de software). AWS recomienda utilizar la página de inicio de sesión de administrador de WordPress (`/wp-login.php`) para la comprobación de estado, porque esta página confirma que el servidor web se está ejecutando y que está configurado para entregar correctamente archivos PHP.

Tiene la posibilidad de crear una página de comprobación de estado personalizada que compruebe otros recursos dependientes, como recursos de base de datos y caché. Para obtener más información, consulte [Comprobaciones de estado de los grupos de destino](#) en la Guía de Application Load Balancer.

La elasticidad es una característica clave de la nube de AWS. Puede lanzar más capacidad de computación (por ejemplo, servidores web) cuando la necesite y ejecutar menos cuando no la necesite. [AWS Auto Scaling](#) es un servicio de AWS que le ayuda a automatizar este aprovisionamiento para aumentar o reducir la capacidad de Amazon EC2 según las condiciones que defina sin necesidad de intervención manual. Puede configurar AWS Auto Scaling para que la cantidad de instancias de EC2 que utiliza aumente sin problemas durante los picos de demanda para mantener el rendimiento y que disminuya automáticamente cuando el tráfico se reduzca con el objetivo de minimizar los costes.

Elastic Load Balancing también admite adiciones dinámicas y la eliminación de hosts de Amazon EC2 de la rotación de equilibrio de carga. El propio Elastic Load Balancing también aumenta y disminuye dinámicamente la capacidad de equilibrio de carga para ajustarse a las demandas del tráfico sin ninguna intervención manual.

## Nivel web sin estado

Para sacar partido de numerosos servidores web en una configuración de escalado automático, su nivel web tiene que ser sin estado. Una aplicación sin estado es una que no necesita conocimiento de interacciones previas y no almacena información de la sesión. En el caso de WordPress, esto significa que todos los usuarios finales recibirán la misma respuesta, sin importar en que servidor web se procesó su solicitud. Una aplicación sin estado puede escalarse horizontalmente ya que cualquier solicitud se puede atender desde cualquier recurso de computación disponible (es decir, instancias de servidor web). Cuando ya no se necesita esa capacidad, es posible terminar con toda seguridad cualquier recurso individual (después de vaciar las tareas en ejecución). Estos recursos no tienen que ser conscientes de la presencia de sus pares, todo lo que se necesita es una forma de distribuir la carga de trabajo entre ellos.

A la hora de almacenar datos de la sesión de usuario, el núcleo de WordPress es completamente sin estado porque confía en cookies que se almacenan en el navegador web del cliente. El almacenamiento de la sesión carece de importancia a menos que haya instalado código personalizado (por ejemplo, un complemento de WordPress) que confía, en cambio, en sesiones PHP nativas.

Sin embargo, WordPress se diseñó originalmente para ejecutarse en un solo servidor. Como resultado, almacena algunos datos en el sistema de archivos local del servidor. Cuando se ejecuta WordPress en una configuración de varios servidores, esto crea un problema porque hay una incoherencia entre los servidores web. Por ejemplo, si un usuario carga una nueva imagen, solo se almacena en uno de los servidores.

Esto demuestra por qué es necesario mejorar la configuración de ejecución de WordPress predeterminada para mover datos importantes al almacenamiento compartido. La arquitectura recomendada tiene una base de datos como una capa separada fuera del servidor web y utiliza el almacenamiento compartido para almacenar las cargas de los usuarios, los temas y los complementos.

## Almacenamiento compartido (Amazon S3 y Amazon EFS)

De forma predeterminada, WordPress almacena las cargas del usuario en el sistema de archivos local y, por tanto, no es sin estado. Por lo tanto, debemos mover la instalación de WordPress y todas las personalizaciones de los usuarios (como la configuración, los complementos, los temas y las cargas generadas por el usuario) a una plataforma de datos compartida para ayudar a reducir la carga en los servidores web y hacer que el nivel web no tenga estado.

[Amazon Elastic File System](#) (Amazon EFS) proporciona sistemas de archivos de red escalables para su uso con instancias de EC2. Los sistemas de archivos de Amazon EFS se distribuyen entre un número ilimitado de servidores de almacenamiento, por lo que pueden crecer de forma elástica, además de permitir un acceso en paralelo masivo desde instancias de EC2. El diseño distribuido de Amazon EFS evita los atascos y las restricciones inherentes a los servidores de archivos tradicionales.

Al mover todo el directorio de instalación de WordPress a un sistema de archivos de EFS y montarlo en cada una de sus instancias de EC2 cuando arrancan, su sitio de WordPress y todos sus datos se almacenarán automáticamente en un sistema de archivos distribuido que no depende de ninguna instancia de EC2, lo que hace que su nivel web sea completamente sin estado. El beneficio de esta arquitectura es que no necesita instalar complementos y temas en cada nuevo lanzamiento de instancia y puede acelerar significativamente la instalación y recuperación de instancias de WordPress. También es más fácil implementar cambios en complementos y temas en WordPress, como se describe en la sección [Consideraciones de implementación](#) de este documento.

Para asegurar el óptimo rendimiento de su sitio web cuando se ejecuta desde un sistema de archivos de EFS, compruebe la configuración recomendada para Amazon EFS y OPcache en la [Arquitectura de referencia de AWS para WordPress](#).

También tiene la opción de descargar todos los recursos estáticos, como imágenes, archivos CSS y JavaScript, a un bucket de S3 con almacenamiento en caché de CloudFront. El mecanismo para hacer esto en una arquitectura de varios servidores es exactamente el mismo que para una arquitectura de un solo servidor, tal y como se explica en la sección [Contenido estático](#) de este documento técnico. Los beneficios son los mismos que en la arquitectura de un solo servidor, puede descargar el trabajo asociado con entregar los activos estáticos a Amazon S3 y CloudFront, lo que permite a sus servidores web centrarse en generar contenido dinámico solo y atender más solicitudes de usuario por servidor web.

## Nivel de datos (Amazon Aurora y Amazon ElastiCache)

Con la instalación de WordPress almacenada en un sistema de archivos de red distribuido, escalable y compartido, y los recursos estáticos entregados desde Amazon S3, ya puede centrarse en el componente con estado que queda: la base de datos. Al igual que con el nivel de almacenamiento, la base de datos no debe depender de ningún servidor individual, por lo que no se puede alojar en uno de los servidores web. En su lugar, aloje la base de datos de WordPress en Amazon Aurora.

[Amazon Aurora](#) es una base de datos relacional compatible con MySQL y PostgreSQL creada para la nube que combina el rendimiento y la disponibilidad que tienen las bases de datos comerciales de

gama alta con la simplicidad y la rentabilidad que tienen las bases de datos de código abierto. Aurora MySQL aumenta el desempeño y la disponibilidad de MySQL al integrar estrechamente el motor de la base de datos con un sistema de almacenamiento distribuido creado en función del propósito, respaldado por SSD. Es tolerante a errores y ofrece recuperación automática, replica sus datos en seis copias en tres zonas de disponibilidad, se ha diseñado para aportar una disponibilidad superior al 99,99 % y hace copias de seguridad de sus datos continuamente en Amazon S3. Amazon Aurora está diseñado para detectar automáticamente los bloqueos de las bases de datos y se reinicia sin necesidad de realizar recuperaciones tras bloqueos ni de recompilar la caché de la base de datos.

Amazon Aurora proporciona varios [tipos de instancias](#) que se adaptan a los diferentes perfiles de las aplicaciones, incluidas las instancias optimizadas para memoria y de rendimiento ampliable. Para mejorar el rendimiento de la base de datos, puede seleccionar un tipo de instancia de gran tamaño para disponer de más recursos de CPU y memoria.

Amazon Aurora gestiona automáticamente la conmutación por error entre la instancia principal y las [réplicas de Aurora](#) para que las operaciones de la base de datos puedan reanudarse rápidamente sin ninguna intervención administrativa manual. La conmutación por error suele tardar menos de 30 segundos.

Después de haber creado al menos una réplica de Aurora, conecte su instancia principal usando el punto de conexión del clúster para permitir que su aplicación utilice la conmutación por error automática en el caso de errores de la instancia principal. Puede crear hasta 15 réplicas de lectura de baja latencia en tres zonas de disponibilidad.

A medida que escala su base de datos, el caché de la base de datos también se escala. Como se indicó en la sección [Almacenamiento en caché de la base de datos](#), ElastiCache tiene características para escalar la caché en numerosos nodos del clúster de ElastiCache y en numerosas zonas de disponibilidad en una región para mejorar la disponibilidad. Cuando escale su clúster de ElastiCache, debe asegurarse de configurar su complemento de almacenamiento en caché para que se conecte mediante el punto de conexión de la configuración de modo que WordPress pueda usar nuevos nodos de clúster conforme se añaden y dejar de usar los nodos de clúster antiguos cuando se eliminan. También necesitará configurar sus servidores web para que utilicen [ElastiCache Cluster Client para PHP](#) y actualizar su AMI para almacenar este cambio.

## Quick Start WordPress high availability by Bitnami on AWS

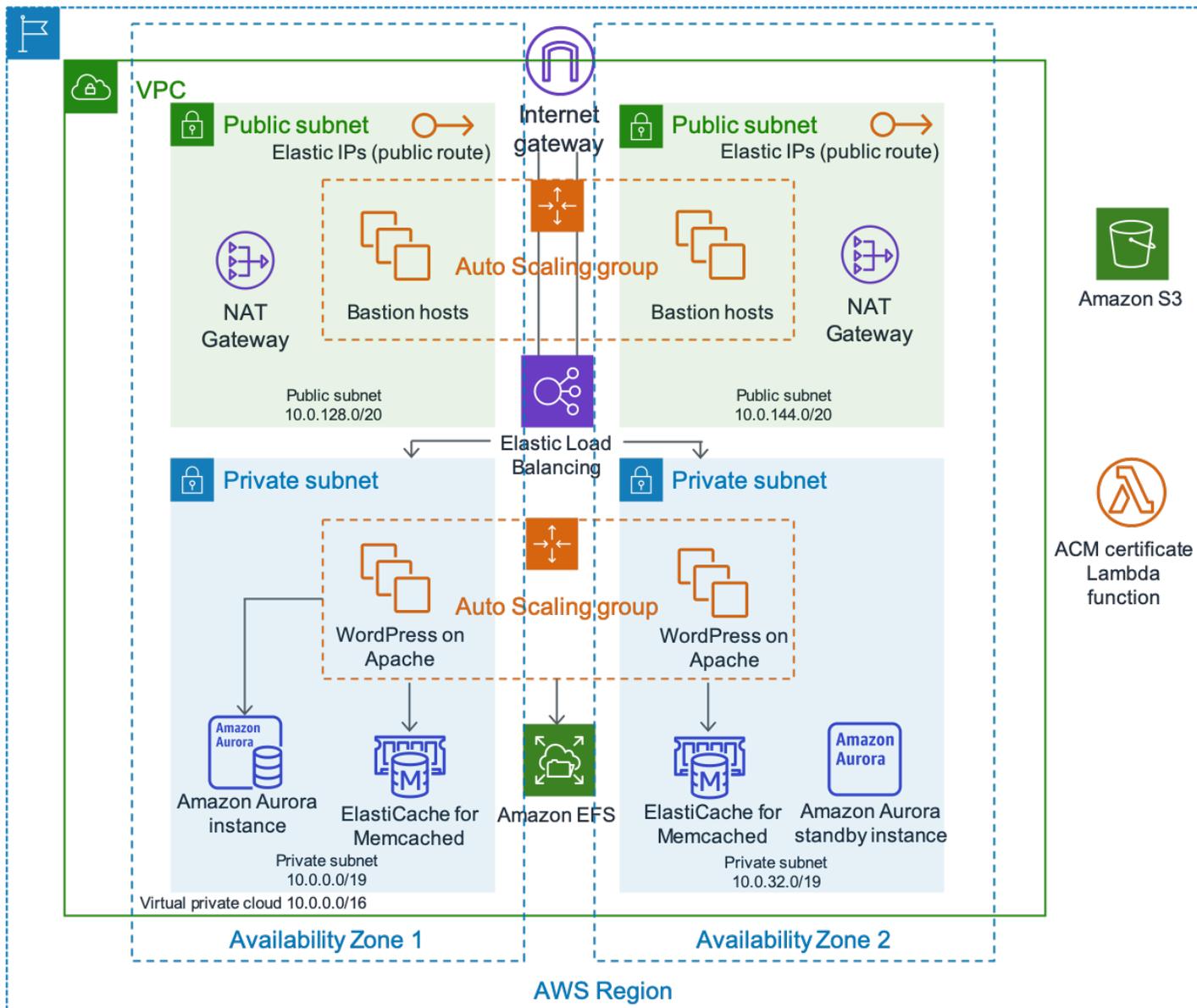
Los socios y arquitectos de soluciones de AWS crean Quick Starts para ayudarlo a implementar tecnologías conocidas en AWS, basadas en prácticas recomendadas de AWS destinadas a

lograr seguridad y alta disponibilidad. Estos aceleradores reducen cientos de procedimientos manuales a unos pocos pasos para que pueda crear su entorno de producción rápidamente y empezar a usarlo de inmediato. Cada Quick Start incluye plantillas de AWS CloudFormation que automatizan la implementación y una guía que describe la arquitectura y proporciona instrucciones de implementación paso a paso.

[El Quick Start WordPress High Availability by Bitnami on AWS](#) configura el siguiente entorno en AWS:

- Una arquitectura altamente disponible que abarca dos zonas de disponibilidad.\*
- Una Virtual Private Cloud (VPC) configurada con subredes públicas y privadas de acuerdo con las prácticas recomendadas de AWS. Esto proporciona la infraestructura de red para la implementación\*.
- Una puerta de enlace de Internet para proporcionar acceso a Internet. Esta puerta de enlace la utilizan los hosts bastión para enviar y recibir tráfico.\*
- En las subredes públicas, puertas de enlace NAT administradas para permitir el acceso saliente a Internet para los recursos en las subredes privadas\*.
- En las subredes públicas, hosts bastión de Linux en un grupo de Auto Scaling que permiten el acceso de entrada mediante Secure Shell (SSH) a las instancias de EC2 en subredes públicas y privadas\*.
- Elastic Load Balancing para distribuir solicitudes HTTP y HTTPS en varias instancias de WordPress.
- En las subredes privadas, las instancias de EC2 que alojan la aplicación de WordPress en Apache. Estas instancias se aprovisionan en un grupo de Auto Scaling para garantizar una alta disponibilidad.
- En las subredes privadas, las instancias de Amazon Aurora DB que administra Amazon Relational Database Service (Amazon RDS).
- En las subredes privadas, Amazon Elastic File System (Amazon EFS) para compartir recursos (como complementos, temas e imágenes) en las instancias de WordPress.
- En las subredes privadas, nodos de Amazon ElastiCache for Memcached para almacenar en caché las consultas de base de datos.

\* La plantilla que implementa el inicio rápido en una VPC existente omite las tareas marcadas con asteriscos y solicita la configuración de su VPC existente.



### Arquitectura de WordPress High Availability by Bitnami

La descripción detallada de la implementación de WordPress High Availability by Bitnami on AWS está fuera del alcance de este documento. Para ver la configuración y las opciones, consulte [WordPress High Availability by Bitnami on AWS](#).

## Conclusión

AWS presenta muchas opciones de arquitectura para ejecutar WordPress. La opción más sencilla es una instalación de un solo servidor para sitios web de poco tráfico. Para sitios web más avanzados, los administradores del sitio pueden añadir varias opciones adicionales; cada una representa una mejora incremental en términos de disponibilidad y escalabilidad. Los administradores pueden seleccionar las características que más se acercan a sus requisitos y presupuesto.

# Colaboradores

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

- Paul Lewis, arquitecto de soluciones, Amazon Web Services
- Ronan Guilfoyle, arquitecto de soluciones, Amazon Web Services
- Andreas Chatzakis, director de arquitectura de soluciones, Amazon Web Services
- Jibril Touzi, gerente técnico de cuentas, Amazon Web Services
- Hakmin Kim, arquitecto de soluciones de socios de migración, Amazon Web Services

## Revisiones del documento

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

update-history-change

[Se ha actualizado el documento técnico](#)

[Publicación inicial](#)

update-history-description

Se ha actualizado para cambiar la arquitectura de referencia y el complemento de AWS para WordPress.

Se ha actualizado para incluir nuevos enfoques de implementación y el complemento de AWS para WordPress.

Se ha actualizado para aclarar los mensajes del producto Amazon Aurora.

Se ha actualizado para incluir los servicios de AWS lanzados desde la primera publicación.

Publicado por primera vez.

update-history-date

19 de octubre de 2021

30 de octubre de 2019

1 de febrero de 2018

1 de diciembre de 2017

1 de diciembre de 2014

# Apéndice A: Configuración de CloudFront

Para obtener una eficiencia y rendimiento óptimos al usar Amazon CloudFront con su sitio web de WordPress, es importante configurar el sitio web correctamente para los diferentes tipos de contenido que se entregan.

Temas

- [Orígenes y comportamientos](#)
- [Creación de la distribución de CloudFront](#)

## Orígenes y comportamientos

Un [origen](#) es una ubicación a la que CloudFront envía solicitudes de contenido que distribuye a través de las ubicaciones de borde. Dependiendo de su implementación, puede tener uno o dos orígenes. Uno para el contenido dinámico (la instancia de Lightsail en la [opción de implementación de un solo servidor](#) o Application Load Balancer en la [opción de implementación elástica](#)) utilizando un origen personalizado. Es posible que tenga un segundo origen al que dirigir CloudFront para el contenido estático. En la [arquitectura de referencia](#) anterior, esto es un bucket de S3. Cuando usa Amazon S3 como origen de su distribución, tiene que usar una [política de buckets](#) para garantizar que el contenido sea de acceso público.

Los [comportamientos](#) le permiten establecer reglas que rigen la forma en que CloudFront almacena en caché el contenido y, a su vez, determinan la eficacia de la caché. Los comportamientos le permiten controlar el protocolo y los métodos HTTP con los que se puede acceder a su sitio web. También le permiten controlar si pasa encabezados HTTP, cookies o cadenas de consultas a su backend (y, si es así, cuáles). Los comportamientos se aplican a patrones de ruta de URL específicos.

## Creación de la distribución de CloudFront

Para crear una distribución web de CloudFront, utilice la distribución. El origen y el comportamiento predeterminados que se crean automáticamente se utilizarán para el contenido dinámico. Cree cuatro comportamientos adicionales para personalizar aún más la forma en que se tratarán las solicitudes estáticas y dinámicas. En la siguiente tabla, se resumen las propiedades de configuración de los cinco comportamientos. También puede omitir esta configuración manual y utilizar el

complemento de AWS para WordPress que se explica en el [Apéndice B: Instalación y configuración de complementos](#). Es la forma más sencilla de configurar CloudFront para acelerar su sitio de WordPress.

Tabla 1: Resumen de las propiedades de configuración de los comportamientos de CloudFront

Propiedad	Estático	Dinámico (admin)	Dinámico (front-end)
Rutas (comportamientos)	wp-content/* wp-includes/*	wp-admin/* wp-login.php	predeterminado (*)
Protocolos	HTTP y HTTPS	Redireccionamiento a HTTPS	HTTP y HTTPS
Métodos HTTP	GET, HEAD	ALL	ALL
Encabezados HTTP	NONE	ALL	Host CloudFront-Forwarded-Proto CloudFront-Is-Mobile-Viewer CloudFront-Is-Tablet-Viewer CloudFront-Is-Desktop-Viewer
Cookies	NONE	ALL	comment_* wordpress_* wp-settings-*
Cadenas de consulta	YES (invalidación)	SÍ	SÍ

Para el comportamiento predeterminado, AWS recomienda la siguiente configuración:

- Permita que la política del protocolo de origen se adapte al espectador, de modo que si los espectadores se conectan a CloudFront mediante HTTPS, CloudFront también se conectará a su origen mediante HTTPS, y se logrará un cifrado de extremo a extremo. Tenga en cuenta que esto requiere que instale un certificado SSL de confianza en el equilibrador de carga. Para obtener más información, consulte [Exigir HTTPS para la comunicación entre CloudFront y su origen personalizado](#).
- Permita todos los métodos HTTP ya que las porciones dinámicas del sitio web requieren tanto solicitudes GET como POST (por ejemplo, para admitir POST para los formularios de envío de comentarios).
- Reenvíe solo las cookies que cambien la salida de WordPress; por ejemplo `>wordpress_*`, `wp-settings-*` y `comment_*`. Debe ampliar esa lista si ha instalado cualquier complemento que dependa de otras cookies que no están en la lista.
- Reenvíe solo los encabezados HTTP que afecten a la salida de WordPress, como por ejemplo, `Host`, `CloudFront-Forwarded-Proto`, `CloudFront-is-Desktop-Viewer`, `CloudFront-is-Mobile-Viewer` y `CloudFront-is-Tablet-Viewer`:
  - `Host` permite que varios sitios web de WordPress se alojen en el mismo origen.
  - `CloudFront-Forwarded-Proto` permite almacenar en caché diferentes versiones de páginas en función de si se accede a ellas a través de HTTP o HTTPS.
  - `CloudFront-is-Desktop-Viewer`, `CloudFront-is-Mobile-Viewer` y `CloudFront-is-Tablet-Viewer` le permiten personalizar la salida de sus temas en función del tipo de dispositivo del usuario final.
- Reenvíe todas las cadenas de consulta a la caché en función de sus valores porque WordPress se basa en ellos; también se pueden usar para invalidar objetos almacenados en caché.

Si desea publicar su sitio web con un nombre de dominio personalizado (es decir, no `*.cloudfront.net`), introduzca los URI apropiados en Alternate Domain Names (Nombres de dominio alternativos) en Distribution Settings (Configuración de distribución). En este caso, necesita también un certificado de SSL para el nombre de dominio personalizado. Puede [solicitar](#) certificados SSL a través de AWS Certificate Manager y configurarlos en una distribución de CloudFront.

Ahora, cree dos comportamientos de caché más para el contenido dinámico: uno para la página de inicio de sesión (patrón de ruta: `wp-login.php`) y otro para el panel de administración (patrón de ruta: `wp-admin/*`). Estos dos comportamientos tienen exactamente los mismos ajustes, como sigue:

- Aplique una política de protocolo del espectador de solo HTTPS.
- Permita todos los métodos HTTP.
- Caché basado en todos los encabezados HTTP.
- Envíe todas las cookies.
- Envíe y almacene en caché en función de las cadenas de consulta.

La razón de esta configuración es que esta sección del sitio web es muy personalizada y normalmente tiene pocos usuarios, por lo que la eficiencia de la memoria caché no es una prioridad. El objetivo es que la configuración sea sencilla para garantizar la máxima compatibilidad con cualquier complemento instalado al pasar todas las cookies y encabezados al origen.

El [complemento de AWS para WordPress](#) que se explica en el Apéndice B crea automáticamente una distribución de CloudFront que se ajusta a la configuración anterior.

De forma predeterminada, WordPress almacena todo localmente en el servidor web, que es el almacenamiento en bloque (Amazon EBS) para la [implementación de un solo servidor](#) y el almacenamiento de archivos (Amazon EFS) para la [implementación elástica](#). Además de reducir los costes de almacenamiento y transferencia de datos, trasladar los recursos estáticos a Amazon S3 permite obtener escalabilidad, disponibilidad de datos, seguridad y rendimiento. Hay varios complementos que facilitan el traslado del contenido estático a Amazon S3; uno de ellos es [W3 Total Cache](#), que también se explica en el [Apéndice B: Instalación y configuración de complementos](#).

# Apéndice B: Instalación y configuración de complementos

## Temas

- [Complemento de AWS para WordPress](#)
- [Configuración de contenido estático](#)

## Complemento de AWS para WordPress

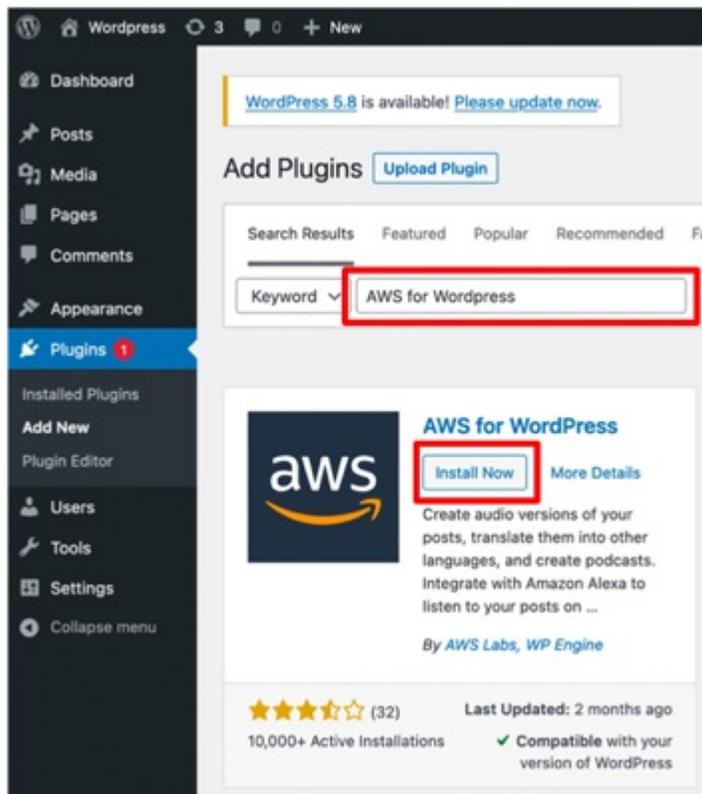
El complemento de AWS para WordPress es el único complemento de WordPress que escribe y mantiene activamente AWS. Permite a los clientes configurar fácilmente [Amazon CloudFront](#) y [AWS Certificate Manager \(ACM\)](#) en sitios web de WordPress para mejorar el rendimiento y la seguridad. El complemento utiliza los servicios de [Amazon Machine Learning](#) para traducir contenido a uno o más idiomas, produce versiones de audio de cada traducción y lee sitios web de WordPress a través de dispositivos Amazon Alexa.

El complemento ya está instalado en [WordPress High Availability by Bitnami on AWS Quick Start](#).

## Instalación y configuración del complemento

Para instalar el complemento:

1. Para utilizar el complemento AWS para WordPress, debe crear un usuario de IAM para el complemento. Un usuario de IAM es una persona o aplicación de una cuenta de AWS que tiene permiso para realizar llamadas de API a servicios de AWS.
2. Necesita un rol de [AWS Identity and Access Management \(IAM\)](#) o un usuario de IAM para controlar la autenticación y la autorización de su cuenta de AWS. Para evitar que usuarios no autorizados obtengan estos permisos, proteja las credenciales de usuario de IAM. Trate la clave de acceso secreta como una contraseña; guárdela en un lugar seguro y no la comparta con nadie. Al igual que una contraseña, [rote la clave de acceso periódicamente](#). Si la clave de acceso secreta se filtró accidentalmente, [elimínela de inmediato](#). A continuación, puede crear una nueva clave de acceso para usarla con el complemento de AWS para WordPress.
3. En el menú Plugins (Complementos) del panel de administración de WordPress, busque AWS for WordPress y elija Install Now (Instalar ahora).



4. Si la instalación del complemento no funciona, es posible que haya un problema con los permisos del usuario. Conéctese al servidor web de WordPress y siga estas instrucciones para resolver el problema.
- Abra el archivo `wp-config.php` en el directorio de instalación de WordPress y escriba el siguiente código al final del archivo `wp-config.php`:

```
define('FS_METHOD', 'direct');
```

- Ejecute el siguiente comando para conceder permiso de escritura:

```
chmod 777 <WordPress install directory>/wp-content
```

Advertencia: Es arriesgado mantener el permiso de escritura como 777. De hacerlo, cualquier persona puede editar o eliminar esta carpeta. Cambie el permiso de escritura a 755 o a un número inferior cuando termine el trabajo en el complemento.

- Si usa la arquitectura de referencia, el directorio de instalación de WordPress es `"/var/www/wordpress/<site directory>"`.

Este documento no tiene como objetivo analizar en detalle todos los ajustes de AWS for WordPress. Para obtener información sobre la configuración y las opciones, consulte [Introducción al complemento de AWS para WordPress](#).

## Amazon CloudFront y AWS Certificate Manager

Para configurar CloudFront y AWS Certificate Manager:

1. En el menú del complemento, elija CloudFront e introduzca los siguientes parámetros:
  - Nombre de dominio de origen: dominio DNS del servidor de origen HTTP del que CloudFront obtiene el contenido de su sitio web (por ejemplo, `example.com`).
  - Nombre de dominio alternativo (CNAME): nombre de dominio que sus visitantes utilizan para que la experiencia sea más rápida en el sitio web. AWS recomienda usar “www” delante del dominio (por ejemplo, `www.example.com`).
2. Elija Initiate Setup (Iniciar configuración) para iniciar la configuración.

El complemento solicita automáticamente un certificado SSL para el CNAME a través de ACM. Una vez que valide el token de ACM [actualizando los registros DNS](#) con las entradas de CNAME, el complemento creará una distribución de CloudFront que se ajuste a las prácticas recomendadas definidas en el [Apéndice A](#).

### Note

El complemento de AWS para WordPress requiere HTTPS para la comunicación entre CloudFront y su origen personalizado. Asegúrese de que su origen tenga un certificado SSL válido para el nombre de dominio de origen. Para obtener más información, consulte [Exigir HTTPS para la comunicación entre CloudFront y su origen personalizado](#).

## Traduzca el contenido y haga que se lea en voz alta

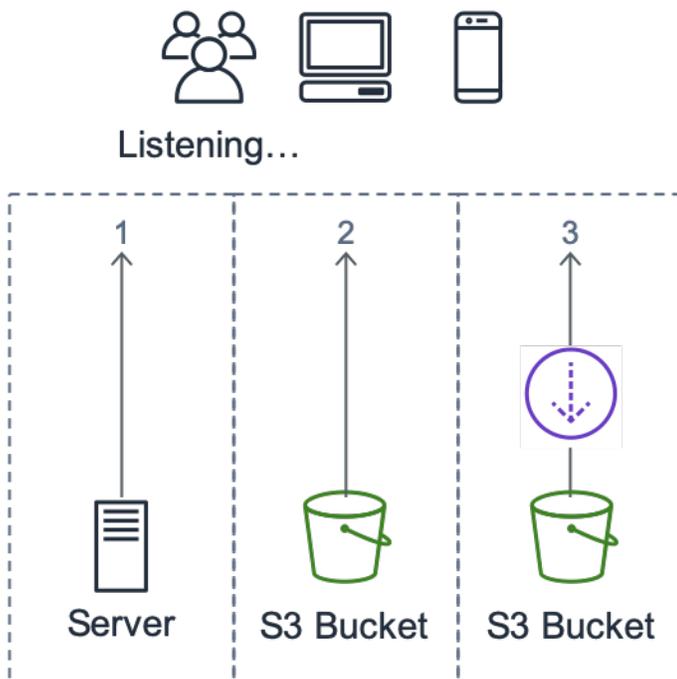
El complemento de AWS para WordPress le permite traducir automáticamente texto a diferentes idiomas y convertir el contenido escrito en formatos de audio multilingües. Estas características utilizan los servicios de Amazon Machine Learning.

[Amazon Polly](#) es un servicio que transforma texto en habla realista. Gracias a las decenas de voces en varios idiomas, puede seleccionar la voz ideal y crear aplicaciones con habla aptas para

numerosos países distintos. Utilice el complemento para crear archivos de audio con cualquiera de las voces e idiomas que admite Amazon Polly. Los visitantes pueden transmitir el audio cuando deseen utilizando reproductores de audio en línea y aplicaciones móviles.

De forma predeterminada, el complemento almacena los archivos de audio nuevos en el servidor web. Puede optar por almacenar los archivos en Amazon S3 o en Amazon CloudFront. Los usuarios disfrutarán de la misma experiencia de escucha independientemente de cómo almacene los archivos de audio. Solo cambia la ubicación de difusión:

- Los archivos de audio almacenados en el servidor de WordPress se transmiten directamente desde el servidor.
- Los archivos almacenados en buckets de S3 se transmiten desde dichos buckets.
- Si usa CloudFront, los archivos se almacenan en Amazon S3 y se transmiten con CloudFront.

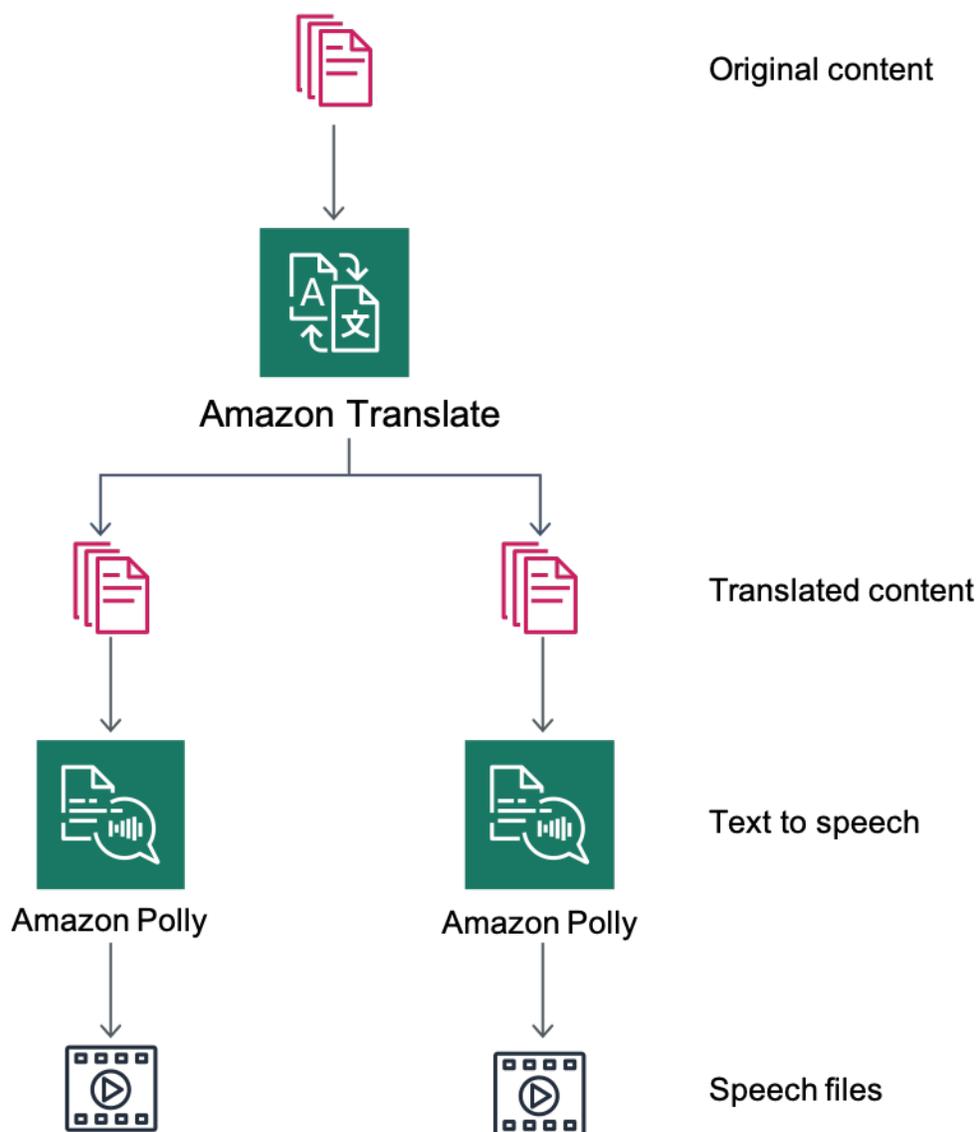


### Ubicación de transmisión

[Amazon Translate](#) es un servicio de traducción automática que permite realizar una traducción lingüística rápida, de gran calidad y asequible. Proporcionar contenido multilingüe es una gran oportunidad para los propietarios de sitios web. A pesar de que el inglés es el idioma dominante en la web, los angloparlantes nativos son tan solo el 26 % del total de la audiencia en línea.

Al ofrecer versiones escritas y orales de su contenido de WordPress en varios idiomas, puede satisfacer las necesidades de una audiencia internacional más amplia. Puede configurar el complemento para hacer lo siguiente:

- Traducir automáticamente el contenido a diferentes idiomas y crear grabaciones de audio de cada traducción del nuevo contenido en el momento de publicarlo, u optar por traducir y crear grabaciones para publicaciones individuales
- Traducir a diferentes idiomas y crear grabaciones de audio para cada traducción del contenido archivado
- Utilizar la fuente RSS de Amazon Pollycast para publicar el contenido de audio como podcast



Información general de la función de traducción de contenido y de conversión de texto a voz

## Transmisión de podcasts mediante Amazon Pollycast

Con las fuentes de Amazon Pollycast, los visitantes pueden escuchar contenido de audio a través de aplicaciones de podcast estándar. Las fuentes de Pollycast, compatibles con RSS 2.0 proporcionan los datos XML necesarios para agregar podcasts mediante aplicaciones móviles de podcasts populares, como iTunes, y directorios de podcast.

Cuando instale el complemento de AWS para WordPress, encontrará la opción de habilitar la generación de fuentes XML en la pestaña de configuración de Podcast. Allí, también encontrará la opción de configurar varias propiedades opcionales. Después de habilitar la funcionalidad, recibirá un enlace para la fuente.

## Lectura de contenido a través de dispositivos Amazon Alexa

Puede ampliar los sitios web y blogs de WordPress con dispositivos Alexa. Esto abre nuevas posibilidades para que los creadores y autores de sitios web lleguen a un público aún más amplio. También permite que la gente escuche más fácilmente sus blogs favoritos con solo pedirle a Alexa que los lea.

Para exponer el sitio web de WordPress a Alexa, debe habilitar lo siguiente:

- El complemento de AWS para WordPress
- Las funcionalidades de conversión de texto a voz y Amazon Pollycast. Estas funcionalidades generan una fuente RSS en su sitio de WordPress que consume Amazon Alexa.
- Amazon S3 como almacenamiento predeterminado de sus archivos en conversión de texto a voz; es importante que su sitio web utilice una conexión HTTPS segura para exponer sus fuentes a Alexa.

El siguiente diagrama presenta el flujo de interacciones y componentes que se necesitan para exponer su sitio web a través de Alexa.



Flujo de interacciones necesarias para exponer los sitios web de WordPress a través de Alexa

1. El usuario invoca una nueva skill de Alexa, por ejemplo, dice: Alexa, pide al blog de demostración la última publicación. La skill en sí se crea con uno de los proyectos de skills de Alexa. Esto le permite exponer su skill a través de los dispositivos Alexa aunque no tenga unos buenos conocimientos técnicos.
2. La skill de Alexa analiza la llamada y la fuente RSS que ha generado el complemento de AWS para WordPress y, a continuación, devuelve el enlace a la versión de audio del último artículo.
3. Según el enlace proporcionado por la fuente, Alexa lee el artículo reproduciendo el archivo de audio guardado en Amazon S3.

Consulte la [página del complemento](#) en el marketplace de WordPress para obtener una guía paso a paso para instalar y configurar el complemento y sus funcionalidades.

## Configuración de contenido estático

De forma predeterminada, WordPress almacena todo localmente en el servidor web, que es el almacenamiento en bloque (Amazon EBS) para la [implementación de un solo servidor](#) y el almacenamiento de archivos (Amazon EFS) para la [implementación elástica](#). Además de reducir los costes de almacenamiento y transferencia de datos, trasladar los recursos estáticos a Amazon S3 permite obtener escalabilidad, disponibilidad de datos, seguridad y rendimiento.

En este ejemplo, el complemento W3 Total Cache (W3TC) se usa para almacenar los recursos estáticos en Amazon S3. Sin embargo, hay otros complementos disponibles con capacidades similares. Si desea utilizar una alternativa, puede ajustar los siguientes pasos en consecuencia. Los pasos solo se refieren a características o configuraciones relevantes para este ejemplo. Este

documento no tiene por objetivo analizar en detalle todos los ajustes. Consulte la [página del complemento W3 Total Cache](#) en [wordpress.org](http://wordpress.org) para obtener más información.

## Creación de usuarios de IAM

Debe crear un usuario de AWS Identity and Access Management (IAM) para que el complemento de WordPress almacene recursos estáticos en Amazon S3. Para ver los pasos, consulte [Creación de un usuario de IAM en su cuenta de AWS](#).

Nota: Los roles de IAM permiten gestionar mejor el acceso a los recursos de AWS, pero en el momento de redactar este documento, el complemento W3 Total Cache no admitía [roles de IAM](#).

Tome nota de las credenciales de seguridad del usuario y almacénelas de manera segura; las necesitará más adelante.

## Creación de buckets de Amazon S3

1. En primer lugar, cree un bucket de Amazon S3 en la región de AWS de su elección. Para ver los pasos, consulte [Creación de un bucket](#). Para habilitar el alojamiento de sitios web estáticos para el bucket, siga el [Tutorial: Configuración de un sitio web estático en Amazon S3](#).
2. Cree una política de IAM para ofrecer acceso al bucket de S3 especificado al usuario de IAM creado previamente y, a continuación, asocie la política al usuario de IAM. Para ver los pasos para crear la siguiente política, consulte [Administración de políticas de IAM](#).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1389783689000",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "s3:DeleteObject",
        "s3:GetObject",
        "s3:GetObjectAcl",
        "s3:ListBucket",
        "s3:PutObject",
        "s3:PutObjectAcl"
      ],
      "Resource": [
        "arn:aws:s3:::wp-demo",
```

```
        "arn:aws:s3:::wp-demo/*"
    ]
}
]
```

3. Instale y active el complemento W3TC desde el panel de administrado de WordPress.
4. Navegue a la sección General Settings (Configuración general) de la configuración del complemento y asegúrese de que están habilitados tanto Browser Cache (Caché del navegador) como CDN.
5. En la lista desplegable de la configuración de CDN, elija Origin Push: Amazon CloudFront (esta opción tiene Amazon S3 como origen).
6. Navegue a la sección de caché del navegador de la configuración del complemento y habilite los encabezados expires (caduca), cache control (control de caché) y entity tag (ETag) (etiqueta de entidad [ETag]).
7. Active también la opción Prevent caching of objects after settings change (Evitar el almacenamiento en caché de objetos después de cambiar la configuración) de modo que siempre que se cambien las configuraciones se genere y se anexe a objetos una nueva cadena de consulta.
8. Navegue a la sección CDN de la configuración del complemento y escriba las credenciales de seguridad del usuario de IAM que creó antes, así como el nombre del bucket de S3.
9. Si va a entregar su sitio web mediante la URL de CloudFront, escriba el nombre del dominio de distribución en la casilla relevante. De lo contrario, escriba uno o más CNAME para los nombres de dominio personalizados.
10. Finalmente, exporte la biblioteca multimedia y cargue los wp-includes, los archivos de temas y los archivos personalizados para Amazon S3 mediante el complemento W3TC. Estas funciones de carga están disponibles en la sección Generalidades de la página de configuración de CDN.

## Creación de origen estático

Ahora que los archivos estáticos están almacenados en Amazon S3, vaya a la configuración de CloudFront en la consola de CloudFront y configure Amazon S3 como el origen del contenido estático. Para ello, añada un segundo origen que apunte al bucket de S3 que creó para ese fin. A continuación, cree dos comportamientos de caché más, uno para cada una de las dos carpetas (wp-content y wp-includes) que deberían usar el origen de S3 en lugar del origen predeterminado para el contenido dinámico. Configure ambas de esta manera:

- Entregue solo solicitudes GET de HTTP.
- Amazon S3 no varía sus salidas en función de cookies o encabezados HTTP, de modo que puede mejorar la eficiencia del almacenamiento en caché al no enviarlas al origen por medio de CloudFront.
- A pesar de que estos comportamientos entregan solo contenido estático (que no acepta parámetros), enviará cadenas de consulta al origen. Esto sirve para que pueda usar cadenas de consultas como identificadores de la versión para invalidar instantáneamente, por ejemplo, los archivos CSS más antiguos al implementar versiones nuevas. Para obtener más información, consulte la [Guía para desarrolladores de Amazon CloudFront](#).

#### Note

Después de añadir los comportamientos de origen estático a su distribución de CloudFront, verifique el orden para asegurarse de que los comportamientos `wp-admin/*` y `wp-login.php` tengan una prioridad mayor que los comportamientos del contenido estático. De otro modo, puede ver algún comportamiento extraño al acceder al panel de administrador.

## Apéndice C: Copia de seguridad y recuperación

En comparación con los entornos de alojamiento tradicionales, la recuperación de un error en AWS es más rápida y fácil. Por ejemplo, puede lanzar un reemplazo en cuestión de minutos en respuesta a un error de hardware o puede usar la conmutación por error automatizada en muchos de nuestros servicios administrados para mitigar el impacto de un reinicio debido al mantenimiento de rutina.

Sin embargo, aún tiene que asegurarse de realizar copias de seguridad de los datos correctos a fin de recuperarlos correctamente. Para reestablecer la disponibilidad de un sitio web de WordPress, tiene que poder recuperar los siguientes componentes:

- Sistema operativo (OS) e instalación y configuración de servicios (Apache, MySQL, etc.)
- Configuración y código de la aplicación de WordPress
- Temas y complementos de WordPress
- Cargas (por ejemplo, archivos multimedia para publicar)
- Contenido de la base de datos (publicaciones, comentarios, etc.)

AWS proporciona diversos métodos para realizar copias de seguridad y restaurar los recursos y los datos de las aplicaciones web.

En este documento técnico, hemos examinado el uso de las instantáneas de Lightsail para proteger todos los datos almacenados en el almacenamiento local de la instancia. Si su sitio web de WordPress solo se ejecuta en la instancia de Lightsail, las instantáneas normales de Lightsail deberían ser suficientes para que pueda recuperar su sitio web de WordPress en su totalidad. Sin embargo, aún perderá los cambios aplicados a su sitio web desde que se tomó la última instantánea si realiza la restauración desde una instantánea.

En una implementación de varios servidores, tiene que hacer copias de seguridad de cada componente que explicamos antes usando diferentes mecanismos. Cada componente puede tener diferentes requisitos relativos a la frecuencia de copia de seguridad. Por ejemplo, la instalación y configuración de WordPress y el sistema operativo cambiarán con mucha menos frecuencia que el contenido generado por el usuario y, por lo tanto, se pueden hacer copias de seguridad menos a menudo sin que se pierdan datos en una recuperación.

Para hacer una copia de seguridad del sistema operativo y la instalación y configuración de los servicios así como de la configuración y el código de la aplicación de WordPress, puede crear una

AMI de una instancia de EC2 correctamente configurada. Las AMI pueden servir para dos propósitos: actuar como copia de seguridad de un estado de instancia y actuar como plantilla al lanzar nuevas instancias.

Para hacer una copia de seguridad de la configuración y el código de la aplicación de WordPress, tiene que usar AMI y también copias de seguridad de Aurora.

Para hacer una copia de seguridad de los complementos y temas de WordPress instalados en su sitio web, tiene que hacer una copia de seguridad del bucket de Amazon S3 o el sistema de archivos de EFS en el que están almacenados.

- Para los temas y complementos almacenados en un bucket de S3, puede habilitar la [replicación entre regiones](#) para que todos los objetos cargados en su bucket principal se repliquen automáticamente en su bucket de copia de seguridad en otra región de AWS. La replicación entre regiones requiere que el control de [versiones](#) esté habilitado tanto en los buckets de origen como en los de destino, lo que le proporciona una capa adicional de protección y le permite volver a una versión anterior de cualquier objeto determinado en su bucket.
- Para los temas y complementos almacenados en un sistema de archivos EFS, puede crear un AWS Data Pipeline para copiar datos desde su sistema de archivos EFS de producción a otro sistema de archivos EFS, como se describe en la página de la documentación [Copia de seguridad de los sistemas de archivos de Amazon EFS](#). También puede realizar copias de seguridad de un sistema de archivos EFS con cualquier aplicación de copia de seguridad con la que ya esté familiarizado.
- Para hacer una copia de seguridad de las cargas del usuario debe seguir los pasos descritos antes para hacer una copia de seguridad de los temas y los complementos de WordPress.
- Para hacer copias de seguridad del contenido de la base de datos, debe utilizar la [copia de seguridad de Aurora](#). Aurora crea copias de seguridad del volumen de clúster automáticamente y conserva los datos de restauración durante el tiempo asignado al periodo de retención de copia de seguridad. Aurora realiza copias de seguridad continuas e incrementales para que se puedan restaurar con rapidez a cualquier punto durante el periodo de retención de copia de seguridad. No se produce ningún impacto en el rendimiento ni ninguna interrupción del servicio de base de datos durante la escritura de los datos de copia de seguridad. Puede especificar un periodo de retención de copia de seguridad con un valor de entre 1 día y 35 días. También puede crear [instantáneas de base de datos manuales](#), que se conservan hasta que las borre. Las instantáneas de base de datos manuales son útiles para copias de seguridad y archivado a largo plazo.

## Apéndice D: Implementación de nuevos complementos y temas

Pocos sitios web son estáticos. En la mayoría de los casos, añadirá periódicamente temas y complementos de WordPress que están disponibles al público o realizará actualizaciones a versiones de WordPress más recientes. En otros casos, desarrollará sus propios temas y complementos a partir de cero.

Siempre que haga un cambio estructural en su instalación de WordPress hay cierto riesgo de introducir problemas imprevistos. Como mínimo, realice una copia de seguridad del código de la aplicación, la configuración y la base de datos antes de realizar cualquier cambio significativo (como instalar un nuevo complemento). Para sitios web de empresas o de valor, primero pruebe esos cambios en un entorno provisional distinto. Con AWS, es fácil replicar la configuración del entorno de producción y ejecutar todo el proceso de implementación de forma segura. Después de hacer las pruebas, simplemente puede dismantelar el entorno de prueba y dejar de pagar por esos recursos. Más adelante, en este documento técnico se analizan algunos aspectos específicos de WordPress.

Algunos complementos escriben la información de configuración en la tabla de la base de datos `wp_options` (o introducen cambios en el esquema de la base de datos), mientras que otros crean archivos de configuración en el directorio de instalación de WordPress. Como hemos movido la base de datos y el almacenamiento a plataformas compartidas, estos cambios estarán disponibles de inmediato para todas las instancias en ejecución sin más esfuerzo por su parte.

Al implementar nuevos temas en WordPress, se necesita algo más de esfuerzo. Si solo usa Amazon EFS para almacenar todos sus archivos de instalación de WordPress, los nuevos temas estarán disponibles de inmediato para todas sus instancias en ejecución. Sin embargo, si va a descargar contenido estático a Amazon S3, debe procesar una copia en la ubicación de bucket correcta. Los complementos como W3 Total Cache ofrecen una forma de iniciar manualmente esa tarea. De otro modo, puede automatizar este paso como parte de un proceso de compilación.

Como los activos de temas se pueden almacenar en caché en CloudFront y en el navegador, necesita una forma de invalidar versiones más antiguas cuando implementa los cambios. La mejor forma de lograrlo es incluir alguna clase de identificador de versión en su objeto. Este identificador puede ser una cadena de consulta con una marca temporal de fecha o una cadena aleatoria. Si usa el complemento W3 Total Cache, puede actualizar la cadena de consulta multimedia que se anexa a las URL de los archivos multimedia.

# Avisos

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

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