



Guía para desarrolladores

AWS Elastic Beanstalk



AWS Elastic Beanstalk: Guía para desarrolladores

Copyright © 2024 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 que menosprecie o desacredite a Amazon. Todas las demás marcas registradas 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

¿Qué es AWS Elastic Beanstalk?	1
Precios	2
Pasos siguientes	2
Introducción	4
Configuración: cree una cuenta AWS	4
Inscríbase en una Cuenta de AWS	4
Creación de un usuario con acceso administrativo	5
Paso 1: Crear	6
Creación de una aplicación y un entorno	6
AWS recursos creados para la aplicación de ejemplo	11
Paso 2: Explorar	13
Paso 3: Implementar una nueva versión	15
Paso 4: Configurar	17
Realizar un cambio de configuración	17
Verificar el cambio de configuración	18
Paso 5: Eliminar	19
Sigüientes pasos	20
Conceptos	24
Aplicación	24
Versión de la aplicación	24
Entorno	24
Capa del entorno	24
Configuración del entorno	25
Configuración guardada	25
Plataforma	25
Entornos del servidor web	25
Entornos de trabajo	27
Consideraciones sobre el diseño	28
Escalabilidad	29
Seguridad	29
Almacenamiento persistente	30
Tolerancia a errores	31
Entrega de contenido	32
Actualizaciones de software y parches	32

Conectividad	32
Permisos	34
Rol de servicio	35
Perfil de instancia	45
Política de usuario	46
Plataformas	47
Glosario de plataformas	47
Modelo de responsabilidad compartida	51
Política de compatibilidad con plataformas	52
Ramificaciones de la plataforma retiradas	52
Después del período de gracia de 90 días	53
Calendario de plataformas	54
Recursos de planificación	54
Próximos lanzamientos de las sucursales de plataformas	55
Programación de sucursales de plataforma de retirada	56
Historial de la ramificación de la plataforma retirada	57
Historial del servidor y del sistema operativo	61
Plataformas admitidas	63
Plataformas admitidas	63
Plataformas Linux	64
Versiones de Amazon Linux compatibles	65
Lista de plataformas Linux de Elastic Beanstalk	66
Ampliación de plataformas Linux	67
Uso de Docker	93
Sucursales de la plataforma Docker	93
Ramificación de la plataforma Docker	95
Ramificación de la plataforma administrada por ECS	129
Contenedores preconfigurados	159
Uso de Go	168
QuickStart para Go	168
Entorno de desarrollo	175
La plataforma Go	176
Uso de Java	184
Introducción	186
Entorno de desarrollo	196
La plataforma Tomcat	199

La plataforma Java SE	217
Añadir una base de datos	228
Conjunto de herramientas de Eclipse	237
Recursos	258
Trabajar con .NET Core en Linux	259
QuickStart para .NET Core en Linux	259
Entorno de desarrollo	267
.NET Core en la plataforma Linux	268
Con la AWS Toolkit for Visual Studio	275
Migración de Windows a Linux	300
Trabajar con .NET en un servidor Windows	301
Ramas de la plataforma Windows 2012 retiradas	303
QuickStart para .NET Core en Windows	305
Tutorial: ASP.NET Core	312
Entorno de desarrollo	325
La plataforma .NET	326
Añadir una base de datos	340
El AWS Toolkit for Visual Studio	344
Migración de la aplicación local	379
Uso de Node.js	380
QuickStart para Node.js	380
Entorno de desarrollo	388
La plataforma Node.js	391
Ejemplos de aplicaciones y tutoriales	408
Tutorial: Express	410
Tutorial: Express con agrupación en clústeres	422
Tutorial - Node.js con DynamoDB	440
Añadir una base de datos	452
Recursos	456
Uso de PHP	456
QuickStart para PHP	457
Entorno de desarrollo	464
Plataforma PHP	467
Ejemplos de aplicaciones y tutoriales	477
Uso de Python	557
Entorno de desarrollo	558

La plataforma Python	561
Tutorial: Flask	570
Tutorial: Django	578
Añadir una base de datos	592
Recursos	595
Uso de Ruby	596
Entorno de desarrollo	596
La plataforma Ruby	599
Tutorial: Rails	607
Tutorial: Sinatra	616
Añadir una base de datos	622
Tutoriales y ejemplos	626
Administración de aplicaciones	629
Consola de administración de la aplicación	631
Administración de versiones de la aplicación	632
Ciclo de vida de la versión	636
Etiquetado de versiones de la aplicación	639
Cree un paquete de código fuente.	641
Creación de un paquete de código fuente desde la línea de comandos	642
Creación de un paquete de código fuente con Git	643
Comprimir archivos en Mac OS X Finder o en el Explorador de Windows	643
Creación de un paquete de código fuente para una aplicación .NET	646
Comprobación del paquete de código fuente	648
Etiquetado de recursos de	648
Propagación de etiquetas para plantillas de lanzamiento	650
Recursos que se pueden etiquetar	651
Etiquetado de aplicaciones	651
Administración de entornos	655
Consola de administración del entorno	656
Información general del entorno	657
Acciones del entorno	660
Eventos	662
Estado	663
Registros	664
Supervisión	665
Alarmas	665

Managed Updates (Actualizaciones administradas)	666
Etiquetas	666
Configuración	667
Creación de entornos	669
El asistente de creación de nuevo entorno	677
Clonación de un entorno	700
Terminación de un entorno	703
Con el AWS CLI	705
Con la API	707
URL de lanzamiento inmediato	711
Compose Environments	717
Implementaciones	720
Elección de una política de implementación	721
Implementación de una nueva versión de la aplicación	724
Nueva implementación de una versión anterior	724
Otras formas de implementar la aplicación	725
Opciones de implementación	725
Implementaciones blue/green (azul/verde)	734
Cambios de configuración	737
Actualizaciones continuas	739
Actualizaciones inmutables	744
Actualizaciones de la plataforma	748
Método 1: actualización de la versión de la plataforma del entorno	752
Método 2: realización de una implementación "blue/green"	754
Managed Updates (Actualizaciones administradas)	755
Actualización de un entorno heredado	763
Migración a AL2023/AL2	765
Preguntas frecuentes sobre la retirada de plataformas	783
Cancelar una actualización	788
Volver a crear un entorno	790
Volver a crear un entorno en ejecución	790
Volver a crear un entorno terminado	791
Tipos de entornos	793
Entorno escalable con balanceo de carga	793
Entorno de una sola instancia	794
Cambiar el tipo de entorno	794

Entornos de trabajo	795
El daemon de SQS del entorno de trabajo	798
Colas de mensajes fallidos	799
Tareas periódicas	800
Uso de Amazon CloudWatch para el escalado automático en capas del entorno de trabajo	802
Configuración de entornos de trabajo	802
Enlaces de entorno	806
Configuración de entornos	809
Configuración usando la consola	811
Página de configuración	811
Página Review Changes (Revisar cambios)	813
Instancias de Amazon EC2	814
Tipos de instancias de Amazon EC2	815
Configuración de instancias de Amazon EC2 para su entorno	816
Configuración de las instancias AWS EC2 para su entorno mediante el AWS CLI	824
Recomendaciones para los entornos de primera oleada Graviton arm64	828
El espacio de nombres <code>aws:autoscaling:launchconfiguration</code>	830
IMDS	831
Grupo de Auto Scaling	834
Compatibilidad con instancias de spot	835
Configuración del grupo de Auto Scaling mediante la consola de Elastic Beanstalk	840
Configuración del grupo de Auto Scaling a través de la CLI de EB	844
Opciones de configuración	845
Desencadenadores	846
Acciones programadas	849
Configuración de comprobaciones de estado	854
Balancedador de carga	855
Equilibrador de carga clásico	857
Equilibrador de carga de aplicación	869
Balancedador de carga de aplicaciones compartido.	891
Network Load Balancer	911
Configuración de registros de acceso	924
Base de datos	924
Ciclo de vida de la base de datos	925
Agregar una instancia de base de datos de Amazon RDS al entorno utilizando la consola ..	926

Conexión a la base de datos	929
Configuración de una instancia de base de datos de RDS integrada mediante la consola	929
Configuración de una instancia de base de datos de RDS integrada utilizando archivos de configuración	930
Desacoplamiento de una instancia de base de datos de RDS utilizando la consola	931
Desacoplamiento de una instancia de base de datos de RDS utilizando archivos de configuración	934
Seguridad	936
Configuración de la seguridad de su entorno	937
Espacios de nombres de configuración de seguridad del entorno	939
Etiquetado de entornos	940
Adición de etiquetas durante la creación del entorno	940
Administración de etiquetas de un entorno existente	941
Propiedades del entorno y configuración de software	944
Configurar la configuración específica de la plataforma	944
Configuración de las propiedades del entorno (variables de entorno)	946
Espacios de nombres para configuración de software	947
Acceso a las propiedades del entorno	949
Debugging	951
Visualización de registros	954
Notificaciones	957
Configuración de notificaciones mediante la consola de Elastic Beanstalk	958
Configuración de las notificaciones con las opciones de configuración	959
Configuración de los permisos para enviar notificaciones	961
Amazon VPC	963
Configuración de la VPC en la consola de Elastic Beanstalk	964
El espacio de nombres aws:ec2:vpc	967
Migración de EC2-Classical a una VPC	968
Nombre de dominio	973
Configuración de entornos (procedimiento avanzado)	975
Opciones de configuración	976
Prioridad	976
Valores recomendados	977
Antes de crear el entorno	979
Durante la creación	986
Después de la creación	993

Opciones generales	1004
Opciones específicas de la plataforma	1093
Opciones personalizadas	1108
.Ebextensions	1109
Opciones de configuración	1112
Servidor Linux	1114
Servidor Windows	1132
Recursos personalizados	1142
Configuraciones guardadas	1171
Etiquetado de configuraciones guardadas	1177
env.yaml	1180
Imagen personalizada	1183
Creación de una AMI personalizada	1183
Limpieza de una AMI personalizada	1188
AMI basada en una plataforma retirada	1188
Archivos estáticos	1195
Configurar archivos estáticos mediante la consola	1195
Configurar archivos estáticos mediante opciones de configuración	1196
HTTPS	1197
Crear un certificado	1199
Carga de un certificado	1202
Terminar conexiones en el balanceador de carga	1204
Terminar en la instancia	1208
Cifrado integral	1245
Paso a través de TCP	1249
Almacenamiento seguro de claves	1250
Redireccionamiento de HTTP a HTTPS	1252
Monitorización de un entorno	1254
Consola de administración	1254
Gráficos de monitorización	1255
Personalización de la consola de monitorización	1256
Informes de estado básicos	1257
Colores de estado	1258
Comprobaciones de estado Elastic Load Balancing	1259
Comprobaciones de estado de entornos de la capa de trabajo y de una sola instancia	1260
Comprobaciones adicionales	1260

CloudWatch Métricas de Amazon	1260
Informes y monitorización de estado mejorados	1262
El agente de estado de Elastic Beanstalk	1265
Factores que determinan el estado de las instancias y del entorno	1266
Personalización de reglas de comprobación de estado	1269
Roles de estado mejorados	1270
Autorización de estado mejorada	1270
Eventos de estado mejorados	1271
Comportamiento de los informes de estado mejorados durante las actualizaciones, implementaciones y escala	1272
Habilitación de informes de estado avanzados	1273
Consola de estado	1278
Colores y estados	1283
Métricas de la instancia	1287
Reglas de informes de estado avanzados	1290
CloudWatch	1295
Usuarios de API	1304
Formato de los registros de estado avanzados	1306
Notificaciones y solución de problemas	1310
Administrar alarmas	1313
Ver historial de cambios	1316
Ver eventos	1318
Monitorización de instancias	1320
Consulta de los registros de instancias	1323
Ubicación de registro en instancias de Amazon EC2	1325
Ubicación de registro en Amazon S3	1326
Configuración de rotación de registros en Linux	1327
Ampliación de la configuración predeterminada de la tarea de registro	1328
Transmisión de archivos de registro a Amazon CloudWatch Logs	1331
Integración de los servicios de AWS	1333
Información general sobre la arquitectura	1333
CloudFront	1334
CloudTrail	1335
Información de Elastic Beanstalk en el historial de CloudTrail	1335
Descripción de las entradas del archivo de registro de Elastic Beanstalk	1336
CloudWatch	1337

Registros de CloudWatch	1338
Requisitos previos para el streaming de registros de instancias a CloudWatch Logs	1340
Cómo Elastic Beanstalk configura CloudWatch Logs	1341
Streaming de registros de instancias a CloudWatch Logs	1347
Solución de problemas con la integración de CloudWatch Logs	1349
Streaming del estado del entorno	1350
EventBridge	1353
Supervisar un recurso de Elastic Beanstalk con EventBridge	1354
Ejemplo de patrones de eventos de Elastic Beanstalk	1357
Ejemplo de eventos de Elastic Beanstalk	1359
Asignación de campos de eventos de Elastic Beanstalk	1360
AWS Config	1364
Configuración de AWS Config	1365
Configuración de AWS Config para registrar recursos de Elastic Beanstalk	1365
Visualización de los detalles de configuración de Elastic Beanstalk en la consola de AWS Config	1366
Evaluación de los recursos de Elastic Beanstalk mediante reglas de AWS Config	1370
DynamoDB	1371
ElastiCache	1372
Amazon EFS	1373
Archivos de configuración	1373
Sistemas de archivos cifrados	1374
Aplicaciones de muestra	1375
Limpieza de sistemas de archivos	1375
IAM	1376
Perfiles de instancias	1376
Roles de servicio	1381
Uso de roles vinculados a servicios	1396
Políticas de usuario	1408
Formato de ARN	1416
Recursos y condiciones	1418
Control de acceso basado en etiquetas	1464
Ejemplo de políticas administradas	1468
Ejemplo de políticas específicas de recursos	1472
Acceso al bucket S3 entre entornos	1482
Amazon RDS	1485

Amazon RDS en VPC predeterminada	1486
Amazon RDS en EC2 Classic	1493
Credenciales de Amazon RDS y Secrets Manager	1499
Limpieza de una instancia externa de Amazon RDS	1500
Simple Storage Service (Amazon S3)	1500
Contenido del bucket de Elastic Beanstalk Amazon S3	1500
Eliminar objetos en el bucket de Elastic Beanstalk Amazon S3	1502
Eliminar el bucket de Elastic Beanstalk Amazon S3	1502
Amazon VPC	1504
VPC pública	1505
VPC pública/privada	1506
VPC privada	1507
Hosts bastión	1509
Amazon RDS	1514
Puntos de conexión de la VPC	1522
Configuración de su máquina de desarrollo	1525
de una carpeta del proyecto	1525
Configurar el control de la fuente	1526
Configurar un repositorio remoto	1526
Instalación de la CLI de EB	1527
Instalación de AWS CLI	1527
CLI DE EB	1528
Instalación de la CLI de EB	1529
Instalación de la CLI de EB con scripts de configuración	1530
Instalación manual	1530
Configuración de la CLI de EB	1541
Ignorar archivos usando .ebignore	1543
Usar perfiles con nombre	1544
Implementar un artefacto en lugar de la carpeta del proyecto	1544
Ajustes de configuración y precedencia	1545
Metadatos de instancia	1545
Conceptos básicos de la CLI de EB	1546
Eb create	1547
Eb status	1547
Eb health	1548
Eb events	1549

Eb logs	1549
Eb open	1550
Eb deploy	1550
Eb config	1551
Eb terminate	1551
CodeBuild	1553
Creación de una aplicación	1553
Creación e implementación del código de la aplicación	1553
Uso de la CLI de EB con Git	1556
Asociación de entornos Elastic Beanstalk con ramas de Git	1556
Implementación de cambios	1556
Uso de submódulos de Git	1557
Asignación de etiquetas de Git a la versión de la aplicación	1558
CodeCommit	1558
Requisitos previos	1559
Crear un repositorio de CodeCommit con la CLI de EB	1560
Implementación desde su repositorio CodeCommit	1561
Configuración de entornos y ramificaciones adicionales	1562
Usar un repositorio de CodeCommit existente	1563
Monitorización del estado	1564
Descripción de la salida	1568
Vista de estado interactiva	1570
Opciones de la vista de estado interactiva	1572
Composición de entornos	1572
Solución de problemas	1575
Solución de problemas de las implementaciones	1575
Comandos de la CLI de EB	1578
eb abort	1579
eb appversion	1581
eb clone	1585
eb codesource	1588
eb config	1590
eb console	1600
eb create	1600
eb deploy	1618
eb events	1620

eb health	1622
eb init	1624
eb labs	1628
eb list	1629
eb local	1630
eb logs	1634
eb open	1638
eb platform	1639
eb printenv	1649
eb restore	1650
eb scale	1652
eb setenv	1653
eb ssh	1654
eb status	1657
eb swap	1659
eb tags	1660
eb terminate	1664
eb upgrade	1666
eb use	1668
Opciones comunes	1668
CLI de EB 2.6 (retirado)	1669
Diferencias de la versión 3 de la CLI de EB	1669
Migración a EB CLI 3 y CodeCommit	1670
CLI de la API de EB (retirada)	1671
Conversión de scripts de la CLI de la API de Elastic Beanstalk	1671
Seguridad	1676
Protección de los datos	1677
Cifrado de datos	1678
Privacidad entre redes	1679
Identity and Access Management	1679
AWS políticas gestionadas	1680
Registro y monitoreo	1692
Informes de estado mejorados	1692
Registros de instancias de Amazon EC2	1692
Notificaciones del entorno	1693
Alarmas de Amazon CloudWatch	1693

Registros de AWS CloudTrail	1693
Debugging deAWS X-Ray	1693
Validación de la conformidad	1694
Resiliencia	1695
Seguridad de la infraestructura	1695
Modelo de responsabilidad compartida	1696
Prácticas recomendadas de seguridad	1696
Prácticas recomendadas de seguridad preventiva	1696
Prácticas recomendadas de detección de seguridad	1698
Solución de problemas	1699
Uso de la herramienta Systems Manager	1699
General guidance	1701
Categorías	1701
Conectividad	1702
Creación del entorno	1702
Implementaciones	1703
Estado	1704
Configuración	1704
Docker	1705
Preguntas frecuentes	1705
Recursos	1708
Aplicaciones de muestra	1709
Historial de la plataforma	1710
Plataformas personalizadas	1710
Historial de documentos	1728
.....	mdccxxx

¿Qué es AWS Elastic Beanstalk?

Amazon Web Services (AWS) incluye más de cien servicios, cada uno de los cuales se ocupa de un área de funcionalidad. Aunque la gran variedad de servicios le brinda la flexibilidad necesaria para decidir cómo quiere administrar la infraestructura de AWS, puede resultar complicado saber qué servicios deben usarse y cómo aprovisionarlos.

Con Elastic Beanstalk, puede implementar y administrar aplicaciones rápidamente en la nube de AWS sin tener que preocuparse por la infraestructura que las ejecuta. Elastic Beanstalk reduce la complejidad de la administración sin restringir la libertad de elección ni el control. Solo tiene que cargar la aplicación y Elastic Beanstalk gestionará de manera automática los detalles de aprovisionamiento de capacidad, balanceo de carga, escalado y monitorización de estado de la aplicación.

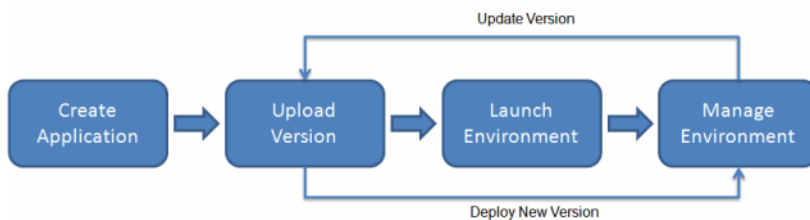
Elastic Beanstalk es compatible con aplicaciones desarrolladas en Go, Java, .NET, Node.js, PHP, Python y Ruby. Cuando implementa su aplicación, Elastic Beanstalk crea la versión de la plataforma compatible seleccionada y aprovisiona uno o varios recursos de AWS, como instancias de Amazon EC2, para ejecutar la aplicación.

Puede interactuar con Elastic Beanstalk a través de la consola de Elastic Beanstalk, la AWS Command Line Interface (AWS CLI) o `eb`, una CLI de nivel superior diseñada específicamente para Elastic Beanstalk.

Para obtener más información sobre cómo implementar una aplicación web de ejemplo con Elastic Beanstalk, consulte [Introducción a AWS: Implementación de una aplicación web](#).

También puede realizar la mayoría de las tareas de implementación, como cambiar el tamaño de la flota de instancias de Amazon EC2 o monitorizar la aplicación, directamente desde la interfaz web de Elastic Beanstalk (consola).

Para utilizar Elastic Beanstalk, debe crear una aplicación, cargar una versión de la aplicación como un paquete de código fuente (por ejemplo, un archivo `Java.war`) en Elastic Beanstalk y proporcionar cierta información sobre la aplicación. Elastic Beanstalk lanza automáticamente un entorno y crea y configura los recursos de AWS necesarios para ejecutar el código. Una vez que se lanza el entorno, puede administrarlo e implementar nuevas versiones de la aplicación. En el siguiente diagrama, se ilustra el flujo de trabajo de Elastic Beanstalk.



Después de crear e implementar la aplicación, puede consultar información sobre ella (por ejemplo, métricas, eventos y el estado del entorno) en la consola de Elastic Beanstalk, las API o las interfaces de línea de comandos, como la AWS CLI unificada.

Precios

No se aplican cargos adicionales por utilizar Elastic Beanstalk. Solo tiene que pagar por los recursos subyacentes de AWS que consume la aplicación. Para obtener información sobre los precios, consulte la [página de detalles del servicio de Elastic Beanstalk](#).

Pasos siguientes

Esta guía contiene información conceptual sobre el servicio web de Elastic Beanstalk, así como información acerca de cómo utilizar el servicio para implementar aplicaciones web. En las diferentes secciones, se explica cómo se utiliza la consola de Elastic Beanstalk, las herramientas de la interfaz de línea de comandos (CLI) y la API para implementar y administrar los entornos de Elastic Beanstalk. En esta guía, también se explica cómo se integra Elastic Beanstalk con otros servicios proporcionados por Amazon Web Services.

Le recomendamos que lea primero [Introducción a Elastic Beanstalk](#) para obtener información sobre cómo empezar a utilizar Elastic Beanstalk. En esta Introducción, aprenderá paso a paso a crear, consultar y actualizar la aplicación de Elastic Beanstalk, así como a editar y terminar el entorno de Elastic Beanstalk. En la Introducción se explican también diferentes mecanismos para obtener acceso a Elastic Beanstalk.

Para obtener más información sobre la aplicación de Elastic Beanstalk y sus componentes, consulte las páginas siguientes.

- [Conceptos de Elastic Beanstalk](#)
- [Glosario de plataformas Elastic Beanstalk](#)
- [Modelo de responsabilidad compartida para el mantenimiento de plataformas Elastic Beanstalk](#)

- [Política de compatibilidad con plataformas Elastic Beanstalk](#)

Introducción a Elastic Beanstalk

Para ayudarle a entender cómo AWS Elastic Beanstalk funciona, en este tutorial se explica cómo crear, explorar, actualizar y eliminar una aplicación de Elastic Beanstalk. Tarda menos de una hora en completarse.

El uso de Elastic Beanstalk es gratuito, AWS pero los recursos que crea para este tutorial están disponibles (y no se ejecutan en un entorno aislado). Se le facturan las tarifas de uso estándar por estos recursos hasta que los termine al final de este tutorial. Los costos totales son, por lo general, inferiores a un dólar. Para obtener información sobre cómo minimizar los costos, consulte [nivel gratuito de AWS](#).

Temas

- [Configuración: cree una cuenta AWS](#)
- [Paso 1: Crear una aplicación de ejemplo](#)
- [Paso 2: Explorar el entorno](#)
- [Paso 3: Implementar una nueva versión de la aplicación](#)
- [Paso 4: Configurar su entorno](#)
- [Paso 5: Eliminar](#)
- [Sigüientes pasos](#)

Configuración: cree una cuenta AWS

Si aún no es AWS cliente, debe crear una AWS cuenta. El registro le permite acceder a Elastic Beanstalk AWS y a otros servicios que necesite.

Inscríbase en una Cuenta de AWS

Si no tiene una Cuenta de AWS, complete los siguientes pasos para crearlo.

Para suscribirse a una Cuenta de AWS

1. Abra <https://portal.aws.amazon.com/billing/signup>.
2. Siga las instrucciones que se le indiquen.

Parte del procedimiento de registro consiste en recibir una llamada telefónica e indicar un código de verificación en el teclado del teléfono.

Cuando te registras en una Cuenta de AWS, Usuario raíz de la cuenta de AWS se crea una. El usuario raíz tendrá acceso a todos los Servicios de AWS y recursos de esa cuenta. Como práctica recomendada de seguridad, asigne acceso administrativo a un usuario y utilice únicamente el usuario raíz para realizar [tareas que requieren acceso de usuario raíz](#).

AWS te envía un correo electrónico de confirmación una vez finalizado el proceso de registro. Puede ver la actividad de la cuenta y administrar la cuenta en cualquier momento entrando en <https://aws.amazon.com/> y seleccionando Mi cuenta.

Creación de un usuario con acceso administrativo

Después de crear un usuario administrativo Cuenta de AWS, asegúrelo Usuario raíz de la cuenta de AWS AWS IAM Identity Center, habilite y cree un usuario administrativo para no usar el usuario root en las tareas diarias.

Proteja su Usuario raíz de la cuenta de AWS

1. Inicie sesión [AWS Management Console](#) como propietario de la cuenta seleccionando el usuario root e introduciendo su dirección de Cuenta de AWS correo electrónico. En la siguiente página, escriba su contraseña.

Para obtener ayuda para iniciar sesión con el usuario raíz, consulte [Signing in as the root user](#) en la Guía del usuario de AWS Sign-In .

2. Active la autenticación multifactor (MFA) para el usuario raíz.

Para obtener instrucciones, consulte [Habilitar un dispositivo MFA virtual para el usuario Cuenta de AWS raíz \(consola\)](#) en la Guía del usuario de IAM.

Creación de un usuario con acceso administrativo

1. Activar IAM Identity Center.

Consulte las instrucciones en [Activar AWS IAM Identity Center](#) en la Guía del usuario de AWS IAM Identity Center .

2. En IAM Identity Center, conceda acceso administrativo a un usuario.

Para ver un tutorial sobre su uso Directorio de IAM Identity Center como fuente de identidad, consulte [Configurar el acceso de los usuarios con la configuración predeterminada Directorio de IAM Identity Center en la](#) Guía del AWS IAM Identity Center usuario.

Iniciar sesión como usuario con acceso de administrador

- Para iniciar sesión con el usuario de IAM Identity Center, utilice la URL de inicio de sesión que se envió a la dirección de correo electrónico cuando creó el usuario de IAM Identity Center.

Para obtener ayuda para iniciar sesión con un usuario del Centro de identidades de IAM, consulte [Iniciar sesión en el portal de AWS acceso](#) en la Guía del AWS Sign-In usuario.

Concesión de acceso a usuarios adicionales

1. En IAM Identity Center, cree un conjunto de permisos que siga la práctica recomendada de aplicar permisos de privilegios mínimos.

Para conocer las instrucciones, consulte [Create a permission set](#) en la Guía del usuario de AWS IAM Identity Center .

2. Asigne usuarios a un grupo y, a continuación, asigne el acceso de inicio de sesión único al grupo.

Para conocer las instrucciones, consulte [Add groups](#) en la Guía del usuario de AWS IAM Identity Center .

Paso 1: Crear una aplicación de ejemplo

En este paso, creará una nueva aplicación a partir de una aplicación de ejemplo preexistente. Elastic Beanstalk admite plataformas para distintos lenguajes de programación, servidores de aplicaciones y contenedores de Docker. Puede elegir una plataforma al crear la aplicación.

Creación de una aplicación y un entorno

Para crear su aplicación de ejemplo, usará el asistente de la consola Create application (Creación de aplicación). Crea una aplicación de Elastic Beanstalk y lanza un entorno dentro de esta. Un entorno es el conjunto de AWS recursos necesarios para ejecutar el código de la aplicación.

Para crear una aplicación de ejemplo

1. Abra la [consola de Elastic Beanstalk](#).
2. Elija Crear aplicación.
3. En Application name (Nombre de la aplicación), escriba **getting-started-app**.
4. Si lo desea, agregue [etiquetas de aplicación](#).
5. En Platform (Plataforma), elija una plataforma.
6. Elija Siguiente.
7. Aparece la página Configurar acceso a servicio.
8. Seleccione Crear un rol de servicio para Rol de servicio.
9. A continuación, nos centraremos en la lista desplegable Perfil de instancia de EC2. Los valores que aparecen en esta lista desplegable pueden variar, en función de si su cuenta ha creado previamente un entorno nuevo.

Seleccione una de las siguientes opciones, dependiendo de los valores que aparezcan en la lista.

- Si `aws-elasticbeanstalk-ec2-role` aparece en la lista desplegable, selecciónelo en la lista desplegable Perfil de instancia de EC2.
- Si aparece otro valor en la lista y es el perfil de instancia de EC2 predeterminado destinado a sus entornos, selecciónelo en la lista desplegable Perfil de instancia de EC2.
- Si en la lista desplegable Perfil de instancia de EC2 no aparecen valores para seleccionar, amplíe el procedimiento siguiente, Crear un rol de IAM para el perfil de instancia de EC2.

Siga los pasos de Crear un rol de IAM para el perfil de instancia de EC2 para crear un rol de IAM que pueda seleccionar posteriormente para el Perfil de instancia de EC2. Luego, vuelva a este paso.

Ahora que ha creado un rol de IAM y ha actualizado la lista, aparece como opción en la lista desplegable. Seleccione el rol de IAM que acaba de crear en la lista desplegable Perfil de instancia de EC2.

10. Seleccione Skip to review (Saltar a revisión) en la página Configure service access (Configurar el acceso al servicio).

Esto permite omitir los pasos opcionales.

11. En la página Review (Revisar), se muestra un resumen de todas sus elecciones.

Elija Submit (Enviar) en la parte inferior de la página.

Crear un rol de IAM para el perfil de instancia de EC2

Configure service access [Info](#)

Service access
IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profiles allow Elastic Beanstalk to create and manage your environment. Both the IAM role and instance profile must be attached to IAM managed policies that contain the required permissions. [Learn more](#)

Service role

Create and use new service role

Use an existing service role

Existing service roles
Choose an existing IAM role for Elastic Beanstalk to assume as a service role. The existing IAM role must have the required IAM managed policies.

aws-elasticbeanstalk-service-role

EC2 key pair
Select an EC2 key pair to securely log in to your EC2 instances. [Learn more](#)

Choose a key pair

EC2 instance profile
Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.

aws-elasticbeanstalk-ec2-role

[View permission details](#)

Cancel [Skip to review](#) [Previous](#) [Next](#)


Para crear un rol de IAM para la selección del perfil de instancia de EC2

1. Seleccione Ver detalles de permisos. Esta opción aparece en la lista desplegable Perfil de instancia de EC2.

Aparece una ventana modal titulada Ver permisos del perfil de instancia. En esta ventana se muestran los perfiles administrados que deberá asociar al nuevo perfil de instancia de EC2 que cree. También proporciona un enlace para ejecutar la consola de IAM.

2. Seleccione el enlace Consola de IAM que aparece en la parte superior de la ventana.
3. En el panel de navegación de la consola de IAM, elija Roles.
4. Elija Crear rol.
5. En Tipo de entidad de confianza, seleccione Servicio de AWS .
6. En Caso de uso, elija EC2.

7. Elija Siguiente.
8. Asocie las políticas administradas adecuadas. Desplácese por la ventana modal Ver permisos del perfil de instancia para ver las políticas administradas. Las políticas también aparecen aquí:
 - `AWSElasticBeanstalkWebTier`
 - `AWSElasticBeanstalkWorkerTier`
 - `AWSElasticBeanstalkMulticontainerDocker`
9. Elija Siguiente.
10. Escriba un nombre para el rol.
11. (Opcional) Añada etiquetas al rol.
12. Elija Crear rol.
13. Vuelva a la ventana de la consola de Elastic Beanstalk que está abierta.
14. Cierre la ventana modal Ver permisos del perfil de instancia.

 Important

No cierre la página del navegador donde se muestra la consola de Elastic Beanstalk.

15. Seleccione



(actualizar), junto a la lista desplegable Perfil de instancia de EC2.

Esto actualiza la lista desplegable, para que el rol que acaba de crear aparezca en ella.

Flujo de trabajo de Elastic Beanstalk

Para implementar y ejecutar la aplicación de ejemplo en AWS los recursos, Elastic Beanstalk lleva a cabo las siguientes acciones. Tarda aproximadamente cinco minutos en completarse.

1. Crea una aplicación de Elastic Beanstalk denominada. `getting-started-app`
2. Lanza un entorno denominado `GettingStartedApp-env` con los siguientes recursos: AWS
 - Una instancia de Amazon Elastic Compute Cloud (Amazon EC2) (máquina virtual)
 - Un grupo de seguridad de Amazon EC2
 - Un bucket de Amazon Simple Storage Service (Amazon S3)
 - CloudWatch Alarmas Amazon

- ¿Una AWS CloudFormation pila
- Un nombre de dominio

Para obtener más información sobre estos AWS recursos, consulte [the section called “AWS recursos creados para la aplicación de ejemplo”](#).

3. Cree una nueva versión de la aplicación llamada Sample Application (Aplicación de muestra). Este es el archivo de aplicación de ejemplo predeterminado de Elastic Beanstalk.
4. Implementa el código de la aplicación de ejemplo en el entorno GettingStartedApp-env.

Durante el proceso de creación del entorno, la consola hace un seguimiento del progreso y muestra los eventos.

The screenshot displays the AWS Elastic Beanstalk console for an environment named 'Gettingstarted-env'. The environment is in a 'Healthy' state, indicated by a green checkmark and the text 'Ok'. The environment ID is 'e-irkuacn9ny' and the application name is 'GettingStarted'. The platform is 'Node.js 16 running on 64bit Amazon Linux 2/5.6.3'. The domain is 'Gettingstarted-env.eba-w2pdx9as.us-east-1.elasticbeanstalk.com'. The console shows a list of 20 events, all of which are 'INFO' type, detailing the environment's initialization and deployment process.

Environment overview

Health: ✔ Ok

Environment ID: e-irkuacn9ny

Domain: Gettingstarted-env.eba-w2pdx9as.us-east-1.elasticbeanstalk.com

Application name: GettingStarted

Platform

Platform: Node.js 16 running on 64bit Amazon Linux 2/5.6.3

Update

Running version: Sample Application

Events (20)

Time	Type	Details
January 8, 2023 19:40:13 (UTC-5)	INFO	Environment health has transitioned from Pending to Ok. Initialization completed 46 seconds ago and took 2 minutes.
January 8, 2023 19:39:29 (UTC-5)	INFO	Successfully launched environment: Gettingstarted-env
January 8, 2023 19:39:28 (UTC-5)	INFO	Application available at Gettingstarted-env.eba-w2pdx9as.us-east-1.elasticbeanstalk.com.
January 8, 2023 19:39:13 (UTC-5)	INFO	Added instance [i-0b1530c3cabd58083] to your environment.
January 8, 2023 19:38:56 (UTC-5)	INFO	Instance deployment completed successfully.
January 8, 2023 19:38:28 (UTC-5)	INFO	Waiting for EC2 instances to launch. This may take a few minutes.
January 8, 2023 19:37:13 (UTC-5)	INFO	Environment health has transitioned to Pending. Initialization in progress (running for 20 seconds). There are no instances.
January 8, 2023 19:37:11 (UTC-5)	INFO	Created security group named: awseb-e-irkuacn9ny-stack-AWSEBSecurityGroup-1TQD00YHCNM7W
January 8, 2023 19:36:55 (UTC-5)	INFO	Created security group named: sg-0d8a4193f0512fe9a
January 8, 2023 19:36:55 (UTC-5)	INFO	Created target group named: arn:aws:elasticloadbalancing:us-east-1:164656829171:targetgroup/awseb-AWSEB-EURAPI3GVX2H/d33ef00e2dc5b0c8
January 8, 2023 19:36:34 (UTC-5)	INFO	Using elasticbeanstalk-us-east-1-164656829171 as Amazon S3 storage bucket for environment data.
January 8, 2023 19:36:33 (UTC-5)	INFO	createEnvironment is starting.

Cuando se lancen todos los recursos y las instancias EC2 que ejecutan la aplicación superen las comprobaciones de estado, los cambios de estado del entorno tendrán el valor Ok. Ahora puede usar el sitio web de su aplicación web.

AWS recursos creados para la aplicación de ejemplo

Al crear la aplicación de ejemplo, Elastic Beanstalk crea los siguientes recursos: AWS

- EC2 instance (Instancia EC2): máquina virtual de Amazon EC2 configurada para ejecutar aplicaciones web en la plataforma que elija.

Cada plataforma ejecuta un conjunto distinto de software, archivos de configuración y scripts compatibles con una determinada versión de lenguaje, marco y contenedor web (o una combinación de ellos). La mayoría de las plataformas utilizan Apache o nginx como un proxy inverso que procesa el tráfico web delante de la aplicación web, reenvía las solicitudes a esta, administra los recursos estáticos y genera registros de acceso y errores.

- Instance security group (Grupo de seguridad de la instancia): grupo de seguridad de Amazon EC2 configurado para permitir el tráfico entrante en el puerto 80. Este recurso permite que el tráfico HTTP procedente del balanceador de carga llegue a la instancia EC2 en la que se ejecuta la aplicación web. De forma predeterminada, el tráfico no está permitido en otros puertos.
- Bucket de Amazon S3: ubicación de almacenamiento para el código fuente, los registros y otros artefactos que se crean al utilizar Elastic Beanstalk.
- CloudWatch Alarmas de Amazon: dos CloudWatch alarmas que monitorean la carga de las instancias de su entorno y se activan si la carga es demasiado alta o demasiado baja. Cuando se activa una alarma, en respuesta, el grupo de Auto Scaling aumenta o reduce los recursos.
- AWS CloudFormation pila: Elastic AWS CloudFormation Beanstalk se utiliza para lanzar los recursos de su entorno y propagar los cambios de configuración. Los recursos se definen en una plantilla que puede verse en la [consola de AWS CloudFormation](#).
- Nombre de dominio: nombre de dominio que direcciona el tráfico a la aplicación web con el formato *subdominio.región.elasticbeanstalk.com*.

Note

Para aumentar la seguridad de las aplicaciones de Elastic Beanstalk, el dominio elasticbeanstalk.com está registrado en la [lista de sufijos públicos \(PSL\)](#). Para mayor seguridad, se recomienda que utilice cookies con un prefijo `__Host-` en caso de que necesite configurar cookies confidenciales en el nombre de dominio predeterminado de sus aplicaciones de Elastic Beanstalk. Esta práctica le ayudará a defender su dominio contra intentos de falsificación de solicitudes entre sitios (CSRF). Para obtener más información, consulte la página de [configuración de cookies](#) en la red de desarrolladores de Mozilla.

Paso 2: Explorar el entorno

Para ver información general del entorno de su aplicación de Elastic Beanstalk, utilice la página Environment overview (Información general del entorno) en la consola de Elastic Beanstalk.

Para ver la información general del entorno

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

La parte superior de la página Environment overview (Información general del entorno) muestra información de nivel superior sobre el entorno. Esto incluye su nombre, su dominio URL, su estado actual, el nombre de la versión actual de la aplicación implementada y la versión de la plataforma en la que se ejecuta la aplicación. Debajo del panel de información general, puede ver los eventos más recientes del entorno en la pestaña Events (Eventos). Las otras pestañas muestran otros detalles principales del entorno.

Para obtener más información acerca de las capas del entorno, las plataformas, las versiones de la aplicación y otros conceptos de Elastic Beanstalk, consulte [Conceptos](#).

The screenshot displays the AWS Elastic Beanstalk console interface. The main content area is titled 'Gettingstarted-env' and includes an 'Environment overview' section with the following details:

- Health: ✔ Ok
- Environment ID: e-irkuacr9ny
- Domain: Gettingstarted-env.eba-w2pdx9as.us-east-1.elasticbeanstalk.com
- Application name: GettingStarted

To the right, the 'Platform' section shows:

- Platform: Node.js 16 running on 64bit Amazon Linux 2/5.7.0
- Running version: Sample Application

Below the overview, there are tabs for 'Events', 'Health', 'Logs', 'Monitoring', 'Alarms', 'Managed updates', and 'Tags'. The 'Events' tab is active, showing a list of events:

Time	Type	Details
March 28, 2023 20:01:06 (UTC-4)	INFO	Environment health has transitioned from Info to Ok. Configuration update completed 47 seconds ago and took 14 minutes.
March 28, 2023 20:00:06 (UTC-4)	INFO	Environment update completed successfully.

Mientras Elastic Beanstalk AWS crea los recursos y lanza la aplicación, el entorno se encuentra en un estado. Pendíng Continúa se añaden mensajes de estado acerca de eventos de lanzamiento a la información general.

El Domain (Dominio) del entorno, o URL, se encuentra en la parte superior de la página Environment overview (Información general del entorno), debajo de Health (Estado) del entorno. Se trata de la dirección URL de la aplicación web que el entorno ejecuta. Elija esta dirección URL para ir a la página Congratulations (Enhorabuena) de la aplicación. En el panel de navegación de la izquierda, se muestra el enlace Go to environment (Ir al entorno) que abre la misma página de la aplicación.

En el panel de navegación de la izquierda también aparece Configuration (Configuración), que muestra la página de información general sobre la configuración. Esta página muestra un resumen de los valores de las opciones de configuración del entorno, agrupados por categoría.

Las pestañas de la parte inferior de la página contienen información más detallada sobre el entorno y proporcionan acceso a otras características adicionales:

- **Events (Eventos):** muestra información o mensajes de error del servicio de Elastic Beanstalk y de otros servicios cuyos recursos usa este entorno.
- **Health (Estado):** muestra el estado y otros detalles de las instancias de Amazon EC2 que ejecutan la aplicación.
- **Logs (Registros):** recupere y descargue registros de Amazon EC2 en su entorno. Puede recuperar registros completos o actividades recientes. Los registros recuperados están disponibles durante 15 minutos.
- **Monitoring (Monitorización):** muestra las estadísticas del entorno, como la latencia media y el uso de CPU.
- **Alarms (Alarmas):** muestra las alarmas que se configuraron para las métricas del entorno. En esta página, puede agregar, modificar o eliminar alarmas.
- **Managed updates (Actualizaciones administradas):** muestra información sobre las actualizaciones administradas de la plataforma, próximas y completadas, y la sustitución de instancias.
- **Tags (Etiquetas):** muestra etiquetas de entorno y le permite administrarlas. Las etiquetas son pares clave-valor que se aplican a su entorno.

Note

El panel de navegación del lado izquierdo de la consola contiene enlaces con el mismo nombre que las pestañas. Al seleccionar cualquiera de estos enlaces, se mostrará el contenido de la pestaña correspondiente.

Paso 3: Implementar una nueva versión de la aplicación

Periódicamente, es posible que tenga que implementar una nueva versión de la aplicación. Puede implementar una nueva versión en cualquier momento, siempre que en el entorno no haya ninguna otra operación de actualización en curso.

La versión de la aplicación con la que inició este tutorial se llama Sample Application (Aplicación de ejemplo).

Para actualizar la versión de la aplicación

1. Descargue la aplicación de ejemplo que coincida con la plataforma de su entorno. Utilice una de las siguientes aplicaciones.
 - Docker: [docker.zip](#)
 - Docker multicontenedor: [2.zip docker-multicontainer-v](#)
 - Docker preconfigurado (Glassfish): [1.zip docker-glassfish-v](#)
 - Go: [go.zip](#)
 - Corretto: [corretto.zip](#)
 - Tomcat: [tomcat.zip](#)
 - .NET Core en Linux: [.zip dotnet-core-linux](#)
 - .NET Core: [dotnet-asp-windows .zip](#)
 - Node.js: [nodejs.zip](#)
 - PHP: [php.zip](#)
 - Python: [python.zip](#)
 - Ruby: [ruby.zip](#)
2. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS

3. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

4. En la página de información general del entorno, elija Upload and deploy (Cargar e implementar).
5. Elija Choose File (Elegir archivo) y, a continuación, cargue el paquete de código fuente de la aplicación de ejemplo que ha descargado.

Upload and deploy

To deploy a previous version, go to the [Application Versions page](#).

Upload application:

File name : **java-tomcat-v3.zip** ✓

Version label:

► **Deployment Preferences**

The application version will be deployed using the **All at once** policy.

Current number of instances: **1**

La consola rellena automáticamente la Version label (Etiqueta de versión) con una nueva etiqueta única. Si escribe su propia etiqueta de versión, asegúrese de que es única.

6. Elija Implementar.

Mientras Elastic Beanstalk implementa el archivo en sus instancias de Amazon EC2, puede ver el estado de la implementación en la información general del entorno. Mientras se actualiza la versión de la aplicación, el estado de Environment Health (Estado del entorno) aparece atenuado. Una vez completada la implementación, Elastic Beanstalk realiza una comprobación de estado de la aplicación. Cuando la aplicación responde a la comprobación de estado, se considera que está en buen estado y este cambia a verde. En la información general del entorno se muestra la nueva Running Version (Versión de ejecución): el nombre que proporcionó como Version label (Etiqueta de la versión).

Elastic Beanstalk también carga la nueva versión de la aplicación y la añade a la tabla de versiones de la aplicación. Para ver la tabla, seleccione Versiones de aplicaciones getting-started-app en el panel de navegación.

Paso 4: Configurar su entorno

Puede configurar el entorno para adaptarlo mejor a su aplicación. Por ejemplo, si tiene una aplicación que requiere un uso intensivo de recursos informáticos, puede cambiar el tipo de instancia de Amazon Elastic Compute Cloud (Amazon EC2) que ejecuta la aplicación. Para aplicar cambios de configuración, Elastic Beanstalk realiza una actualización del entorno.

Hay cambios de configuración que puede realizar fácilmente y que se aplican con rapidez. Algunos cambios requieren eliminar y volver a crear los AWS recursos, lo que puede tardar varios minutos. Al cambiar las opciones de configuración, Elastic Beanstalk le advierte sobre el posible tiempo de inactividad de las aplicaciones.

Realizar un cambio de configuración

En este ejemplo de un cambio de configuración, edite la configuración de la capacidad del entorno. Puede configurar un entorno escalable con balanceo de carga que tenga entre dos y cuatro instancias de Amazon EC2 en su grupo de Auto Scaling y, a continuación, comprobar que el cambio se ha producido. Elastic Beanstalk crea una instancia adicional de Amazon EC2, que se añade a la única instancia que creó inicialmente. A continuación, Elastic Beanstalk asocia ambas instancias con el balanceador de carga del entorno. Como resultado, la capacidad de respuesta de su aplicación mejora y su disponibilidad aumenta.

Para cambiar la capacidad del entorno

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS

2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

 Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Instance traffic and scaling (Escalado y tráfico de instancias), elija Edit (Editar).
5. Contraiga la sección Instances (Instancias) para poder ver con más facilidad la sección Capacity (Capacidad). En la sección Auto Scaling group (Grupo de escalado automático), cambie el Environment type (Tipo de entorno) a Load balanced (Con equilibrador de carga).
6. En la fila Instances (Instancias), cambie Max (Máx.) a **4** y, a continuación, cambie Min (Mín.) a **2**.
7. Para guardar los cambios, elija Aplicar en la parte inferior de la página.
8. Una advertencia le indica que esta actualización reemplaza todas sus instancias actuales. Elija Confirmar.
9. Aparecerá la página Environment overview (Información general del entorno), que muestra la pestaña Events (Eventos).

El entorno puede tardar unos minutos en actualizarse. Para averiguar si se ha completado, busque el evento Successfully deployed new configuration to environment en la lista de eventos. Esto confirma que el número mínimo de instancias de Auto Scaling se ha establecido en 2. Elastic Beanstalk lanza automáticamente la segunda instancia.

Verificar el cambio de configuración

Cuando se haya completado la actualización del entorno y este esté listo, compruebe su cambio.

Para comprobar la capacidad superior

1. Seleccione Health (Estado) ya sea desde la lista de pestañas o desde el panel de navegación izquierdo.
2. Consulte la sección Enhanced instance health (Estado mejorado de las instancias).

Verá que aparecen dos instancias de Amazon EC2. Se ha aumentado la capacidad del entorno a dos instancias.

Instance ID	Status	Running time	Deployment ID	Requests/sec	2xx Responses
i-04a22cd25ba2f7c4e	Ok	January 10, 2023 01:20:26 (UTC-5)	1	2	2
i-0b1530c3cabd58083	Ok	January 8, 2023 19:37:28 (UTC-5)	1	1	1

Paso 5: Eliminar

¡Enhorabuena! Implementó correctamente una aplicación de muestra en la AWS nube, cargó una nueva versión y modificó su configuración para agregar una segunda instancia de Auto Scaling. Para asegurarse de que no se le cobra por servicios que no utiliza, elimine todas las versiones de la aplicación y finalice el entorno. Esto también elimina los AWS recursos que el entorno creó para usted.

Para eliminar la aplicación y todos los recursos asociados

1. Borre todas las versiones de la aplicación.
 - a. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
 - b. En el panel de navegación, elija Aplicaciones y, a continuación, elija. getting-started-app
 - c. En el panel de navegación, busque el nombre de la aplicación y elija Application versions (Versiones de la aplicación).
 - d. En la página Application Versions (Versiones de la aplicación), seleccione todas las versiones de la aplicación que desee eliminar.
 - e. Elija Acciones y, a continuación, elija Eliminar.
 - f. Active Delete versions from Amazon S3 (Eliminar versiones de Amazon S3).
 - g. Elija Delete (Eliminar) y a continuación haga clic en Done (Listo).
2. Finalice el entorno.

- a. En el panel de navegación, elija `getting-started-app`, a continuación, elija `GettingStartedApp-env` en la lista de entornos.
 - b. Elija `Actions` (Acciones) y, a continuación, `Terminate Environment` (Terminar entorno).
 - c. Confirme que desea terminar `GettingStartedApp-env` escribiendo el nombre del entorno y, a continuación, elija `Terminate`.
3. Elimine la aplicación `getting-started-app` .
 - a. En el panel de navegación, elija `getting-started-app`.
 - b. Elija `Acciones` y, a continuación, elija `Eliminar aplicación`.
 - c. Confirme que desea eliminar `getting-started-app` escribiendo el nombre de la aplicación y, a continuación, elija `Eliminar`.

Siguientes pasos

Ahora que sabe cómo crear una aplicación y un entorno de Elastic Beanstalk, le recomendamos que lea [Conceptos](#). Este tema ofrece información acerca de los componentes y la arquitectura de Elastic Beanstalk, y describe consideraciones de diseño importantes para la aplicación de Elastic Beanstalk.

Además de la consola de Elastic Beanstalk, puede utilizar las siguientes herramientas para crear y administrar entornos de Elastic Beanstalk.

CLI DE EB

La CLI de EB es una herramienta de línea de comando para crear y administrar entornos. Para obtener más información, consulte [Uso de la interfaz de línea de comandos de Elastic Beanstalk \(CLI de EB\)](#).

AWS SDK for Java

AWS SDK for Java Proporciona una API de Java que puede utilizar para crear aplicaciones que utilizan servicios de AWS infraestructura. Con él AWS SDK for Java, puede empezar en cuestión de minutos con un único paquete descargable que incluye la biblioteca AWS Java, ejemplos de código y documentación.

AWS SDK for Java Requiere el kit de desarrollo J2SE 5.0 o una versión posterior. Puede descargar el software de Java más reciente de <http://developers.sun.com/downloads/>. El SDK también requiere

Apache Commons (Codec, HttpClient y Logging), así como paquetes de terceros de Saxon-HE, que se incluyen el directorio de herramientas de terceros del SDK.

Para obtener más información, consulte el [SDK para Java de AWS](#).

AWS Toolkit for Eclipse

AWS Toolkit for Eclipse Se trata de un complemento de código abierto para el IDE Java de Eclipse. Puede usarla para crear proyectos web de AWS Java preconfigurados con Elastic Beanstalk y AWS SDK for Java, a continuación, implementar las aplicaciones web en Elastic Beanstalk. El complemento de Elastic Beanstalk se basa en Eclipse Web Tools Platform (WTP). El conjunto de herramientas cuenta con una plantilla de aplicaciones web de ejemplo, "Travel Log", donde se ilustra el uso de Amazon S3 y Amazon SNS.

Para asegurarse de que cuenta con todas las dependencias de WTP, le recomendamos que empiece con la distribución de Java EE de Eclipse. Puede descargarlo desde <http://eclipse.org/downloads/>.

Para obtener más información sobre el uso del complemento de Elastic Beanstalk para Eclipse, consulte [AWS Toolkit for Eclipse](#). Para empezar a crear una aplicación de Elastic Beanstalk con Eclipse, consulte [Creación e implementación de aplicaciones Java en Elastic Beanstalk](#).

AWS SDK for .NET

AWS SDK for .NET Esto le permite crear aplicaciones que utilizan servicios de infraestructura. AWS Con él AWS SDK for .NET, puede empezar en cuestión de minutos con un único paquete descargable que incluye la AWS biblioteca.NET, ejemplos de código y documentación.

Para obtener más información, consulte el [SDK para .NET de AWS](#). Para ver las versiones de .NET Framework y Visual Studio admitidas, consulte la [Guía para desarrolladores de AWS SDK for .NET](#).

AWS Toolkit for Visual Studio

Con el AWS Toolkit for Visual Studio complemento, puede implementar una aplicación.NET existente en Elastic Beanstalk. También puede crear proyectos con las AWS plantillas preconfiguradas con. AWS SDK for .NET

Para obtener información sobre los requisitos previos y la instalación, consulte [Toolkit for Visual Studio de AWS](#). Para empezar a crear una aplicación de Elastic Beanstalk con Visual Studio, consulte [Creación e implementación de aplicaciones.NET Windows en Elastic Beanstalk](#).

AWS SDK para JavaScript en Node.js

El AWS SDK para JavaScript Node.js le permite crear aplicaciones sobre los servicios de AWS infraestructura. Con el AWS SDK de JavaScript Node.js, puede empezar en cuestión de minutos con un único paquete descargable que incluye la biblioteca AWS Node.js, ejemplos de código y documentación.

Para obtener más información, consulta el [AWS SDK JavaScript de Node.js](#).

AWS SDK for PHP

AWS SDK for PHP Le permite crear aplicaciones sobre los servicios de AWS infraestructura. Con él AWS SDK for PHP, puede empezar en cuestión de minutos con un único paquete descargable que incluye la biblioteca AWS PHP, ejemplos de código y documentación.

AWS SDK for PHP Requiere PHP 5.2 o posterior. Para obtener más información sobre la descarga, consulte <http://php.net/>.

Para obtener más información, consulte el [SDK de AWS para PHP](#).

AWS SDK for Python (Boto)

Con él AWS SDK for Python (Boto), puede empezar en cuestión de minutos con un único paquete descargable que incluye la biblioteca de AWS Python, ejemplos de código y documentación. Puede crear aplicaciones de Python utilizando las API y evitar la complejidad que supone escribir directamente el código en las interfaces de servicios web.

La all-in-one biblioteca proporciona API de Python fáciles de usar para desarrolladores que ocultan muchas de las tareas de nivel inferior asociadas con la programación para la AWS nube, como la autenticación, los reintentos de solicitudes y la gestión de errores. El SDK ofrece ejemplos prácticos en Python sobre el uso de las bibliotecas para crear aplicaciones.

Para obtener más información sobre Boto, así como fragmentos de código de ejemplo, documentación, herramientas y otros recursos, consulte [Centro de desarrolladores de Python](#).

AWS SDK for Ruby

Puedes empezar en cuestión de minutos con un único paquete descargable que incluye la biblioteca de AWS Ruby, ejemplos de código y documentación. Puede crear aplicaciones de Ruby utilizando las API y evitar la complejidad que supone escribir directamente el código en las interfaces de servicios web.

La all-in-one biblioteca proporciona API de Ruby fáciles de usar para los desarrolladores que ocultan muchas de las tareas de nivel inferior asociadas a la programación para la AWS nube, como la autenticación, los reintentos de solicitudes y la gestión de errores. El SDK ofrece ejemplos prácticos en Ruby sobre el uso de las bibliotecas para crear aplicaciones.

Para obtener información sobre el SDK, así como fragmentos de código de ejemplo, documentación, herramientas y otros recursos, consulte [Centro de desarrolladores de Ruby](#).

Conceptos de Elastic Beanstalk

AWS Elastic Beanstalk lo habilita para administrar todos los recursos que ejecutan su aplicación como entornos. Estos son algunos conceptos clave de Elastic Beanstalk.

Aplicación

Una aplicación Elastic Beanstalk es una colección lógica de componentes de Elastic Beanstalk, que incluye entornos, versiones, y configuraciones de entorno. En Elastic Beanstalk, una aplicación es conceptualmente similar a una carpeta.

Versión de la aplicación

En Elastic Beanstalk, una versión de la aplicación es una iteración específica y designada de un código implementable de una aplicación web. Las versiones de la aplicación apuntan a un objeto de Amazon Simple Storage Service (Amazon S3) que contiene el código implementable, como un archivo de Java WAR. Las versiones de la aplicación forman parte de una aplicación. Las aplicaciones pueden tener muchas versiones y cada versión de la aplicación es única. En un entorno en ejecución, puede implementar cualquier versión que ya se haya cargado en la aplicación o puede cargar e instalar inmediatamente una nueva versión de la aplicación. También puede cargar varias versiones de la aplicación para ver las diferencias entre una y otra versión de la aplicación web.

Entorno

Un entorno es un conjunto de recursos de AWS que ejecutan una versión de la aplicación. Cada entorno ejecuta una versión de la aplicación al mismo tiempo; sin embargo, puede haber varios entornos que ejecuten simultáneamente la misma versión de la aplicación o versiones de la aplicación diferentes. Cuando se crea un entorno, Elastic Beanstalk aprovisiona los recursos necesarios para ejecutar la versión de la aplicación especificada.

Capa del entorno

Cuando inicie un entorno de Elastic Beanstalk, primero elija un nivel de entorno. La capa del entorno designa el tipo de aplicación que ejecuta el entorno y determina qué recursos aprovisiona Elastic Beanstalk para admitirla. Una aplicación que atiende solicitudes HTTP se ejecuta en una [capa del](#)

[entorno de servidor web](#). Un entorno backend que extrae tareas de Amazon Simple Queue Service (Amazon SQS) se ejecuta en un [nivel de entorno de trabajo](#).

Configuración del entorno

Una configuración del entorno identifica una serie de parámetros y opciones de configuración que determinan cómo se comporta un entorno y sus recursos asociados. Cuando se actualiza una opción de configuración del entorno, Elastic Beanstalk aplica automáticamente los cambios a los recursos existentes o elimina e implementa nuevos recursos (en función del tipo de cambio).

Configuración guardada

Una configuración guardada es una plantilla que puede utilizar como punto de partida para crear configuraciones de entorno únicas. Puede crear y modificar configuraciones guardadas y aplicarlas a los entornos mediante la consola de Elastic Beanstalk, la CLI de EB, la AWS CLI o la API. La API y la AWS CLI remiten a las configuraciones guardadas como plantillas de configuración.

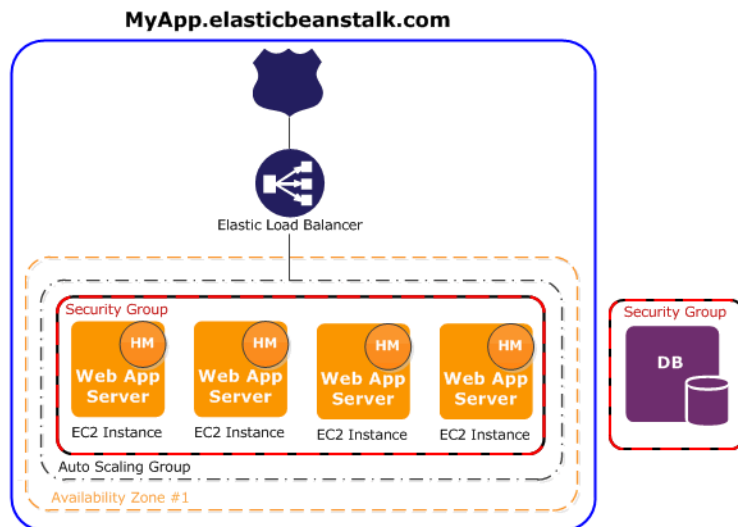
Plataforma

Una plataforma es una combinación de un sistema operativo, el tiempo de ejecución de un lenguaje de programación, un servidor web, un servidor de aplicaciones y componentes de Elastic Beanstalk. Usted diseña y orienta su aplicación web a una plataforma. Elastic Beanstalk proporciona una variedad de plataformas en las que puede crear sus aplicaciones.

Para obtener más información, consulte [Plataformas Elastic Beanstalk](#).

Entornos del servidor web

En el diagrama siguiente se muestra un ejemplo de arquitectura de Elastic Beanstalk para un nivel de entorno de servidor web y se muestra cómo interactúan los componentes de ese tipo de entorno.



El entorno es la piedra angular de la aplicación. En el diagrama, el entorno se muestra en la línea continua de nivel superior. Al crear un entorno, Elastic Beanstalk aprovisiona los recursos necesarios para ejecutar la aplicación. Entre los recursos que AWS crea para un entorno se incluye un elastic load balancer (ELB en el diagrama), un grupo de Auto Scaling y una o varias instancias de Amazon Elastic Compute Cloud (Amazon EC2).

Todos los entornos tienen un CNAME (URL) que apunta a un balanceador de carga. El entorno tiene una URL; por ejemplo, `myapp.us-west-2.elasticbeanstalk.com`. Esta URL tiene alias en [Amazon Route 53](#) a una URL de Elastic Load Balancing, algo así como, `abcdef-123456.us-west-2.elb.amazonaws.com` mediante un registro CNAME. [Amazon Route 53](#) es un servicio web de sistema de nombres de dominio (DNS) escalable y de alta disponibilidad. Se encarga de direccionar el tráfico a la infraestructura de forma segura y fiable. El nombre de dominio registrado con el proveedor de DNS reenviará las solicitudes a CNAME.

El balanceador de carga se encuentra delante de las instancias de Amazon EC2, que forman parte de un grupo de Auto Scaling. Amazon EC2 Auto Scaling inicia automáticamente más instancias de Amazon EC2 para acomodar el aumento de la carga que registra la aplicación. Si la carga de la aplicación se reduce, Amazon EC2 Auto Scaling detiene las instancias, aunque siempre deja en ejecución al menos una.

El conjunto de software que se ejecuta en las instancias de Amazon EC2 depende del tipo de contenedor. Un tipo de contenedor define la topología de la infraestructura y el conjunto de software que se va a utilizar en ese entorno. Por ejemplo, un entorno de Elastic Beanstalk con un contenedor de Apache Tomcat utilizará el sistema operativo Amazon Linux, un servidor web Apache y software de Apache Tomcat. Para ver una lista de los tipos de contenedores admitidos, consulte [Plataformas](#)

[compatibles con Elastic Beanstalk](#). Cada instancia de servidor de Amazon EC2 que ejecuta la aplicación utiliza uno de estos tipos de contenedores. Además, en cada instancia de servidor de Amazon EC2, se ejecuta un componente de software llamado administrador del host (HM). El administrador de host es responsable de lo siguiente:

- Implementar la aplicación
- Agregar eventos y métricas para recuperar información a través de la consola, la API o la línea de comando
- Generar eventos de nivel de instancia
- Monitorizar los archivos log de la aplicación en busca de errores críticos
- Monitorizar el servidor de aplicaciones
- Instalar parches en los componentes de la instancia
- Rotar los archivos log de la aplicación y publicarlos en Amazon S3

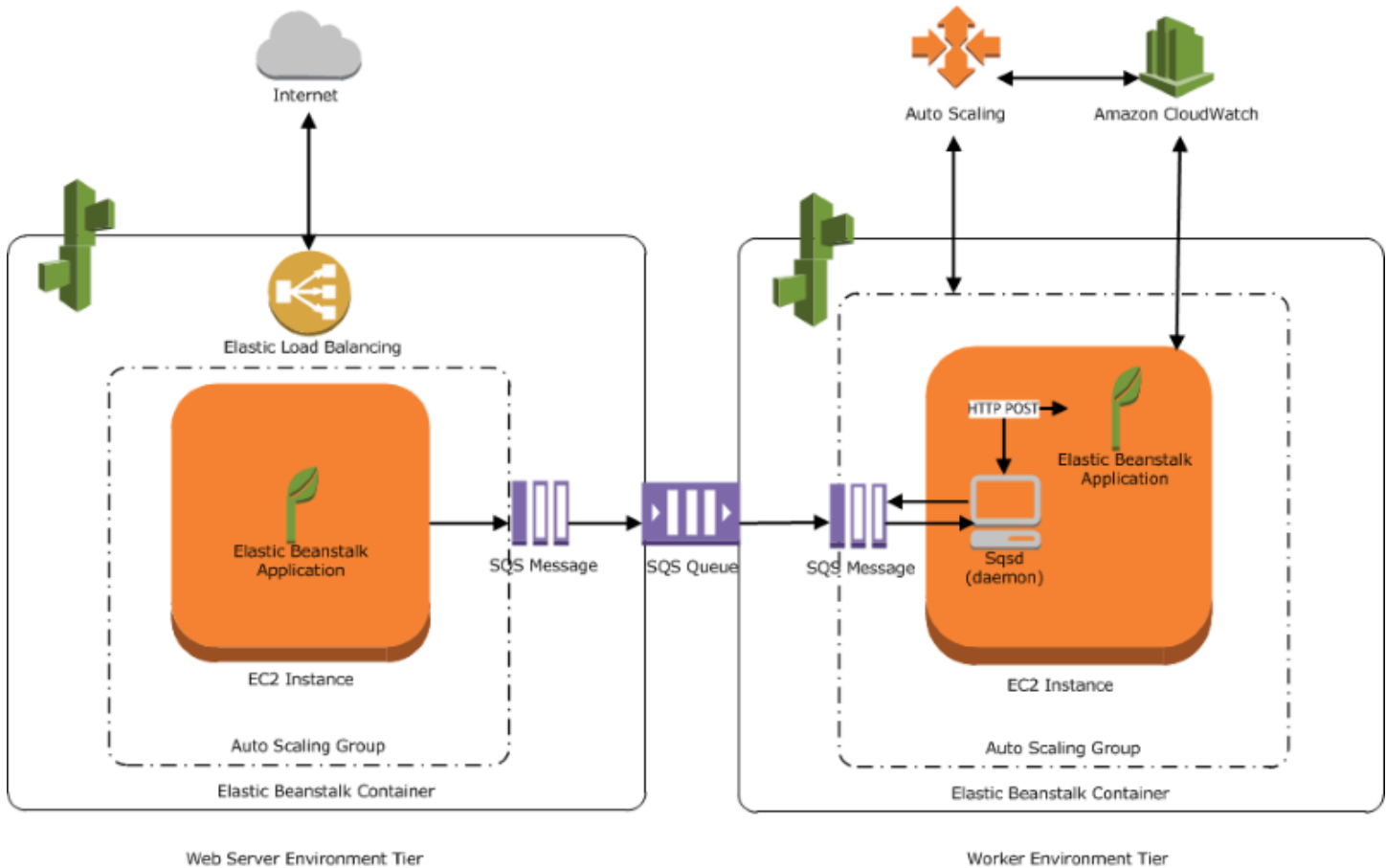
El administrador del host registra métricas, errores, eventos y el estado de las instancias del servidor, que están disponibles a través de la consola de Elastic Beanstalk, las API y las CLI.

Las instancias de Amazon EC2 que se muestran en el diagrama forman parte de un grupo de seguridad. Los grupos de seguridad definen las reglas del firewall de las instancias. De forma predeterminada, Elastic Beanstalk define un grupo de seguridad, lo que permite a todos los usuarios conectarse a través del puerto 80 (HTTP). Puede definir varios grupos de seguridad. Por ejemplo, puede definir un grupo de seguridad para el servidor de base de datos. Si desea obtener más información sobre los grupos de seguridad de Amazon EC2 y cómo configurarlos para la aplicación de Elastic Beanstalk, consulte [Grupos de seguridad](#).

Entornos de trabajo

Entre los recursos de AWS que se crean para un nivel de entorno de trabajo se incluye un grupo de Auto Scaling, una o varias instancias de Amazon EC2 y un rol de IAM. En el nivel de entorno de trabajo, Elastic Beanstalk también crea y aprovisiona una cola de Amazon SQS si aún no hay ninguna. Cuando se lanza un entorno de trabajo, Elastic Beanstalk instala los archivos auxiliares necesarios para el lenguaje de programación elegido y un daemon en cada instancia EC2 del grupo de Auto Scaling. El daemon lee mensajes de una cola de Amazon SQS. El demonio envía los datos de cada mensaje que lee a la aplicación web que se ejecuta en el entorno de trabajo para su procesamiento. Si hay varias instancias en el entorno de trabajo, cada instancia tendrá su propio daemon, pero todas consultarán la misma cola de Amazon SQS.

En el siguiente diagrama se muestran los diferentes componentes y sus interacciones entre entornos y servicios de AWS.



Amazon CloudWatch se utiliza para las alarmas y para la monitorización de estado. Para obtener más información, consulte [Informes de estado básicos](#).

Para obtener más información acerca de cómo funciona el nivel de entorno de trabajo, consulte [Entornos de trabajo de Elastic Beanstalk](#).

Consideraciones sobre el diseño

Dado que las aplicaciones que se implementan con AWS Elastic Beanstalk se ejecutan en recursos en Nube de AWS, debe tener en cuenta varios factores de configuración para optimizar sus aplicaciones: escalabilidad, seguridad, almacenamiento persistente, tolerancia a errores, entrega de contenido, actualizaciones de software/parches y conectividad. Cada uno de ellos se trata por separado en este tema. Para obtener una lista completa de documentos técnicos de AWS en los que se tratan temas como arquitectura, seguridad y economía, consulte los [documentos técnicos sobre computación en la nube de AWS](#).

Escalabilidad

Al operar en un entorno de hardware físico, a diferencia de lo que ocurre con un entorno en la nube, se puede enfocar la escalabilidad de una de estas dos maneras. Se puede escalar mediante el escalado vertical o el escalado horizontal. El enfoque de escalado vertical requiere que se invierta en un hardware potente, que pueda soportar las crecientes demandas de su empresa. El enfoque de escalado horizontal requiere que se siga un modelo de inversión distribuido. Por lo tanto, las adquisiciones de hardware y aplicaciones se podrán orientar mejor, los conjuntos de datos estarán federados y el diseño estará orientado a los servicios. El enfoque de escalado vertical puede ser costoso y existe el riesgo de que la demanda sea mayor que la capacidad. En este sentido, el enfoque de escalado horizontal suele ser más eficaz. Sin embargo, al usarlo, se debe ser capaz de predecir la demanda a intervalos regulares e implementar la infraestructura en fragmentos para satisfacer esa demanda. Por lo tanto, a menudo este enfoque puede dar lugar a que no se utilice la capacidad y podría requerir una supervisión pormenorizada.

Al migrar a la nube, puede hacer que su infraestructura se ajuste mejor a la demanda aprovechando la elasticidad de la nube. La elasticidad ayuda a agilizar la adquisición y liberación de recursos. Con él, su infraestructura se puede reducir y escalar horizontalmente según fluctúe la demanda. Para utilizarlo, configure los ajustes de Auto Scaling para aumentar o disminuir el escalado en función de las métricas de los recursos del entorno. Por ejemplo, puede establecer métricas como la utilización del servidor o la E/S de red. Puede utilizar Auto Scaling para que la capacidad de computación se active automáticamente cuando el uso aumente y para que se desactive cuando el uso disminuya. Puede publicar las métricas del sistema (por ejemplo, CPU, memoria, E/S de disco y E/S de red) en Amazon CloudWatch. A continuación, puede utilizar CloudWatch para configurar alarmas que activen acciones de Auto Scaling o envíen notificaciones basadas en estas métricas. Para obtener instrucciones sobre cómo configurar Auto Scaling, consulte [Grupo de Auto Scaling para su entorno de Elastic Beanstalk](#).

También recomendamos que, en la medida de lo posible, diseñe todas las aplicaciones de Elastic Beanstalk sin estado, utilizando componentes tolerantes a errores acoplados de manera laxa y que puedan escalarse horizontalmente en función de las necesidades. Para obtener más información sobre el diseño de arquitecturas de aplicaciones escalables para AWS, consulte [AWS Well-Architected Framework](#).

Seguridad

La seguridad en AWS es una [responsabilidad compartida](#). Amazon Web Services protege los recursos físicos del entorno y se asegura de que la nube sea un lugar seguro para que puedan

ejecutarse las aplicaciones. Usted es responsable de la seguridad de los datos de entrada y salida del entorno de Elastic Beanstalk y de la seguridad de la aplicación.

Configure SSL para proteger el flujo de información entre la aplicación y los clientes. Para configurar SSL, necesita un certificado gratuito de AWS Certificate Manager (ACM). Si ya tiene un certificado expedido por una entidad de certificación (CA) externa, puede utilizar ACM para importar ese certificado. De lo contrario, puede importarlo mediante la AWS CLI.

Si ACM no está [disponible en su Región de AWS](#), puede adquirir un certificado de una CA externa, como VeriSign o Entrust. A continuación, utilice la AWS Command Line Interface (AWS CLI) para cargar un certificado de terceros o autofirmado y una clave privada en AWS Identity and Access Management (IAM). La clave pública del certificado autentica el servidor en el navegador. También sirve de base para crear la clave de sesión compartida que cifra los datos en ambas direcciones. Para obtener instrucciones sobre cómo crear, cargar y asignar un certificado SSL para el entorno, consulte [Configuración de HTTPS para su entorno de Elastic Beanstalk](#).

Cuando configura un certificado SSL para el entorno, los datos se cifran entre el cliente y el balanceador de carga de Elastic Load Balancing para el entorno. De forma predeterminada, el cifrado termina en el balanceador de carga, por lo que el tráfico entre el balanceador de carga y las instancias Amazon EC2 permanece sin cifrar.

Almacenamiento persistente

Las aplicaciones de Elastic Beanstalk se ejecutan en instancias de Amazon EC2 que no cuentan con un almacenamiento local persistente. Cuando las instancias de Amazon EC2 terminan, el sistema de archivos local no se guarda. Las nuevas instancias de Amazon EC2 comienzan con un sistema de archivos predeterminado. Le recomendamos que configure la aplicación para almacenar datos en una fuente de datos persistente. AWS ofrece una serie de servicios de almacenamiento persistente que puede utilizar con su aplicación. En la siguiente tabla se enumeran.

Storage service (Servicio de almacenamiento)	Documentación de servicio	Integración de Elastic Beanstalk
Amazon S3	Documentación sobre Amazon Simple Storage Service	Usar Elastic Beanstalk con Amazon S3
Amazon Elastic File System	Documentación de Amazon Elastic File System	Uso de Elastic Beanstalk con Amazon Elastic File System

Storage service (Servicio de almacenamiento)	Documentación de servicio	Integración de Elastic Beanstalk
Amazon Elastic Block Store (EBS)	Amazon Elastic Block Store (EBS) Feature Guide: Elastic Block Store	
Amazon DynamoDB	Documentación sobre Amazon DynamoDB	Uso de Elastic Beanstalk con Amazon DynamoDB
Amazon Relational Database Service (RDS)	Documentación sobre Amazon Relational Database Service	Uso de Elastic Beanstalk con Amazon RDS

Note

Elastic Beanstalk crea un usuario webapp para que lo configure como propietario de los directorios de aplicaciones en las instancias de EC2. En el caso de las versiones de la plataforma Amazon Linux 2 que se publiquen a partir del [3 de febrero de 2022](#), Elastic Beanstalk asigna al usuario webapp un valor uid (id de usuario) y gid (id de grupo) de 900 para los nuevos entornos. Hace lo mismo con los entornos existentes tras una actualización de la versión de la plataforma. Este enfoque mantiene un permiso de acceso consistente para el usuario webapp al almacenamiento permanente del sistema de archivos.

En la improbable situación de que otro usuario o proceso ya esté utilizando 900, el sistema operativo establece de forma predeterminada el uid y gid del usuario webapp en otro valor. Ejecute el comando `id webapp` de Linux en las instancias de EC2 para verificar los valores uid y gid asignados al usuario webapp.

Tolerancia a errores

Por lo general, conviene adoptar una actitud pesimista al diseñar la arquitectura de la nube. Aproveche la elasticidad que ofrece. Planee siempre el diseño y la implementación como si hubiera que recuperarse automáticamente de un error. Utilice varias zonas de disponibilidad para las instancias de Amazon EC2 y Amazon RDS. El concepto de las zonas de disponibilidad es similar al de los centros de datos lógicos. Utilice Amazon CloudWatch para obtener más información sobre

el estado de la aplicación de Elastic Beanstalk y poder emprender las acciones adecuadas en caso de que se produzca un problema en el hardware o disminuya el desempeño. Establezca la configuración de Auto Scaling para mantener la flota de instancias de Amazon EC2 en un tamaño fijo, de modo que las instancias de Amazon EC2 en mal estado se reemplacen por otras nuevas. Si utiliza Amazon RDS, defina el período de retención de las copias de seguridad para que Amazon RDS pueda realizar copias de seguridad automatizadas.

Entrega de contenido

Cuando los usuarios se conectan al sitio web, las solicitudes pueden direccionarse a través de una serie de redes diferentes. Como resultado, los usuarios podrían experimentar un rendimiento deficiente debido a una elevada latencia. Amazon CloudFront puede ayudar a mejorar los problemas de latencia distribuyendo el contenido web, como imágenes y vídeos, por una red de ubicaciones de borde repartidas por todo el mundo. Las solicitudes de los usuarios se dirigen a la ubicación de borde más cercana para que el contenido se entregue con el mejor rendimiento posible. CloudFront funciona perfectamente con Amazon S3, que almacena de forma duradera las versiones originales y definitivas de sus archivos. Para obtener más información sobre Amazon CloudFront, consulte la [Guía para desarrolladores de Amazon CloudFront](#).

Actualizaciones de software y parches

AWS Elastic Beanstalk publica de forma periódica [actualizaciones de las plataformas](#) para proporcionar correcciones, actualizaciones de software y nuevas características. Elastic Beanstalk ofrece varias opciones para gestionar las actualizaciones de la plataforma. Con las [actualizaciones de plataforma administradas](#), el entorno se actualiza automáticamente a la última versión de una plataforma durante un periodo de mantenimiento programado mientras la aplicación sigue en servicio. Para los entornos que se crearon el 25 de noviembre de 2019 o con posterioridad a través de la consola de Elastic Beanstalk, de forma predeterminada y siempre que sea posible, las actualizaciones administradas están habilitadas. También puede iniciar manualmente las actualizaciones utilizando la consola de Elastic Beanstalk o la CLI de EB.

Conectividad

Elastic Beanstalk necesita conectarse a las instancias del entorno para poder completar las implementaciones. Cuando se implementa una aplicación de Elastic Beanstalk dentro de una Amazon VPC, la configuración necesaria para habilitar la conectividad depende del tipo de entorno de Amazon VPC que se ha creado:

- En el caso de los entornos de una sola instancia, no se requiere ninguna configuración adicional. Esto se debe a que, con estos entornos, Elastic Beanstalk asigna a cada instancia de Amazon EC2 una dirección IP elástica pública que permite a la instancia comunicarse directamente con Internet.
- En el caso de los entornos escalables con balanceo de carga de una Amazon VPC donde hay subredes públicas y privadas, debe hacer lo siguiente:
 - Cree un balanceador de carga en la subred pública para direccionar el tráfico entrante procedente de Internet a las instancias de Amazon EC2.
 - Cree un dispositivo de traducción de direcciones de red (NAT) para direccionar el tráfico saliente procedente de las instancias de Amazon EC2 en subredes privadas a Internet.
 - Crear reglas de direccionamiento entrantes y salientes para las instancias de Amazon EC2 en la subred privada.
 - Si se utiliza una instancia NAT, configurar los grupos de seguridad de la instancia NAT y las instancias de Amazon EC2 para habilitar la comunicación con Internet.
- Si se trata de un entorno con balanceo de carga y escalable en una Amazon VPC que solo tiene una subred pública, no se requiere ninguna configuración adicional. Esto se debe a que, con este entorno, las instancias de Amazon EC2 están configuradas con una dirección IP pública que permite a las instancias comunicarse con Internet.

Para obtener más información sobre el uso de Elastic Beanstalk con Amazon VPC, consulte [Uso de Elastic Beanstalk con Amazon VPC](#).

Roles de servicio, perfiles de instancia y políticas de usuario

Cuando crea un entorno, AWS Elastic Beanstalk le pide que proporcione los siguientes roles AWS Identity and Access Management (IAM):

- [Rol de servicio](#): el rol de servicio lo asume Elastic Beanstalk para usar otros servicios de Servicios de AWS en su nombre.
- [Perfil de instancia](#): Elastic Beanstalk aplica el perfil de instancias a las instancias de su entorno. Les permite realizar las siguientes tareas:
 - Recuperar [versiones de aplicaciones](#) de Amazon Simple Storage Service (Amazon S3)
 - Cargar registros en Simple Storage Service (Amazon S3).
 - Realizar otras tareas que varían según el tipo de entorno y la plataforma.

Rol de servicio

Cuando crea un entorno en la consola de Elastic Beanstalk o la CLI de Elastic Beanstalk EB, se crean y asignan los roles de servicio necesarios a las [políticas administradas](#). Estas políticas incluyen todos los permisos necesarios. Ahora, supongamos que el rol de servicio ya existe en su cuenta y, a continuación, crea un nuevo entorno en la consola de Elastic Beanstalk o mediante la CLI de Elastic Beanstalk. Si esto sucede, el rol de servicio existente se asigna automáticamente al nuevo entorno.

Perfil de instancia

Si su cuenta de AWS no tiene un perfil de instancia de EC2, debe crear uno mediante el servicio IAM. Después, puede asignar el perfil de instancia de EC2 a los nuevos entornos que cree. El asistente Crear entorno proporciona información que guía a través del servicio IAM, para que pueda crear un perfil de instancia de EC2 con los permisos necesarios. Después de crear el perfil de instancia, puede volver a la consola para seleccionarlo como perfil de instancia de EC2 y proseguir con los pasos para crear su entorno.

Note

Anteriormente, Elastic Beanstalk creaba un perfil de instancia de EC2 predeterminado denominado `aws-elasticbeanstalk-ec2-role` la primera vez que una cuenta de AWS creaba un entorno. Este perfil de instancia incluía políticas administradas predeterminadas. Si su cuenta ya tiene este perfil de instancia, seguirá estando disponible para que lo asigne a sus entornos.

No obstante, las recientes directrices de seguridad de AWS no permiten que un servicio de AWS cree automáticamente roles con políticas de confianza para otros servicios de AWS, en este caso EC2. Debido a estas directrices de seguridad, Elastic Beanstalk ya no crea un perfil de instancia `aws-elasticbeanstalk-ec2-role` predeterminado.

Políticas de usuario

Además de los dos roles que asigna a su entorno, también puede crear [políticas de usuario](#) y aplicarlas a los usuarios y grupos de IAM de su cuenta. La aplicación de políticas de usuario permite a los usuarios crear y administrar entornos y aplicaciones de Elastic Beanstalk. Elastic Beanstalk también proporciona políticas administradas para el acceso completo y el acceso de solo lectura. Para obtener más información sobre estas políticas, consulte [the section called “Políticas de usuario”](#).

Perfiles de instancia y políticas de usuario adicionales

Puede crear sus propios perfiles de instancia y políticas de usuario para escenarios avanzados. Si las instancias necesitan tener acceso a servicios que no están incluidos en las políticas predeterminadas, puede añadir políticas adicionales a las políticas predeterminadas. Si la política administrada es demasiado permisiva para sus necesidades, también puede crear políticas de usuario más restrictivas. Para obtener más información sobre los permisos de AWS, consulte la [Guía del usuario de IAM](#).

Temas

- [Rol de servicio de Elastic Beanstalk](#)
- [Perfil de instancia de Elastic Beanstalk](#)
- [Política de usuario de Elastic Beanstalk](#)

Rol de servicio de Elastic Beanstalk

Un rol de servicio es el rol de IAM que asume Elastic Beanstalk cuando llama a otros servicios en su nombre. Por ejemplo, Elastic Beanstalk utiliza el rol de servicio cuando llama a las API de Amazon Elastic Compute Cloud (Amazon EC2), Elastic Load Balancing y Amazon EC2 Auto Scaling para recopilar información. El rol de servicio que utiliza Elastic Beanstalk es el que especificó cuando creó el entorno de Elastic Beanstalk.

Hay dos políticas administradas que están asociadas al rol de servicio. Estas políticas proporcionan los permisos que habilitan a Elastic Beanstalk a acceder a los recursos de AWS necesarios para

crear y administrar sus entornos. Una política administrada brinda permisos para el [monitoreo de estado mejorado](#) y la compatibilidad con el nivel de trabajo de Amazon SQS, y otra política provee los permisos adicionales necesarios para las [actualizaciones de plataforma administradas](#).

AWSElasticBeanstalkEnhancedHealth

Esta política otorga todos los permisos que necesita Elastic Beanstalk para monitorear el estado del entorno. También incluye acciones de Amazon SQS para permitir que Elastic Beanstalk monitoree la actividad de cola para entornos de trabajo.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticloadbalancing:DescribeInstanceHealth",
        "elasticloadbalancing:DescribeLoadBalancers",
        "elasticloadbalancing:DescribeTargetHealth",
        "ec2:DescribeInstances",
        "ec2:DescribeInstanceStatus",
        "ec2:GetConsoleOutput",
        "ec2:AssociateAddress",
        "ec2:DescribeAddresses",
        "ec2:DescribeSecurityGroups",
        "sqs:GetQueueAttributes",
        "sqs:GetQueueUrl",
        "autoscaling:DescribeAutoScalingGroups",
        "autoscaling:DescribeAutoScalingInstances",
        "autoscaling:DescribeScalingActivities",
        "autoscaling:DescribeNotificationConfigurations",
        "sns:Publish"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

AWS Elastic Beanstalk Managed Updates Customer Role Policy

Esta política concede permisos a Elastic Beanstalk para actualizar los entornos en su nombre para realizar actualizaciones de plataformas administradas.

Agrupaciones de permisos de nivel de servicio

Esta política se agrupa en instrucciones basadas en el conjunto de permisos proporcionados.

- *ElasticBeanstalkPermissions*: este grupo de permisos es para llamar a las acciones del servicio Elastic Beanstalk (API de Elastic Beanstalk).
- *AllowPassRoleToElasticBeanstalkAndDownstreamServices*: este grupo de permisos permite que cualquier rol se transfiera a Elastic Beanstalk y a otros servicios descendentes como AWS CloudFormation.
- *ReadOnlyPermissions*: este grupo de permisos sirve para recopilar información sobre el entorno en ejecución.
- **OperationPermissions*: los grupos con este patrón de nomenclatura son para llamar a las operaciones necesarias a fin de llevar a cabo actualizaciones de plataforma.
- **BroadOperationPermissions*: los grupos con este patrón de nomenclatura son para llamar a las operaciones necesarias a fin de llevar a cabo actualizaciones de plataforma. También incluyen amplios permisos para admitir entornos heredados.
- **TagResource*: los grupos con este patrón de nomenclatura son para llamadas que utilizan las API de creación de etiquetas para adjuntar etiquetas a los recursos que se están creando en un entorno de Elastic Beanstalk.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ElasticBeanstalkPermissions",
      "Effect": "Allow",
      "Action": [
        "elasticbeanstalk:*"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowPassRoleToElasticBeanstalkAndDownstreamServices",
      "Effect": "Allow",
```

```

    "Action": "iam:PassRole",
    "Resource": "arn:aws:iam::*:role/*",
    "Condition": {
      "StringEquals": {
        "iam:PassedToService": [
          "elasticbeanstalk.amazonaws.com",
          "ec2.amazonaws.com",
          "ec2.amazonaws.com.cn",
          "autoscaling.amazonaws.com",
          "elasticloadbalancing.amazonaws.com",
          "ecs.amazonaws.com",
          "cloudformation.amazonaws.com"
        ]
      }
    }
  },
  {
    "Sid": "ReadOnlyPermissions",
    "Effect": "Allow",
    "Action": [
      "autoscaling:DescribeAccountLimits",
      "autoscaling:DescribeAutoScalingGroups",
      "autoscaling:DescribeAutoScalingInstances",
      "autoscaling:DescribeLaunchConfigurations",
      "autoscaling:DescribeLoadBalancers",
      "autoscaling:DescribeNotificationConfigurations",
      "autoscaling:DescribeScalingActivities",
      "autoscaling:DescribeScheduledActions",
      "ec2:DescribeAccountAttributes",
      "ec2:DescribeAddresses",
      "ec2:DescribeAvailabilityZones",
      "ec2:DescribeImages",
      "ec2:DescribeInstanceAttribute",
      "ec2:DescribeInstances",
      "ec2:DescribeKeyPairs",
      "ec2:DescribeLaunchTemplates",
      "ec2:DescribeLaunchTemplateVersions",
      "ec2:DescribeSecurityGroups",
      "ec2:DescribeSnapshots",
      "ec2:DescribeSpotInstanceRequests",
      "ec2:DescribeSubnets",
      "ec2:DescribeVpcClassicLink",
      "ec2:DescribeVpcs",
      "elasticloadbalancing:DescribeInstanceHealth",

```

```

        "elasticloadbalancing:DescribeLoadBalancers",
        "elasticloadbalancing:DescribeTargetGroups",
        "elasticloadbalancing:DescribeTargetHealth",
        "logs:DescribeLogGroups",
        "rds:DescribeDBEngineVersions",
        "rds:DescribeDBInstances",
        "rds:DescribeOrderableDBInstanceOptions",
        "sns:ListSubscriptionsByTopic"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Sid": "EC2BroadOperationPermissions",
    "Effect": "Allow",
    "Action": [
        "ec2:AllocateAddress",
        "ec2:AssociateAddress",
        "ec2:AuthorizeSecurityGroupEgress",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:CreateLaunchTemplate",
        "ec2:CreateLaunchTemplateVersion",
        "ec2:CreateSecurityGroup",
        "ec2>DeleteLaunchTemplate",
        "ec2>DeleteLaunchTemplateVersions",
        "ec2>DeleteSecurityGroup",
        "ec2:DisassociateAddress",
        "ec2:ReleaseAddress",
        "ec2:RevokeSecurityGroupEgress",
        "ec2:RevokeSecurityGroupIngress"
    ],
    "Resource": "*"
},
{
    "Sid": "EC2RunInstancesOperationPermissions",
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": "*",
    "Condition": {
        "ArnLike": {
            "ec2:LaunchTemplate": "arn:aws:ec2:*:*:launch-template/*"
        }
    }
}

```

```

    },
    {
      "Sid": "EC2TerminateInstancesOperationPermissions",
      "Effect": "Allow",
      "Action": [
        "ec2:TerminateInstances"
      ],
      "Resource": "arn:aws:ec2:*:*:instance/*",
      "Condition": {
        "StringLike": {
          "ec2:ResourceTag/aws:cloudformation:stack-id": [
            "arn:aws:cloudformation:*:*:stack/awseb-e-*",
            "arn:aws:cloudformation:*:*:stack/eb-*"
          ]
        }
      }
    }
  ],
  {
    "Sid": "ECSBroadOperationPermissions",
    "Effect": "Allow",
    "Action": [
      "ecs:CreateCluster",
      "ecs:DescribeClusters",
      "ecs:RegisterTaskDefinition"
    ],
    "Resource": "*"
  },
  {
    "Sid": "ECSDeleteClusterOperationPermissions",
    "Effect": "Allow",
    "Action": "ecs>DeleteCluster",
    "Resource": "arn:aws:ecs:*:*:cluster/awseb-*"
  },
  {
    "Sid": "ASGOperationPermissions",
    "Effect": "Allow",
    "Action": [
      "autoscaling:AttachInstances",
      "autoscaling:CreateAutoScalingGroup",
      "autoscaling:CreateLaunchConfiguration",
      "autoscaling:CreateOrUpdateTags",
      "autoscaling>DeleteLaunchConfiguration",
      "autoscaling>DeleteAutoScalingGroup",
      "autoscaling>DeleteScheduledAction",

```



```

        "autoscaling:DetachInstances",
        "autoscaling>DeletePolicy",
        "autoscaling:PutScalingPolicy",
        "autoscaling:PutScheduledUpdateGroupAction",
        "autoscaling:PutNotificationConfiguration",
        "autoscaling:ResumeProcesses",
        "autoscaling:SetDesiredCapacity",
        "autoscaling:SuspendProcesses",
        "autoscaling:TerminateInstanceInAutoScalingGroup",
        "autoscaling:UpdateAutoScalingGroup"
    ],
    "Resource": [
        "arn:aws:autoscaling:*:*:launchConfiguration:*:launchConfigurationName/awseb-e-*",
        "arn:aws:autoscaling:*:*:launchConfiguration:*:launchConfigurationName/eb-*",
        "arn:aws:autoscaling:*:*:autoScalingGroup:*:autoScalingGroupName/awseb-e-*",
        "arn:aws:autoscaling:*:*:autoScalingGroup:*:autoScalingGroupName/eb-*"
    ]
},
{
    "Sid": "CFNOperationPermissions",
    "Effect": "Allow",
    "Action": [
        "cloudformation:*"
    ],
    "Resource": [
        "arn:aws:cloudformation:*:*:stack/awseb-*",
        "arn:aws:cloudformation:*:*:stack/eb-*"
    ]
},
{
    "Sid": "ELBOperationPermissions",
    "Effect": "Allow",
    "Action": [
        "elasticloadbalancing:AddTags",
        "elasticloadbalancing:ApplySecurityGroupsToLoadBalancer",
        "elasticloadbalancing:ConfigureHealthCheck",
        "elasticloadbalancing>CreateLoadBalancer",
        "elasticloadbalancing>DeleteLoadBalancer",
        "elasticloadbalancing:DeregisterInstancesFromLoadBalancer",
        "elasticloadbalancing:DeregisterTargets",
        "elasticloadbalancing:RegisterInstancesWithLoadBalancer",

```

```

        "elasticloadbalancing:RegisterTargets"
    ],
    "Resource": [
        "arn:aws:elasticloadbalancing:*:*:targetgroup/awseb-*",
        "arn:aws:elasticloadbalancing:*:*:targetgroup/eb-*",
        "arn:aws:elasticloadbalancing:*:*:loadbalancer/awseb-*",
        "arn:aws:elasticloadbalancing:*:*:loadbalancer/eb-*",
        "arn:aws:elasticloadbalancing:*:*:loadbalancer/*/awseb-*/**",
        "arn:aws:elasticloadbalancing:*:*:loadbalancer/*/eb-*/**"
    ]
},
{
    "Sid": "CWLogsOperationPermissions",
    "Effect": "Allow",
    "Action": [
        "logs:CreateLogGroup",
        "logs>DeleteLogGroup",
        "logs:PutRetentionPolicy"
    ],
    "Resource": "arn:aws:logs:*:*:log-group:/aws/elasticbeanstalk/*"
},
{
    "Sid": "S3ObjectOperationPermissions",
    "Effect": "Allow",
    "Action": [
        "s3:DeleteObject",
        "s3:GetObject",
        "s3:GetObjectAcl",
        "s3:GetObjectVersion",
        "s3:GetObjectVersionAcl",
        "s3:PutObject",
        "s3:PutObjectAcl",
        "s3:PutObjectVersionAcl"
    ],
    "Resource": "arn:aws:s3:::elasticbeanstalk-*/*"
},
{
    "Sid": "S3BucketOperationPermissions",
    "Effect": "Allow",
    "Action": [
        "s3:GetBucketLocation",
        "s3:GetBucketPolicy",
        "s3:ListBucket",
        "s3:PutBucketPolicy"
    ]
}

```

```
    ],
    "Resource": "arn:aws:s3:::elasticbeanstalk-*"
  },
  {
    "Sid": "SNSOperationPermissions",
    "Effect": "Allow",
    "Action": [
      "sns:CreateTopic",
      "sns:GetTopicAttributes",
      "sns:SetTopicAttributes",
      "sns:Subscribe"
    ],
    "Resource": "arn:aws:sns:*:*:ElasticBeanstalkNotifications-*"
  },
  {
    "Sid": "SQSOperationPermissions",
    "Effect": "Allow",
    "Action": [
      "sqs:GetQueueAttributes",
      "sqs:GetQueueUrl"
    ],
    "Resource": [
      "arn:aws:sqs:*:*:awseb-e-*",
      "arn:aws:sqs:*:*:eb-*"
    ]
  },
  {
    "Sid": "CWPutMetricAlarmOperationPermissions",
    "Effect": "Allow",
    "Action": [
      "cloudwatch:PutMetricAlarm"
    ],
    "Resource": [
      "arn:aws:cloudwatch:*:*:alarm:awseb-*",
      "arn:aws:cloudwatch:*:*:alarm:eb-*"
    ]
  },
  {
    "Sid": "AllowECSTagResource",
    "Effect": "Allow",
    "Action": [
      "ecs:TagResource"
    ],
    "Resource": "*"
  }
```

```
    "Condition": {
      "StringEquals": {
        "ecs:CreateAction": [
          "CreateCluster",
          "RegisterTaskDefinition"
        ]
      }
    }
  ]
}
```

Puede crear un entorno de Elastic Beanstalk con cualquiera de los siguientes enfoques. Cada sección describe cómo el enfoque administra el rol de servicio.

Consola de Elastic Beanstalk

Si crea un entorno mediante la consola de Elastic Beanstalk, Elastic Beanstalk le pide que cree un rol de servicio denominado `aws-elasticbeanstalk-service-role`. Cuando se crea con Elastic Beanstalk, este rol incluye una política de confianza que permite a Elastic Beanstalk asumir el rol de servicio. Las dos políticas administradas antes mencionadas en este tema también se asocian al rol.

Interfaz de la línea de comandos de Elastic Beanstalk (CLI de EB)

Puede crear un entorno mediante el comando [the section called “eb create”](#) de la interfaz de línea de comandos de Elastic Beanstalk (CLI de EB). Si no especifica un rol de servicio mediante la opción `--service-role`, Elastic Beanstalk crea el mismo rol de servicio `aws-elasticbeanstalk-service-role` predeterminado. Si el rol de servicio predeterminado ya existe, Elastic Beanstalk lo utiliza para el entorno nuevo. Cuando se crea con Elastic Beanstalk, este rol incluye una política de confianza que permite a Elastic Beanstalk asumir el rol de servicio. Las dos políticas administradas antes mencionadas en este tema también se asocian al rol.

API de Elastic Beanstalk

Puede crear un entorno mediante la acción `CreateEnvironment` de la API de Elastic Beanstalk. Si no especifica un rol de servicio, Elastic Beanstalk crea un rol vinculado a servicios de supervisión. Se trata de un tipo único de rol de servicio predefinido por Elastic Beanstalk que incluye todos los permisos que el servicio requiere para llamar a otros Servicios de AWS en su nombre. El rol vinculado al servicio está asociado a su cuenta. Elastic Beanstalk lo crea una vez y, a continuación, vuelve a utilizarlo al crear entornos adicionales. También puede utilizar IAM para crear un rol vinculado al servicio de supervisión de la cuenta con antelación. Cuando su cuenta tiene un rol

vinculado al servicio de supervisión, puede utilizarlo para crear un entorno mediante la consola de Elastic Beanstalk, la API de Elastic Beanstalk o la CLI de EB. Para obtener instrucciones sobre cómo utilizar roles vinculados a servicios con entornos de Elastic Beanstalk, consulte [Uso de roles vinculados a servicios para Elastic Beanstalk](#).

Para obtener más información acerca de los roles de servicio, consulte [Administración de roles de servicio de Elastic Beanstalk](#).

Perfil de instancia de Elastic Beanstalk

Un perfil de instancia es un rol de IAM que se aplica a las instancias de Amazon EC2 que se lanzan en el entorno de Elastic Beanstalk. Cuando se crea un entorno de Elastic Beanstalk, se especifica el perfil de instancia que se utiliza cuando las instancias de EC2 realizan las siguientes acciones:

- Recuperar [versiones de aplicaciones](#) de Amazon Simple Storage Service (Amazon S3)
- Escribir registros en Amazon S3
- En [entornos integrados de AWS X-Ray](#), cargue datos de depuración en X-Ray.
- En entornos Docker administrados por Amazon ECS, coordine implementaciones de contenedores con Amazon Elastic Container Service (Amazon ECS)
- En entornos de trabajo, lea desde una cola de Amazon Simple Queue Service (Amazon SQS)
- En entornos de trabajo, realizan la elección del líder con Amazon DynamoDB
- En entornos de trabajo, publican métricas de estado de las instancias en Amazon CloudWatch

Elastic Beanstalk proporciona un conjunto de políticas administradas que permiten a las instancias de EC2 del entorno realizar las operaciones requeridas. Las políticas administradas requeridas para los casos de uso básicos son las siguientes.

- `AWSElasticBeanstalkWebTier`
- `AWSElasticBeanstalkWorkerTier`
- `AWSElasticBeanstalkMulticontainerDocker`

Estas políticas se asocian al perfil de instancia que se crea cuando se lanza un entorno en la consola de Elastic Beanstalk por primera vez.

Si la aplicación web requiere acceso a otros Servicios de AWS adicionales, agregue instrucciones o políticas administradas al perfil de instancia que permitan el acceso a dichos servicios.

Para obtener más información sobre los perfiles de instancia, consulte [Administración de perfiles de instancia de Elastic Beanstalk](#).

Política de usuario de Elastic Beanstalk

Cree usuarios de IAM para cada usuario que utiliza Elastic Beanstalk para evitar usar la cuenta raíz o compartir credenciales. Como práctica recomendada, conceda solo a estos usuarios permiso para tener acceso a los servicios y las características que necesitan.

Elastic Beanstalk requiere permisos no solo para sus propias acciones de la API, sino también para otros servicios de AWS. Elastic Beanstalk utiliza permisos de usuario para lanzar recursos en un entorno. Estos recursos incluyen instancias de EC2, un balanceador de carga de Elastic Load Balancer y un grupo de Auto Scaling. Elastic Beanstalk también utiliza permisos de usuario para guardar registros y plantillas en Amazon Simple Storage Service (Amazon S3), enviar notificaciones a Amazon SNS, asignar perfiles de instancia y publicar métricas en CloudWatch. Elastic Beanstalk requiere permisos de AWS CloudFormation para organizar las implementaciones de recursos y las actualizaciones. También requiere permisos de Amazon RDS para crear bases de datos cuando sea necesario y permisos de Amazon SQS para crear colas para entornos de trabajo.

Para obtener más información sobre las políticas de usuario, consulte [Administración de políticas de usuario de Elastic Beanstalk](#).

Plataformas Elastic Beanstalk

AWS Elastic Beanstalk proporciona una variedad de plataformas en las que puede crear sus aplicaciones. Diseñe su aplicación web con una de estas plataformas y Elastic Beanstalk implementará su código en la versión de la plataforma que seleccione para crear un entorno de aplicación activo.

Elastic Beanstalk proporciona plataformas para distintos lenguajes de programación, servidores de aplicación y contenedores de Docker. Algunas plataformas disponen de varias versiones compatibles a la vez.

Temas

- [Glosario de plataformas Elastic Beanstalk](#)
- [Modelo de responsabilidad compartida para el mantenimiento de plataformas Elastic Beanstalk](#)
- [Política de compatibilidad con plataformas Elastic Beanstalk](#)
- [Calendario de lanzamientos de la plataforma Elastic Beanstalk](#)
- [Plataformas compatibles con Elastic Beanstalk](#)
- [Plataformas Linux de Elastic Beanstalk](#)
- [Implementación de aplicaciones de Elastic Beanstalk desde contenedores Docker](#)
- [Creación e implementación de aplicaciones Go en Elastic Beanstalk](#)
- [Creación e implementación de aplicaciones Java en Elastic Beanstalk](#)
- [Trabajar con .NET Core en Linux](#)
- [Creación e implementación de aplicaciones .NET Windows en Elastic Beanstalk](#)
- [Implementación de aplicaciones Node.js en Elastic Beanstalk](#)
- [Creación e implementación de aplicaciones PHP en Elastic Beanstalk](#)
- [Uso de Python](#)
- [Creación e implementación de aplicaciones Ruby en Elastic Beanstalk](#)

Glosario de plataformas Elastic Beanstalk

A continuación se muestran términos clave relacionados con las plataformas de AWS Elastic Beanstalk y el ciclo de vida de estas.

Tiempo de ejecución

El software de tiempo de ejecución específico a los lenguajes de programación (marco, bibliotecas, intérprete, vm, etc.) necesario para ejecutar el código de la aplicación.

Componentes Elastic Beanstalk

Componentes de software que Elastic Beanstalk agrega a una plataforma para habilitar la funcionalidad de Elastic Beanstalk. Por ejemplo, el agente de estado avanzado es necesario para recopilar información sanitaria y generar informes sobre ella.

Plataforma

Una plataforma es una combinación de un sistema operativo (SO), un tiempo de ejecución, un servidor web, un servidor de aplicaciones y componentes de Elastic Beanstalk. Las plataformas proporcionan componentes que están disponibles para ejecutar la aplicación.

Versión de la plataforma

Una combinación de versiones específicas de un sistema operativo (SO), un tiempo de ejecución, un servidor web, un servidor de aplicaciones y componentes de Elastic Beanstalk. Crea un entorno de Elastic Beanstalk basado en una versión de la plataforma e implementa la aplicación en él.

Una versión de la plataforma tiene un número de versión semántica con la forma X.Y.Z, donde X es la versión principal, Y es la versión secundaria y Z es la versión del parche.

Una versión de la plataforma puede estar en uno de los siguientes estados:

- **Compatible:** una versión de la plataforma que consta únicamente de componentes compatibles. Los componentes no han llegado al final de su vida útil (EOL) como lo indican sus respectivos proveedores (propietarios: AWS o terceros o comunidades). Reciben parches regulares o actualizaciones menores de sus proveedores. Elastic Beanstalk pone a su disposición versiones de plataforma compatibles para la creación de entornos.
- **Retirada:** una versión de la plataforma con uno o más componentes retirados, que han llegado al final de su vida útil (EOL), tal y como indican sus proveedores. Las versiones de la plataforma retiradas no están disponibles para su uso en entornos de Elastic Beanstalk para los clientes nuevos o existentes.

Para obtener más información, consulte [the section called “Política de compatibilidad con plataformas”](#).

Ramificación de la plataforma

Una línea de versiones de la plataforma que comparten versiones específicas (normalmente principales) de algunos de sus componentes, como el sistema operativo (SO), el tiempo de ejecución o los componentes de Elastic Beanstalk. Por ejemplo: Python 3.6 en ejecución en Amazon Linux de 64 bits; IIS 10.0 en ejecución en Windows Server 2016 de 64 bits. Cada versión sucesiva de la plataforma en la ramificación es una actualización de la anterior.

La versión más reciente de la plataforma en cada ramificación de la plataforma está disponible sin condiciones para la creación del entorno. Las versiones anteriores de la plataforma en la ramificación siguen siendo compatibles: puede crear un entorno basado en una versión anterior de la plataforma si la ha utilizado en un entorno en los últimos 30 días. Pero estas versiones anteriores de la plataforma no tienen los componentes más recientes y, por ello, no se recomienda su uso.

Una ramificación de la plataforma puede estar en uno de los siguientes estados:

- **Compatible:** una ramificación actual de la plataforma. Consta únicamente de componentes compatibles. Recibe continuas actualizaciones de la plataforma y se recomienda su uso en entornos de producción. Para obtener una lista de las ramificaciones compatibles de la plataforma, consulte [Plataformas compatibles con Elastic Beanstalk](#) en la guía Plataformas de AWS Elastic Beanstalk.
- **Beta:** una versión preliminar de la ramificación de la plataforma. Es de naturaleza experimental. Puede recibir continuas actualizaciones de la plataforma durante un tiempo, pero no tiene soporte a largo plazo. No se recomienda utilizar una ramificación beta de la plataforma en entornos de producción. Úsela solo para evaluación. Para obtener una lista de las ramificaciones beta de la plataforma, consulte [Versiones beta públicas de la plataforma Elastic Beanstalk](#) en la guía AWS Elastic Beanstalk Plataformas.
- **Obsoleta:** una ramificación de la plataforma con uno o más componentes obsoletos. Recibe continuas actualizaciones de la plataforma, pero no se recomienda su uso en entornos de producción. Para obtener una lista de las ramificaciones obsoletas de la plataforma, consulte [Versiones de la plataforma Elastic Beanstalk programadas para su retiro](#) en la guía Plataformas de AWS Elastic Beanstalk.
- **Retirada:** una ramificación de la plataforma con uno o más componentes retirados. Ya no recibe actualizaciones de la plataforma y no se recomienda su uso en entornos de producción. Las ramificaciones de plataformas retiradas no aparecen en la guía Plataformas de AWS Elastic Beanstalk. Elastic Beanstalk no pone a su disposición versiones de plataforma de sucursales de plataforma retiradas para la creación de entornos.

Un componente compatible no tiene una fecha de retirada programada por su proveedor (propietario o comunidad). El proveedor puede ser AWS o un tercero. Un componente obsoleto tiene una fecha de retirada programada por su proveedor. Un componente retirado ha llegado al final de su vida útil (EOL) y ya no es admitido por su proveedor. Para obtener más información, consulte [the section called “Política de compatibilidad con plataformas”](#).

Si su entorno utiliza una ramificación obsoleta o retirada de la plataforma, le recomendamos que la actualice a una versión de la plataforma de una ramificación de la plataforma compatible. Para obtener más información, consulte [the section called “Actualizaciones de la plataforma”](#).

Actualización de plataforma

La publicación de una nueva versión de la plataforma que contiene actualizaciones de algunos componentes de la plataforma: SO, tiempo de ejecución, servidor web, servidor de aplicaciones y componentes de Elastic Beanstalk. Las actualizaciones de la plataforma siguen una taxonomía de versión semántica y pueden tener varios niveles:

- **Actualización principal:** una actualización que contiene cambios incompatibles con las versiones de la plataforma existentes. Es posible que tenga que modificar su aplicación para que se ejecute correctamente en una nueva versión principal. Una actualización principal tiene un nuevo número de versión principal de la plataforma.
- **Actualización secundaria:** una actualización que añade una funcionalidad que es compatible con una versión anterior de la plataforma. No tendrá que modificar su aplicación para que se ejecute correctamente en una nueva versión secundaria. Una actualización secundaria tiene un nuevo número de versión secundaria de la plataforma.
- **Actualización de parche:** una actualización que consta de versiones de mantenimiento (solución de errores, actualizaciones de seguridad y mejoras de rendimiento) compatibles con versiones anteriores de la plataforma. Una actualización de parche tiene un nuevo número de versión de parche de la plataforma.

Managed Updates

Una característica de Elastic Beanstalk que aplica automáticamente actualizaciones secundarias y parches a los componentes del sistema operativo (SO), el tiempo de ejecución, el servidor web, el servidor de aplicaciones y los componentes de Elastic Beanstalk para una versión de plataforma compatible con Elastic Beanstalk. Una actualización administrada aplica una versión de la plataforma más reciente de la misma ramificación de la plataforma a su entorno. Puede configurar las actualizaciones de plataforma administradas para aplicar solo las actualizaciones de parches o para las actualizaciones secundarias y de parches. También puede deshabilitar las actualizaciones administradas por completo.

Para obtener más información, consulte [Actualizaciones de plataforma administradas](#).

Modelo de responsabilidad compartida para el mantenimiento de plataformas Elastic Beanstalk

AWS y nuestros clientes comparten la responsabilidad de lograr un alto nivel de seguridad y conformidad de los componentes de software. Este modelo compartido reduce su carga operativa.

Para obtener más información, consulte el [modelo de responsabilidad AWS compartida](#).

AWS Elastic Beanstalk le ayuda a cumplir su parte del modelo de responsabilidad compartida al proporcionar una función de actualizaciones gestionadas. Esta función aplica automáticamente actualizaciones secundarias y de parche para una versión de plataforma compatible de Elastic Beanstalk. Si se produce un error en una actualización administrada, Elastic Beanstalk le notifica el error para garantizar que sea consciente del mismo y pueda emprender acciones inmediatas.

Para obtener más información, consulte [Actualizaciones de plataforma administradas](#).

Además, Elastic Beanstalk hace lo siguiente:

- Publica su [política de compatibilidad con plataformas](#) y la programación de retiradas para los siguientes 12 meses.
- Publica actualizaciones principales, secundarias y de parches para componentes de sistema operativo (SO), tiempo de ejecución, servidor de aplicaciones y servidor web, normalmente en el plazo de 30 días tras estar disponibles. Elastic Beanstalk se encarga de crear actualizaciones para componentes de Elastic Beanstalk que estén presentes en sus versiones de plataforma compatibles. Todas las demás actualizaciones vienen directamente de sus proveedores (los propietarios o la comunidad).

Anunciamos todas las actualizaciones de las plataformas compatibles en nuestras [notas de la versión](#) en la guía Notas de la versión de AWS Elastic Beanstalk . También proporcionamos una lista de todas las plataformas compatibles y sus componentes, junto con un historial de la plataforma, en la guía Plataformas de AWS Elastic Beanstalk . Para más información, consulte [Plataformas admitidas](#).

Usted será responsable de lo siguiente:

- Actualice todos los componentes que controla (identificados como Cliente en el [modelo de responsabilidad AWS compartida](#)). Esto incluye garantizar la seguridad de la aplicación, los datos y los componentes que necesite la aplicación y que haya descargado.
- Asegurarse de que sus entornos de Elastic Beanstalk se ejecuten en una versión de plataforma compatible, y migrar cualquier entorno que se ejecute en una versión de la plataforma retirada a una versión compatible.
- Resolver todos los problemas que surgen en intentos fallidos de actualización administrada y volver a intentar la actualización.
- Aplicar usted mismo parches al SO, el tiempo de ejecución, el servidor de aplicaciones y el servidor web si ha decidido no optar por las actualizaciones administradas de Elastic Beanstalk. Puede hacer esto [aplicando manualmente actualizaciones de plataforma](#) o aplicando parches directamente a los componentes en todos los recursos del entorno pertinentes.
- [Gestione la seguridad y el cumplimiento de todos AWS los servicios que utilice fuera de Elastic Beanstalk de acuerdo con AWS el modelo de responsabilidad compartida.](#)

Política de compatibilidad con plataformas Elastic Beanstalk

AWS Elastic Beanstalk proporciona una variedad de plataformas para ejecutar aplicaciones AWS. Elastic Beanstalk admite las ramificaciones de la plataforma que siguen recibiendo actualizaciones secundarias y de parches continuas por parte de sus proveedores (propietarios o comunidad). Para obtener una definición completa de los términos relacionados, consulte [Glosario de plataformas Elastic Beanstalk](#).

Ramificaciones de la plataforma retiradas

Cuando el proveedor marca el fin de su vida útil (EOL) de un componente de una rama de plataforma compatible, Elastic Beanstalk marca la rama de plataforma como retirada. Los componentes de una rama de plataforma incluyen los siguientes: sistema operativo (SO), versión en lenguaje de ejecución, servidor de aplicaciones o servidor web.

Una vez que una rama de plataforma se marca como retirada, se aplican las siguientes políticas:

- Elastic Beanstalk deja de proporcionar actualizaciones de mantenimiento, incluidas las actualizaciones de seguridad.
- Elastic Beanstalk ya no proporciona soporte técnico a las sucursales de plataforma retiradas.

- Elastic Beanstalk ya no pone la rama de plataforma a disposición de los nuevos clientes de Elastic Beanstalk para su implementación en nuevos entornos. Hay un período de gracia de 90 días a partir de la fecha de retirada publicada para los clientes existentes con entornos activos que se ejecutan en ramificaciones de las plataformas retiradas.

Note

Una rama de plataforma retirada no estará disponible en la consola de Elastic Beanstalk. Sin embargo, estará disponible a través de la AWS CLI CLI de EB y la API de EB para los clientes que tengan entornos existentes basados en la rama de plataforma retirada. Los clientes actuales también pueden usar las consolas del [entorno Clone](#) y [Rebuild](#).

Para obtener una lista de las ramas de la plataforma cuya retirada está programada, consulte el [Programación de sucursales de plataforma de retirada](#) tema de programación de la plataforma de Elastic Beanstalk que aparece a continuación.

Para obtener más información sobre qué esperar cuando la rama de plataforma de su entorno se retire, consulte. [Preguntas frecuentes sobre la retirada de plataformas](#)

Después del período de gracia de 90 días

Nuestra política para las sucursales de plataforma retiradas no elimina el acceso a los entornos ni elimina los recursos. Sin embargo, los clientes actuales que ejecutan un entorno de Elastic Beanstalk en una sucursal de plataforma retirada deben ser conscientes de los riesgos que conlleva hacerlo. Estos entornos pueden terminar en una situación impredecible, ya que Elastic Beanstalk no puede proporcionar actualizaciones de seguridad, soporte técnico o revisiones a las sucursales de la plataforma retiradas debido a que el proveedor marca el EOL de sus componentes.

Por ejemplo, puede surgir una vulnerabilidad de seguridad perjudicial y crítica en un entorno que se ejecuta en una ramificación de la plataforma retirada. O una acción de la API de EB puede dejar de funcionar para el entorno si con el tiempo se vuelve incompatible con el servicio de Elastic Beanstalk. La posibilidad de que se produzcan este tipo de riesgos aumenta cuanto más tiempo permanezca activo un entorno en una ramificación de la plataforma retirada. Para que pueda seguir beneficiándose de las mejoras importantes de seguridad, rendimiento y funcionalidad ofrecidas por los proveedores de componentes en las versiones más recientes, le recomendamos encarecidamente que actualice todos los entornos de Elastic Beanstalk a una versión compatible de la plataforma.

Si su aplicación presenta problemas al ejecutarse en una rama de plataforma retirada y no puede migrarla a una plataforma compatible, tendrá que considerar otras alternativas. Las soluciones alternativas incluyen encapsular la aplicación en una imagen de Docker para ejecutarla como contenedor de Docker. Esto permitiría a un cliente utilizar cualquiera de nuestras soluciones de Docker, como nuestras plataformas Docker de Elastic Beanstalk AL2023/AL2, u otros servicios basados en Docker, como Amazon ECS o Amazon EKS. Las alternativas que no son de Docker incluyen nuestro AWS CodeDeploy servicio, que permite una personalización completa de los tiempos de ejecución que desee.

Calendario de lanzamientos de la plataforma Elastic Beanstalk

Para garantizar que sus aplicaciones se ejecuten en un entorno seguro y compatible, Elastic Beanstalk proporciona actualizaciones periódicas para sus plataformas gestionadas, tal y como se describe en el tema anterior. [Modelo de responsabilidad compartida](#) Además del lanzamiento mensual de las nuevas versiones de las sucursales de la plataforma, el mantenimiento de nuestras versiones también incluye los siguientes procesos:

- Lanzamiento de nuevas ramas de plataforma: por lo general, introducen una nueva versión principal de un lenguaje, sistema operativo o servidor de aplicaciones en tiempo de ejecución.
- Retirada de las sucursales de plataforma: debemos retirar una rama de plataforma cuando uno de sus componentes llegue al final de su vida útil (EOL). Para obtener más información sobre nuestra política para las sucursales retiradas, consulta [Política de compatibilidad con plataformas Elastic Beanstalk](#)

Temas

- [Recursos de planificación](#)
- [Próximos lanzamientos de las sucursales de plataformas](#)
- [Programación de sucursales de plataforma de retirada](#)
- [Historial de la ramificación de la plataforma retirada](#)
- [Historial de servidores y sistemas operativos retirados](#)

Recursos de planificación

Además de los programas siguientes, hay recursos adicionales que pueden ayudarle a planificar el mantenimiento y el soporte de la aplicación que se ejecuta en una plataforma de Elastic Beanstalk.

Para obtener más información sobre los componentes de nuestra plataforma, las fechas importantes y los anuncios de lanzamiento, consulte los siguientes recursos:

- [AWS Elastic Beanstalk Guía de plataformas](#): esta guía proporciona una lista detallada de los componentes de cada una de las sucursales de nuestra plataforma. También proporciona un historial de plataformas por fecha de lanzamiento con los mismos detalles. Esta guía puede informarle cuando se modifiquen componentes específicos de la rama de su plataforma. Si su aplicación comienza a comportarse de forma diferente, también puede hacer una referencia cruzada de la fecha en la que ocurrió en la guía de plataformas para ver si hubo algún cambio en la plataforma que pudiera haber afectado a su aplicación.
- [AWS Elastic Beanstalk Notas de la versión](#): nuestras notas de la versión anuncian todas las versiones de nuestra plataforma, tanto las secundarias como las principales. Esto incluye nuestras actualizaciones mensuales de la plataforma, las versiones de seguridad, las correcciones y los anuncios de retirada. Puede suscribirse a nuestras fuentes RSS desde la documentación de las notas de la versión.

Próximos lanzamientos de las sucursales de plataformas

En la siguiente tabla se enumeran las próximas sucursales de la plataforma Elastic Beanstalk y su fecha de lanzamiento prevista. Estas fechas son provisionales y están sujetas a cambios.

Versión en tiempo de ejecución o rama de plataforma	Sistema operativo	Fecha de lanzamiento prevista
Corretto 21 with Tomcat 10 AL2023	Amazon Linux 2023	Septiembre de 2024
PHP 8.3 AL2023	Amazon Linux 2023	Septiembre de 2024
Python 3.12 AL2023	Amazon Linux 2023	Septiembre de 2024
Ruby 3.3 AL2023	Amazon Linux 2023	Noviembre de 2024

Programación de sucursales de plataforma de retirada

En la siguiente tabla se enumeran las sucursales de la plataforma Elastic Beanstalk cuya retirada está programada porque algunos de sus componentes están llegando al final de su vida útil (EOL).

Para obtener una lista más detallada de las sucursales de la plataforma que se retiran, que incluye sus componentes específicos, consulte las versiones de [plataforma que se retiran](#) en la guía de plataformas.AWS Elastic Beanstalk

Versión en tiempo de ejecución o rama de plataforma	Sistema operativo	Fecha prevista de jubilación
Corretto 8 with Tomcat 8.5 AL2	Amazon Linux 2	30 de septiembre de 2024
Corretto 11 with Tomcat 8.5 AL2	Amazon Linux 2	30 de septiembre de 2024
.NET 6 AL2023	Amazon Linux 2023	31 de enero de 2025
Node.js 14 AL2	Amazon Linux 2	30 de septiembre de 2024
Node.js 16 AL2	Amazon Linux 2	30 de septiembre de 2024
Ruby 2.7 AL2	Amazon Linux 2	30 de septiembre de 2024
Ruby 3.0 AL2	Amazon Linux 2	30 de septiembre de 2024
PHP 8.0 AL2	Amazon Linux 2	30 de septiembre de 2024
PHP 8.1 AL2	Amazon Linux 2	31 de enero de 2025
PHP 8.1 AL2023	Amazon Linux 2023	31 de enero de 2025
Python 3.7 AL2	Amazon Linux 2	30 de septiembre de 2024
Python 3.8 AL2	Amazon Linux 2	31 de enero de 2025

Historial de la ramificación de la plataforma retirada

En las siguientes tablas se enumeran las ramas de la plataforma Elastic Beanstalk que ya están retiradas. Puede ver un historial detallado de estas ramas de la plataforma y sus componentes en el [historial de la plataforma de la guía](#) de plataformas.AWS Elastic Beanstalk

Amazon Linux 2 (AL2)

Versión en tiempo de ejecución o rama de plataforma	Fecha de retirada		
Corretto 11 with Tomcat 7 AL2	29 de junio de 2022		
Corretto 8 with Tomcat 7 AL2	29 de junio de 2022		
Node.js 12 AL2	23 de diciembre de 2022		
Node.js 10 AL2	29 de junio de 2022		
PHP 7.4 AL2	9 de junio de 2023		
PHP 7.3 AL2	29 de junio de 2022		
PHP 7.2 AL2	29 de junio de 2022		
Ruby 2.6 AL2	23 de diciembre de 2022		
Ruby 2.5 AL2	29 de junio de 2022		

Amazon Linux AMI (AL1)

Versión en tiempo de ejecución/rama de plataforma	Fecha de retirada		
Single Container Docker	18 de julio de 2022		
Multicontainer Docker	18 de julio de 2022		
Preconfigured Docker - GlassFish 5.0 with Java 8	18 de julio de 2022		
Go 1	18 de julio de 2022		
Java 8	18 de julio de 2022		
Java 7	18 de julio de 2022		
Java 8 with Tomcat 8.5	18 de julio de 2022		
Java 7 with Tomcat 7	18 de julio de 2022		
Node.js	18 de julio de 2022		
PHP 7.2 - 7.3	18 de julio de 2022		
Python 3.6	18 de julio de 2022		
Ruby 2.4, 2.5, 2.6 with Passenger	18 de julio de 2022		

Versión en tiempo de ejecución/rama de plataforma	Fecha de retirada		
Ruby 2.4, 2.5, 2.6 with Puma	18 de julio de 2022		
Go 1.3–1.10	31 de octubre de 2020		
Java 6	31 de octubre de 2020		
Node.js 4.x–8.x	31 de octubre de 2020		
PHP 5.4–5.6	31 de octubre de 2020		
PHP 7.0–7.1	31 de octubre de 2020		
Python 2.6, 2.7, 3.4	31 de octubre de 2020		
Ruby 1.9.3	31 de octubre de 2020		
Ruby 2.0–2.3	31 de octubre de 2020		

Note

El [18 de julio de 2022](#), Elastic Beanstalk estableció el estado de todas las ramificaciones de la plataforma basadas en la AMI de Amazon Linux (AL1) como retirado. Para obtener más información, consulte [Preguntas frecuentes sobre la retirada de plataformas](#).

Windows Server

Versión en tiempo de ejecución/rama de plataforma	Fecha de retirada		
IIS 8.5 en Windows Server (y Core) 2012 R2 de 64 bits, versión 0.1.0	29 de junio de 2022		
IIS 8.5 en Windows Server (y Core) 2012 R2 de 64 bits, versión 1.2.0	29 de junio de 2022		
IIS 10.0 en Windows Server 2016 de 64 bits (y Core), versión 1.2.0	29 de junio de 2022		
IIS 8 en la ramificación de la plataforma de Windows Server 2012 R1 de 64 bits	22 de junio de 2022		
IIS 8 en Windows Server 2012 R1 de 64 bits, versión 0.1.0	22 de junio de 2022		

Versión en tiempo de ejecución/rama de plataforma	Fecha de retirada		
IIS 8 en Windows Server 2012 R1 de 64 bits, versión 1.2.0	22 de junio de 2022		

Note

Para obtener más información sobre la retirada de las ramificaciones de la plataforma Windows 2012 R2, consulte las ramificaciones de la [plataforma Windows Server 2012 R2 retiradas](#) en las notas de la versión AWS Elastic Beanstalk .

Historial de servidores y sistemas operativos retirados

Las siguientes tablas proporcionan un historial de los sistemas operativos, servidores de aplicaciones y servidores web que ya no son compatibles con las plataformas Elastic Beanstalk. Todas las sucursales de la plataforma que utilizaban estos componentes ya están retiradas. Las fechas reflejan la fecha de retirada de la última sucursal de la plataforma Elastic Beanstalk que incluyó el componente.

Sistemas operativos

Versión del sistema operativo	Fecha de retirada de la plataforma		
Windows Server 2012 R2 running IIS 8.5	4 de diciembre de 2023		

Versión del sistema operativo	Fecha de retirada de la plataforma		
Windows Server Core 2012 R2 running IIS 8.5	4 de diciembre de 2023		
Amazon Linux AMI (AL1)	18 de julio de 2022		
Windows Server 2012 R1	22 de junio de 2022		
Windows Server 2008 R2	28 de octubre de 2019		

Servidores de aplicaciones

Versión del servidor de aplicaciones	Fecha de retirada de la plataforma		
Tomcat 7	29 de junio de 2022 para plataformas de Amazon Linux 2 (AL2) 18 de julio de 2022 para las plataformas de la AMI de Amazon Linux (AL1)		
Tomcat 8	31 de octubre de 2020		
Tomcat 6	31 de octubre de 2020		

Servidores web

Versión del servidor web	Fecha de retirada de la plataforma		
IIS 8 en Windows Server de 64 bits	22 de junio de 2022		
Servidor HTTP de Apache 2.2	31 de octubre de 2020		
Nginx 1.12.2	31 de octubre de 2020		

Plataformas compatibles con Elastic Beanstalk

AWS Elastic Beanstalk proporciona una variedad de plataformas en las que puede crear sus aplicaciones. Diseñe su aplicación web con una de estas plataformas y Elastic Beanstalk implementará su código en la versión de la plataforma que seleccione para crear un entorno de aplicación activo.

Elastic Beanstalk proporciona plataformas para los lenguajes de programación (Go, Java, Node.js, PHP, Python y Ruby), servidores de aplicaciones (Tomcat, Passenger y Puma) y contenedores de Docker. Algunas plataformas disponen de varias versiones compatibles a la vez.

Elastic Beanstalk aprovisiona los recursos necesarios para ejecutar la aplicación, incluida una o varias instancias de Amazon EC2. La pila de software que se ejecuta en las instancias de Amazon EC2 depende de la versión de la plataforma específica que ha seleccionado para su entorno.

Puede personalizar y configurar el software del que depende la aplicación en la plataforma. Puede obtener más información en [Personalización de software en servidores Linux](#) y [Personalización de software en servidores Windows](#). Las notas de la versión detalladas de los últimos lanzamientos están disponibles en [Notas de la versión de AWS Elastic Beanstalk](#).

Plataformas admitidas

La guía de AWS Elastic Beanstalk plataformas muestra todas las versiones actuales de las ramas de plataforma en la sección Plataformas compatibles con [Elastic Beanstalk](#). La guía de plataformas también incluye un historial de plataformas para cada plataforma, que incluye una lista de las

versiones anteriores de las plataformas de sucursal. Para ver el historial de cada plataforma, selecciona uno de los siguientes enlaces.

- [Docker](#)
- [Go](#)
- [Java SE](#)
- [Tomcat \(con Java SE\)](#)
- [.NET Core en Linux](#)
- [.NET en Windows Server](#)
- [Node.js](#)
- [PHP](#)
- [Python](#)
- [Ruby](#)

El nombre de la pila de soluciones de una rama de plataforma

[Puede usar el nombre de la pila de soluciones para una versión de rama de plataforma determinada para lanzar un entorno con la CLI de EB, la API de Elastic Beanstalk o la CLI.AWS](#) La guía de AWS Elastic Beanstalk plataformas incluye el nombre de la pila de soluciones en la versión de la rama de plataforma en las secciones [Plataformas compatibles de Elastic Beanstalk](#) e [Historial de plataformas](#).

Para recuperar todos los nombres de las pilas de soluciones que puede usar para crear un entorno, utilice la [ListAvailableSolutionStacksAPI](#) o [aws elasticbeanstalk list-available-solution-stacks](#) la AWS CLI.

Plataformas Linux de Elastic Beanstalk

La mayoría de las plataformas que admite Elastic Beanstalk están basadas en el sistema operativo Linux. En concreto, estas plataformas se basan en Amazon Linux, una distribución de Linux proporcionada por AWS. Las plataformas Linux de Elastic Beanstalk utilizan instancias de Amazon Elastic Compute Cloud (Amazon EC2) y estas instancias ejecutan Amazon Linux.

Las plataformas de Elastic Beanstalk Linux ofrecen una gran cantidad de funcionalidades para usar. Puede ampliar las plataformas de diferentes maneras para que admitan su aplicación. Para obtener más detalles, consulte [the section called “Ampliación de plataformas Linux”](#).

Temas

- [Versiones de Amazon Linux compatibles](#)
- [Lista de plataformas Linux de Elastic Beanstalk](#)
- [Ampliación de las plataformas Linux de Elastic Beanstalk](#)

Versiones de Amazon Linux compatibles

AWS Elastic Beanstalk admite plataformas basadas en Amazon Linux 2 y Amazon Linux 2023.

A partir del [19 de octubre de 2023](#), Elastic Beanstalk ofrece las plataformas AL2023 para todos los lenguajes de programación que también son compatibles con las plataformas Amazon Linux 2. Beanstalk también es compatible con las plataformas Docker y Docker basado en ECS tanto en Amazon Linux 2 como en Amazon Linux 2023.

Para obtener más información acerca de Amazon Linux 2 y Amazon Linux 2023, consulte lo siguiente:

- Amazon Linux 2 — [Amazon Linux](#) en la guía del usuario de Amazon EC2.
- Amazon Linux 2023: [¿Qué es Amazon Linux 2023?](#) en la Guía del usuario de Amazon Linux 2023.

Para obtener más información sobre las versiones de plataforma admitidas actualmente, consulte [Plataformas compatibles con Elastic Beanstalk](#).

Note

Puede migrar su aplicación desde una ramificación de la plataforma AL1 o AL2 de Elastic Beanstalk a la ramificación de la plataforma AL2023 equivalente. Para obtener más información, consulte [Migración de su aplicación de Linux de Elastic Beanstalk a Amazon Linux 2023 o Amazon Linux 2](#).

Amazon Linux 2023

AWS anunció la [disponibilidad general](#) de Amazon Linux 2023 en marzo de 2023. La Guía del usuario de Linux Amazon 2023 ofrece un resumen de las diferencias clave ente Amazon Linux 2 y Amazon Linux 2023. Para obtener más información, consulte [Comparación de Amazon Linux 2 y Amazon Linux 2023](#) en la guía del usuario.

Existe un alto grado de compatibilidad entre las plataformas Amazon Linux 2 y Amazon Linux 2023 de Elastic Beanstalk. Sin embargo, hay algunas diferencias a destacar:

- Instance Metadata Service Version 1 (IMDSv1): La configuración de la opción [DisableIMDSv1](#) establece `true` como valor predeterminado en las plataformas AL2023. El valor predeterminado es `false` en las plataformas AL2.
- Herramienta de instancia pkg-repo: la herramienta [pkg-repo](#) no está disponible para entornos que se ejecutan en plataformas AL2023. Sin embargo, puede aplicar manualmente las actualizaciones del paquete y del sistema operativo a una instancia de AL2023. Para obtener más información, consulte [Administración de paquetes y actualizaciones del sistema operativo](#) en la Guía del usuario de Amazon Linux 2023
- Configuración HTTPd de Apache: el archivo `httpd.conf` de Apache para las plataformas AL2023 tiene algunos ajustes de configuración diferentes a los de AL2:
 - Denegar el acceso a todo el sistema de archivos del servidor de forma predeterminada. Esta configuración se describe en Proteger los archivos del servidor de forma predeterminada en la página [Consejos de seguridad](#) del sitio web de Apache.
 - Impedir que los usuarios anulen las funciones de seguridad que se hayan configurado. La configuración deniega el acceso a la configuración `.htaccess` en todos los directorios, excepto en los habilitados específicamente. Esta configuración se describe en Proteger la configuración del sistema en la página [Consejos de seguridad](#) del sitio web de Apache. La página [Tutorial del servidor HTTP Apache: archivos.htaccess](#) indica que esta configuración puede ayudar a mejorar el rendimiento.
 - Denegar el acceso a los archivos con `.ht*` como patrón de nombre. Esta configuración impide que los clientes web visualicen los archivos `.htaccess` y `.htpasswd`.

Puede cambiar cualquiera de las opciones de configuración de entorno anteriores. Para obtener más información, consulte [Ampliación de las plataformas Linux de Elastic Beanstalk](#). Amplíe el tema Proxy inverso para ver la sección Configuración de HTTPD de Apache.

Lista de plataformas Linux de Elastic Beanstalk

La siguiente lista menciona las plataformas de Linux que Elastic Beanstalk admite para diferentes lenguajes de programación y para contenedores de Docker. Elastic Beanstalk ofrece las plataformas basadas en Amazon Linux 2 y Amazon Linux 2023 para todos ellos. Para obtener más información sobre una plataforma, seleccione el enlace correspondiente.

- [Docker \(y Docker basado en ECS\)](#)
- [Go](#)
- [Tomcat \(con Java SE\)](#)
- [Java SE](#)
- [.NET Core en Linux](#)
- [Node.js](#)
- [PHP](#)
- [Python](#)
- [Ruby](#)

Ampliación de las plataformas Linux de Elastic Beanstalk

Las [plataformas Linux de AWS Elastic Beanstalk](#) cuentan con muchas funcionalidades listas para usar diseñadas para desarrollar y ejecutar la aplicación. Si es necesario, puede ampliar las plataformas de diferentes maneras para configurar opciones, instalar software, agregar archivos y comandos de inicio, proporcionar instrucciones de compilación y ejecución, e incorporar scripts de inicialización que se ejecuten en varias etapas de aprovisionamiento de instancias de Amazon Elastic Compute Cloud (Amazon EC2) del entorno.

Buildfile y Procfile

Algunas plataformas permiten personalizar el modo en que se crea o prepara la aplicación y permiten especificar los procesos que ejecutan la aplicación. Cada tema individual de la plataforma menciona específicamente Buildfile y/o Procfile si la plataforma es compatible. Busque su plataforma específica en [Plataformas](#).

Para todas las plataformas de soporte, la sintaxis y la semántica son idénticas, y son como se describe en esta página. Los temas individuales de la plataforma mencionan el uso específico de estos archivos para crear y ejecutar aplicaciones en sus respectivos idiomas.

Buildfile

Si desea especificar un comando de compilación y configuración personalizado para la aplicación, coloque un archivo llamado `Buildfile` en el directorio raíz del origen de la aplicación. El nombre de archivo distingue entre mayúsculas y minúsculas. Utilice la siguiente sintaxis con el archivo `Buildfile`.

```
<process_name>: <command>
```

El comando del archivo `Buildfile` debe ajustarse a la siguiente expresión regular: `^[A-Za-z0-9_-]+:\s*[\s]*.*$`.

Elastic Beanstalk no monitoriza la aplicación que se ejecuta con un `Buildfile`. Utilice un archivo `Buildfile` con comandos que se ejecuten durante cortos periodos de tiempo y terminen después de completar las tareas. En el caso de los procesos de larga duración de la aplicación que no deben terminar, utilice en su lugar el archivo [Procfile](#).

Todas las rutas de `Buildfile` son rutas relativas que hacen referencia a la raíz del paquete de código fuente. En el siguiente ejemplo de `Buildfile`, `build.sh` es un script de shell que se encuentra en la raíz del paquete de código fuente.

Example Buildfile

```
make: ./build.sh
```

Si desea incluir pasos de compilación personalizados, le recomendamos que, en lugar de `Buildfile`, utilice enlaces de plataforma `predeploy` para todo excepto para los comandos más simples. Los enlaces de plataforma admiten secuencias de comandos más elaboradas y posibilitan una mejor administración de los errores. Los enlaces de plataforma se describen en la siguiente sección.

Procfile

Si desea especificar comandos personalizados para iniciar y ejecutar la aplicación, coloque un archivo llamado `Procfile` en el directorio raíz del código fuente de la aplicación. El nombre de archivo distingue entre mayúsculas y minúsculas. Utilice la siguiente sintaxis con el archivo `Procfile`. Puede especificar uno o varios comandos.

```
<process_name1>: <command1>  
<process_name2>: <command2>  
...
```

Cada línea del archivo `Procfile` debe ajustarse a la siguiente expresión regular: `^[A-Za-z0-9_-]+:\s*[\s]*.*$`.

Utilice un archivo `Procfile` para procesos de aplicación de larga ejecución que no deberían salir. Elastic Beanstalk espera que los procesos del archivo `Procfile` se ejecuten de forma continua.

Elastic Beanstalk monitoriza estos procesos y reinicia cualquier proceso que finalice. Para los procesos de corta ejecución, utilice [Buildfile](#).

Todas las rutas de `Procfile` son rutas relativas que hacen referencia a la raíz del paquete de código fuente. En el siguiente ejemplo, `Procfile` define tres procesos. El primero, llamado `web` en el ejemplo, es la aplicación web principal.

Example Procfile

```
web: bin/myserver  
cache: bin/mycache  
foo: bin/fooapp
```

Elastic Beanstalk configura el servidor proxy para reenviar solicitudes a su aplicación web principal en el puerto 5000 y puede configurar este número de puerto. Un uso habitual de `Procfile` es pasar este número de puerto a su aplicación como un argumento de la línea de comandos. Para obtener más información sobre la configuración del proxy, expanda la sección Configuración del proxy inverso que se encuentra en esta página.

Elastic Beanstalk captura flujos de error y salida estándar de procesos del archivo `Procfile` en archivos de registro. Elastic Beanstalk nombra los archivos de registro después del proceso y los almacena en `/var/log`. Por ejemplo, el proceso `web` del ejemplo anterior genera los logs llamados `web-1.log` y `web-1.error.log` para `stdout` y `stderr`, respectivamente.

Enlaces de la plataforma

Los enlaces de la plataforma están diseñados específicamente para ampliar la plataforma del entorno. Se trata de scripts personalizados y otros archivos ejecutables que se implementan como parte del código fuente de la aplicación y que Elastic Beanstalk ejecuta durante diversas etapas de aprovisionamiento de instancias.

Note

Los enlaces de la plataforma no son compatibles con las versiones de la plataforma de la AMI de Amazon Linux (anteriores a Amazon Linux 2).

Enlaces de la plataforma de implementación de aplicaciones

Una implementación de aplicación se produce cuando se proporciona un nuevo paquete de origen para la implementación o cuando se realiza un cambio de configuración que requiere la terminación y recreación de todas las instancias del entorno.

Para proporcionar enlaces de la plataforma que se ejecutan durante una implementación de aplicaciones, coloque los archivos en el directorio `.platform/hooks` del paquete de origen, en uno de los subdirectorios siguientes.

- `prebuild`: los archivos de este subdirectorio se ejecutan después de que el motor de la plataforma Elastic Beanstalk se descarga y extrae el paquete de código fuente de la aplicación y antes de que configure la aplicación y el servidor web.

Los archivos de `prebuild` se ejecutan después de los comandos que se encuentran en la sección [commands](#) de un archivo de configuración y antes de los comandos de `Buildfile`.

- `predeploy`: los archivos de este subdirectorio se ejecutan después de que el motor de la plataforma Elastic Beanstalk se configura y prepara la aplicación y el servidor web, y antes de que los implemente en la ubicación del entorno de ejecución final.

Los archivos de `predeploy` se ejecutan después de los comandos que se encuentran en la sección [container_commands](#) de un archivo de configuración y antes de que se ejecuten los comandos de `Procfile`.

- `postdeploy`: los archivos de este subdirectorio se ejecutan después de que el motor de la plataforma Elastic Beanstalk implementa la aplicación y el servidor proxy.

Este es el último paso del flujo de trabajo de implementación.

Enlaces de la plataforma de implementación de configuración

Una implementación de configuración se produce cuando se realizan cambios de configuración que solo actualizan las instancias del entorno sin volver a crearlas. Las siguientes actualizaciones de opciones provocan una actualización de configuración.

- [Propiedades del entorno y configuración específica de la plataforma](#)
- [Archivos estáticos](#)
- [Daemon de AWS X-Ray](#)
- [Almacenamiento y streaming de registros](#)

- Puerto de aplicación (para obtener más información, amplíe la sección Configuración de proxy inverso en esta página)

Para proporcionar enlaces que se ejecutan durante una implementación de configuración, colóquelos en el directorio `.platform/confighooks` del paquete de origen. Se aplican los mismos tres subdirectorios que para los enlaces de implementación de aplicaciones.

Más información sobre enlaces de plataforma

Los archivos de enlace pueden ser archivos binarios o archivos de script que comienzan con una línea `#!` que contiene la ruta del intérprete; por ejemplo `#!/bin/bash`. Todos los archivos deben tener permiso de ejecución. Utilice `chmod +x` para configurar el permiso de ejecución de los archivos de enlace. En todas las versiones de la plataforma basadas en Amazon Linux 2023 y Amazon Linux 2 que se lanzaron el 29 de abril de 2022 o en fecha posterior, Elastic Beanstalk concede automáticamente permisos de ejecución a todos los scripts de enlace de plataforma. En este caso, no tiene que otorgar permisos de ejecución manualmente. Para obtener una lista de estas versiones de plataforma, consulte las notas de la versión de Linux del [29 de abril de 2022](#) en la Guía de notas de la versión de AWS Elastic Beanstalk.

Elastic Beanstalk ejecuta los archivos de cada uno de estos directorios en orden lexicográfico en función del nombre de archivo. Todos los archivos se ejecutan con el usuario `root`. El directorio de trabajo actual (`cwd`) de los enlaces de plataforma es el directorio raíz de la aplicación. En los archivos `prebuild` y `predeploy`, es el directorio de almacenamiento provisional de la aplicación, mientras que en los archivos `postdeploy` es el directorio actual de la aplicación. Si se producen errores en uno de los archivos (sale con un código de salida distinto de cero), la implementación se cancela y no se ejecuta correctamente.

El script de texto de enlace de plataforma puede fallar si contiene caracteres de salto de línea Retorno de carro/Avance de línea (CRLF, por sus siglas en inglés) de Windows. Si un archivo se guardó en un host de Windows y luego se transfirió a un servidor de Linux, puede contener saltos de línea CRLF de Windows. Para las plataformas lanzadas a partir del [29 de diciembre de 2022](#), Elastic Beanstalk convierte de forma automática los CRLF de Windows en caracteres de avance de línea (LF) de Linux en los archivos de texto de enlace de plataforma. Si la aplicación se ejecuta en cualquier plataforma de Amazon Linux 2 que se haya lanzado antes de esta fecha, tendrá que convertir los caracteres CRLF de Windows en caracteres LF de Linux. Un modo de lograrlo es crear y guardar el archivo de script en un host de Linux. Las herramientas que convierten estos caracteres también están disponibles en Internet.

Los archivos de enlace tienen acceso a todas las propiedades de entorno definidas en las opciones de la aplicación y a las variables de entorno del sistema HOME, PATH y PORT.

Para obtener valores de variables de entorno y otras opciones de configuración en los scripts de enlace de plataforma, puede utilizar la utilidad `get-config` que Elastic Beanstalk proporciona en las instancias de entorno. Para obtener más información, consulte [the section called “Herramientas de script de plataforma”](#).

Archivos de configuración

Puede añadir [archivos de configuración](#) al directorio `.ebextensions` del código fuente de la aplicación para configurar diferentes aspectos del entorno de Elastic Beanstalk. Entre otras cosas, los archivos de configuración le permiten personalizar el software y otros archivos de las instancias del entorno, así como ejecutar comandos de inicialización en las instancias. Para obtener más información, consulte [the section called “Servidor Linux”](#).

También puede definir [opciones de configuración](#) utilizando archivos de configuración. Muchas de las opciones controlan el comportamiento de la plataforma y algunas de estas opciones son [específicas de la plataforma](#).

En las plataformas basadas en Amazon Linux 2 y Amazon Linux 2023, se recomienda utilizar Buildfile, Procfile y enlaces de plataforma para configurar y ejecutar código personalizado en las instancias de su entorno durante el aprovisionamiento de instancias. Estos mecanismos se describen en las secciones anteriores de esta página. Puede seguir utilizando comandos y comandos de contenedor en los archivos de configuración `.ebextensions`, pero no resulta fácil trabajar con ellos. Por ejemplo, puede resultar complicado escribir scripts de comandos en un archivo YAML desde un punto de vista sintáctico. Tendrá que seguir utilizando los archivos de configuración `.ebextensions` con cualquier script que necesite una referencia a un recurso de AWS CloudFormation.

Configuración del proxy inverso

Todas las versiones de la plataforma Amazon Linux 2 y Amazon Linux 2023 utilizan nginx como servidor proxy inverso predeterminado. Las plataformas Tomcat, Node.js, PHP y Python también admiten Apache HTTPD como alternativa. Para seleccionar Apache en estas plataformas, establezca la opción `ProxyServer` en el espacio de nombres `aws:elasticbeanstalk:environment:proxy` en `apache`. Todas las plataformas habilitan la configuración del servidor proxy de manera uniforme, como se describe en esta sección.

Note

En las versiones de la plataforma AMI de Amazon Linux (antes de Amazon Linux 2), es posible que tenga que configurar servidores proxy de manera diferente. Puede encontrar estos detalles heredados en los [temas respectivos de la plataforma](#) en esta guía.

Elastic Beanstalk configura el servidor proxy de las instancias del entorno para que reenvíen el tráfico web a la aplicación web principal de la URL raíz del entorno; por ejemplo, `http://my-env.elasticbeanstalk.com`.

De forma predeterminada, Elastic Beanstalk configura el proxy para que reenvíe las solicitudes entrantes en el puerto 80 a la aplicación web principal en el puerto 5000. Puede configurar este número de puerto con la propiedad de entorno `PORT` utilizando el espacio de nombres [aws:elasticbeanstalk:application:environment](#) en un archivo de configuración, tal y como se muestra en el ejemplo siguiente.

```
option_settings:
  - namespace: aws:elasticbeanstalk:application:environment
    option_name: PORT
    value: <main_port_number>
```

Para obtener más información sobre cómo configurar las variables de la aplicación, consulte [the section called “Opciones de configuración”](#).

La aplicación debe escuchar el puerto que esté configurado para ella en el proxy. Si cambia el puerto predeterminado utilizando la propiedad de entorno `PORT`, el código podrá acceder a él a través del valor de la variable de entorno `PORT`. Por ejemplo, llame a `os.Getenv("PORT")` en Go o `System.getenv("PORT")` en Java. Si configura el proxy para que envíe tráfico a diferentes procesos de la aplicación, puede configurar varias propiedades de entorno y utilizar sus valores tanto en la configuración del proxy como en el código de la aplicación. Otra opción es pasar el valor del puerto al proceso como un argumento de comando en el archivo `Procfile`. Para obtener detalles al respecto, expanda la sección `Buildfile` y `Procfile` en esta página.

Configuración de nginx

Elastic Beanstalk utiliza nginx como proxy inverso predeterminado para asignar la aplicación al balanceador de carga de Elastic Load Balancing. Elastic Beanstalk proporciona una configuración de nginx predeterminada que puede ampliar o invalidar completamente con su propia configuración.

Note

Cuando agregue o edite un archivo de configuración `.conf` de nginx, asegúrese de codificarlo como UTF-8.

Para ampliar la configuración predeterminada de nginx de Elastic Beanstalk, añada los archivos de configuración `.conf` en una carpeta llamada `.platform/nginx/conf.d/` del paquete de código fuente de su aplicación. La configuración de nginx de Elastic Beanstalk incluye archivos `.conf` en esta carpeta automáticamente.

```
~/workspace/my-app/  
|-- .platform  
|   |-- nginx  
|       |-- conf.d  
|           |-- myconf.conf  
|-- other source files
```

Para anular completamente la configuración predeterminada de nginx de Elastic Beanstalk, incluya una configuración en el paquete de código fuente de `.platform/nginx/nginx.conf`:

```
~/workspace/my-app/  
|-- .platform  
|   |-- nginx  
|       |-- nginx.conf  
|-- other source files
```

Si invalida la configuración de nginx de Elastic Beanstalk, agregue la siguiente línea al archivo `nginx.conf` para extraer las configuraciones de Elastic Beanstalk de [Informes y monitorización de estado mejorados](#), los mapeos automáticos de la aplicación y los archivos estáticos.

```
include conf.d/elasticbeanstalk/*.conf;
```

Configuración de Apache HTTPD

Las plataformas Tomcat, Node.js, PHP y Python le permiten elegir el servidor proxy Apache HTTPD como alternativa a nginx. Este no es el valor predeterminado. En el siguiente ejemplo se configura Elastic Beanstalk para que utilice Apache HTTPD.

Example .ebextensions/httpd-proxy.config

```
option_settings:
  aws:elasticbeanstalk:environment:proxy:
    ProxyServer: apache
```

Puede ampliar la configuración de Apache predeterminada de Elastic Beanstalk con archivos de configuración adicionales. Si lo prefiere, puede anular por completo la configuración de Apache predeterminada de Elastic Beanstalk.

Para ampliar la configuración de Apache predeterminada de Elastic Beanstalk, añada los archivos de configuración `.conf` a una carpeta con el nombre `.platform/httpd/conf.d` en el paquete de código fuente de su aplicación. La configuración de Apache de Elastic Beanstalk incluye archivos `.conf` en esta carpeta automáticamente.

```
~/workspace/my-app/
|-- .ebextensions
|   -- httpd-proxy.config
|-- .platform
|   -- httpd
|       -- conf.d
|           -- port5000.conf
|           -- ssl.conf
-- index.jsp
```

Por ejemplo, la siguiente configuración de Apache 2.4 agrega un agente de escucha en el puerto 5000.

Example .platform/httpd/conf.d/port5000.conf

```
listen 5000
<VirtualHost *:5000>
  <Proxy *>
    Require all granted
  </Proxy>
  ProxyPass / http://localhost:8080/ retry=0
  ProxyPassReverse / http://localhost:8080/
  ProxyPreserveHost on

  ErrorLog /var/log/httpd/elasticbeanstalk-error_log
</VirtualHost>
```

Para anular por completo la configuración de Apache predeterminada de Elastic Beanstalk, incluya una configuración en el paquete de código fuente en `.platform/httpd/conf/httpd.conf`.

```
~/workspace/my-app/  
|-- .ebextensions  
|   -- httpd-proxy.config  
|-- .platform  
|   `-- httpd  
|       `-- conf  
|           `-- httpd.conf  
`-- index.jsp
```

Note

Si invalida la configuración de Apache de Elastic Beanstalk, agregue la siguiente línea al archivo `httpd.conf` para extraer las configuraciones de Elastic Beanstalk de [Informes y monitorización de estado mejorados](#), los mapeos automáticos de la aplicación y los archivos estáticos.

```
IncludeOptional conf.d/elasticbeanstalk/*.conf
```

Note

Si va a migrar su aplicación de Elastic Beanstalk a una plataforma Amazon Linux 2 o Amazon Linux 2023, asegúrese de leer también la información de [the section called “Migración a AL2023/AL2”](#).

Temas

- [Ejemplo de aplicación con extensiones](#)
- [Flujo de trabajo de implementación de instancia](#)
- [Flujo de trabajo de implementación de instancias para ECS sobre Amazon Linux 2 y posterior](#)
- [Herramientas de script de plataforma](#)

Ejemplo de aplicación con extensiones

En el siguiente ejemplo, se muestra un paquete de código fuente de aplicaciones con varias características de extensibilidad con las que son compatibles las plataformas Amazon Linux 2 y Amazon Linux 2023 de Elastic Beanstalk: un archivo Procfile, archivos de configuración .ebextensions, enlaces personalizados y archivos de configuración de proxy.

```
~/my-app/
|-- web.jar
|-- Procfile
|-- readme.md
|-- .ebextensions/
|   |-- options.config          # Option settings
|   `-- cloudwatch.config      # Other .ebextensions sections, for example files and
    container commands
`-- .platform/
    |-- nginx/                  # Proxy configuration
    |   |-- nginx.conf
    |   `-- conf.d/
    |       `-- custom.conf
    |-- hooks/                  # Application deployment hooks
    |   |-- prebuild/
    |   |   |-- 01_set_secrets.sh
    |   |   `-- 12_update_permissions.sh
    |   |-- predeploy/
    |   |   `-- 01_some_service_stop.sh
    |   `-- postdeploy/
    |       |-- 01_set_tmp_file_permissions.sh
    |       |-- 50_run_something_after_app_deployment.sh
    |       `-- 99_some_service_start.sh
    `-- confighooks/           # Configuration deployment hooks
        |-- prebuild/
        |   `-- 01_set_secrets.sh
        |-- predeploy/
        |   `-- 01_some_service_stop.sh
        `-- postdeploy/
            |-- 01_run_something_after_config_deployment.sh
            `-- 99_some_service_start.sh
```

Note

Algunas de estas extensiones no son compatibles con las versiones de la plataforma de la AMI de Amazon Linux (antes de Amazon Linux 2).

Flujo de trabajo de implementación de instancia

Note

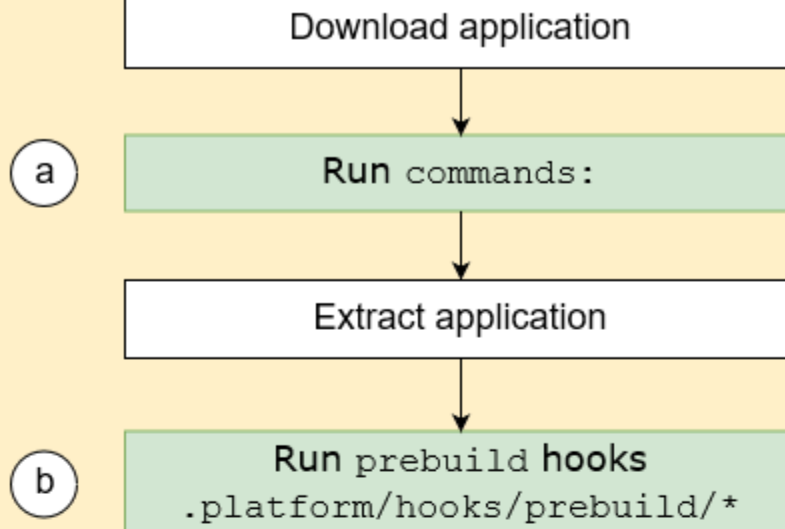
La información de esta sección no se aplica a la ramificación de la plataforma ECS sobre Amazon Linux 2 y Amazon Linux 2023. Para obtener más información, consulte la siguiente sección, [Flujo de trabajo de implementación de instancias para ECS sobre Amazon Linux 2 y posterior](#).

Dado que hay muchos mecanismos para ampliar la plataforma el entorno, resulta útil saber qué sucede cuando Elastic Beanstalk aprovisiona una instancia o ejecuta una implementación en una instancia. En el diagrama siguiente, se muestra todo este flujo de trabajo de implementación. En él se representan las diferentes fases de una implementación y los pasos que Elastic Beanstalk realiza en cada fase.

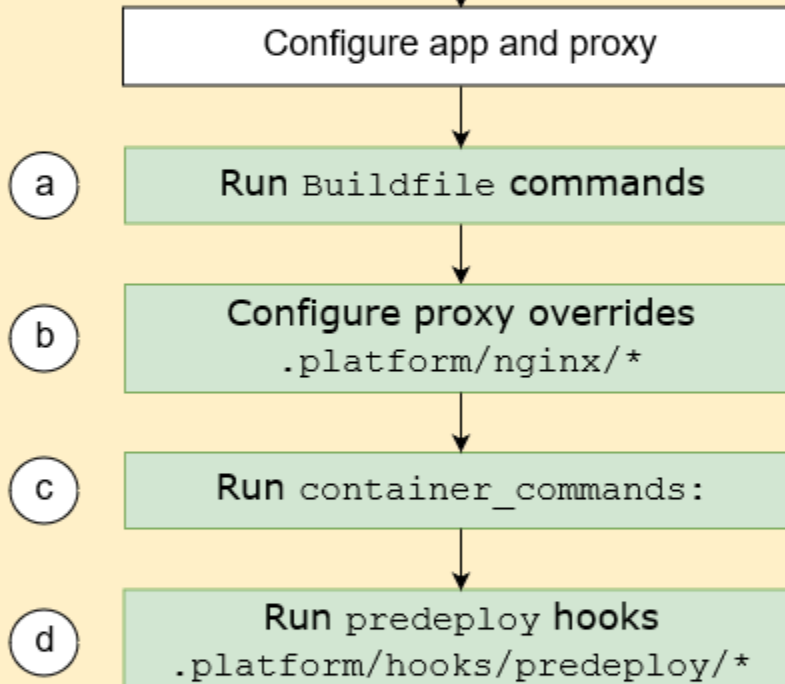
Notas

- El diagrama no representa todos los pasos que Elastic Beanstalk lleva a cabo en las instancias del entorno durante la implementación. El objetivo de este diagrama es meramente ilustrativo y pretende proporcionarle el orden y el contexto de la ejecución de las personalizaciones.
- Para simplificar, el diagrama menciona solo los subdirectorios de enlace `.platform/hooks/*` (para implementaciones de aplicaciones) y no los subdirectorios de enlace de `.platform/confighooks/*` (para implementaciones de configuración). Los enlaces de estos últimos subdirectorios se ejecutan exactamente en los mismos pasos que los enlaces en los subdirectorios correspondientes que se muestran en el diagrama.

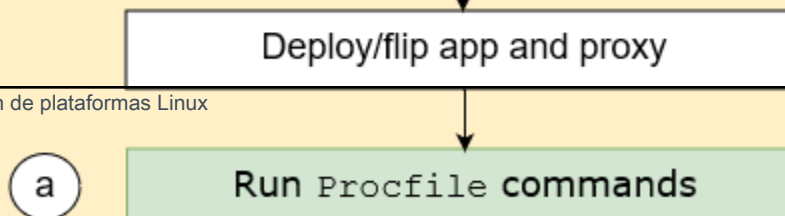
1. Initial steps



2. Configure



3. Deploy



En la lista siguiente, se detallan las fases y los pasos de la implementación.

1. Pasos iniciales

Elastic Beanstalk descarga y extrae su aplicación. Después de cada uno de estos pasos, Elastic Beanstalk ejecuta uno de los pasos de extensibilidad.

- a. Ejecuta los comandos que se encuentran en la sección [commands:](#) de los archivos de configuración.
- b. Ejecuta cualquier archivo ejecutable que se encuentre en el directorio `.platform/confighooks/prebuild` del paquete de origen (`.platform/hooks/prebuild` para una implementación de configuración).

2. Configuración

Elastic Beanstalk configura la aplicación y el servidor proxy.

- a. Ejecuta los comandos que se encuentran en el directorio `Buildfile` del paquete de código fuente.
- b. Copia los archivos de configuración de proxy personalizados, si tiene alguno en el directorio `.platform/nginx` del paquete de código fuente, en la ubicación del entorno de ejecución.
- c. Ejecuta los comandos que se encuentran en la sección [container_commands:](#) de los archivos de configuración.
- d. Ejecuta cualquier archivo ejecutable que se encuentre en el directorio `.platform/confighooks/predeploy` del paquete de origen (`.platform/hooks/predeploy` para una implementación de configuración).

3. implementación

Elastic Beanstalk implementa y ejecuta la aplicación y el servidor proxy.

- a. Ejecuta el comando que se encuentra en el archivo `Procfile` del paquete de código fuente.
- b. Ejecuta o vuelve a ejecutar el servidor proxy con los archivos de configuración proxy personalizados, si los hay.
- c. Ejecuta cualquier archivo ejecutable que se encuentre en el directorio `.platform/confighooks/postdeploy` del paquete de origen (`.platform/hooks/postdeploy` para una implementación de configuración).

Flujo de trabajo de implementación de instancias para ECS sobre Amazon Linux 2 y posterior

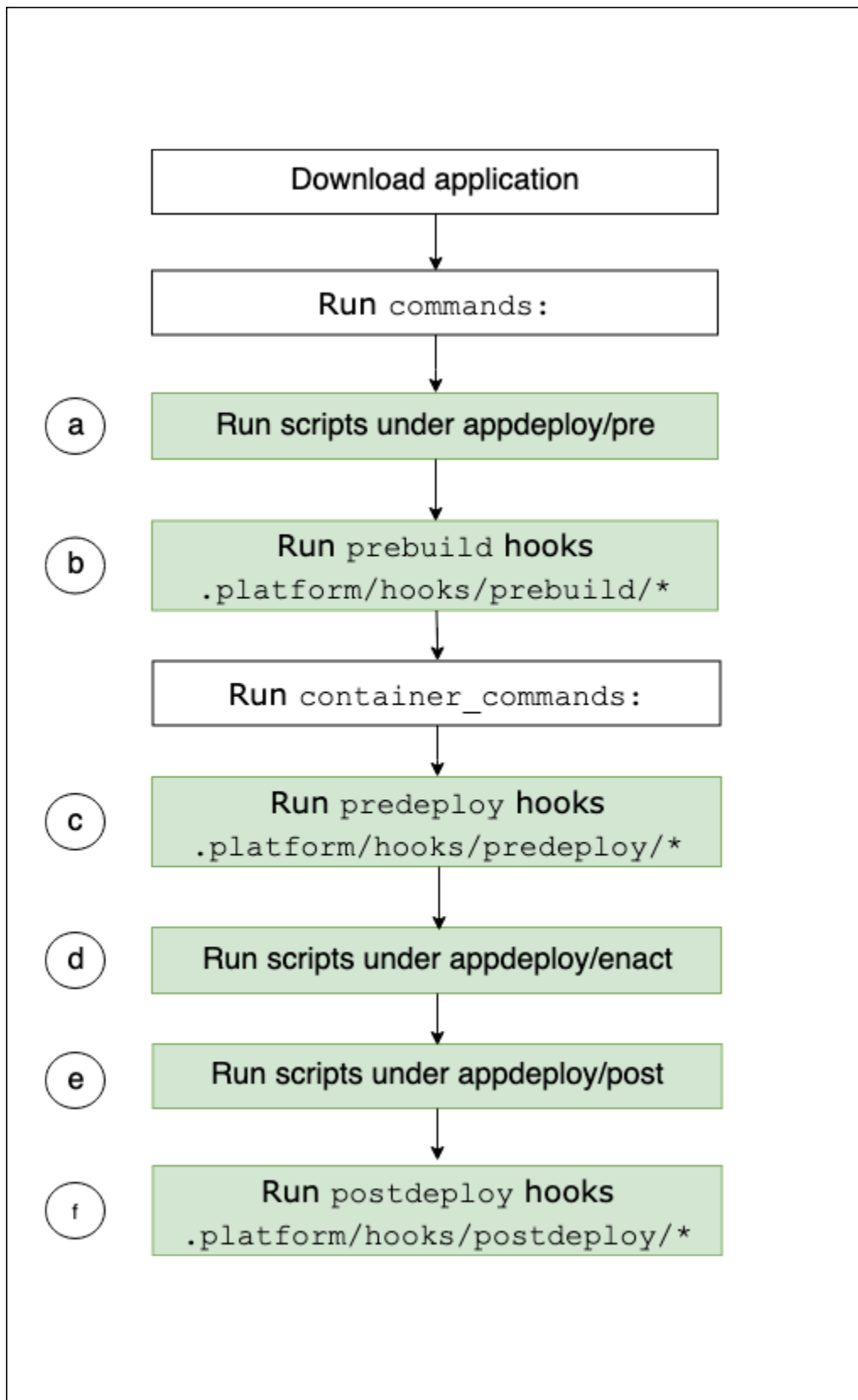
En la sección anterior se describen las características de extensibilidad que se admiten a lo largo de las fases del flujo de trabajo de la implementación de la aplicación. Existen algunas diferencias para las ramificaciones de la plataforma Docker de [ECS sobre Amazon Linux 2 y posterior](#). En esta sección se explica cómo se aplican esos conceptos a esta ramificación de la plataforma específica.

Dado que hay muchos mecanismos para ampliar la plataforma el entorno, resulta útil saber qué sucede cuando Elastic Beanstalk aprovisiona una instancia o ejecuta una implementación en una instancia. En el siguiente diagrama, se muestra todo este flujo de trabajo de implementación para un entorno basado en las ramificaciones de la plataforma ECS sobre Amazon Linux 2 y ECS sobre Amazon Linux 2023. En él se representan las diferentes fases de una implementación y los pasos que Elastic Beanstalk realiza en cada fase.

A diferencia del flujo de trabajo descrito en la sección anterior, la fase de configuración de la implementación no admite las siguientes características de extensibilidad: comandos `Buildfile`, comandos `Profile`, configuración del proxy inverso.

Notas

- El diagrama no representa todos los pasos que Elastic Beanstalk lleva a cabo en las instancias del entorno durante la implementación. El objetivo de este diagrama es meramente ilustrativo y pretende proporcionarle el orden y el contexto de la ejecución de las personalizaciones.
- Para simplificar, el diagrama menciona solo los subdirectorios de enlace `.platform/hooks/*` (para implementaciones de aplicaciones) y no los subdirectorios de enlace de `.platform/confighooks/*` (para implementaciones de configuración). Los enlaces de estos últimos subdirectorios se ejecutan exactamente en los mismos pasos que los enlaces en los subdirectorios correspondientes que se muestran en el diagrama.



En la siguiente lista, se detallan los pasos del flujo de trabajo de implementación.

- a. Ejecuta los archivos ejecutables que se encuentran en el directorio `appdeploy/pre` en `EBhooksDir`.
- b. Ejecuta cualquier archivo ejecutable que se encuentre en el directorio `.platform/fighooks/prebuild` del paquete de origen (`.platform/hooks/prebuild` para una implementación de configuración).
- c. Ejecuta cualquier archivo ejecutable que se encuentre en el directorio `.platform/fighooks/predeploy` del paquete de origen (`.platform/hooks/predeploy` para una implementación de configuración).
- d. Ejecuta los archivos ejecutables que se encuentran en el directorio `appdeploy/enact` en `EBhooksDir`.
- e. Ejecuta los archivos ejecutables que se encuentran en el directorio `appdeploy/post` en `EBhooksDir`.
- f. Ejecuta cualquier archivo ejecutable que se encuentre en el directorio `.platform/fighooks/postdeploy` del paquete de origen (`.platform/hooks/postdeploy` para una implementación de configuración).

La referencia a `EBhooksDir` representa la ruta del directorio de enlaces de la plataforma. Para recuperar el nombre de la ruta de directorio, utilice la herramienta de script [get-config](#) en la línea de comandos de la instancia del entorno como se muestra a continuación:

```
$ /opt/elasticbeanstalk/bin/get-config platformconfig -k EBhooksDir
```

Herramientas de script de plataforma

En este tema se describen las herramientas que se AWS Elastic Beanstalk proporcionan para los entornos que utilizan las plataformas Amazon Linux. Las herramientas se encuentran en las instancias de Amazon EC2 de los entornos de Elastic Beanstalk.

get-config

Utilice la herramienta `get-config` para recuperar valores de variables de entorno y otra información de plataformas e instancias. La herramienta está disponible en `/opt/elasticbeanstalk/bin/get-config`.

comandos get-config

Cada comando de herramienta de `get-config` devuelve un tipo específico de información. Utilice la siguiente sintaxis para ejecutar los comandos de cualquiera de las herramientas.

```
$ /opt/elasticbeanstalk/bin/get-config command [ options ]
```

En el siguiente ejemplo se ejecuta el comando `environment`.

```
$ /opt/elasticbeanstalk/bin/get-config environment -k PORT
```

En función del comando y las opciones que elija, la herramienta devuelve un objeto (JSON o YAML) con pares de valor de clave o un valor único.

Puede probar `get-config` mediante SSH para conectarse a una instancia de EC2 en su entorno de Elastic Beanstalk.

Note

Cuando se ejecuta `get-config` para pruebas, algunos comandos pueden requerir privilegios de usuario raíz para acceder a la información subyacente. Si obtiene un error de permiso de acceso, ejecute el comando de nuevo en `sudo`.

No es necesario agregar `sudo` cuando utiliza la herramienta en los scripts que implementa en su entorno. Elastic Beanstalk ejecuta todos sus scripts como usuario raíz.

En las siguientes secciones se describen los comandos de las herramientas.

`optionsettings`: opciones de configuración

El comando `get-config optionsettings` devuelve un objeto que enumera las opciones de configuración establecidas en el entorno y utilizadas por la plataforma en instancias de entorno. Se organizan por espacio de nombres.

```
$ /opt/elasticbeanstalk/bin/get-config optionsettings
{"aws:elasticbeanstalk:application:environment":
{"JDBC_CONNECTION_STRING":"","aws:elasticbeanstalk:container:tomcat:jvmoptions":{"JVM
Options":"","Xms":"256m","Xmx":"256m"},"aws:elasticbeanstalk:environment:proxy":
{"ProxyServer":"nginx","StaticFiles":
[""]},"aws:elasticbeanstalk:healthreporting:system":
```

```
{"SystemType":"enhanced"},"aws:elasticbeanstalk:hostmanager":  
{"LogPublicationControl":"false"}}
```

Para devolver un valor de opción de configuración específico, use la opción `--namespace (-n)` para especificar un espacio de nombres y la opción `--option-name (-o)` para especificar un nombre de opción.

```
$ /opt/elasticbeanstalk/bin/get-config optionsettings -  
n aws:elasticbeanstalk:container:php:phpini -o memory_limit  
256M
```

`environment`: propiedades del entorno

El comando `get-config environment` devuelve un objeto que contiene una lista de propiedades de entorno. Estas incluyen tanto las propiedades configuradas por el usuario como las proporcionadas por Elastic Beanstalk.

```
$ /opt/elasticbeanstalk/bin/get-config environment  
{"JDBC_CONNECTION_STRING":"","RDS_PORT":"3306","RDS_HOSTNAME":"anj9aw1b0tbj6b.cijbpanmxz5u.us-  
west-2.rds.amazonaws.com","RDS_USERNAME":"testusername","RDS_DB_NAME":"ebdb","RDS_PASSWORD":"te
```

Por ejemplo, Elastic Beanstalk proporciona propiedades de entorno para conectarse a una instancia de base de datos de Amazon RDS integrada (como `RDS_HOSTNAME`). Estas propiedades de conexión RDS aparecen en la salida de `get-config environment`. No obstante, no aparecen en la salida de `get-config optionsettings`. El motivo es que no se establecieron en las opciones de configuración.

Para devolver una propiedad específica del entorno, utilice la opción `--key (-k)` para especificar la clave de una propiedad.

```
$ /opt/elasticbeanstalk/bin/get-config environment -k TESTPROPERTY  
testvalue
```

`container`: valores de configuración en instancia

El comando `get-config container` devuelve un objeto que enumera los valores de configuración del entorno y de la plataforma para las instancias de entorno.

En el siguiente ejemplo se muestra la salida del comando en un entorno Tomcat de Amazon Linux 2.

```
$ /opt/elasticbeanstalk/bin/get-config container
```

```
{
  "common_log_list": ["/var/log/eb-engine.log", "/var/log/eb-hooks.log"],
  "default_log_list": ["/var/log/nginx/access.log", "/var/log/nginx/error.log"],
  "environment_name": "myenv-1da84946",
  "instance_port": "80",
  "log_group_name_prefix": "aws/elasticbeanstalk",
  "proxy_server": "nginx",
  "static_files": [""],
  "xray_enabled": "false"
}
```

Para devolver el valor de una clave específica, utilice la opción `--key` (`-k`) para especificar la clave.

```
$ /opt/elasticbeanstalk/bin/get-config container -k environment_name
myenv-1da84946
```

`addons`: valores de configuración de complementos

El comando `get-config addons` devuelve un objeto que contiene información de configuración de complementos de entorno. Utilícelo para recuperar la configuración de una base de datos de Amazon RDS asociada al entorno.

```
$ /opt/elasticbeanstalk/bin/get-config addons
{"rds":{"Description":"RDS Environment variables","env":{"RDS_DB_NAME":"ebdb","RDS_HOSTNAME":"ea13k2wimu1dh8i.c18mnpu5rwwg.us-east-2.rds.amazonaws.com","RDS_PASSWORD":"password","RDS_PORT":"3306","RDS_USERNAME":"user"}}}
```

Puede restringir el resultado de dos maneras. Para recuperar valores de un complemento específico, utilice la opción `--add-on` (`-a`) para especificar el nombre del complemento.

```
$ /opt/elasticbeanstalk/bin/get-config addons -a rds
{"Description":"RDS Environment variables","env":{"RDS_DB_NAME":"ebdb","RDS_HOSTNAME":"ea13k2wimu1dh8i.c18mnpu5rwwg.us-east-2.rds.amazonaws.com","RDS_PASSWORD":"password","RDS_PORT":"3306","RDS_USERNAME":"user"}}}
```

Para devolver el valor de una clave específica dentro de un complemento, agregue la opción `--key` (`-k`) para especificar la clave.

```
$ /opt/elasticbeanstalk/bin/get-config addons -a rds -k RDS_DB_NAME
ebdb
```

`platformconfig`: valores de configuración constante

El comando `get-config platformconfig` devuelve un objeto que contiene información de configuración de plataforma que es constante a la versión de esta última. La salida es la misma en

todos los entornos que ejecutan la misma versión de plataforma. El objeto de salida del comando contiene dos objetos incrustados:

- **GeneralConfig**: contiene información que es constante en las versiones más recientes de todas las ramificaciones de la plataforma Amazon Linux 2 y Amazon Linux 2023.
- **PlatformSpecificConfig**: contiene información que es constante para la versión de la plataforma y que es específica de ella.

En el siguiente ejemplo se muestra la salida del comando en un entorno que utiliza la ramificación de la plataforma Tomcat 8.5 que ejecuta Corretto 11.

```
$ /opt/elasticbeanstalk/bin/get-config platformconfig
{"GeneralConfig":{"AppUser":"webapp","AppDeployDir":"/var/app/
current/","AppStagingDir":"/var/app/
staging/","ProxyServer":"nginx","DefaultInstancePort":"80"},"PlatformSpecificConfig":
{"ApplicationPort":"8080","JavaVersion":"11","TomcatVersion":"8.5"}}
```

Para devolver el valor de una clave específica, utilice la opción `--key (-k)` para especificar la clave. Estas claves son únicas en los dos objetos incrustados. No es necesario especificar el objeto que contiene la clave.

```
$ /opt/elasticbeanstalk/bin/get-config platformconfig -k AppStagingDir
/var/app/staging/
```

Opciones de salida de get-config

Utilice la opción `--output` para especificar el formato del objeto de salida. Los valores válidos son JSON (predeterminado) y YAML. Se trata de una opción general. Debe especificarla antes del nombre del comando.

En el siguiente ejemplo se devuelven los valores de las opciones de configuración en formato YAML.

```
$ /opt/elasticbeanstalk/bin/get-config --output YAML optionsettings
aws:elasticbeanstalk:application:environment:
  JDBC_CONNECTION_STRING: ""
aws:elasticbeanstalk:container:tomcat:jvmoptions:
  JVM Options: ""
  Xms: 256m
  Xmx: 256m
aws:elasticbeanstalk:environment:proxy:
```

```
ProxyServer: nginx
StaticFiles:
  - ""
aws:elasticbeanstalk:healthreporting:system:
  SystemType: enhanced
aws:elasticbeanstalk:hostmanager:
  LogPublicationControl: "false"
```

pkg-repo

Note

La herramienta `pkg-repo` no está disponible para entornos basados en las plataformas Amazon Linux 2023. Sin embargo, puede aplicar manualmente las actualizaciones del paquete y del sistema operativo a una instancia de AL2023. Para obtener más información, consulte [Administración de paquetes y actualizaciones del sistema operativo](#) en la Guía del usuario de Amazon Linux 2023

En algunos casos urgentes, es posible que tenga que actualizar sus instancias de Amazon EC2 mediante la aplicación de un parche de seguridad de Amazon Linux 2 que aún no se haya lanzado con las versiones necesarias de la plataforma de Elastic Beanstalk. No puede realizar una actualización manual en sus entornos de Elastic Beanstalk de forma predeterminada. Esto se debe a que las versiones de la plataforma se encuentran bloqueadas en una versión específica del repositorio de Amazon Linux 2. Con este bloqueo se garantiza que las instancias ejecuten versiones de software compatibles y coherentes. En casos urgentes, la herramienta `pkg-repo` permite una solución para actualizar manualmente los paquetes yum en Amazon Linux 2 si necesita instalarlo en un entorno antes de que se lance en una nueva versión de la plataforma de Elastic Beanstalk.

La herramienta `pkg-repo` en las plataformas de Amazon Linux 2 proporciona la capacidad de desbloquear los repositorios de paquetes yum. A continuación, puede realizar manualmente una `yum update` para aplicar un parche de seguridad. Por el contrario, puede seguir la actualización mediante el uso de la herramienta correspondiente a fin de bloquear los repositorios de paquetes yum y así evitar nuevas actualizaciones. La herramienta `pkg-repo` está disponible en el directorio `/opt/elasticbeanstalk/bin/pkg-repo` de todas las instancias de EC2 en sus entornos de Elastic Beanstalk.

Los cambios que se realizan con la herramienta `pkg-repo` solo se llevan a cabo en la instancia de EC2 en la que se utiliza dicha herramienta. No afectan a otras instancias ni impiden futuras

actualizaciones del entorno. Los ejemplos que se proporcionan más adelante en este tema explican el modo de aplicar los cambios en todas las instancias mediante la llamada a los comandos `pkg-repo` desde los scripts y los archivos de configuración.

Warning

No se recomienda esta herramienta para la mayoría de los usuarios. Los cambios manuales aplicados a una versión de plataforma desbloqueada se consideran fuera de banda. Esta opción solo es posible para aquellos usuarios en circunstancias excepcionales que puedan asumir los siguientes riesgos:

- No se puede garantizar que las versiones de los paquetes sean coherentes en todas las instancias en los entornos.
- No se garantiza que los entornos modificados mediante la herramienta `pkg-repo` funcionen correctamente. No se han probado ni verificado en las plataformas compatibles con Elastic Beanstalk.

Se recomienda aplicar las prácticas recomendadas, que incluyen pruebas y planes de retroceso. A fin de facilitar las prácticas recomendadas, puede utilizar la consola de Elastic Beanstalk y la CLI de EB para así clonar un entorno e intercambiar las direcciones URL de este. Para obtener más información sobre cómo usar estas operaciones, consulte [Implementaciones azul/verde](#) en el capítulo relativo a la Administración de entornos de esta guía.

Si planea editar manualmente los archivos de configuración del repositorio yum, ejecute la herramienta `pkg-repo` primero. Es posible que la herramienta `pkg-repo` no funcione como se esperaba en un entorno de Amazon Linux 2 con archivos de configuración de repositorios yum editados manualmente. Esto se debe a que la herramienta podría no reconocer los cambios de configuración.

Para obtener más información sobre el repositorio de paquetes de Amazon Linux, consulte el tema sobre el [repositorio de paquetes](#) en la Guía del usuario de Amazon EC2.

comandos `pkg-repo`

Utilice la siguiente sintaxis para ejecutar los comandos de la herramienta `pkg-repo`.

```
$ /opt/elasticbeanstalk/bin/pkg-repo command [options]
```

Los comandos `pkg-repo` son los siguientes:

- `lock`: bloquea los repositorios de paquetes yum a una versión específica
- `unlock`: desbloquea los repositorios de paquetes yum desde una versión específica
- `status`: enumera todos los repositorios de paquetes yum y su estado de bloqueo actual
- `help`: muestra ayuda general o ayuda para un comando

Las opciones se aplican a los comandos de la siguiente manera:

- `lock`, `unlock` y `status` – opciones: `-h`, `--help` o ninguno (predeterminado).
- `help` – opciones: `lock`, `unlock`, `status` o ninguno (predeterminado).

En el siguiente ejemplo se ejecuta el comando `unlock`.

```
$ sudo /opt/elasticbeanstalk/bin/pkg-repo unlock
Amazon Linux 2 core package repo successfully unlocked
Amazon Linux 2 extras package repo successfully unlocked
```

En el siguiente ejemplo se ejecuta el comando `lock`.

```
$ sudo /opt/elasticbeanstalk/bin/pkg-repo lock
Amazon Linux 2 core package repo successfully locked
Amazon Linux 2 extras package repo successfully locked
```

En el siguiente ejemplo se ejecuta el comando `status`.

```
$ sudo /opt/elasticbeanstalk/bin/pkg-repo status
Amazon Linux 2 core package repo is currently UNLOCKED
Amazon Linux 2 extras package repo is currently UNLOCKED
```

En el siguiente ejemplo se ejecuta el comando `help` para el comando `lock`.

```
$ sudo /opt/elasticbeanstalk/bin/pkg-repo help lock
```

En el siguiente ejemplo se ejecuta el comando `help` para la herramienta `pkg-repo`.

```
$ sudo /opt/elasticbeanstalk/bin/pkg-repo help
```

Puede probar `pkg-repo` mediante SSH para conectarse a una instancia de su entorno de Elastic Beanstalk. Una opción SSH es el comando [eb ssh](#) de la CLI de EB.

Note

La herramienta `pkg-repo` requiere privilegios de usuario raíz para ejecutarse. Si obtiene un error de permiso de acceso, ejecute el comando de nuevo en `sudo`. No es necesario agregar `sudo` cuando utiliza la herramienta en los archivos de configuración o scripts que implementa en su entorno. Elastic Beanstalk ejecuta todos sus scripts como usuario raíz.

ejemplos pkg-repo

En la sección anterior se proporcionan ejemplos de línea de comandos para realizar pruebas en una instancia de EC2 individual de un entorno de Elastic Beanstalk. Este enfoque puede resultar práctico para las pruebas. Sin embargo, actualiza solo una instancia a la vez, por lo que no resulta práctico aplicar cambios en todas las instancias de un entorno.

Un enfoque más pragmático es utilizar scripts de [enlaces de plataforma](#) o un archivo de configuración [.ebextensions](#) que permita aplicar los cambios en todas las instancias de forma coherente.

En el siguiente ejemplo se llama a `pkg-repo` desde un archivo de configuración en la carpeta [.ebextensions](#). Elastic Beanstalk ejecuta los comandos en el archivo `update_package.config` cuando se implementa el paquete de código fuente de la aplicación.

```
.ebextensions
### update_package.config
```

Para recibir la última versión del paquete `docker`, esta configuración especifica el paquete `docker` en el comando `yum update`.

```
### update_package.config ###

commands:
  update_package:
```

```
command: |
  /opt/elasticbeanstalk/bin/pkg-repo unlock
  yum update docker -y
  /opt/elasticbeanstalk/bin/pkg-repo lock
  yum clean all -y
  rm -rf /var/cache/yum
```

En esta configuración no se especifica ningún paquete en el comando yum update. En consecuencia, se aplican todas las actualizaciones disponibles.

```
### update_package.config ###

commands:
  update_package:
    command: |
      /opt/elasticbeanstalk/bin/pkg-repo unlock
      yum update -y
      /opt/elasticbeanstalk/bin/pkg-repo lock
      yum clean all -y
      rm -rf /var/cache/yum
```

En el siguiente ejemplo se llama a pkg-repo desde un script bash como [enlace de la plataforma](#). Elastic Beanstalk ejecuta el archivo de script update_package.sh que se encuentra en el subdirectorio prebuild.

```
.platform
### hooks
  ### prebuild
    ### update_package.sh
```

Para recibir la última versión del paquete docker, este script especifica el paquete docker en el comando yum update. Si se omite el nombre del paquete, se aplican todas las actualizaciones disponibles. En el ejemplo del archivo de configuración anterior se demuestra esto.

```
### update_package.sh ###

#!/bin/bash

/opt/elasticbeanstalk/bin/pkg-repo unlock
yum update docker -y
/opt/elasticbeanstalk/bin/pkg-repo lock
```

```
yum clean all -y
rm -rf /var/cache/yum
```

download-source-bundle (Solo AMI de Amazon Linux)

En las ramificaciones de la plataforma AMI de Amazon Linux (antes de Amazon Linux 2), Elastic Beanstalk proporciona una herramienta adicional, que es `download-source-bundle`. Utilice esta herramienta para descargar el código fuente de su aplicación cuando implemente su plataforma. La herramienta está disponible en `/opt/elasticbeanstalk/bin/download-source-bundle`.

El script de ejemplo `00-unzip.sh` se encuentra en la carpeta `appdeploy/pre` de las instancias de entorno. Se demuestra cómo utilizar `download-source-bundle` para descargar el código fuente de la aplicación en la carpeta `/opt/elasticbeanstalk/deploy/appsource` durante la implementación.

Implementación de aplicaciones de Elastic Beanstalk desde contenedores Docker

En este capítulo se explica cómo puede utilizar Elastic Beanstalk para implementar aplicaciones web desde contenedores de Docker. Los contenedores de Docker son autónomos y contienen todo el software y la información de configuración que la aplicación web necesita para ejecutarse. Con los contenedores Docker, puede definir su propio entorno de ejecución. También puede elegir su propio lenguaje de programación y las dependencias de las aplicaciones, como gestores de paquetes o herramientas, que normalmente no son compatibles con otras plataformas de Elastic Beanstalk.

Siga los pasos que se indican [QuickStart para Docker](#) para crear una aplicación «Hello World» de Docker e implementarla en un entorno de Elastic Beanstalk mediante la CLI de EB.

Temas

- [Sucursales de la plataforma Docker](#)
- [Uso de la ramificación de la plataforma Docker](#)
- [Uso de la ramificación de la plataforma de Amazon ECS](#)
- [Contenedores de Docker preconfigurados \(AMI de Amazon Linux\)](#)

Sucursales de la plataforma Docker

La plataforma de Docker en Elastic Beanstalk admite las siguientes ramificaciones de la plataforma:

Docker sobre Amazon Linux 2 y Docker sobre AL2023

Elastic Beanstalk implementa los contenedores de Docker y el código fuente en las instancias de EC2 y los administra. Estas ramificaciones de la plataforma ofrecen compatibilidad para múltiples contenedores. Puede utilizar la herramienta Docker Compose para simplificar la configuración, las pruebas y la implementación de su aplicación. Para obtener más información sobre esta ramificación de la plataforma, consulte [the section called “Ramificación de la plataforma Docker”](#).

ECS sobre Amazon Linux 2 y ECS sobre AL2023

Se ofrece esta ramificación para los clientes que necesiten una ruta de migración a AL2023/AL2 desde la ramificación de la plataforma retirada Docker multicontenedor sobre (AMI de Amazon Linux). Las ramificaciones de la plataforma más recientes admiten todas las características de la ramificación de la plataforma retirada. No es necesario realizar cambios en el código fuente. Para obtener más información, consulte [Migración de Docker multicontenedor sobre Amazon Linux a ECS sobre Amazon Linux 2023](#). Si no tiene un entorno de Elastic Beanstalk que se ejecute en una ramificación de la plataforma basada en ECS, le recomendamos que utilice la ramificación de la plataforma Docker que se ejecuta en AL2023 de 64 bits. Esto ofrece un enfoque más sencillo y requiere menos recursos.

Ramificaciones de la plataforma retiradas que se ejecutan en la AMI de Amazon Linux (AL1)

El [18 de julio de 2022](#), Elastic Beanstalk estableció el estado de todas las ramificaciones de la plataforma basadas en la AMI de Amazon Linux (AL1) como retirado. Amplíe cada una de las siguientes secciones para obtener más información sobre cada ramificación de la plataforma retirada y su ruta de migración a la ramificación de la plataforma más reciente sobre Amazon Linux 2 o Amazon Linux 2023 (recomendada).

Docker (AMI de Amazon Linux)

Esta ramificación de la plataforma puede implementar una imagen de Docker, descrita en un Dockerfile o en una definición `DockerRun.aws.json` de la v1. Esta ramificación de la plataforma ejecuta solo un contenedor para cada instancia. Las ramificaciones de la plataforma posteriores, Docker que se ejecuta en AL2023 de 64 bits y Docker que se ejecuta en Amazon Linux 2 de 64 bits, son compatibles con múltiples contenedores de Docker por instancia.

Le recomendamos que cree sus entornos con la ramificación de la plataforma más reciente y compatible, Docker que se ejecuta en AL2023 de 64 bits. Luego, puede migrar la aplicación al

entorno recién creado. Para obtener más información sobre la creación de estos entornos, consulte [the section called “Ramificación de la plataforma Docker”](#). Para obtener más información sobre cómo migrar, consulte [Migración de su aplicación de Linux de Elastic Beanstalk a Amazon Linux 2023 o Amazon Linux 2](#).

Multicontainer Docker (AMI de Amazon Linux)

Esta ramificación de la plataforma utiliza Amazon ECS para coordinar una implementación de varios contenedores de Docker en un clúster de Amazon ECS dentro de un entorno de Elastic Beanstalk. Si actualmente utiliza esta ramificación de la plataforma retirada, se recomienda que migre a la ramificación de la plataforma ECS sobre Amazon Linux 2023 más reciente. La ramificación de la plataforma más reciente es compatible con todas las características de esta ramificación de la plataforma suspendida. No es necesario realizar cambios en el código fuente. Para obtener más información, consulte [Migración de Docker multicontenedor sobre Amazon Linux a ECS sobre Amazon Linux 2023](#).

Contenedores de Docker preconfigurados

Además de las plataformas Docker mencionadas anteriormente, también existe la rama de GlassFish plataforma Docker preconfigurada que se ejecuta en el sistema operativo AMI Amazon Linux (AL1).

Esta ramificación de la plataforma se reemplazó por las ramificaciones de la plataforma Docker que se ejecuta en AL2023 de 64 bits y Docker que se ejecuta en Amazon Linux 2 de 64 bits. Para obtener más información, consulte [Implementación de una GlassFish aplicación](#) en la plataforma Docker.

Uso de la ramificación de la plataforma Docker

AWS Elastic Beanstalk puede lanzar entornos Docker creando una imagen descrita en una imagen Docker remota `Dockerfile` o extrayendo una imagen Docker remota. Si va a implementar una imagen de Docker remota, no tiene que incluir un archivo `Dockerfile`. En su lugar, si también utiliza Docker Compose, use un archivo `docker-compose.yml`, que especifica la imagen que se va a usar y las opciones de configuración adicionales. Si no utiliza Docker Compose con sus entornos de Docker, use un archivo `DockerRun.aws.json` en su lugar.

Temas

- [QuickStart: Implementar una aplicación de Docker en Elastic Beanstalk](#)
- [Configuración de Docker](#)

- [Configuración de los entornos de Docker](#)

QuickStart: Implementar una aplicación de Docker en Elastic Beanstalk

Este QuickStart tutorial explica el proceso de creación de una aplicación Docker e implementarla en un AWS Elastic Beanstalk entorno.

Note

Este QuickStart tutorial está destinado a fines de demostración. No utilice la aplicación creada en este tutorial para el tráfico de producción.

Secciones

- [Tu AWS cuenta](#)
- [Requisitos previos](#)
- [Paso 1: Cree una aplicación y un contenedor de Docker](#)
- [Paso 2: ejecuta tu aplicación localmente](#)
- [Paso 3: Implemente la aplicación Docker con la CLI de EB](#)
- [Paso 4: Ejecute la aplicación en Elastic Beanstalk](#)
- [Paso 5: Eliminar](#)
- [AWS recursos para su aplicación](#)
- [Siguiendo pasos](#)
- [Implemente con la consola de Elastic Beanstalk](#)

Tu AWS cuenta

Si aún no eres AWS cliente, debes crear una AWS cuenta. El registro le permite acceder a Elastic Beanstalk AWS y a otros servicios que necesite.

Si ya tiene una AWS cuenta, puede pasar a [Requisitos previos](#)

Crea una AWS cuenta

Inscríbese en una Cuenta de AWS

Si no tiene una Cuenta de AWS, complete los siguientes pasos para crearlo.

Para suscribirse a una Cuenta de AWS

1. Abra <https://portal.aws.amazon.com/billing/signup>.
2. Siga las instrucciones que se le indiquen.

Parte del procedimiento de registro consiste en recibir una llamada telefónica e indicar un código de verificación en el teclado del teléfono.

Cuando te registras en una Cuenta de AWS, Usuario raíz de la cuenta de AWS se crea una. El usuario raíz tendrá acceso a todos los Servicios de AWS y recursos de esa cuenta. Como práctica recomendada de seguridad, asigne acceso administrativo a un usuario y utilice únicamente el usuario raíz para realizar [tareas que requieren acceso de usuario raíz](#).

AWS te envía un correo electrónico de confirmación una vez finalizado el proceso de registro. Puede ver la actividad de la cuenta y administrar la cuenta en cualquier momento entrando en <https://aws.amazon.com/> y seleccionando Mi cuenta.

Creación de un usuario con acceso administrativo

Después de crear un usuario administrativo Cuenta de AWS, asegúrelo Usuario raíz de la cuenta de AWS AWS IAM Identity Center, habilite y cree un usuario administrativo para no usar el usuario root en las tareas diarias.

Proteja su Usuario raíz de la cuenta de AWS

1. Inicie sesión [AWS Management Console](#) como propietario de la cuenta seleccionando el usuario root e introduciendo su dirección de Cuenta de AWS correo electrónico. En la siguiente página, escriba su contraseña.

Para obtener ayuda para iniciar sesión con el usuario raíz, consulte [Iniciar sesión como usuario raíz](#) en la Guía del usuario de AWS Sign-In .

2. Active la autenticación multifactor (MFA) para el usuario raíz.

Para obtener instrucciones, consulte [Habilitar un dispositivo MFA virtual para el usuario Cuenta de AWS raíz \(consola\)](#) en la Guía del usuario de IAM.

Creación de un usuario con acceso administrativo

1. Activar IAM Identity Center.

Consulte las instrucciones en [Activar AWS IAM Identity Center](#) en la Guía del usuario de AWS IAM Identity Center .

2. En IAM Identity Center, conceda acceso administrativo a un usuario.

Para ver un tutorial sobre su uso Directorio de IAM Identity Center como fuente de identidad, consulte [Configurar el acceso de los usuarios con la configuración predeterminada Directorio de IAM Identity Center en la](#) Guía del AWS IAM Identity Center usuario.

Iniciar sesión como usuario con acceso de administrador

- Para iniciar sesión con el usuario de IAM Identity Center, utilice la URL de inicio de sesión que se envió a la dirección de correo electrónico cuando creó el usuario de IAM Identity Center.

Para obtener ayuda para iniciar sesión con un usuario del Centro de identidades de IAM, consulte [Iniciar sesión en el portal de AWS acceso](#) en la Guía del AWS Sign-In usuario.

Concesión de acceso a usuarios adicionales

1. En IAM Identity Center, cree un conjunto de permisos que siga la práctica recomendada de aplicar permisos de privilegios mínimos.

Para conocer las instrucciones, consulte [Create a permission set](#) en la Guía del usuario de AWS IAM Identity Center .

2. Asigne usuarios a un grupo y, a continuación, asigne el acceso de inicio de sesión único al grupo.

Para conocer las instrucciones, consulte [Add groups](#) en la Guía del usuario de AWS IAM Identity Center .

Requisitos previos

Para seguir los procedimientos de esta guía, necesitará un shell o un terminal de línea de comando donde pueda ejecutar los comandos. Los comandos aparecen en listas y van precedidos del símbolo del sistema (\$) y del nombre del directorio actual, si es aplicable.

```
~/eb-project$ this is a command  
this is output
```

En Linux y macOS, puede utilizar el administrador de shell y paquetes preferido. En Windows, puede [instalar el subsistema de Windows para Linux](#) y obtener una versión de Ubuntu y Bash integrada en Windows.

CLI DE EB

En este tutorial también se utiliza la interfaz de línea de comandos de Elastic Beanstalk (CLI de EB). Para obtener detalles sobre la instalación y configuración de la CLI de EB, consulte [Instalación de la CLI de EB](#) y [Configuración de la CLI de EB](#).

Docker

Para seguir este tutorial, necesitarás una instalación local de Docker que funcione. Para obtener más información, consulte [Descripción de Docker](#) en la página de documentos de Docker.

Compruebe que el daemon de Docker esté en funcionamiento ejecutando el siguiente comando.

```
~$ docker info
```

Paso 1: Cree una aplicación y un contenedor de Docker

Para este ejemplo, creamos una imagen de Docker de la aplicación Flask de muestra a la que también se hace referencia en [Implementación de una aplicación Flask en Elastic Beanstalk](#)

La aplicación consta de dos archivos:

- `app.py`— el archivo Python que contiene el código que se ejecutará en el contenedor.
- `Dockerfile`— el Dockerfile para crear la imagen.

Coloca ambos archivos en la raíz de un directorio.

```
~/eb-docker-flask/  
|-- Dockerfile  
|-- app.py
```

Añada los siguientes contenidos a su `Dockerfile`.

Example `~/eb-docker-flask/Dockerfile`

```
FROM python:3.12  
COPY . /app
```

```
WORKDIR /app
RUN pip install Flask==3.0.2
EXPOSE 5000
CMD [ "python3", "-m" , "flask", "run", "--host=0.0.0.0"]
```

Agregue los siguientes contenidos a su `app.py` archivo.

Example `~/eb-docker-flask/app.py`

```
from flask import Flask
app = Flask(__name__)
@app.route('/')
def hello_world():
    return 'Hello Elastic Beanstalk! This is a Docker application'
```

Construye tu contenedor de Docker etiquetando la imagen con. `eb-docker-flask`

```
~/eb-docker-flask$ docker build -t eb-docker-flask
```

Paso 2: ejecuta tu aplicación localmente

Usa el comando [docker build](#) para crear tu imagen de contenedor localmente, etiquetando la imagen con. `eb-docker-flask` El punto (.) al final del comando especifica que la ruta es un directorio local.

```
~/eb-docker-flask$ docker run -dp 127.0.0.1:5000:5000 eb-docker-flask .
```

Ejecute su contenedor con el [comando docker run](#). El comando imprimirá el ID del contenedor en ejecución. La `-d` opción ejecuta el docker en modo de fondo. La `-p` opción expone la aplicación en el puerto 5000. Elastic Beanstalk envía el tráfico al puerto 5000 de la plataforma Docker de forma predeterminada.

```
~/eb-docker-flask$ docker run -dp 127.0.0.1:5000:5000 eb-docker-flask container-id
```

Navigate hasta en su `http://127.0.0.1:5000/` navegador. Debería ver el texto «¡Hola, Elastic Beanstalk! Esta es una aplicación de Docker».

Ejecute el comando [docker kill](#) para terminar el contenedor.

```
~/eb-docker-flask$ docker kill container-id
```

Paso 3: Implemente la aplicación Docker con la CLI de EB

Ejecute los siguientes comandos para crear un entorno de Elastic Beanstalk para esta aplicación.

Para crear un entorno e implementar su aplicación Docker

1. Inicialice el repositorio de la CLI de EB con el comando `eb init`.

```
~/eb-docker-flask$ eb init -p docker docker-tutorial us-east-2  
Application docker-tutorial has been created.
```

Este comando crea una aplicación con un nombre `docker-tutorial` y configura su repositorio local para crear entornos con la última versión de la plataforma Docker.

2. (Opcional) Ejecute de nuevo `eb init` para configurar un par de claves predeterminadas de forma que pueda usar SSH para conectarse a la instancia EC2 donde se ejecuta la aplicación.

```
~/eb-docker-flask$ eb init  
Do you want to set up SSH for your instances?  
(y/n): y  
Select a keypair.  
1) my-keypair  
2) [ Create new KeyPair ]
```

Seleccione un par de claves si ya tiene uno o siga las instrucciones para crear uno. Si no ve el símbolo del sistema o más adelante necesita cambiar la configuración, ejecute `eb init -i`.

3. Cree un entorno e implemente la aplicación en él con `eb create`. Elastic Beanstalk crea automáticamente un archivo zip para la aplicación y lo inicia en el puerto 5000.

```
~/eb-docker-flask$ eb create docker-tutorial
```

Elastic Beanstalk tarda unos cinco minutos en crear el entorno.

Paso 4: Ejecute la aplicación en Elastic Beanstalk

Cuando finalice el proceso de creación del entorno, abra su sitio web con `eb open`

```
~/eb-docker-flask$ eb open
```

¡Enhorabuena! ¡Ha implementado una aplicación de Docker con Elastic Beanstalk! Se abre una ventana del navegador con el nombre de dominio creado para su aplicación.

Paso 5: Eliminar

Puede cerrar el entorno cuando termine de trabajar con la aplicación. Elastic Beanstalk AWS cancela todos los recursos asociados a su entorno.

Para finalizar el entorno de Elastic Beanstalk con la CLI de EB, ejecute el siguiente comando.

```
~/eb-docker-flask$ eb terminate
```

AWS recursos para su aplicación

Acabas de crear una aplicación de instancia única. Sirve como una aplicación de muestra sencilla con una sola instancia EC2, por lo que no requiere balanceo de carga ni escalado automático. Para las aplicaciones de instancia única, Elastic Beanstalk crea los siguientes recursos: AWS

- EC2 instance (Instancia EC2): máquina virtual de Amazon EC2 configurada para ejecutar aplicaciones web en la plataforma que elija.

Cada plataforma ejecuta un conjunto distinto de software, archivos de configuración y scripts compatibles con una determinada versión de lenguaje, marco y contenedor web (o una combinación de ellos). La mayoría de las plataformas utilizan Apache o nginx como un proxy inverso que procesa el tráfico web delante de la aplicación web, reenvía las solicitudes a esta, administra los recursos estáticos y genera registros de acceso y errores.

- Instance security group (Grupo de seguridad de la instancia): grupo de seguridad de Amazon EC2 configurado para permitir el tráfico entrante en el puerto 80. Este recurso permite que el tráfico HTTP procedente del balanceador de carga llegue a la instancia EC2 en la que se ejecuta la aplicación web. De forma predeterminada, el tráfico no está permitido en otros puertos.
- Bucket de Amazon S3: ubicación de almacenamiento para el código fuente, los registros y otros artefactos que se crean al utilizar Elastic Beanstalk.
- CloudWatch Alarmas de Amazon: dos CloudWatch alarmas que monitorean la carga de las instancias de su entorno y se activan si la carga es demasiado alta o demasiado baja. Cuando se activa una alarma, en respuesta, el grupo de Auto Scaling aumenta o reduce los recursos.
- AWS CloudFormation pila: Elastic AWS CloudFormation Beanstalk se utiliza para lanzar los recursos de su entorno y propagar los cambios de configuración. Los recursos se definen en una plantilla que puede verse en la [consola de AWS CloudFormation](#).

- Nombre de dominio: nombre de dominio que direcciona el tráfico a la aplicación web con el formato *subdominio.región.elasticbeanstalk.com*.

Elastic Beanstalk administra todos estos recursos. Cuando termina su entorno, Elastic Beanstalk termina todos los recursos que este contiene.

Siguientes pasos

Una vez que disponga de un entorno que ejecute una aplicación, podrá implementar una nueva versión de la aplicación o una aplicación distinta en cualquier momento. La implementación de una nueva versión de la aplicación es una tarea muy rápida, ya que no se requiere aprovisionar ni reiniciar instancias EC2. También puede explorar su nuevo entorno con la consola de Elastic Beanstalk. Para ver los pasos detallados, consulte [Explore su entorno](#) en el capítulo Introducción de esta guía.

Una vez que haya implementado una o dos aplicaciones de muestra y esté listo para empezar a desarrollar y ejecutar aplicaciones Docker de forma local, consulte

Implemente con la consola de Elastic Beanstalk

También puede usar la consola de Elastic Beanstalk para iniciar la aplicación de muestra. Para ver los pasos detallados, consulte [Crear una aplicación de ejemplo](#) en el capítulo Introducción de esta guía.

Configuración de Docker

En esta sección se describe cómo preparar su imagen y contenedor Docker y para implementarlos en Elastic Beanstalk.

Entorno de Docker con Docker Compose

En esta sección se describe cómo preparar su imagen y contenedor Docker y para implementarlos en Elastic Beanstalk. Cualquier aplicación web que implemente en Elastic Beanstalk en un entorno de Docker debe incluir un archivo `docker-compose.yml` si también utiliza la herramienta Docker Compose. Puede implementar su aplicación web como un servicio en contenedor en Elastic Beanstalk realizando una de las siguientes acciones:

- Cree un archivo `docker-compose.yml` para implementar una imagen de Docker de un repositorio alojado en Elastic Beanstalk. No se requieren otros archivos si todas las implementaciones provienen de imágenes de repositorios públicos. (Si la implementación

debe provenir de una imagen de un repositorio privado, debe incluir archivos de configuración adicionales para la autenticación. Para obtener más información, consulte [Utilizar imágenes de un repositorio privado](#)). Para obtener más información sobre el archivo `docker-compose.yml`, consulte [Compose file reference](#) en el sitio web de Docker.

- Cree un `Dockerfile` para que Elastic Beanstalk cree y ejecute una imagen personalizada. Este archivo es opcional y depende de los requisitos de implementación. Para obtener más información acerca de `Dockerfile`, consulte [Dockerfile reference \(Referencia de Dockerfile\)](#) en el sitio web de Docker.
- Cree un archivo `.zip` que contenga los archivos de la aplicación, todas las dependencias de los archivos de la aplicación, el archivo `Dockerfile` y el archivo `docker-compose.yml`. Si utiliza la CLI de EB para implementar la aplicación, se creará un archivo `.zip` automáticamente. Los dos archivos deben estar en la raíz, o en el nivel superior, del archivo `.zip`.

Si utiliza únicamente un archivo `docker-compose.yml` para implementar la aplicación, no es necesario crear un archivo `.zip`.

Este tema es una referencia de sintaxis. Para obtener procedimientos detallados sobre el inicio de entornos Docker mediante Elastic Beanstalk, consulte [Uso de la ramificación de la plataforma Docker](#).

Para obtener más información sobre Docker Compose y cómo instalarlo, consulte los sitios de Docker [Overview of Docker Compose](#) e [Install Docker Compose](#).

Note

Si no usa Docker Compose para configurar sus entornos de Docker, tampoco debe usar el archivo `docker-compose.yml`. En su lugar, utilice el archivo `DockerRun.aws.json`, `Dockerfile` o ambos.

Para obtener más información, consulte [the section called “Configuración de plataformas de Docker \(sin Docker Compose\)”](#).

Uso de imágenes de un repositorio privado

Elastic Beanstalk debe autenticarse con el registro en línea que aloja el repositorio privado antes de que pueda extraer e implementar las imágenes de un repositorio privado. Ofrecemos ejemplos de dos opciones para almacenar y recuperar las credenciales de su entorno de Elastic Beanstalk para autenticarse en un repositorio.

- El AWS Secrets Manager
- El archivo `Dockerrun.aws.json v3`

Usando AWS Secrets Manager

Puede configurar Elastic Beanstalk para que inicie sesión en su repositorio privado antes de que inicie el proceso de implementación. Esto permite a Elastic Beanstalk acceder a las imágenes desde el repositorio e implementar estas imágenes en su entorno Elastic Beanstalk.

Esta configuración inicia eventos en la fase de precompilación del proceso de implementación de Elastic Beanstalk. Esto se configura en el directorio de configuración [.ebextensions](#). La configuración utiliza scripts de [enlace de plataforma](#) que llaman a `docker login` para la autenticación en el registro en línea que aloja el repositorio privado. A continuación, se presenta un desglose detallado de estos pasos de configuración.

Configuración de Elastic Beanstalk para autenticarse en su repositorio privado con AWS Secrets Manager

Note

Se deben otorgar permisos específicos para completar estos pasos. Para obtener más información, consulte las siguientes referencias.

- En Paso 2, necesitará permisos para crear un secreto. Para obtener más información, consulte [Ejemplo: permiso para crear secretos](#) en la Guía del usuario de AWS Secrets Manager .
- En Paso 3, necesitará permisos para recuperar secretos mediante las referencias dinámicas `secretsmanager:`. Para obtener más información, consulte [Ejemplo: Permiso para recuperar valores de secretos](#) en la Guía del usuario de AWS Secrets Manager .

1. Cree su estructura de directorios `.ebextensions` de la siguiente manera.

```
### .ebextensions
#   ### env.config
### .platform
#   ### confighooks
# #   ### prebuild
```

```
# #      ### 01login.sh
#      ### hooks
#      ### prebuild
#      ### 01login.sh
### docker-compose.yml
```

2. Úselo AWS Secrets Manager para guardar las credenciales de su repositorio privado para que Elastic Beanstalk pueda recuperarlas cuando sea necesario. Para ello, ejecute el comando [create-secret](#) AWS CLI de Secrets Manager.

```
aws secretsmanager create-secret \
    --name MyTestSecret \
    --description "My image repo credentials created with the CLI." \
    --secret-string "{\"USER\": \"EXAMPLE-USERNAME\", \"PASSWD\": \"EXAMPLE-PASSWORD\"}"
```

3. Cree el siguiente archivo `env.config` y colóquelo en el directorio `.ebextensions` como se muestra en la estructura de directorios anterior. Esta configuración utiliza el espacio de nombres [aws:elasticbeanstalk:application:environment](#) para iniciar las variables `USER` y `PASSWD` del entorno de Elastic Beanstalk con las referencias dinámicas a AWS Secrets Manager. Para obtener más información sobre las referencias `secretsmanager` dinámicas, consulte [Recuperar un AWS Secrets Manager secreto de un AWS CloudFormation recurso en la Guía del AWS Secrets Manager usuario](#).

Note

`USER` y `PASSWD` en el script deben coincidir con las mismas cadenas que se utilizan en el comando `secretsmanager create-secret` anterior.

```
option_settings:
  aws:elasticbeanstalk:application:environment:
    USER: '{{resolve:secretsmanager:MyTestSecret:SecretString:USER}}'
    PASSWD: '{{resolve:secretsmanager:MyTestSecret:SecretString:PASSWD}}'
```

4. Cree el siguiente archivo de script `01login.sh` y colóquelo en los siguientes directorios (también se muestra en la estructura de directorios anterior):
 - `.platform/confighooks/prebuild`
 - `.platform/hooks/prebuild`

```
### example 01login.sh
#!/bin/bash
USER=/opt/elasticbeanstalk/bin/get-config environment -k USER
/opt/elasticbeanstalk/bin/get-config environment -k PASSWD | docker login -u $USER
--password-stdin
```

El script `01login.sh` llama al script de la plataforma [get-config](#) para recuperar las credenciales del repositorio y, luego, inicia sesión en el repositorio. Almacena el nombre de usuario en la variable `USER` del script. En la siguiente línea, recupera la contraseña. En lugar de almacenar la contraseña en una variable, el script la envía directamente al comando `docker login` del flujo de entrada `stdin`. La opción `--password-stdin` usa el flujo de entrada, por lo que no tiene que almacenar la contraseña en una variable. Para obtener más información sobre la autenticación con la interfaz de la línea de comandos de Docker, consulte [Docker login](#) (Inicio de sesión de Docker) en el sitio web de documentación de Docker.

Notas

- Todos los archivos de script deben tener permiso de ejecución. Utilice `chmod +x` para configurar el permiso de ejecución de los archivos de enlace. Para todas las versiones de plataformas basadas en Amazon Linux 2 que se publicaron el 29 de abril de 2022 o en fecha posterior, Elastic Beanstalk automáticamente otorga permisos de ejecución a todos los scripts de enlace de plataforma. En este caso, no tiene que otorgar permisos de ejecución manualmente. Para obtener una lista de estas versiones de plataforma, consulte las notas de la versión del [29 de abril de 2022 - Plataforma Linux](#) en la AWS Elastic Beanstalk Guía de notas de la versión.
- Los archivos de enlace pueden ser archivos binarios o archivos de script y deben comenzar con una línea `#!` que contenga la ruta del intérprete, como `#!/bin/bash`.
- Para obtener más información, consulte [the section called “Enlaces de la plataforma”](#) en Ampliación de plataformas Linux de Elastic Beanstalk.

Cuando Elastic Beanstalk se autentique con el registro en línea que aloja el repositorio privado, podrá implementar y extraer sus imágenes.

Uso del archivo `Dockerrun.aws.json v3`

En esta sección se describe otro enfoque para que Elastic Beanstalk se autentique en un repositorio privado. Con este enfoque, se genera un archivo de autenticación con el comando Docker y, a continuación, se carga el archivo de autenticación en un bucket de Amazon S3. También debe incluir la información del bucket en el archivo `Dockerrun.aws.json v3`.

Para generar y proporcionar un archivo de autenticación a Elastic Beanstalk

1. Genere un archivo de autenticación con el comando `docker login`. En el caso de los repositorios de Docker Hub, ejecute `docker login`:

```
$ docker login
```

En el caso de otros registros, incluya la URL del servidor del registro:

```
$ docker login registry-server-url
```

Note

Si su entorno de Elastic Beanstalk utiliza la versión de la plataforma Docker AMI de Amazon Linux (precede a Amazon Linux 2), lea la información relevante en [the section called “Configuración de Docker en la AMI de Amazon Linux \(anterior a Amazon Linux 2\)”](#).

Para obtener más información sobre el archivo de autenticación, consulte [Store images on Docker Hub](#) y [docker login](#) en el sitio web de Docker.

2. Cargue una copia del archivo de autenticación denominado `.dockercfg` en un bucket de Amazon S3 seguro.
 - El bucket de Amazon S3 debe alojarse en el Región de AWS mismo entorno que el que lo utiliza. Elastic Beanstalk no puede descargar archivos desde un bucket de Amazon S3 alojado en otras regiones.
 - Conceda permisos para la operación `s3:GetObject` al rol de IAM en el perfil de instancia. Para obtener más información, consulte [Administración de perfiles de instancia de Elastic Beanstalk](#).

- Incluya la información del bucket de Amazon S3 en el parámetro `Authentication` en el archivo `Dockerrun.aws.json v3`.

A continuación, se muestra un ejemplo de un archivo `Dockerrun.aws.json v3`.

```
{
  "AWSEBDockerrunVersion": "3",
  "Authentication": {
    "bucket": "DOC-EXAMPLE-BUCKET",
    "key": "mydockercfg"
  }
}
```

Note

El parámetro `AWSEBDockerrunVersion` indica la versión del archivo `Dockerrun.aws.json`.

- La plataforma de Docker Amazon Linux 2 utiliza el archivo `Dockerrun.aws.json v3` para los entornos que usan Docker Compose. Utiliza el archivo `Dockerrun.aws.json v1` para los entornos que no usan Docker Compose.
- La plataforma de la AMI Multicontainer Docker de Amazon Linux utiliza el archivo `Dockerrun.aws.json v2`.

Cuando Elastic Beanstalk pueda autenticarse con el registro en línea que aloja el repositorio privado, se podrán implementar y extraer sus imágenes.

Creación de imágenes personalizadas con un Dockerfile

Tendrá que crear un archivo `Dockerfile` si aún no dispone de una imagen alojada en un repositorio.

El siguiente fragmento es un ejemplo del archivo `Dockerfile`. Cuando siga las instrucciones de [Uso de la ramificación de la plataforma Docker](#), podrá cargar este archivo `Dockerfile` tal como está escrito. Elastic Beanstalk ejecuta el juego 2048 cuando se utiliza este `Dockerfile`.

Para obtener más información sobre las instrucciones que puede incluir en el archivo `Dockerfile`, consulte la [referencia de Dockerfile](#) en el sitio web de Docker.

```
FROM ubuntu:12.04

RUN apt-get update
RUN apt-get install -y nginx zip curl

RUN echo "daemon off;" >> /etc/nginx/nginx.conf
RUN curl -o /usr/share/nginx/www/master.zip -L https://codeload.github.com/gabrielecirulli/2048/zip/master
RUN cd /usr/share/nginx/www/ && unzip master.zip && mv 2048-master/* . && rm -rf 2048-master master.zip

EXPOSE 80

CMD ["/usr/sbin/nginx", "-c", "/etc/nginx/nginx.conf"]
```

Note

Puede ejecutar compilaciones de varias etapas a partir de un solo Dockerfile para producir imágenes de menor tamaño con una reducción significativa de la complejidad. Para obtener más información, consulte [Utilizar compilaciones de varias etapas](#) en el sitio web de documentación de Docker.

Configuración de plataformas de Docker (sin Docker Compose)

Si su entorno de Docker en Elastic Beanstalk no utiliza Docker Compose, consulte la información adicional de las secciones siguientes.

Configuración de la plataforma de Docker: sin Docker Compose

Cualquier aplicación web que se implemente en Elastic Beanstalk en un entorno de Docker debe incluir un archivo `Dockerfile` o `Dockerrun.aws.json`. Puede implementar su aplicación web desde un contenedor de Docker en Elastic Beanstalk realizando alguna de las siguientes acciones:

- Cree un `Dockerfile` para que Elastic Beanstalk cree y ejecute una imagen personalizada.
- Cree un archivo `Dockerrun.aws.json` para implementar una imagen de Docker de un repositorio alojado en Elastic Beanstalk.
- Cree un archivo `.zip` que contenga los archivos de la aplicación, todas las dependencias de los archivos de la aplicación, el archivo `Dockerfile` y el archivo `Dockerrun.aws.json`. Si utiliza la CLI de EB para implementar la aplicación, se creará un archivo `.zip` automáticamente.

Si utiliza únicamente un archivo `Dockerfile` o solo un archivo `Dockerrun.aws.json` para implementar la aplicación, no es necesario comprimir el archivo en un archivo `.zip`.

Este tema es una referencia de sintaxis. Para obtener información detallada sobre cómo iniciar entornos de Docker, consulte [Uso de la ramificación de la plataforma Docker](#).

Dockerrun.aws.json v1

Un archivo `Dockerrun.aws.json` describe cómo implementar una imagen remota de Docker como una aplicación de Elastic Beanstalk. Este archivo JSON es específico de Elastic Beanstalk. Si la aplicación se ejecuta en una imagen que está disponible en un repositorio alojado, puede especificar la imagen en un archivo `Dockerrun.aws.json v1` y omitir el archivo `Dockerfile`.

Las claves y los valores válidos para el archivo `Dockerrun.aws.json v1` se definen con las siguientes operaciones:

AWSEBDockerrunVersion

(Obligatorio) Especifica el número de versión como el valor 1 para entornos Docker de un solo contenedor.

Autenticación

(Obligatorio solo para repositorios privados) Especifica el objeto de Amazon S3 que almacena el archivo `.dockercfg`.

Consulte [Uso de imágenes de un repositorio privado](#).

Imagen

Especifica la imagen base de Docker en un repositorio Docker existente desde el que va a crear un contenedor Docker. Especifique el valor de la clave Name (Nombre) con el formato *<organización>/nombre de imagen* para las imágenes de Docker Hub o *<sitio>/nombre de organización/<nombre de imagen>* para otros sitios.

Cuando especifica una imagen en el archivo `Dockerrun.aws.json`, cada instancia del entorno de Elastic Beanstalk ejecuta `docker pull` para ejecutar la imagen. Si lo desea, incluya la clave Update (Actualizar). El valor predeterminado es `true` e indica a Elastic Beanstalk que consulte el repositorio, inserte las actualizaciones en la imagen y sobrescriba las imágenes en caché.

Cuando utilice un archivo `Dockerfile`, no especifique la clave `Image` en el archivo `Dockerrun.aws.json`. Elastic Beanstalk siempre crea y usa la imagen descrita en el archivo `Dockerfile` si hay alguna.

Puertos

(Obligatorio si especifica la clave `Image`) Muestra los puertos que se exponen en el contenedor Docker. Elastic Beanstalk `ContainerPort` el valor para conectar el contenedor de Docker al proxy inverso que se ejecuta en el host.

Puede especificar varios puertos de contenedor, pero Elastic Beanstalk solo utiliza el primer puerto. Utiliza este puerto para conectar el contenedor al proxy inverso del host y para enrutar las solicitudes de la Internet pública. Si utiliza un **Dockerfile**, el primer `ContainerPort` debe coincidir con la primera entrada de la lista `EXPOSE` **Dockerfile**.

Si lo desea, puede especificar una lista de puertos de entrada `HostPort`. `HostPort` las entradas especifican los puertos del host a `ContainerPort` los que se asignan los valores. Si no especificas un `HostPort` valor, el valor predeterminado será el `ContainerPort` valor.

```
{
  "Image": {
    "Name": "image-name"
  },
  "Ports": [
    {
      "ContainerPort": 8080,
      "HostPort": 8000
    }
  ]
}
```

Volúmenes

Asigne volúmenes de una instancia EC2 a su contenedor Docker. Especifique una o varias matrices de volúmenes que desee asignar.

```
{
  "Volumes": [
    {
      "HostDirectory": "/path/inside/host",
      "ContainerDirectory": "/path/inside/container"
    }
  ]
}
```



```
    }  
  ]  
  ...
```

Registro

Especifique el directorio dentro del contenedor en el que la aplicación escribe los registros. Elastic Beanstalk carga todos los registros de este directorio en Amazon S3 cuando solicita registros de cola o paquete. Si rota los logs en una carpeta con el nombre `rotated` dentro de este directorio, también puede configurar Elastic Beanstalk para que cargue los logs rotados en Amazon S3 para su almacenamiento permanente. Para obtener más información, consulte [Visualización de registros de instancias de Amazon EC2 en su entorno de Elastic Beanstalk](#).

Comando

Especifique un comando para ejecutarlo en el contenedor. Si especifica un punto de entrada y, a continuación, se añade `Command` (Comando) como argumento en el punto de entrada. Para obtener más información, consulte [CMD](#) en la documentación de Docker.

Punto de entrada

Especifique un comando predeterminado para ejecutarse cuando se inicia el contenedor. Para obtener más información, consulte [ENTRYPOINT](#) en la documentación de Docker.

El siguiente fragmento es un ejemplo que ilustra la sintaxis del archivo `Dockerrun.aws.json` para un solo contenedor.

```
{  
  "AWSEBDockerrunVersion": "1",  
  "Image": {  
    "Name": "janedoe/image",  
    "Update": "true"  
  },  
  "Ports": [  
    {  
      "ContainerPort": "1234"  
    }  
  ],  
  "Volumes": [  
    {
```

```
    "HostDirectory": "/var/app/mydb",
    "ContainerDirectory": "/etc/mysql"
  }
],
"Logging": "/var/log/nginx",
"Entrypoint": "/app/bin/myapp",
"Command": "--argument"
}
```

Puede proporcionar a Elastic Beanstalk solo el archivo `Dockerrun.aws.json` o un archivo `.zip` que contenga los archivos `Dockerrun.aws.json` y `Dockerfile`. Cuando proporciona ambos archivos, el archivo `Dockerfile` describe la imagen de Docker y el archivo `Dockerrun.aws.json` proporciona información adicional para la implementación tal y como se describe más adelante en esta sección.

Note

Los dos archivos deben estar en la raíz, o en el nivel superior, del archivo `.zip`. No cree el archivo de almacenamiento desde un directorio que contenga los archivos. En lugar de ello, desplácese hasta ese directorio y cree allí el archivo de almacenamiento.

Cuando proporcione ambos archivos, no especifique una imagen en el archivo `Dockerrun.aws.json`. Elastic Beanstalk crea y usa la imagen descrita en `Dockerfile` e ignora la imagen especificada en el archivo `Dockerrun.aws.json`.

Uso de imágenes de un repositorio privado

Añada la información sobre el bucket de Amazon S3; que contiene el archivo de autenticación en el parámetro `Authentication` del archivo `Dockerrun.aws.json v1`. Asegúrese de que el parámetro `Authentication` contiene un bucket de Amazon S3 y una clave válidos. El bucket de Amazon S3 debe estar alojado en la misma Región de AWS que el entorno que lo está utilizando. Elastic Beanstalk no descarga archivos de buckets de Amazon S3 alojados en otras regiones.

Para obtener información sobre cómo generar y cargar el archivo de autenticación, consulte [Uso de imágenes de un repositorio privado](#).

El siguiente ejemplo muestra el uso de un archivo de autenticación denominado `mydockercfg` en un bucket denominado `DOC-EXAMPLE-BUCKET` para utilizar una imagen privada de un registro de terceros.

```
{
  "AWSEBDockerrunVersion": "1",
  "Authentication": {
    "Bucket": "DOC-EXAMPLE-BUCKET",
    "Key": "mydockercfg"
  },
  "Image": {
    "Name": "quay.io/johndoe/private-image",
    "Update": "true"
  },
  "Ports": [
    {
      "ContainerPort": "1234"
    }
  ],
  "Volumes": [
    {
      "HostDirectory": "/var/app/mydb",
      "ContainerDirectory": "/etc/mysql"
    }
  ],
  "Logging": "/var/log/nginx"
}
```

Configuración de los entornos de Docker

Hay varias maneras de configurar el comportamiento de su entorno de Docker de Elastic Beanstalk.

Note

Si su entorno Elastic Beanstalk utiliza una versión de la plataforma Docker AMI de Amazon Linux (anterior a Amazon Linux 2), asegúrese de leer la información adicional en [the section called “Configuración de Docker en la AMI de Amazon Linux \(anterior a Amazon Linux 2\)”](#).

Secciones

- [Configuración de software en entornos de Docker](#)
- [Referencia a variables de entorno en contenedores](#)
- [Uso de la característica de interpolación para variables de entorno \(Docker Compose\)](#)
- [Creación de registros para informes de estado avanzados \(Docker Compose\)](#)

- [Registro personalizado del contenedor de Docker \(Docker Compose\)](#)
- [Imágenes de Docker](#)
- [Configuración de actualizaciones administradas para entornos de Docker](#)
- [Espacios de nombres de la configuración de Docker](#)
- [Configuración de Docker en la AMI de Amazon Linux \(anterior a Amazon Linux 2\)](#)

Configuración de software en entornos de Docker

Puede utilizar la consola de Elastic Beanstalk para configurar el software que se ejecuta en las instancias del entorno.

Para configurar el entorno Docker en la consola de Elastic Beanstalk

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Actualizaciones, supervisión y registro, seleccione Editar.
5. Realice los cambios de configuración necesarios.
6. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

Para obtener información sobre cómo configurar el software en cualquier entorno, consulte [the section called “Propiedades del entorno y configuración de software”](#). En las siguientes secciones se aborda información específica de Docker.

Opciones de contenedor

La sección Container options (Opciones de contenedor) contiene opciones específicas de la plataforma. Para los entornos de Docker, le permite elegir si su entorno incluirá o no el servidor proxy NGINX.

Entornos con Docker Compose

Si administra su entorno de Docker con Docker Compose, Elastic Beanstalk presupone que va a ejecutar un servidor proxy como contenedor. Por lo tanto, Elastic Beanstalk establece el valor None (Ninguno) en la opción Proxy server (Servidor proxy) y no proporciona una configuración NGINX.

Note

Incluso si selecciona NGINX como servidor proxy, esta configuración se omite en un entorno con Docker Compose. La opción Proxy server (Servidor proxy) seguirá teniendo el valor predeterminado None (Ninguno).

Como el proxy del servidor web NGINX está deshabilitado para la plataforma de Docker en Amazon Linux 2 con Docker Compose, debe seguir las instrucciones para generar registros para informes de estado mejorados. Para obtener más información, consulte [Creación de registros para informes de estado avanzados \(Docker Compose\)](#).

Propiedades del entorno y variables de entorno

La sección Environment Properties (Propiedades de entorno) le permite especificar opciones de configuración del entorno en las instancias de Amazon Elastic Compute Cloud (Amazon EC2) que ejecutan la aplicación. Las propiedades del entorno se pasan como pares de clave-valor a la aplicación. En un entorno de Docker, Elastic Beanstalk pasa las propiedades de entorno a contenedores como variables de entorno.

El código de la aplicación que se ejecuta en un contenedor puede hacer referencia a una variable de entorno por su nombre y leer su valor. El código fuente que lee estas variables de entorno variará en función del lenguaje de programación. Encontrará instrucciones para leer los valores de las variables de entorno en los lenguajes de programación que son compatibles con las plataformas administradas por Elastic Beanstalk en el tema correspondiente a la plataforma de que se trate. Para obtener una lista de enlaces a estos temas, consulte [the section called “Propiedades del entorno y configuración de software”](#).

Entornos con Docker Compose

Si administra su entorno de Docker con Docker Compose, debe realizar alguna configuración adicional para recuperar las variables de entorno en los contenedores. Para que los ejecutables que se ejecutan en su contenedor tengan acceso a estas variables de entorno, debe hacer referencia a ellas en el archivo `docker-compose.yml`. Para obtener más información, consulte [Referencia a variables de entorno en contenedores](#).

Referencia a variables de entorno en contenedores

Si utiliza la herramienta Docker Compose en la plataforma de Docker en Amazon Linux 2, Elastic Beanstalk genera un archivo de entorno de Docker Compose llamado `.env` en el directorio raíz del proyecto de aplicación. Este archivo almacena las variables de entorno que ha configurado para Elastic Beanstalk.

Note

Si incluye un archivo `.env` en el paquete de la aplicación, Elastic Beanstalk no generará un archivo `.env`.

Para que un contenedor haga referencia a las variables de entorno que define en Elastic Beanstalk, debe seguir uno o ambos de estos enfoques de configuración.

- Agregue el archivo `.env` generado por Elastic Beanstalk a la opción de configuración `env_file` del archivo `docker-compose.yml`.
- Defina directamente las variables de entorno en el archivo `docker-compose.yml`.

Los siguientes archivos proporcionan un ejemplo. El archivo `docker-compose.yml` de ejemplo muestra ambos enfoques.

- Si define las propiedades del entorno `DEBUG_LEVEL=1` y `LOG_LEVEL=error`, Elastic Beanstalk genera el siguiente archivo `.env` automáticamente:

```
DEBUG_LEVEL=1
LOG_LEVEL=error
```

- En este archivo `docker-compose.yml`, la opción de configuración `env_file` apunta al archivo `.env` y también define la variable de entorno `DEBUG=1` directamente en el archivo `docker-compose.yml`.

```
services:
  web:
    build: .
    environment:
      - DEBUG=1
    env_file:
```

```
- .env
```

Notas

- Si establece la misma variable de entorno en ambos archivos, la variable definida en el archivo `docker-compose.yml` prevalece sobre la variable definida en el archivo `.env`.
- Tenga cuidado de no dejar espacios entre el signo igual (=) y el valor asignado a la variable para evitar que se agreguen espacios a la cadena.

Para obtener más información sobre las variables de entorno en Docker Compose, consulte [Environment variables in Compose](#).

Uso de la característica de interpolación para variables de entorno (Docker Compose)

A partir de la versión de plataforma del [28 de julio de 2023](#), la ramificación de la plataforma Docker Amazon Linux 2 ofrece la característica de interpolación de Docker Compose. Con esta característica, los valores de un archivo de Compose se pueden establecer mediante variables e interpolarlos en el tiempo de ejecución. Para obtener más información sobre esta característica, consulte [Interpolación](#) en el sitio web de documentación de Docker.

Important

Si quiere utilizar esta característica con sus aplicaciones, tenga en cuenta que tendrá que implementar un enfoque que utilice enlaces de plataforma. Esto es necesario debido a una mitigación que implementamos en el motor de la plataforma. La mitigación garantiza la compatibilidad con versiones anteriores para los clientes que no conocen la nueva característica de interpolación y que ya tienen aplicaciones que utilizan variables de entorno con el carácter `$`. El motor de plataforma actualizado evita la interpolación de forma predeterminada al reemplazar el personaje `$` por caracteres `$$`.

El siguiente es un ejemplo de un script de enlace de plataforma que puede configurar para permitir el uso de la característica de interpolación.

```
#!/bin/bash
```

```
: '
example data format in .env file
key1=value1
key2=value2
'

envfile="/var/app/staging/.env"
tempfile=$(mktemp)

while IFS= read -r line; do
  # split each env var string at '='
  split_str=(${line//=/ })
  if [ ${#split_str[@]} -eq 2 ]; then
    # replace '$$' with '$'
    replaced_str=${split_str[1]/\$/\$/}
    # update the value of env var using ${replaced_str}
    line="${split_str[0]}=${replaced_str}"
  fi
  # append the updated env var to the tempfile
  echo "${line}" #"${tempfile}"
done < "${envfile}"
# replace the original .env file with the tempfile
mv "${tempfile}" "${envfile}"
```

Coloque los enlaces de la plataforma en estos dos directorios:

- `.platform/confighooks/predeploy/`
- `.platform/hooks/predeploy/`

Para obtener más información, consulte [Enlaces de la plataforma](#) en el tema Ampliar las plataformas Linux de esta guía.

Creación de registros para informes de estado avanzados (Docker Compose)

El [agente de salud de Elastic Beanstalk](#) proporciona métricas de estado del sistema operativo y de las aplicaciones para entornos de Elastic Beanstalk. Se basa en los formatos de registro del servidor web que transmiten información en un formato específico.

Elastic Beanstalk presupone que ejecuta un proxy de servidor web como un contenedor. Por consiguiente, el proxy del servidor web NGINX está deshabilitado para los entornos de Docker que ejecutan Docker Compose. Debe configurar el servidor para que escriba registros en la ubicación y

con el formato que utiliza el agente de estado de Elastic Beanstalk. De esta forma, podrá aprovechar al máximo los informes de estado mejorados, aunque el proxy del servidor web esté deshabilitado.

Para obtener instrucciones al respecto, consulte [Configuración de los registros de servidor web](#)

Registro personalizado del contenedor de Docker (Docker Compose)

Para solucionar problemas de manera eficiente y monitorear sus servicios en contenedores, puede [solicitar registros de instancias](#) de Elastic Beanstalk desde la consola de administración del entorno o desde la CLI de EB. Los registros de instancias se componen de registros de paquetes y registros de cola, combinados y empaquetados, para permitirle ver registros y eventos recientes de una manera eficiente y rápida.

Elastic Beanstalk crea los directorios de registro en la instancia del contenedor, uno para cada servicio definido en el archivo `docker-compose.yml`, en `/var/log/eb-docker/containers/<service name>`. Si utiliza la característica Docker Compose en la plataforma de Docker en Amazon Linux 2, puede montar estos directorios en la ubicación dentro de la estructura de archivos del contenedor donde se escriben los registros. Si monta directorios de registro para escribir datos de registro, Elastic Beanstalk podrá recopilar datos de registro de estos directorios.

Si sus aplicaciones están en una plataforma de Docker que no utiliza Docker Compose, puede seguir el procedimiento estándar descrito en [Registro personalizado del contenedor de Docker \(Docker Compose\)](#).

Para configurar los archivos de registros del servicio como archivos de cola y registros de paquetes recuperables

1. Edite el archivo `docker-compose.yml`.
2. Debajo de la clave `volumes` de su servicio, agregue un montaje de enlace de la manera siguiente:

```
"${EB_LOG_BASE_DIR}/<service name>:<log directory inside container>
```

En el siguiente archivo `docker-compose.yml` de ejemplo:

- `nginx-proxy` es *<nombre del servicio>*
- `/var/log/nginx` es *<directorio de registro dentro del contenedor>*

```
services:
```

```
nginx-proxy:
  image: "nginx"
  volumes:
    - "${EB_LOG_BASE_DIR}/nginx-proxy:/var/log/nginx"
```

- El directorio `var/log/nginx` contiene los registros del servicio `nginx-proxy` en el contenedor y se corresponde con el directorio `/var/log/eb-docker/containers/nginx-proxy` del host.
- Todos los registros de este directorio ahora se pueden recuperar como registros de paquete y cola a través de la funcionalidad [Solicitar registros de instancia](#) de Elastic Beanstalk.

Notas

- `EB_LOG_BASE_DIR` es una variable de entorno establecida por Elastic Beanstalk con el valor `/var/log/eb-docker/containers`.
- Elastic Beanstalk crea automáticamente el directorio `/var/log/eb-docker/containers/<service name>` para cada servicio del archivo `docker-compose.yml`.

Imágenes de Docker

Las ramificaciones de la plataforma Docker administradas por Docker y ECS para Elastic Beanstalk admiten el uso de imágenes de Docker almacenadas en un repositorio de imágenes en línea público o privado.

Especifique las imágenes por su nombre en `DockerRun.aws.json`. Tenga en cuenta estas convenciones:

- Las imágenes de los repositorios oficiales de Docker Hub utilizan un solo nombre (por ejemplo, `ubuntu` o `mongo`).
- Las imágenes de otros repositorios de Docker Hub se identifican con un nombre de organización (por ejemplo, `amazon/amazon-ecs-agent`).
- Las imágenes de otros repositorios online se designan además con un nombre de dominio (por ejemplo, `quay.io/assemblyline/ubuntu` o `account-id.dkr.ecr.us-east-2.amazonaws.com/ubuntu:trusty`).

En entornos que utilizan únicamente la plataforma Docker, también puede crear su propia imagen durante la creación del entorno con un archivo Dockerfile. Para obtener más información, consulte [Creación de imágenes personalizadas con un Dockerfile](#). La plataforma Multicontainer Docker no admite esta funcionalidad.

Uso de imágenes de un repositorio de Amazon ECR

Puede almacenar sus imágenes de Docker personalizadas AWS con [Amazon Elastic Container Registry](#) (Amazon ECR). Cuando guarda sus imágenes de Docker en Amazon ECR, Elastic Beanstalk se autentica automáticamente en el registro de Amazon ECR con el [perfil de instancia](#) del entorno para que no sea necesario [generar un archivo de autenticación](#) y cargarlo en Amazon Simple Storage Service (Amazon S3).

No obstante, sí es necesario que proporcione a las instancias permiso para obtener acceso a las imágenes del repositorio de Amazon ECR. Para ello, agregue permisos al perfil de instancia del entorno. Puede adjuntar la política ContainerRegistryReadOnly gestionada de [AmazonEC2](#) al perfil de la instancia para proporcionar acceso de solo lectura a todos los repositorios de Amazon ECR de su cuenta, o conceder el acceso a un único repositorio mediante la siguiente plantilla para crear una política personalizada:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowEbAuth",
      "Effect": "Allow",
      "Action": [
        "ecr:GetAuthorizationToken"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Sid": "AllowPull",
      "Effect": "Allow",
      "Resource": [
        "arn:aws:ecr:us-east-2:account-id:repository/repository-name"
      ],
      "Action": [
        "ecr:GetAuthorizationToken",
        "ecr:BatchCheckLayerAvailability",
```

```

        "ecr:GetDownloadUrlForLayer",
        "ecr:GetRepositoryPolicy",
        "ecr:DescribeRepositories",
        "ecr:ListImages",
        "ecr:BatchGetImage"
    ]
}
]
}

```

Sustituya el nombre de recurso de Amazon (ARN) en la política anterior por el ARN del repositorio.

En el archivo `Dockerrun.aws.json`, haga referencia a la imagen utilizando la URL. En la [plataforma Docker](#), la URL va en la definición de Image:

```

"Image": {
    "Name": "account-id.dkr.ecr.us-east-2.amazonaws.com/repository-name:latest",
    "Update": "true"
},

```

En la [plataforma Multicontainer Docker](#), utilice la clave `image` en un objeto de la definición de contenedor:

```

"containerDefinitions": [
    {
        "name": "my-image",
        "image": "account-id.dkr.ecr.us-east-2.amazonaws.com/repository-name:latest",

```

Uso de imágenes de un repositorio privado

Para utilizar una imagen de Docker de un repositorio privado hospedado por un registro online, debe proporcionar un archivo de autenticación que contenga la información necesaria para autenticarse con el registro.

Genere un archivo de autenticación con el comando `docker login`. En el caso de los repositorios de Docker Hub, ejecute `docker login`:

```
$ docker login
```

En el caso de otros registros, incluya la URL del servidor del registro:

```
$ docker login registry-server-url
```

Note

Si su entorno de Elastic Beanstalk utiliza una versión de la plataforma Docker AMI de Amazon Linux (anterior a Amazon Linux 2), lea la información adicional en [the section called “Configuración de Docker en la AMI de Amazon Linux \(anterior a Amazon Linux 2\)”](#).

Cargue una copia llamada `.dockerrcfg` del archivo de autenticación en un bucket de Amazon S3 seguro. El bucket de Amazon S3 debe estar alojado en la misma AWS región que el entorno que lo utiliza. Elastic Beanstalk no puede descargar archivos desde un bucket de Amazon S3 alojado en otras regiones. Conceda permisos para la operación `s3:GetObject` al rol de IAM en el perfil de instancia. Para obtener más información, consulte [Administración de perfiles de instancia de Elastic Beanstalk](#).

Incluya la información del bucket de Amazon S3 en el parámetro `Authentication (v1)` o `authentication (v2)` del archivo `Dockerrun.aws.json`.

Para obtener más información sobre el formato `Dockerrun.aws.json` de los entornos Docker, consulte [Configuración de Docker](#). En el caso de entornos con varios contenedores, consulte [Configuración de Docker administrado por ECS](#).

Para obtener más información sobre el archivo de autenticación, consulte [Store images on Docker Hub](#) y [docker login](#) en el sitio web de Docker.

Configuración de actualizaciones administradas para entornos de Docker

Con las [actualizaciones de plataforma administradas](#), puede configurar el entorno para que se actualice automáticamente a la última versión de una plataforma de manera programada.

En el caso de los entornos de Docker, es posible que decida si una actualización de plataforma automática debe suceder en varias versiones de Docker, si la nueva versión de la plataforma incluye una nueva versión de Docker. Elastic Beanstalk admite actualizaciones de plataformas administradas en todas las versiones de Docker cuando se actualiza desde un entorno que ejecuta una versión de plataforma Docker más reciente que 2.9.0. Cuando una nueva versión de la plataforma incluye una nueva versión de Docker, Elastic Beanstalk aumenta el número de versión de actualización secundaria. Por lo tanto, para permitir las actualizaciones de plataforma administradas en diferentes versiones de Docker, habilítelas para las actualizaciones de versiones secundarias y de parche. Para

evitar las actualizaciones de plataforma administradas en diferentes versiones de Docker, habilítelas para que solo apliquen las actualizaciones de versión de parche.

Por ejemplo, el siguiente [archivo de configuración](#) habilita las actualizaciones de plataforma administradas a las 9:00 h UTC cada martes para las actualizaciones de versiones secundarias y de parche, lo que permite las actualizaciones administradas en diferentes versiones de Docker:

Example `.ebextensions/ .config managed-platform-update`

```
option_settings:
  aws:elasticbeanstalk:managedactions:
    ManagedActionsEnabled: true
    PreferredStartTime: "Tue:09:00"
  aws:elasticbeanstalk:managedactions:platformupdate:
    UpdateLevel: minor
```

En los entornos que ejecutan versiones de plataforma de Docker 2.9.0 o anteriores, Elastic Beanstalk nunca realiza las actualizaciones de plataforma administradas si la nueva versión de plataforma incluye una nueva versión de Docker.

Espacios de nombres de la configuración de Docker

Puede usar un [archivo de configuración](#) para definir opciones de configuración y realizar otras tareas de configuración en las instancias durante las implementaciones. Las opciones de configuración se pueden definir a través del servicio de Elastic Beanstalk o la plataforma que utilice y están organizadas por espacios de nombres.

Note

Esta información solo se aplica al entorno de Docker que no ejecuta Docker Compose. Esta opción tiene un comportamiento diferente con los entornos de Docker que ejecutan Docker Compose. Para obtener más información sobre los servicios proxy con Docker Compose, consulte [Opciones de contenedor](#).

La plataforma Docker admite las opciones de los siguientes espacios de nombres, además de las [opciones admitidas para todos los entornos de Elastic Beanstalk](#):

- `aws:elasticbeanstalk:environment:proxy`: Elija el servidor proxy para su entorno. Docker admite que se ejecute Nginx o ningún servidor proxy.

El siguiente archivo de configuración de ejemplo configura un entorno de Docker de modo que no ejecute ningún servidor proxy.

Example `.ebextensions/docker-settings.config`

```
option_settings:
  aws:elasticbeanstalk:environment:proxy:
    ProxyServer: none
```

Configuración de Docker en la AMI de Amazon Linux (anterior a Amazon Linux 2)

Si su entorno de Elastic Beanstalk Docker utiliza una versión de la plataforma AMI de Amazon Linux (anterior a Amazon Linux 2), lea la información adicional de esta sección.

Uso de un archivo de autenticación para un repositorio privado

Esta información le interesa si está [usando imágenes de un repositorio privado](#). A partir de la versión 1.7 de Docker, el comando `docker login` cambió el nombre del archivo de autenticación y el formato del archivo. Las versiones de la plataforma Docker en la AMI de Amazon Linux (anteriores a Amazon Linux 2) requieren el archivo de configuración anterior en formato `~/.dockercfg`.

Con la versión 1.7 de Docker y otras posteriores, el comando `docker login` crea el archivo de autenticación en `~/.docker/config.json` con el siguiente formato.

```
{
  "auths":{
    "server":{
      "auth":"key"
    }
  }
}
```

Con la versión 1.6.2 de Docker y anteriores, el comando `docker login` crea el archivo de autenticación en `~/.dockercfg` con el siguiente formato.

```
{
  "server" :
  {
    "auth" : "auth_token",
    "email" : "email"
  }
}
```

```
}
```

Para convertir un archivo `config.json`, elimine la clave `auths` externa, añada una clave `email` y adapte el documento JSON para que se ajuste al formato antiguo.

En las versiones de la plataforma Docker de Amazon Linux 2, Elastic Beanstalk utiliza el nombre y el formato más recientes del archivo de autenticación. Si está utilizando una versión de plataforma Docker de Amazon Linux 2, puede usar el archivo de autenticación que el comando `docker login` crea sin aplicar ninguna conversión.

Configuración de volúmenes de almacenamiento adicionales

Para mejorar el rendimiento de la AMI de Amazon Linux, Elastic Beanstalk configura dos volúmenes de almacenamiento de Amazon EBS para las instancias de Amazon EC2 del entorno de Docker. Además del volumen raíz que se aprovisiona para todos los entornos de Elastic Beanstalk, se aprovisiona un segundo volumen de 12 GB llamado `xvdcz` para el almacenamiento de imágenes en entornos de Docker.

Si necesita más espacio de almacenamiento o un mayor número de IOPS para las imágenes de Docker, puede personalizar el volumen de almacenamiento de imágenes utilizando la opción de configuración `BlockDeviceMapping` del espacio de nombres [aws:autoscaling:launchconfiguration](#).

Por ejemplo, el siguiente [archivo de configuración](#) aumenta el tamaño del volumen de almacenamiento a 100 GB con 500 IOPS provisionadas:

Example `.ebextensions/blockdevice-xvdcz.config`

```
option_settings:
  aws:autoscaling:launchconfiguration:
    BlockDeviceMappings: /dev/xvdcz=:100::io1:500
```

Si utiliza la opción `BlockDeviceMappings` para configurar volúmenes adicionales para la aplicación, debería incluir un mapeo de `xvdcz` para asegurarse de que se ha creado. En el siguiente ejemplo, se configuran dos volúmenes: el volumen de almacenamiento de imágenes `xvdcz` con la configuración predeterminada y un volumen de aplicaciones adicional de 24 GB llamado `sdh`:

Example `.ebextensions/blockdevice-sdh.config`

```
option_settings:
  aws:autoscaling:launchconfiguration:
```



```
BlockDeviceMappings: /dev/xvdcz=:12:true:gp2,/dev/sdh=:24
```

Note

Cuando cambie la configuración de este espacio de nombres, Elastic Beanstalk sustituirá todas las instancias del entorno por instancias que ejecuten la nueva configuración. Para obtener más información, consulte [Cambios de configuración](#).

Uso de la ramificación de la plataforma de Amazon ECS

En este tema se aborda la ramificación de la plataforma Amazon ECS on Amazon Linux 2 y la ramificación de la plataforma que reemplaza, Multicontainer Docker on AL1 (también administrada por ECS). A menos que se indique lo contrario, toda la información de este tema se aplica a ambas ramificaciones de la plataforma.

Note

El [18 de julio de 2022](#), Elastic Beanstalk estableció el estado de todas las ramificaciones de la plataforma basadas en la AMI de Amazon Linux (AL1) como retirado.

Migrar desde Docker de múltiples contenedores en AL1

Si actualmente utiliza la ramificación de la plataforma retirada Docker multicontenedor sobre AL1, puede migrar a la ramificación de la plataforma ECS sobre AL2023 más reciente. La ramificación de la plataforma más reciente es compatible con todas las características de esta ramificación de la plataforma discontinuada. No es necesario realizar cambios en el código fuente. Para obtener más información, consulte [Migración de Docker multicontenedor sobre Amazon Linux a ECS sobre Amazon Linux 2023](#).

Temas

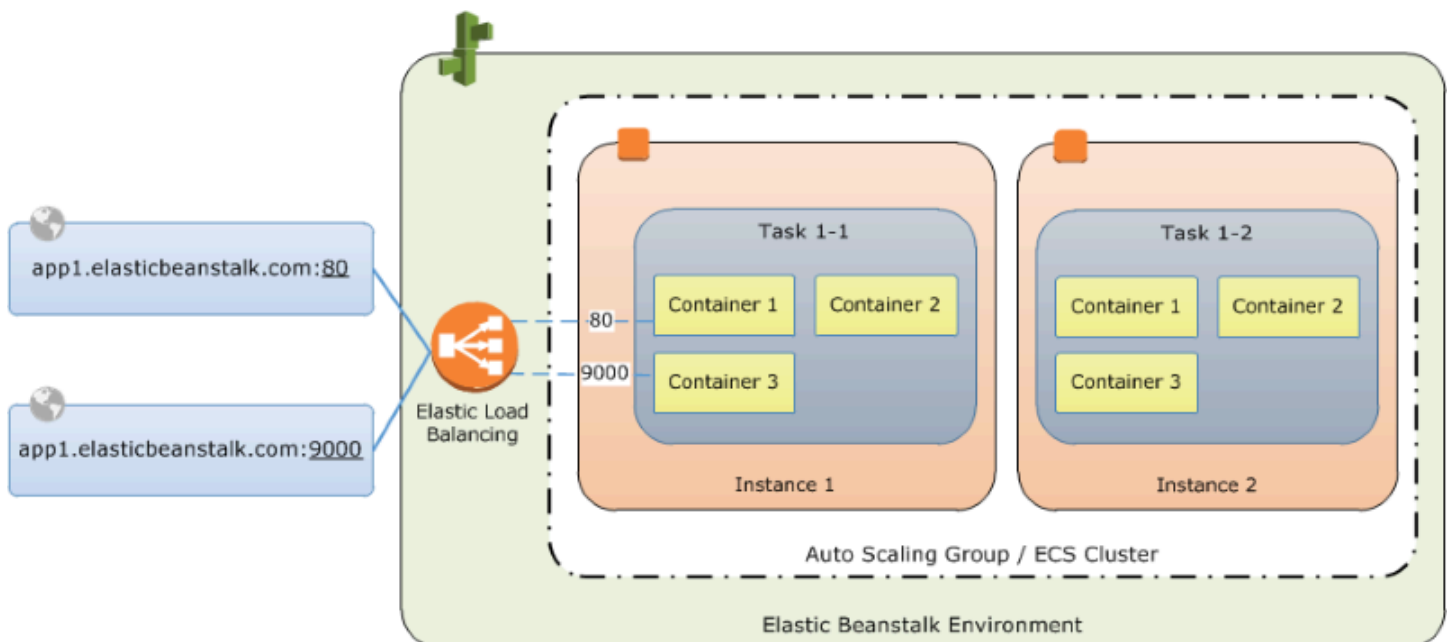
- [Plataforma Docker administrada por ECS](#)
- [Dockerrun.aws.json file](#)
- [Imágenes de Docker](#)
- [Rol de la instancia de contenedor](#)
- [Recursos de Amazon ECS creados por Elastic Beanstalk](#)

- [Uso de varios agentes de escucha de Elastic Load Balancing](#)
- [Implementaciones de contenedores con errores](#)
- [Configuración de Docker administrado por ECS](#)
- [Entornos de Docker administrados por ECS con la consola de Elastic Beanstalk](#)
- [Migración de Docker multicontenedor sobre Amazon Linux a ECS sobre Amazon Linux 2023](#)
- [Migración \(heredada\) a la ramificación de la plataforma Docker que se ejecuta en Amazon Linux 2 desde Multicontainer Docker que se ejecuta en Amazon Linux](#)

Plataforma Docker administrada por ECS

Elastic Beanstalk utiliza Amazon Elastic Container Service (Amazon ECS) para coordinar las implementaciones de contenedores en entornos de Docker administrados por ECS. Amazon ECS dispone de herramientas para administrar un clúster de instancias que ejecutan contenedores de Docker. Elastic Beanstalk se encarga de las tareas de Amazon ECS, incluida la creación de clústeres, la definición y la ejecución de tareas. Todas las instancias del entorno ejecutan el mismo conjunto de contenedores, que está definido en un archivo `DockerTaskDefinition.json` de la v2. Para sacar el máximo provecho a Docker, Elastic Beanstalk le permite crear un entorno en el que las instancias Amazon EC2 ejecutan varios contenedores de Docker en paralelo.

El siguiente diagrama muestra un entorno de ejemplo de Elastic Beanstalk configurado con tres contenedores Docker que se ejecutan en cada instancia de Amazon EC2 de un grupo de Auto Scaling:



Note

Elastic Beanstalk ofrece características de extensibilidad para todas sus plataformas que se pueden utilizar para personalizar la implementación y la ejecución de sus aplicaciones. Para la ramificación de la plataforma ECS que se ejecuta en Amazon Linux 2, la implementación del flujo de trabajo de la implementación de las instancias de estas características varía de las demás plataformas. Para obtener más información, consulte [Flujo de trabajo de implementación de instancias para ECS sobre Amazon Linux 2 y posterior](#).

Dockerrun.aws.json file

Las instancias de contenedor (instancias de Amazon EC2 que ejecutan Docker administrado por ECS en un entorno de Elastic Beanstalk) requieren un archivo de configuración denominado `Dockerrun.aws.json`. Este archivo es específico para Elastic Beanstalk y puede utilizarse solo o junto con el código fuente y el contenido de un [paquete de código fuente](#) para crear un entorno en una plataforma de Docker.

Note

La versión 1 del formato `Dockerrun.aws.json` se utiliza para lanzar un único contenedor de Docker en un entorno de Elastic Beanstalk que se ejecuta en la AMI de Amazon Linux (la versión que precede a Amazon Linux 2). El entorno se basa en la ramificación de la plataforma Docker que se ejecuta en Amazon Linux de 64 bits, que se retirará el 18 de julio de 2022. Para obtener más información sobre el formato `Dockerrun.aws.json` de la v1, consulte [Configuración de la plataforma de Docker: sin Docker Compose](#).

El formato de la versión 2 de `Dockerrun.aws.json` agrega compatibilidad con varios contenedores en cada instancia de Amazon EC2 y solo puede utilizarse con la plataforma Docker administrada por ECS. El formato difiere de forma considerable de la versión anterior.

Consulte [Dockerrun.aws.json v2](#) para obtener más información sobre el formato actualizado y un archivo de ejemplo.

Imágenes de Docker

La plataforma Docker administrada por ECS de Elastic Beanstalk necesita imágenes prediseñadas y almacenadas en un repositorio de imágenes en línea público o privado.

Note

En la plataforma Docker administrada por ECS de Elastic Beanstalk, no se pueden crear imágenes personalizadas durante la implementación con un archivo `Dockerfile`. Cree las imágenes e impleméntelas en un repositorio online antes de crear un entorno de Elastic Beanstalk.

Especifique las imágenes por su nombre en `Dockerrun.aws.json` de la v2. Tenga en cuenta estas convenciones:

- Las imágenes de los repositorios oficiales de Docker Hub utilizan un solo nombre (por ejemplo, `ubuntu` o `mongo`).
- Las imágenes de otros repositorios de Docker Hub se identifican con un nombre de organización (por ejemplo, `amazon/amazon-ecs-agent`).
- Las imágenes de otros registros online se identifican además con un nombre de dominio (por ejemplo, `quay.io/assemblyline/ubuntu`).

Si desea configurar Elastic Beanstalk para que se autentique en un repositorio privado, incluya el parámetro `authentication` en el archivo `Dockerrun.aws.json` de la v2.

Rol de la instancia de contenedor

Elastic Beanstalk utiliza una AMI optimizada para Amazon ECS con un agente contenedor de Amazon ECS que se ejecuta en un contenedor Docker. El agente se comunica con Amazon ECS para coordinar las implementaciones de contenedores. Para poder comunicarse con Amazon ECS, cada instancia de Amazon EC2; debe tener los permisos correspondientes en IAM;. Estos permisos se adjuntan al [perfil de instancias](#) predeterminado cuando se crea un entorno en la consola de Elastic Beanstalk:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ECSAccess",
      "Effect": "Allow",
      "Action": [
        "ecs:Poll",
        "ecs:StartTask",
```

```
        "ecs:StopTask",
        "ecs:DiscoverPollEndpoint",
        "ecs:StartTelemetrySession",
        "ecs:RegisterContainerInstance",
        "ecs:DeregisterContainerInstance",
        "ecs:DescribeContainerInstances",
        "ecs:Submit*"
    ],
    "Resource": "*"
}
]
```

Si crea su propio perfil de instancia, puede adjuntar la política administrada `AWSElasticBeanstalkMulticontainerDocker` para asegurarse de que los permisos se mantienen actualizados. Para obtener instrucciones sobre la creación de políticas y roles de IAM, consulte [Creating IAM Roles](#) en la Guía del usuario de IAM;

Recursos de Amazon ECS creados por Elastic Beanstalk

Cuando se crea un entorno con la plataforma Docker administrada por ECS, Elastic Beanstalk crea y configura de forma automática varios recursos de Amazon Elastic Container Service mientras crea el entorno. De este modo, se crean los contenedores necesarios en cada instancia de Amazon EC2.

- Clúster de Amazon ECS las instancias de contenedor de Amazon ECS están organizadas en clústeres. Cuando se utiliza con Elastic Beanstalk, siempre se crea un clúster para cada entorno de Docker administrado por ECS.
- Definición de tareas de Amazon ECS: Elastic Beanstalk utiliza el archivo `Dockerrun.aws.json` de la v2 del proyecto para generar la definición de tareas de Amazon ECS que se utiliza para configurar las instancias de contenedor del entorno.
- Tarea de Amazon ECS: Elastic Beanstalk se comunica con Amazon ECS para ejecutar una tarea en cada instancia del entorno para coordinar la implementación de contenedores. En un entorno escalable, Elastic Beanstalk inicia una nueva tarea siempre que se agrega una instancia al clúster. En casos excepcionales, es posible que tenga que aumentar la cantidad de espacio reservada para contenedores e imágenes. Obtenga más información en la sección [Configuración de los entornos de Docker](#).
- Agente de contenedor de Amazon ECS: el agente se ejecuta en un contenedor de Docker de las instancias del entorno. El agente sondea el servicio Amazon ECS y espera a que se ejecute una tarea.

- Volúmenes de datos de Amazon ECS: Elastic Beanstalk inserta definiciones de volúmenes (además de los volúmenes definidos en `DockerRun.aws.json` de la v2) en la definición de tareas para facilitar la recopilación de registros.

Elastic Beanstalk crea volúmenes de logs en la instancia de contenedor, uno para cada contenedor, en `/var/log/containers/containername`. Estos volúmenes se llaman `awseb-logs-containername` y se proporcionan para los contenedores que se van a montar. Consulte [Formato de definición de contenedor](#) para obtener más información acerca de cómo montarlos.

Uso de varios agentes de escucha de Elastic Load Balancing

Puede configurar varios agentes de escucha de Elastic Load Balancing en un entorno de Docker administrado por ECS para admitir el tráfico entrante en los servidores proxy y otros servicios que no se ejecutan en el puerto HTTP predeterminado.

Cree una carpeta `.ebextensions` en el paquete de código fuente y agregue un archivo con la extensión `.config`. En el ejemplo siguiente, se muestra un archivo de configuración que crea un agente de escucha de Elastic Load Balancing en el puerto 8080.

.ebextensions/elb-listener.config

```
option_settings:
  aws:elb:listener:8080:
    ListenerProtocol: HTTP
    InstanceProtocol: HTTP
    InstancePort: 8080
```

Si su entorno se ejecuta en una [Amazon Virtual Private Cloud](#) (Amazon VPC) personalizada que creó, Elastic Beanstalk se encarga del resto. En una VPC predeterminada, debe configurar el grupo de seguridad de la instancia para que permita la entrada desde el balanceador de carga. Utilice un segundo archivo de configuración que agregue una regla de entrada al grupo de seguridad:

.ebextensions/elb-ingress.config

```
Resources:
  port8080SecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
```

```
IpProtocol: tcp
ToPort: 8080
FromPort: 8080
SourceSecurityGroupName: { "Fn::GetAtt": ["AWSEBLoadBalancer",
"SourceSecurityGroup.GroupName"] }
```

Para obtener más información sobre el formato de los archivos de configuración, consulte [Añadir y personalizar recursos del entorno de Elastic Beanstalk](#) y [Opciones de configuración](#).

Además de agregar un agente de escucha a la configuración de Elastic Load Balancing y de abrir un puerto en el grupo de seguridad, debe asignar el puerto de la instancia del host al puerto del contenedor de Docker en la sección `containerDefinitions` del archivo `Dockerrun.aws.json` de la v2. En el fragmento siguiente, se muestra un ejemplo:

```
"portMappings": [
  {
    "hostPort": 8080,
    "containerPort": 8080
  }
]
```

Consulte [Dockerrun.aws.json v2](#) para obtener más información sobre el formato del archivo `Dockerrun.aws.json` de la v2.

Implementaciones de contenedores con errores

Si falla una tarea de Amazon ECS, uno o más contenedores en su entorno de Elastic Beanstalk no se iniciarán. Elastic Beanstalk no restaura entornos de varios contenedores debido a una tarea de Amazon ECS que no se realizó. Si un contenedor no puede iniciarse en el entorno, vuelva a implementar la versión actual o una versión anterior que funcionara desde la consola de Elastic Beanstalk.

Para implementar una versión existente

1. Abra la consola de Elastic Beanstalk en la región de su entorno.
2. Haga clic en Actions (Acciones) a la derecha del nombre de la aplicación y luego en View application versions (Ver versiones de la aplicación).
3. Seleccione una versión de la aplicación y haga clic en Deploy (Implementar).

Configuración de Docker administrado por ECS

`Dockerrun.aws.json` es un archivo de configuración de Elastic Beanstalk que describe cómo implementar un conjunto de contenedores de Docker alojados en un clúster ECS en un entorno de Elastic Beanstalk. La plataforma Elastic Beanstalk crea una definición de tarea de ECS, que incluye una definición de contenedor de ECS. Estas definiciones se describen en el archivo de configuración `Dockerrun.aws.json`.

La definición de contenedor del archivo `Dockerrun.aws.json` describe los contenedores que se van a implementar en cada instancia de Amazon EC2 del clúster de ECS. En este caso, una instancia de Amazon EC2 también se denomina instancia de contenedor de host, porque aloja los contenedores de Docker. El archivo de configuración también describe los volúmenes de datos que se deben crear en la instancia de contenedor de host para el montaje de los contenedores de Docker. Para obtener más información y un diagrama de los componentes de un entorno de Docker gestionado por ECS en Elastic Beanstalk, consulte la sección [Plataforma Docker administrada por ECS](#) anteriormente en este capítulo.

Se puede usar un archivo `Dockerrun.aws.json` tal como está o incluirlo en un solo archivo junto con el código fuente adicional. El código fuente archivado con un `Dockerrun.aws.json` se implementa en instancias de contenedor de Amazon EC2; y está accesible en el directorio `/var/app/current/`.

Temas

- [Dockerrun.aws.json v2](#)
- [Formato del volumen](#)
- [Formato de definición de contenedor](#)
- [Formato de autenticación: uso de imágenes de un repositorio privado](#)
- [Ejemplo Dockerrun.aws.json v2](#)

Dockerrun.aws.json v2

En el archivo `Dockerrun.aws.json` se incluyen las siguientes secciones:

AWSEBDockerrunVersion

Especifica el número de versión como el valor 2 para los entornos Docker administrados por ECS.

volumes

Crea volúmenes a partir de carpetas de la instancia de contenedor de Amazon EC2 o a partir del paquete de código fuente (implementado en `/var/app/current`). Monte estos volúmenes en rutas que estén dentro de los contenedores de Docker mediante `mountPoints` en la sección `containerDefinitions`.

containerDefinitions

Una matriz de definiciones de contenedor.

autenticación (opcional)

La ubicación en Amazon S3 de un archivo `.dockercfg` que contiene los datos de autenticación de un repositorio privado.

Las secciones `Definición de contenedor` y `Volúmenes` de `Dockerrun.aws.json` utilizan el mismo formato que las secciones correspondientes de un archivo de definición de tareas de Amazon ECS. Para obtener más información sobre el formato de definición de tarea y una lista completa de los parámetros de definición de tareas, consulte [Definiciones de tareas de Amazon ECS](#) en la Guía para el desarrollador del servicio de contenedores de Amazon Elastic.

Formato del volumen

El parámetro `volumen` crea volúmenes a partir de carpetas de la instancia de contenedor de Amazon EC2 o a partir del paquete de código fuente (implementado en `/var/app/current`).

Los volúmenes se especifican en el siguiente formato:

```
"volumes": [  
  {  
    "name": "volumename",  
    "host": {  
      "sourcePath": "/path/on/host/instance"  
    }  
  }  
],
```

Monte estos volúmenes en rutas que estén dentro de los contenedores de Docker mediante `mountPoints` en la definición de contenedor.

Elastic Beanstalk configura volúmenes adicionales para los logs, uno para cada contenedor. Estos los deben montar los contenedores de Docker para poder escribir logs en la instancia del host.

Para obtener más información, consulte el campo `mountPoints` de la sección Formato de definición de contenedor que aparece a continuación.

Formato de definición de contenedor

Los siguientes ejemplos muestran un subconjunto de parámetros que se utilizan habitualmente en la sección Definiciones de contenedor. Hay disponibles otros parámetros opcionales.

La plataforma Beanstalk crea una definición de tarea de ECS, que incluye una definición de contenedor de ECS. Beanstalk admite un subconjunto de parámetros para la definición del contenedor de ECS. Para obtener más información, consulte [Definiciones de contenedor](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

Un archivo `DockerRun.aws.json` contiene una matriz de uno o varios objetos de definición de contenedor con los siguientes campos:

name

El nombre del contenedor. Consulte [Standard Container Definition Parameters \(Parámetros de definición de contenedores estándar\)](#) para obtener información sobre la longitud máxima y los caracteres permitidos.

image

El nombre de una imagen de Docker en un repositorio Docker insertado desde el que va a crear un contenedor Docker. Tenga en cuenta estas convenciones:

- Las imágenes de los repositorios oficiales de Docker Hub utilizan un solo nombre (por ejemplo, `ubuntu` o `mongo`).
- Las imágenes de otros repositorios de Docker Hub se identifican con un nombre de organización (por ejemplo, `amazon/amazon-ecs-agent`).
- Las imágenes de otros repositorios online se cualifican más con un nombre de dominio (por ejemplo, `quay.io/assemblyline/ubuntu`).

environment

Una matriz de variables de entorno que se va a pasar al contenedor.

Por ejemplo, la siguiente entrada define una variable de entorno con el nombre **Container** y el valor **PHP**:

```
"environment": [  
  {  
    "name": "Container",  
    "value": "PHP"  
  }  
],
```

essential

Es True si la tarea se debe detener si el contenedor produce un error. Los contenedores Nonessential puedan terminar o bloquearse sin que ello afecte al resto de los contenedores de la instancia.

memory

Cantidad de memoria de la instancia de contenedor que se va a reservar para el contenedor. Especifique un entero distinto de cero para uno o ambos parámetros memory o memoryReservation en las definiciones de contenedor.

memoryReservation

El límite flexible (en MiB) de memoria que reservar para el contenedor. Especifique un entero distinto de cero para uno o ambos parámetros memory o memoryReservation en las definiciones de contenedor.

mountPoints

Volúmenes de la instancia de contenedor de Amazon EC2; que se van a montar y la ubicación en el sistema de archivos del contenedor de Docker en la que se van a montar. Cuando se montan volúmenes que incluyen contenido de aplicaciones, el contenedor puede leer los datos que se cargan en el paquete de código fuente. Cuando se montan volúmenes de logs para escribir datos de log, Elastic Beanstalk puede recopilar datos de log de estos volúmenes.

Elastic Beanstalk crea volúmenes de logs en la instancia de contenedor, uno para cada contenedor de Docker, en `/var/log/containers/containername`. Estos volúmenes se denominan `awseb-logs-containername` y deben montarse en la ubicación dentro de la estructura de archivos del contenedor donde se escriben los logs.

Por ejemplo, el siguiente punto de montaje asigna la ubicación de los registros de nginx en el contenedor al volumen generado por Elastic Beanstalk para el contenedor `nginx-proxy`.

```
{
```

```
"sourceVolume": "awseb-logs-nginx-proxy",
"containerPath": "/var/log/nginx"
}
```

portMappings

Asigna puertos de red del contenedor a puertos del host.

links

Lista de contenedores que se van a vincular. Los contenedores vinculados pueden detectarse entre sí y comunicarse de forma segura.

volumesFrom

Esta opción le permite montar todos los volúmenes desde un contenedor diferente. Por ejemplo, para montar volúmenes desde un contenedor con el nombre web:

```
"volumesFrom": [
  {
    "sourceContainer": "web"
  }
],
```

Formato de autenticación: uso de imágenes de un repositorio privado

La sección `authentication` contiene los datos de autenticación de un repositorio privado. Esta entrada es opcional.

Añada la información sobre el bucket de Amazon S3; que contiene el archivo de autenticación en el parámetro `authentication` del archivo `Dockerrun.aws.json`. Asegúrese de que el parámetro `authentication` contiene un bucket de Amazon S3 y una clave válidos. El bucket de Amazon S3; debe estar alojado en la misma región que el entorno que lo está utilizando. Elastic Beanstalk no descargará archivos de los buckets de Amazon S3 alojados en otras regiones.

Usa el siguiente formato:

```
"authentication": {
  "bucket": "DOC-EXAMPLE-BUCKET",
  "key": "mydockercfg"
},
```

Para obtener información sobre cómo generar y cargar el archivo de autenticación, consulte [Uso de imágenes de un repositorio privado](#) en el tema Configuración del entorno de este capítulo.

Ejemplo Dockerrun.aws.json v2

El siguiente fragmento es un ejemplo que ilustra la sintaxis del archivo `Dockerrun.aws.json` para una instancia con dos contenedores.

```
{
  "AWSEBDockerrunVersion": 2,
  "volumes": [
    {
      "name": "php-app",
      "host": {
        "sourcePath": "/var/app/current/php-app"
      }
    },
    {
      "name": "nginx-proxy-conf",
      "host": {
        "sourcePath": "/var/app/current/proxy/conf.d"
      }
    }
  ],
  "containerDefinitions": [
    {
      "name": "php-app",
      "image": "php:fpm",
      "environment": [
        {
          "name": "Container",
          "value": "PHP"
        }
      ],
      "essential": true,
      "memory": 128,
      "mountPoints": [
        {
          "sourceVolume": "php-app",
          "containerPath": "/var/www/html",
          "readOnly": true
        }
      ]
    }
  ]
}
```

```
    },
    {
      "name": "nginx-proxy",
      "image": "nginx",
      "essential": true,
      "memory": 128,
      "portMappings": [
        {
          "hostPort": 80,
          "containerPort": 80
        }
      ],
      "links": [
        "php-app"
      ],
      "mountPoints": [
        {
          "sourceVolume": "php-app",
          "containerPath": "/var/www/html",
          "readOnly": true
        },
        {
          "sourceVolume": "nginx-proxy-conf",
          "containerPath": "/etc/nginx/conf.d",
          "readOnly": true
        },
        {
          "sourceVolume": "awseb-logs-nginx-proxy",
          "containerPath": "/var/log/nginx"
        }
      ]
    }
  ]
}
```

Entornos de Docker administrados por ECS con la consola de Elastic Beanstalk

Puede lanzar un clúster de instancias de varios contenedores en un entorno de Elastic Beanstalk de instancia única o escalable mediante la consola de Elastic Beanstalk. En este tutorial, se explica la configuración de contenedores y la preparación de código fuente para un entorno que utiliza dos contenedores.

Los contenedores, una aplicación PHP y un proxy nginx, se ejecutan uno al lado del otro en cada una de las instancias de Amazon Elastic Compute Cloud (Amazon EC2) en un entorno de Elastic Beanstalk. Después de crear el entorno y comprobar que las aplicaciones están en ejecución, se conectará a una instancia de contenedor para ver cómo encaja todo.

Secciones

- [Definir los contenedores de Docker administrados por ECS](#)
- [Agregar contenido](#)
- [Implementar en Elastic Beanstalk](#)
- [Conexión con una instancia de contenedor](#)
- [Inspeccione el agente de contenedor de Amazon ECS](#)

Definir los contenedores de Docker administrados por ECS

El primer paso para crear un nuevo entorno de Docker es crear un directorio para los datos de la aplicación. Esta carpeta puede estar ubicada en cualquier lugar del equipo local y tener cualquier nombre. Además de un archivo de configuración del contenedor, esta carpeta incluirá el contenido que se carga en Elastic Beanstalk y se implementa en el entorno.

Note

Todo el código de este tutorial está disponible en el repositorio awslabs de GitHub, en <https://github.com/awslabs/eb-docker-nginx-proxy>.

El archivo que Elastic Beanstalk utiliza para configurar los contenedores en una instancia de Amazon EC2 es un archivo de texto con formato JSON denominado `Dockerrun.aws.json`. Cree un archivo de texto con este nombre en la raíz de la aplicación y agregue el siguiente texto:

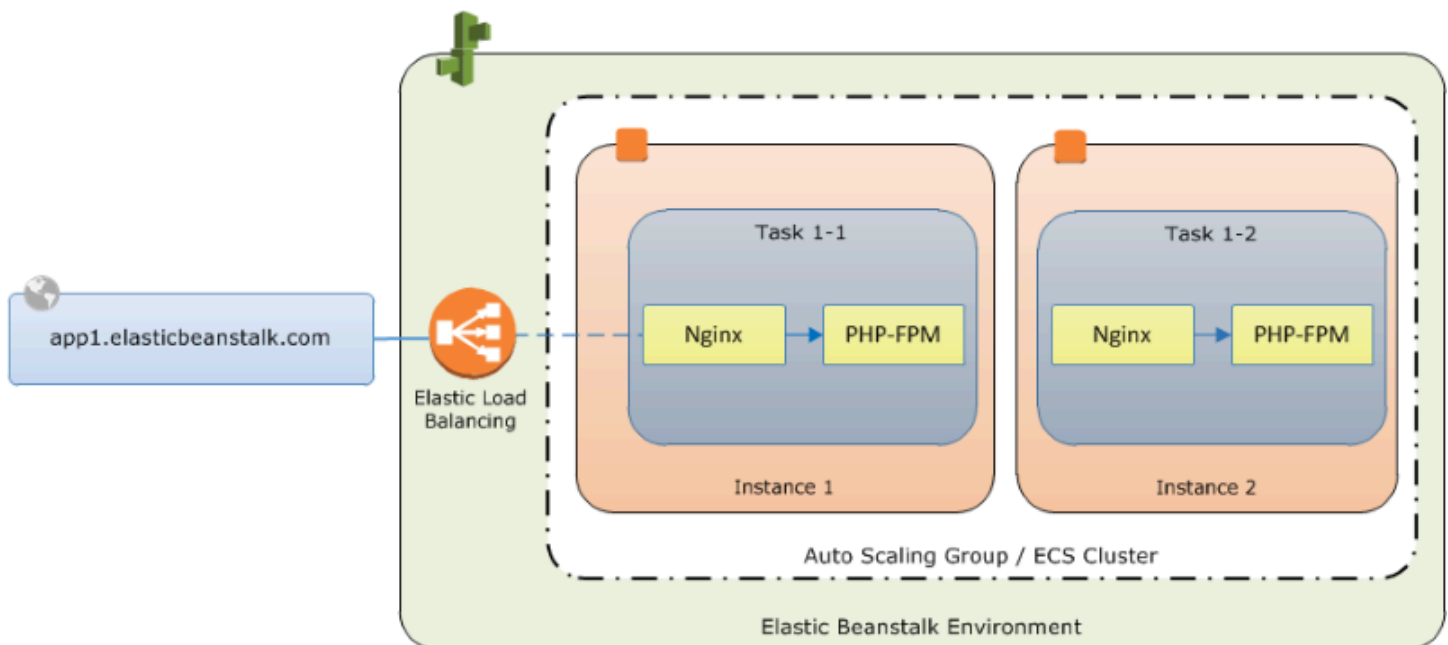
```
{
  "AWSEBDockerrunVersion": 2,
  "volumes": [
    {
      "name": "php-app",
      "host": {
        "sourcePath": "/var/app/current/php-app"
      }
    }
  ],
}
```

```
{
  "name": "nginx-proxy-conf",
  "host": {
    "sourcePath": "/var/app/current/proxy/conf.d"
  }
},
"containerDefinitions": [
  {
    "name": "php-app",
    "image": "php:fpm",
    "essential": true,
    "memory": 128,
    "mountPoints": [
      {
        "sourceVolume": "php-app",
        "containerPath": "/var/www/html",
        "readOnly": true
      }
    ]
  },
  {
    "name": "nginx-proxy",
    "image": "nginx",
    "essential": true,
    "memory": 128,
    "portMappings": [
      {
        "hostPort": 80,
        "containerPort": 80
      }
    ],
    "links": [
      "php-app"
    ],
    "mountPoints": [
      {
        "sourceVolume": "php-app",
        "containerPath": "/var/www/html",
        "readOnly": true
      },
      {
        "sourceVolume": "nginx-proxy-conf",
        "containerPath": "/etc/nginx/conf.d",
```



```
    "readOnly": true
  },
  {
    "sourceVolume": "awseb-logs-nginx-proxy",
    "containerPath": "/var/log/nginx"
  }
]
}
]
```

Esta configuración de ejemplo define dos contenedores y un sitio web de PHP con un proxy nginx delante. Estos dos contenedores se ejecutarán en paralelo en los contenedores de Docker de cada instancia del entorno de Elastic Beanstalk y obtendrán acceso al contenido compartido (el contenido del sitio web) desde los volúmenes de la instancia del host, que también se define en este archivo. Los propios contenedores se crean a partir de imágenes hospedadas en repositorios oficiales de Docker Hub. El entorno resultante se parece al siguiente:



Los volúmenes definidos en la configuración corresponden al contenido que va a crear a continuación y a cargar como parte del paquete de código fuente de la aplicación. Los contenedores obtienen acceso al contenido del host montando volúmenes en la sección `mountPoints` de las definiciones de contenedor.

Para obtener más información sobre el formato de `DockerRun.aws.json` y sus parámetros, consulte [Formato de definición de contenedor](#).

Agregar contenido

A continuación, va a agregar contenido en el sitio de PHP para que lo vean los visitantes y un archivo de configuración para el proxy nginx.

php-app/index.php

```
<h1>Hello World!!!</h1>
<h3>PHP Version <pre><?= phpversion()?></pre></h3>
```

php-app/static.html

```
<h1>Hello World!</h1>
<h3>This is a static HTML page.</h3>
```

proxy/conf.d/default.conf

```
server {
    listen 80;
    server_name localhost;
    root /var/www/html;

    index index.php;

    location ~ [^/]\.php(/|$) {
        fastcgi_split_path_info ^(.+?\.php)(/.*)$;
        if (!-f $document_root$fastcgi_script_name) {
            return 404;
        }

        include fastcgi_params;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        fastcgi_param PATH_INFO $fastcgi_path_info;
        fastcgi_param PATH_TRANSLATED $document_root$fastcgi_path_info;

        fastcgi_pass php-app:9000;
        fastcgi_index index.php;
    }
}
```

Implementar en Elastic Beanstalk

Ahora, la carpeta de la aplicación contiene los siguientes archivos:

```
### Dockerrun.aws.json
### php-app
#   ### index.php
#   ### static.html
### proxy
    ### conf.d
        ### default.conf
```

Esto es todo lo que necesita para crear el entorno Elastic Beanstalk. Cree un archivo .zip con los documentos y carpetas anteriores (no incluya la carpeta del proyecto del nivel superior). Para crear el archivo en el Explorador de Windows, seleccione el contenido de la carpeta del proyecto, haga clic con el botón derecho, seleccione Enviar a y haga clic en Carpeta comprimida (en zip).

Note

Para obtener más información sobre la estructura de archivos e instrucciones para la creación de archivos en otros entornos, consulte [Cree el paquete de código fuente de la aplicación.](#)

A continuación, cargue el paquete de origen en Elastic Beanstalk y cree su entorno. En Platform (Plataforma), seleccione Docker. Para la Platform branch (Ramificación de la plataforma), seleccione ECS running on 64bit Amazon Linux 2 (ECS que se ejecuta en Amazon Linux de 64 bits).

Para lanzar un entorno (consola)

1. Abra la consola de Elastic Beanstalk con este enlace preconfigurado: console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=tutorials&environmentType=LoadBalanced
2. En Platform (Plataforma), seleccione la plataforma y la ramificación de plataforma que coincidan con el idioma utilizado por su aplicación, o la plataforma Docker para aplicaciones basadas en contenedores.
3. En Application code (Código de aplicación), seleccione Upload your code (Cargar su propio código).
4. Elija Local file (Archivo local), seleccione Choose file (Seleccionar archivo) y abra el paquete de código fuente.
5. Elija Review and launch (Revisar y lanzar).

6. Revise la configuración disponible y, a continuación, elija Create app (Crear aplicación).

La consola de Elastic Beanstalk le redirigirá al panel de administración del nuevo entorno. En esta pantalla, se muestra el estado del entorno y la salida de eventos del servicio Elastic Beanstalk. Si el estado es verde, haga clic en la URL que está situada junto al nombre del entorno para ver el nuevo sitio web.

Conexión con una instancia de contenedor

A continuación, va a conectarse a una instancia de Amazon EC2 del entorno de Elastic Beanstalk para ver algunos de sus componentes en acción.

La forma más sencilla de conectarse a una instancia del entorno es utilizar la CLI de EB. Para utilizarla, [instale la CLI de EB](#), si aún no lo ha hecho. También necesitará configurar su entorno con un par de claves SSH de Amazon EC2. Para ello, utilice la [página de configuración de seguridad](#) de la consola o el comando `eb init` de la CLI de EB. Para conectarse a una instancia del entorno, utilice el comando `eb ssh` de la CLI de EB.

Ahora que se ha conectado a la instancia Amazon EC2 que hospeda los contenedores de Docker, puede ver cómo funciona todo. Ejecute `ls` en `/var/app/current`:

```
[ec2-user@ip-10-0-0-117 ~]$ ls /var/app/current
Dockerrun.aws.json  php-app  proxy
```

Este directorio contiene los archivos del paquete de código fuente que cargó en Elastic Beanstalk durante la creación del entorno.

```
[ec2-user@ip-10-0-0-117 ~]$ ls /var/log/containers
nginx-proxy      nginx-proxy-4ba868dbb7f3-stdouterr.log
php-app          php-app-dcc3b3c8522c-stdouterr.log      rotated
```

Aquí es donde se crean los logs de la instancia de contenedor y los recopila Elastic Beanstalk. Elastic Beanstalk crea un volumen en este directorio para cada contenedor, que se monta en la ubicación del contenedor donde se escriben los logs.

También puede consultar Docker para ver los contenedores en ejecución mediante `docker ps`.

```
[ec2-user@ip-10-0-0-117 ~]$ sudo docker ps
```

CONTAINER ID STATUS	IMAGE	PORTS	COMMAND	NAMES	CREATED
4ba868dbb7f3 Up 4 minutes	nginx	0.0.0.0:80->80/tcp,	"/docker-entrypoint..."	ecs-awseb-Tutorials-env-	4 minutes ago
dc2aywfjwg-1-nginx-proxy-acca84ef87c4aca15400		:::80->80/tcp			
dcc3b3c8522c Up 4 minutes	php:fpm	9000/tcp	"docker-php-entrypoi..."	ecs-awseb-Tutorials-env-	4 minutes ago
dc2aywfjwg-1-php-app-b8d38ae288b7b09e8101					
d9367c0baad6 Up 5 minutes (healthy)	amazon/amazon-ecs-agent:latest		"/agent"	ecs-agent	5 minutes ago

Aquí pueden verse los dos contenedores en ejecución que implementó, así como el agente de contenedor de Amazon ECS que coordinó la implementación.

Inspeccione el agente de contenedor de Amazon ECS

Las instancias de Amazon EC2 de un entorno de Docker administrado por ECS de Elastic Beanstalk ejecutan un proceso del agente en un contenedor de Docker. Este agente se conecta con el servicio Amazon ECS; para coordinar las implementaciones de contenedores. Estas implementaciones se ejecutan como tareas de Amazon ECS, que están configuradas en los archivos de definición de tareas. Elastic Beanstalk crea estos archivos de definición de tareas en función del archivo `Dockerrun.aws.json` que se carga en el paquete de código fuente.

Compruebe el estado del agente de contenedor con una solicitud HTTP GET en `http://localhost:51678/v1/metadata`:

```
[ec2-user@ip-10-0-0-117 ~]$ curl http://localhost:51678/v1/metadata
{
  "Cluster": "awseb-Tutorials-env-dc2aywfjwg",
  "ContainerInstanceArn": "arn:aws:ecs:us-west-2:123456789012:container-instance/awseb-Tutorials-env-dc2aywfjwg/db7be5215cd74658aacfcb292a6b944f",
  "Version": "Amazon ECS Agent - v1.57.1 (089b7b64)"
}
```

Esta estructura muestra el nombre del clúster de Amazon ECS y el ARN ([nombre de recurso de Amazon](#)) de la instancia del clúster (instancia de Amazon EC2 a la que está conectado).

Para obtener más información, realice una solicitud HTTP GET en `http://localhost:51678/v1/tasks`:

```
[ec2-user@ip-10-0-0-117 ~]$ curl http://localhost:51678/v1/tasks
```

```
{
  "Tasks":[
    {
      "Arn":"arn:aws:ecs:us-west-2:123456789012:task/awseb-Tutorials-env-dc2aywfjwg/
bbde7ebe1d4e4537ab1336340150a6d6",
      "DesiredStatus":"RUNNING",
      "KnownStatus":"RUNNING",
      "Family":"awseb-Tutorials-env-dc2aywfjwg",
      "Version":"1",
      "Containers":[
        {
          "DockerId":"dcc3b3c8522cb9510b7359689163814c0f1453b36b237204a3fd7a0b445d2ea6",
          "DockerName":"ecs-awseb-Tutorials-env-dc2aywfjwg-1-php-app-
b8d38ae288b7b09e8101",
          "Name":"php-app",
          "Volumes":[
            {
              "Source":"/var/app/current/php-app",
              "Destination":"/var/www/html"
            }
          ]
        },
        {
          "DockerId":"4ba868dbb7f3fb3328b8afeb2cb6cf03e3cb1cdd5b109e470f767d50b2c3e303",
          "DockerName":"ecs-awseb-Tutorials-env-dc2aywfjwg-1-nginx-proxy-
acca84ef87c4aca15400",
          "Name":"nginx-proxy",
          "Ports":[
            {
              "ContainerPort":80,
              "Protocol":"tcp",
              "HostPort":80
            },
            {
              "ContainerPort":80,
              "Protocol":"tcp",
              "HostPort":80
            }
          ],
          "Volumes":[
            {
              "Source":"/var/app/current/php-app",
```

```
        "Destination": "/var/www/html"
      },
      {
        "Source": "/var/log/containers/nginx-proxy",
        "Destination": "/var/log/nginx"
      },
      {
        "Source": "/var/app/current/proxy/conf.d",
        "Destination": "/etc/nginx/conf.d"
      }
    ]
  }
]
}
```

Esta estructura describe la tarea que se ejecuta para implementar los dos contenedores en el proyecto de ejemplo de este tutorial. Se muestra la siguiente información:

- **KnownStatus:** el estado `RUNNING` indica que los contenedores siguen activos.
- **Family (Familia):** nombre de la definición de tarea a partir de la que Elastic Beanstalk creó `Dockerrun.aws.json`.
- **Version:** versión de la definición de tareas. Esta aumenta cada vez que el archivo de definición de tareas se actualiza.
- **Containers:** información sobre los contenedores que se ejecutan en la instancia.

Hay aún más información disponible en el mismo servicio Amazon ECS, que puede llamar con la AWS Command Line Interface. Para obtener instrucciones sobre el uso de la AWS CLI con Amazon ECS e información sobre Amazon ECS en general, consulte la [Guía del usuario de Amazon ECS](#).

Migración de Docker multicontenedor sobre Amazon Linux a ECS sobre Amazon Linux 2023

El [18 de julio de 2022](#), Elastic Beanstalk estableció el estado de todas las ramificaciones de la plataforma basadas en la AMI de Amazon Linux (AL1) como retirado. Esto incluye la ramificación de plataforma Multi-container Docker running on 64bit Amazon Linux (Multi-container Docker ejecutándose en Amazon Linux de 64 bits). Este tema sirve de guía para la migración de

aplicaciones de esta ramificación de la plataforma retirada a ECS sobre AL2023 de 64 bits. Esta ramificación de la plataforma de destino está actualizada y es compatible.

Al igual que la ramificación Docker multicontenedor AL1 anterior, la nueva ramificación de la plataforma ECS AL2023 utiliza Amazon ECS para coordinar la implementación de múltiples contenedores de Docker en un clúster de Amazon ECS dentro de un entorno de Elastic Beanstalk. La nueva ramificación de la plataforma ECS AL2023 es compatible con todas las características de la ramificación de la plataforma Docker multicontenedor AL1 anterior. Además, es compatible el mismo archivo `DockerRun.aws.json` de la v2.

Secciones

- [Migración con la consola de Elastic Beanstalk](#)
- [Migración con la AWS CLI](#)

Migración con la consola de Elastic Beanstalk

Para migrar utilizando la consola de Elastic Beanstalk, implemente el mismo código fuente en un nuevo entorno que esté basado en la ramificación de la plataforma ECS sobre AL2023. No es necesario realizar cambios en el código fuente.

Para migrar a la ramificación de la plataforma ECS sobre Amazon Linux 2023

1. Cree el paquete de código fuente de la aplicación con el código fuente de la aplicación que ya se está implementado en el entorno anterior. Puede utilizar el mismo paquete de código fuente de la aplicación y el mismo archivo `DockerRun.aws.json` de la v2.
2. Cree un entorno nuevo utilizando la ramificación de la plataforma ECS sobre Amazon Linux 2023. Utilice el paquete de código fuente del paso anterior para el código de la aplicación. Para obtener instrucciones más detalladas, consulte [Implementar en Elastic Beanstalk](#) en el tutorial de Docker administrado por ECS que aparece anteriormente en este capítulo.

Migración con la AWS CLI

También tiene la opción de utilizar AWS Command Line Interface (AWS CLI) para migrar su entorno existente de Docker multicontenedor Amazon Linux Docker a la ramificación de la plataforma ECS AL2023 más reciente. En este caso, no es necesario crear un entorno nuevo ni volver a implementar el código fuente. Solo debe ejecutar el comando [update-environment](#) de AWS CLI. Realizará una actualización de la plataforma para migrar su entorno existente a la ramificación de la plataforma ECS Amazon Linux 2023.

Utilice la siguiente sintaxis para migrar el entorno a una ramificación de la plataforma nueva.

```
aws elasticbeanstalk update-environment \  
--environment-name my-env \  
--solution-stack-name "64bit Amazon Linux 2023 version running ECS" \  
--region my-region
```

El siguiente es un ejemplo del comando para migrar el entorno beta-101 a la versión 3.0.0 de la ramificación de la plataforma ECS Amazon Linux 2023 en la región us-east-1.

```
aws elasticbeanstalk update-environment \  
--environment-name beta-101 \  
--solution-stack-name "64bit Amazon Linux 2023 v4.0.0 running ECS" \  
--region us-east-1
```

El parámetro `solution-stack-name` proporciona la ramificación de la plataforma y su versión. Utilice la versión de la ramificación de la plataforma más reciente y especifique el nombre de la pila de soluciones adecuado. La versión de cada ramificación de la plataforma se incluye en el nombre de la pila de soluciones, como se muestra en el ejemplo anterior. Para obtener una lista de las pilas de soluciones más actuales para la plataforma de Docker, consulte [Plataformas admitidas](#) en la guía de Plataformas de AWS Elastic Beanstalk.

Note

El comando [list-available-solution-stacks](#) proporciona una lista de las versiones de plataforma disponibles para su cuenta en una región de AWS.

```
aws elasticbeanstalk list-available-solution-stacks --region us-east-1 --query  
SolutionStacks
```

Para obtener más información sobre AWS CLI, consulte la [Guía del usuario de AWS Command Line Interface](#). Para obtener más información sobre los comandos de la AWS CLI de Elastic Beanstalk, consulte la [Referencia de comandos de la AWS CLI para Elastic Beanstalk](#).

Migración (heredada) a la ramificación de la plataforma Docker que se ejecuta en Amazon Linux 2 desde Multicontainer Docker que se ejecuta en Amazon Linux

Antes del lanzamiento de la ramificación de la plataforma ECS que se ejecuta en Amazon Linux 2 de 64 bits, Elastic Beanstalk ofrecía una ruta de migración alternativa a Amazon Linux 2 para los clientes con entornos basados en la ramificación de la plataforma Multicontainer Docker que se ejecuta en Amazon Linux de 64 bits. En este tema se describe esa ruta de migración y se incluye en este documento como referencia para cualquier cliente que haya completado esa ruta de migración.

Ahora recomendamos que los clientes con entornos basados en la ramificación de la plataforma Multicontainer Docker que se ejecuta en Amazon Linux de 64 bits migren a la ramificación de la plataforma ECS que se ejecuta en Amazon Linux 2 de 64 bits. A diferencia de la ruta de migración alternativa, este enfoque continúa utilizando Amazon ECS para coordinar las implementaciones de los contenedores en entornos de Docker administrados por ECS. Este aspecto permite un enfoque más sencillo. No se requieren cambios en el código fuente y se admite el mismo `DockerRun.aws.json` de la v2. Para obtener más información, consulte [Migración de Docker multicontenedor sobre Amazon Linux a ECS sobre Amazon Linux 2023](#).

Migración heredada de Multicontainer Docker en Amazon Linux a la ramificación de la plataforma Docker Amazon Linux 2

Puede migrar las aplicaciones que se ejecutan en la [plataforma Docker de varios contenedores en la AMI de Amazon Linux](#) a la plataforma Amazon Linux 2 Docker. La plataforma Docker Multi-Container en la AMI de Amazon Linux requiere que especifique imágenes de aplicación preconfiguradas para que se ejecuten como contenedores. Después de la migración, ya no tendrá esta limitación, ya que la plataforma de Docker en Amazon Linux 2 también permite a Elastic Beanstalk crear imágenes de contenedor durante la implementación. Sus aplicaciones seguirán ejecutándose en entornos de varios contenedores con los beneficios agregados de la herramienta Docker Compose.

Docker Compose es una herramienta para definir y ejecutar aplicaciones Docker de varios contenedores. Para obtener más información sobre Docker Compose y cómo instalarlo, consulte los sitios de Docker [Overview of Docker Compose](#) e [Install Docker Compose](#).

El archivo `docker-compose.yml`

La herramienta Docker Compose utiliza el archivo `docker-compose.yml` para la configuración de los servicios de aplicaciones. Este archivo reemplaza el archivo `DockerRun.aws.json v2` en el directorio del proyecto de la aplicación y el paquete de código fuente de la aplicación. Cree

el archivo `docker-compose.yml` manualmente. Le resultará útil hacer referencia a su archivo `Dockerrun.aws.json v2` para la mayoría de los valores de parámetros.

A continuación, se muestra un ejemplo de un archivo `docker-compose.yml` y el archivo `Dockerrun.aws.json v2` correspondiente para la misma aplicación. Para obtener más información sobre el archivo `docker-compose.yml`, consulte [Compose file reference](#). Para obtener más información sobre el archivo `Dockerrun.aws.json v2`, consulte [Dockerrun.aws.json v2](#).

docker-compose.yml

```
version: '2.4'
services:
  php-app:
    image: "php:fpm"
    volumes:
      - "./php-app:/var/www/html:ro"
      - "${EB_LOG_BASE_DIR}/php-app:/var/log/sample-app"
    mem_limit: 128m
    environment:
      Container: PHP
  nginx-proxy:
    image: "nginx"
    ports:
      - "80:80"
    volumes:
      - "./php-app:/var/www/html:ro"
      - "./proxy/conf.d:/etc/nginx/conf.d:ro"
      - "${EB_LOG_BASE_DIR}/nginx-proxy:/var/log/nginx"
    mem_limit: 128m
    links:
      - php-app
```

Dockerrun.aws.json v2

```
{
  "AWSEBDockerrunVersion": 2,
  "volumes": [
    {
      "name": "php-app",
      "host": {
        "sourcePath": "/var/app/current/php-app"
      }
    },
    {
      "name": "nginx-proxy-conf",
      "host": {
        "sourcePath": "/var/app/current/proxy/conf.d"
      }
    }
  ],
  "containerDefinitions": [
    {
      "name": "php-app",
      "image": "php:fpm",
      "environment": [
        {
          "name": "Container",
          "value": "PHP"
        }
      ],
      "essential": true,
      "memory": 128,
      "mountPoints": [
```

docker-compose.yml**Dockerrun.aws.json v2**

```
        {
            "sourceVolume": "php-app"
        },
        {
            "containerPath": "/var/www
/html",
            "readOnly": true
        }
    ],
},
{
    "name": "nginx-proxy",
    "image": "nginx",
    "essential": true,
    "memory": 128,
    "portMappings": [
        {
            "hostPort": 80,
            "containerPort": 80
        }
    ],
    "links": [
        "php-app"
    ],
    "mountPoints": [
        {
            "sourceVolume": "php-app"
        },
        {
            "containerPath": "/var/www
/html",
            "readOnly": true
        },
        {
            "sourceVolume": "nginx-pr
oxy-conf",
            "containerPath": "/etc/ngi
nx/conf.d",
            "readOnly": true
        },
        {
            "sourceVolume": "awseb-lo
gs-nginx-proxy",
```

<code>docker-compose.yml</code>	<code>Dockerrun.aws.json v2</code>
	<pre> "containerPath": "/var/log /nginx" }] } } </pre>

Consideraciones adicionales sobre la migración

La plataforma Docker Amazon Linux 2 y la plataforma Docker de contenedores múltiples Amazon Linux AMI implementan propiedades del entorno de forma diferente. Estas dos plataformas también tienen directorios de registro diferentes a los que Elastic Beanstalk crea para cada uno de sus contenedores. Después de migrar desde la plataforma Docker de contenedores múltiples de Amazon Linux AMI, deberá tener en cuenta estas diferentes implementaciones para su nuevo entorno de plataforma Docker de Amazon Linux 2.

Area	Plataforma Docker en Amazon Linux 2 con Docker Compose	Plataforma Docker de varios contenedores en la AMI de Amazon Linux
Propiedades del entorno	<p>Para que los contenedores accedan a las propiedades del entorno, debe agregar una referencia al archivo <code>.env</code> en el archivo <code>docker-compose.yml</code>. Elastic Beanstalk genera el archivo <code>.env</code>, enumerando cada una de las propiedades como variables de entorno. Para obtener más información, consulte Referencia a variables de entorno en contenedores.</p>	<p>Elastic Beanstalk puede pasar directamente las propiedades del entorno al contenedor. El código que se ejecuta en el contenedor puede acceder a estas propiedades como variables de entorno sin ninguna configuración adicional.</p>
Directorios de registro	<p>Para cada contenedor, Elastic Beanstalk crea un directorio de registro llamado <code>/var/log/eb-docker/</code></p>	<p>Para cada contenedor, Elastic Beanstalk crea un directorio de registro llamado <code>/var/log/container</code></p>

Area	Plataforma Docker en Amazon Linux 2 con Docker Compose	Plataforma Docker de varios contenedores en la AMI de Amazon Linux
	containers/ <i><service name></i> (o <code>\${EB_LOG_BASE_DIR}/<service name></code>). Para obtener más información, consulte Registro personalizado del contenedor de Docker (Docker Compose) .	s/ <i><containername></i> . Para obtener más información, consulte mountPoints en Formato de definición de contenedor .

Pasos para realizar la migración

Para migrar a la plataforma Docker de Amazon Linux 2

1. Cree el archivo `docker-compose.yml` para su aplicación, basado en el archivo `Dockerrun.aws.json v2` existente. Para obtener más información, consulte la sección anterior [El archivo `docker-compose.yml`](#).
2. En el directorio raíz de la carpeta del proyecto de la aplicación, reemplace el archivo `Dockerrun.aws.json v2` por el archivo `docker-compose.yml` que acaba de crear.

La estructura de directorios debe ser similar a la siguiente.

```
~/myApplication
|-- docker-compose.yml
|-- .ebextensions
|-- php-app
|-- proxy
```

3. Utilice el comando `eb init` para configurar el directorio local para la implementación en Elastic Beanstalk.

```
~/myApplication$ eb init -p docker application-name
```

4. Utilice el comando `eb create` para crear un entorno e implementar la imagen de Docker.

```
~/myApplication$ eb create environment-name
```

5. Si la aplicación es una aplicación web, después de que se inicie el entorno, utilice el comando `eb open` para verla en un navegador web.

```
~/myApplication$ eb open environment-name
```

6. Puede mostrar el estado del entorno recién creado mediante el comando `eb status`.

```
~/myApplication$ eb status environment-name
```

Contenedores de Docker preconfigurados (AMI de Amazon Linux)

Note

El [18 de julio de 2022](#), Elastic Beanstalk estableció el estado de todas las ramificaciones de la plataforma basadas en la AMI de Amazon Linux (AL1) como retirado. Para obtener más información sobre la migración a una ramificación de la plataforma Amazon Linux 2023 actual y totalmente compatible, consulte [Migración de su aplicación de Linux de Elastic Beanstalk a Amazon Linux 2023 o Amazon Linux 2](#).

Ya no se admite la rama de GlassFish plataforma Docker preconfigurada que se ejecuta en la AMI de Amazon Linux (AL1). Para migrar su GlassFish aplicación a una plataforma Amazon Linux 2023 compatible, despliegue GlassFish y el código de la aplicación en una imagen de Docker de Amazon Linux 2023. Para obtener más información, consulte el siguiente tema, [the section called “Tutorial: GlassFish en Docker: camino a Amazon Linux 2023”](#).

Introducción a los contenedores de Docker preconfigurados en la AMI de Amazon Linux (precede a Amazon Linux 2)

En esta sección se muestra cómo desarrollar una aplicación de ejemplo localmente e implementarla después en Elastic Beanstalk con un contenedor Docker preconfigurado.

Configurar su entorno de desarrollo local

Para este tutorial, utilizamos una GlassFish aplicación de ejemplo.

Para configurar el entorno

1. Cree una nueva carpeta para la aplicación de ejemplo.

```
~$ mkdir eb-preconf-example
```

```
~$ cd eb-preconf-example
```

2. Descargue el código de la aplicación de ejemplo en la nueva carpeta.

```
~$ wget https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/samples/docker-glassfish-v1.zip
~$ unzip docker-glassfish-v1.zip
~$ rm docker-glassfish-v1.zip
```

Desarrollar y probar la aplicación localmente

Para desarrollar una aplicación de ejemplo GlassFish

1. Añada un `Dockerfile` en la carpeta raíz de la aplicación. En el archivo, especifique la imagen base de AWS Elastic Beanstalk Docker que se utilizará para ejecutar su contenedor Docker local preconfigurado. Más adelante, implementará la aplicación en una versión de plataforma Docker preconfigurada de Elastic GlassFish Beanstalk. Elija la imagen base de Docker que utiliza esta versión de la plataforma. Para averiguar la imagen de Docker actual de la versión de la plataforma, consulte la sección [Docker preconfigurado](#) de la página Plataformas compatibles con AWS Elastic Beanstalk en la guía Plataformas de AWS Elastic Beanstalk .

Example b-preconf-example~/E /Dockerfile

```
# For Glassfish 5.0 Java 8
FROM amazon/aws-eb-glassfish:5.0-a1-onbuild-2.11.1
```

Para obtener más información sobre cómo usar un `Dockerfile`, consulte [Configuración de Docker](#).

2. Cree la imagen de Docker.

```
~/eb-preconf-example$ docker build -t my-app-image .
```

3. Ejecute el contenedor Docker desde la imagen.

Note

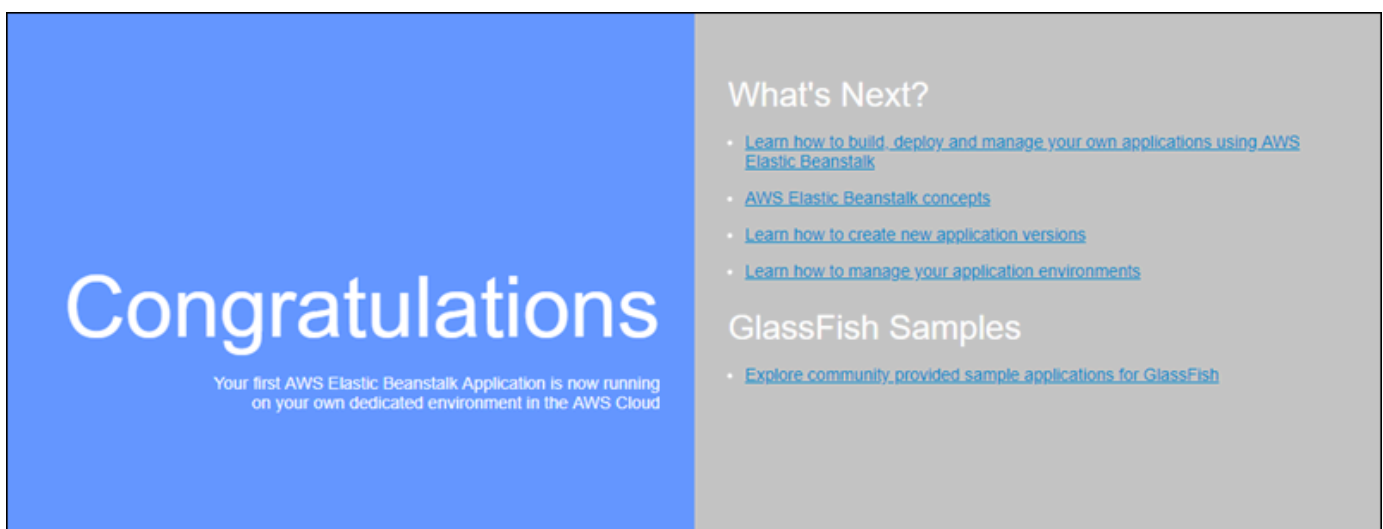
Debe incluir la marca `-p` para asignar el puerto 8080 del contenedor al puerto 3000 del host local. Los contenedores Docker de Elastic Beanstalk siempre exponen la aplicación en el puerto 8080 del contenedor. Las marcas `-it` ejecutan la imagen como un proceso

interactivo. La marca `--rm` limpia el sistema de archivos del contenedor cuando el contenedor existe. De forma opcional, puede incluir una marca `-d` para ejecutar la imagen como un daemon.

```
$ docker run -it --rm -p 3000:8080 my-app-image
```

4. Para ver la aplicación de ejemplo, escriba la siguiente URL en el navegador web.

```
http://localhost:3000
```



Implementar en Elastic Beanstalk

Después de probar la aplicación, ya puede implementarla en Elastic Beanstalk.

Para implementar su aplicación en Elastic Beanstalk

1. En la carpeta raíz de la aplicación, cambie el nombre de `Dockerfile` a `Dockerfile.local`. Este paso es necesario para que Elastic Beanstalk utilice el `Dockerfile` que contiene las instrucciones correctas para que Elastic Beanstalk cree una imagen personalizada de Docker en cada instancia de Amazon EC2 en su entorno de Elastic Beanstalk.

Note

No es necesario realizar este paso si su `Dockerfile` incluye instrucciones que modifican la imagen de Docker base de la versión de la plataforma. No necesita usar un `Dockerfile` en absoluto si su `Dockerfile` incluye solo una línea `FROM` para especificar la imagen base desde la que se crea el contenedor. En tal caso, el `Dockerfile` es redundante.

2. Cree el paquete de código fuente de la aplicación.

```
~/eb-preconf-example$ zip myapp.zip -r *
```

3. [Abra la consola de Elastic Beanstalk con este enlace preconfigurado: console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=Tutorials &EnvironmentType=LoadBalanced](https://console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=Tutorials&EnvironmentType=LoadBalanced)
4. En Platform (Plataforma), en Preconfigured Docker (Docker preconfigurado), elija Glassfish.
5. En Application code, seleccione Upload your code y, luego, Upload.
6. Elija Local file (Archivo local), seleccione Browse (Examinar) y abra el paquete de código fuente de la aplicación que acaba de crear.
7. Seleccione Upload.
8. Elija Review and launch (Revisar y lanzar).
9. Revise la configuración disponible y, a continuación, elija Create app (Crear aplicación).
10. Cuando se crea el entorno, puede ver la aplicación implementada. Elija la URL del entorno que se muestra en la parte superior del panel de la consola.

Implementación de una GlassFish aplicación en la plataforma Docker: una ruta de migración a Amazon Linux 2023

El objetivo de este tutorial es proporcionar a los clientes que utilizan la GlassFish plataforma Docker preconfigurada (basada en la AMI de Amazon Linux) una ruta de migración a Amazon Linux 2023. Puede migrar su GlassFish aplicación a Amazon Linux 2023 mediante la implementación GlassFish y el código de la aplicación a una imagen de Docker de Amazon Linux 2023.

El tutorial explica el uso de la plataforma AWS Elastic Beanstalk Docker para implementar una aplicación basada en el [servidor de aplicaciones Java EE en un GlassFish entorno](#) de Elastic Beanstalk.

Demostramos dos enfoques para construir una imagen de Docker:

- Sencillo: proporciona el código fuente de GlassFish la aplicación y deja que Elastic Beanstalk cree y ejecute una imagen de Docker como parte del aprovisionamiento del entorno. Esto es fácil de configurar, a costa de un mayor tiempo de aprovisionamiento de instancias.
- Avanzado: cree una imagen de Docker personalizada que contenga el código de la aplicación y las dependencias, y proporciónela a Elastic Beanstalk para usarla en su entorno. Este enfoque es un poco más complejo y reduce el tiempo de aprovisionamiento de las instancias en su entorno.

Requisitos previos

En este tutorial, se da por sentado que usted posee algunos conocimientos sobre las operaciones básicas de la interfaz de la línea de comandos de Elastic Beanstalk (CLI de EB) y Docker. Si aún no lo ha hecho, siga las instrucciones que se indican en [Introducción a Elastic Beanstalk](#) para lanzar su primer entorno de Elastic Beanstalk. En este tutorial se utiliza la [CLI de EB](#), pero también puede crear entornos y cargar aplicaciones mediante la consola de Elastic Beanstalk.

Para realizar este tutorial, también necesitará los siguientes componentes de Docker:

- Una instalación local activa de Docker. Para obtener más información, consulte [Descripción de Docker](#) en la página de documentos de Docker.
- Acceso a Docker Hub. Deberá crear un ID de Docker para poder acceder a Docker Hub. Para obtener más información, consulte [Compartir la aplicación](#) en la página de documentos de Docker.

Para obtener más información acerca de la configuración de los entornos Docker en las plataformas Elastic Beanstalk, consulte [Configuración de Docker](#) en este mismo capítulo.

Ejemplo sencillo: proporcione su código de aplicación

Esta es una forma sencilla de implementar la aplicación. GlassFish Proporciona el código fuente de la aplicación junto con el `Dockerfile` incluido en este tutorial. Elastic Beanstalk crea una imagen de Docker que incluye la aplicación y la pila de software. GlassFish A continuación, Elastic Beanstalk ejecuta la imagen en las instancias de su entorno.

Un problema de este enfoque es que Elastic Beanstalk crea la imagen de Docker localmente cada vez que crea una instancia para su entorno. La creación de la imagen aumenta el tiempo de aprovisionamiento de instancias. Este impacto no se limita a la creación inicial del entorno, sino que también ocurre durante las acciones de escalado ascendente.

Para lanzar un entorno con una aplicación de ejemplo GlassFish

1. Descargue el `docker-glassfish-al2-v1.zip` de ejemplo y, a continuación, expanda el archivo `.zip` en un directorio del entorno de desarrollo.

```
~$ curl https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/samples/docker-glassfish-al2-v1.zip --output docker-glassfish-al2-v1.zip
~$ mkdir glassfish-example
~$ cd glassfish-example
~/glassfish-example$ unzip ../docker-glassfish-al2-v1.zip
```

La estructura de directorios debe ser similar a la siguiente.

```
~/glassfish-example
|-- Dockerfile
|-- Dockerrun.aws.json
|-- glassfish-start.sh
|-- index.jsp
|-- META-INF
|   |-- LICENSE.txt
|   |-- MANIFEST.MF
|   `-- NOTICE.txt
|-- robots.txt
`-- WEB-INF
    `-- web.xml
```

Los siguientes archivos son clave para crear y ejecutar un contenedor de Docker en su entorno:

- `Dockerfile`: proporciona instrucciones que Docker utiliza para crear una imagen con la aplicación y las dependencias requeridas.
- `glassfish-start.sh`: un script de shell que ejecuta la imagen de Docker para iniciar la aplicación.
- `Dockerrun.aws.json`— Proporciona una clave de registro para incluir las GlassFish [solicitudes del archivo](#) de registro del servidor de aplicaciones. Si no le interesan GlassFish los registros, puede omitir este archivo.

2. Configure el directorio local para su implementación en Elastic Beanstalk.

```
~/glassfish-example$ eb init -p docker glassfish-example
```

3. (Opcional) Utilice el comando `eb local run` para crear y ejecutar el contenedor de forma local.

```
~/glassfish-example$ eb local run --port 8080
```

Note

Para obtener más información acerca del comando `eb local`, consulte [the section called “eb local”](#). El comando no es compatible con Windows. Si lo prefiere, puede crear y ejecutar el contenedor con los comandos `docker build` y `docker run`. Para obtener más información, consulte la [documentación de Docker](#).

4. (Opcional) Aunque el contenedor se está ejecutando, utilice el comando `eb local open` para ver la aplicación en un navegador web. Si lo prefiere, abra <http://localhost:8080/> en un navegador web.

```
~/glassfish-example$ eb local open
```

5. Use el comando `eb create` para crear un entorno e implementar su aplicación.

```
~/glassfish-example$ eb create glassfish-example-env
```

6. Una vez que se lanza el entorno, utilice el comando `eb open` para verlo en un explorador Web.

```
~/glassfish-example$ eb open
```

Cuando termine de trabajar con el ejemplo, termine el entorno y elimine los recursos relacionados.

```
~/glassfish-example$ eb terminate --all
```

Ejemplo avanzado: proporcione una imagen de Docker predefinida

Esta es una forma más avanzada de implementar la GlassFish aplicación. A partir del primer ejemplo, se crea una imagen de Docker que contiene el código de la aplicación y la pila de GlassFish

software, y se envía a Docker Hub. Después de realizar este paso único, puede lanzar entornos de Elastic Beanstalk basados en la imagen personalizada.

Cuando lanza un entorno y proporciona la imagen de Docker, las instancias del entorno descargan y usan esta imagen directamente y no necesitan crear una imagen de Docker. Por lo tanto, se reduce el tiempo de aprovisionamiento de instancias.

Notas

- Los siguientes pasos crearán una imagen de Docker disponible públicamente.
- Utilizará los comandos Docker de su instalación local de Docker, junto con las credenciales de Docker Hub. Para obtener más información, consulte la sección anterior sobre los requisitos previos de este tema.

Para lanzar un entorno con una imagen de Docker de una GlassFish aplicación prediseñada

1. Descargue y expanda el ejemplo `docker-glassfish-a12-v1.zip` como en el [ejemplo sencillo](#) anterior. Si ha completado dicho ejemplo, puede usar el directorio que ya tiene.
2. Cree una imagen de Docker y envíela a Docker Hub. Ingrese su ID de Docker en `docker-id` para iniciar sesión en Docker Hub.

```
~/glassfish-example$ docker build -t docker-id/beanstalk-glassfish-example:latest .  
~/glassfish-example$ docker push docker-id/beanstalk-glassfish-example:latest
```

Note

Antes de enviar la imagen, es posible que deba ejecutar `docker login`. Se le pedirán las credenciales de Docker Hub si ejecuta el comando sin parámetros.

3. Cree un directorio adicional.

```
~$ mkdir glassfish-prebuilt  
~$ cd glassfish-prebuilt
```

4. Copie el siguiente ejemplo en un archivo denominado `Dockerrun.aws.json`.

Example `~/glassfish-prebuilt/Dockerrun.aws.json`

```
{
  "AWSEBDockerrunVersion": "1",
  "Image": {
    "Name": "docker-username/beanstalk-glassfish-example"
  },
  "Ports": [
    {
      "ContainerPort": 8080,
      "HostPort": 8080
    }
  ],
  "Logging": "/usr/local/glassfish5/glassfish/domains/domain1/logs"
}
```

5. Configure el directorio local para su implementación en Elastic Beanstalk.

```
~/glassfish-prebuilt$ eb init -p docker glassfish-prebuilt
```

6. (Opcional) Utilice el comando `eb local run` para ejecutar el contenedor localmente.

```
~/glassfish-prebuilt$ eb local run --port 8080
```

7. (Opcional) Aunque el contenedor se está ejecutando, utilice el comando `eb local open` para ver la aplicación en un navegador web. Si lo prefiere, abra <http://localhost:8080/> en un navegador web.

```
~/glassfish-prebuilt$ eb local open
```

8. Utilice el comando `eb create` para crear un entorno e implementar la imagen de Docker.

```
~/glassfish-prebuilt$ eb create glassfish-prebuilt-env
```

9. Una vez que se lanza el entorno, utilice el comando `eb open` para verlo en un explorador Web.

```
~/glassfish-prebuilt$ eb open
```

Cuando termine de trabajar con el ejemplo, termine el entorno y elimine los recursos relacionados.

```
~/glassfish-prebuilt$ eb terminate --all
```

Creación e implementación de aplicaciones Go en Elastic Beanstalk

AWS Elastic Beanstalk for Go facilita la implementación, la administración y el escalado de sus aplicaciones web Go mediante Amazon Web Services. Elastic Beanstalk for Go está disponible para cualquier persona que desarrolle o hospede una aplicación web con Go. En este capítulo se proporcionan step-by-step instrucciones para implementar la aplicación web en Elastic Beanstalk.

Después de implementar la aplicación Elastic Beanstalk, puede utilizar la CLI de EB para administrar la aplicación y el entorno, o bien puede utilizar la consola de Elastic Beanstalk, la AWS CLI o las API.

Temas

- [QuickStart: Implementar una aplicación Go en Elastic Beanstalk](#)
- [Configuración del entorno de desarrollo de Go](#)
- [Uso de la plataforma Elastic Beanstalk Go](#)

QuickStart: Implementar una aplicación Go en Elastic Beanstalk

Este QuickStart tutorial explica el proceso de creación de una aplicación Go e implementarla en un AWS Elastic Beanstalk entorno.

Note

Este QuickStart tutorial está destinado a fines de demostración. No utilice la aplicación creada en este tutorial para el tráfico de producción.

Secciones

- [Tu AWS cuenta](#)
- [Requisitos previos](#)
- [Paso 1: Crea una aplicación Go](#)
- [Paso 2: Implemente la aplicación Go con la CLI de EB](#)

- [Paso 3: Ejecute la aplicación en Elastic Beanstalk](#)
- [Paso 4: Limpiar](#)
- [AWS recursos para su aplicación](#)
- [Siguiendo pasos](#)
- [Implemente con la consola de Elastic Beanstalk](#)

Tu AWS cuenta

Si aún no eres AWS cliente, debes crear una AWS cuenta. El registro le permite acceder a Elastic Beanstalk AWS y a otros servicios que necesite.

Si ya tiene una AWS cuenta, puede pasar a [Requisitos previos](#)

Crea una AWS cuenta

Inscríbese en una Cuenta de AWS

Si no tiene una Cuenta de AWS, complete los siguientes pasos para crearlo.

Para suscribirte a una Cuenta de AWS

1. Abra <https://portal.aws.amazon.com/billing/signup>.
2. Siga las instrucciones que se le indiquen.

Parte del procedimiento de registro consiste en recibir una llamada telefónica e indicar un código de verificación en el teclado del teléfono.

Cuando te registras en una Cuenta de AWS, Usuario raíz de la cuenta de AWS se crea una. El usuario raíz tendrá acceso a todos los Servicios de AWS y recursos de esa cuenta. Como práctica recomendada de seguridad, asigne acceso administrativo a un usuario y utilice únicamente el usuario raíz para realizar [tareas que requieren acceso de usuario raíz](#).

AWS te envía un correo electrónico de confirmación una vez finalizado el proceso de registro. Puede ver la actividad de la cuenta y administrar la cuenta en cualquier momento entrando en <https://aws.amazon.com/> y seleccionando Mi cuenta.

Creación de un usuario con acceso administrativo

Después de registrarte en un usuario Cuenta de AWS, protege Usuario raíz de la cuenta de AWS AWS IAM Identity Center, habilita y crea un usuario administrativo para que no utilices el usuario root en las tareas diarias.

Proteja su Usuario raíz de la cuenta de AWS

1. Inicie sesión [AWS Management Console](#) como propietario de la cuenta seleccionando el usuario root e introduciendo su dirección de Cuenta de AWS correo electrónico. En la siguiente página, escriba su contraseña.

Para obtener ayuda para iniciar sesión con el usuario raíz, consulte [Iniciar sesión como usuario raíz](#) en la Guía del usuario de AWS Sign-In .

2. Active la autenticación multifactor (MFA) para el usuario raíz.

Para obtener instrucciones, consulte [Habilitar un dispositivo MFA virtual para el usuario Cuenta de AWS raíz \(consola\)](#) en la Guía del usuario de IAM.

Creación de un usuario con acceso administrativo

1. Activar IAM Identity Center.

Consulte las instrucciones en [Activar AWS IAM Identity Center](#) en la Guía del usuario de AWS IAM Identity Center .

2. En IAM Identity Center, conceda acceso administrativo a un usuario.

Para ver un tutorial sobre su uso Directorio de IAM Identity Center como fuente de identidad, consulte [Configurar el acceso de los usuarios con la configuración predeterminada Directorio de IAM Identity Center en la](#) Guía del AWS IAM Identity Center usuario.

Iniciar sesión como usuario con acceso de administrador

- Para iniciar sesión con el usuario de IAM Identity Center, utilice la URL de inicio de sesión que se envió a la dirección de correo electrónico cuando creó el usuario de IAM Identity Center.

Para obtener ayuda para iniciar sesión con un usuario del Centro de identidades de IAM, consulte [Iniciar sesión en el portal de AWS acceso](#) en la Guía del AWS Sign-In usuario.

Concesión de acceso a usuarios adicionales

1. En IAM Identity Center, cree un conjunto de permisos que siga la práctica recomendada de aplicar permisos de privilegios mínimos.

Para conocer las instrucciones, consulte [Create a permission set](#) en la Guía del usuario de AWS IAM Identity Center .

2. Asigne usuarios a un grupo y, a continuación, asigne el acceso de inicio de sesión único al grupo.

Para conocer las instrucciones, consulte [Add groups](#) en la Guía del usuario de AWS IAM Identity Center .

Requisitos previos

Para seguir los procedimientos de esta guía, necesitará un shell o un terminal de línea de comando donde pueda ejecutar los comandos. Los comandos aparecen en listas y van precedidos del símbolo del sistema (\$) y del nombre del directorio actual, si es aplicable.

```
~/eb-project$ this is a command  
this is output
```

En Linux y macOS, puede utilizar el administrador de shell y paquetes preferido. En Windows, puede [instalar el subsistema de Windows para Linux](#) y obtener una versión de Ubuntu y Bash integrada en Windows.

CLI DE EB

En este tutorial también se utiliza la interfaz de línea de comandos de Elastic Beanstalk (CLI de EB). Para obtener detalles sobre la instalación y configuración de la CLI de EB, consulte [Instalación de la CLI de EB](#) y [Configuración de la CLI de EB](#).

Paso 1: Crea una aplicación Go

Cree el directorio del proyecto.

```
~$ mkdir eb-go  
~$ cd eb-go
```

A continuación, vamos a crear una aplicación que implementará con Elastic Beanstalk. Crearemos el servicio web RESTful "Hello World".

En este ejemplo, se imprime un saludo personalizado que varía en función de la ruta utilizada para obtener acceso al servicio.

En este directorio, cree un archivo de texto llamado `application.go` con el contenido siguiente:

Example `~/eb-go/application.go`

```
package main

import (
    "fmt"
    "net/http"
)

func handler(w http.ResponseWriter, r *http.Request) {
    if r.URL.Path == "/" {
        fmt.Fprintf(w, "Hello World! Append a name to the URL to say hello. For example, use %s/Mary to say hello to Mary.", r.Host)
    } else {
        fmt.Fprintf(w, "Hello, %s!", r.URL.Path[1:])
    }
}

func main() {
    http.HandleFunc("/", handler)
    http.ListenAndServe(":5000", nil)
}
```

Paso 2: Implemente la aplicación Go con la CLI de EB

A continuación, cree el entorno de la aplicación e implemente su aplicación configurada con Elastic Beanstalk.

Para crear un entorno e implementar la aplicación de Go

1. Inicialice el repositorio de la CLI de EB con el comando `eb init`.

```
~/eb-go$ eb init -p go go-tutorial --region us-east-2
```

```
Application go-tutorial has been created.
```

Este comando crea una aplicación con un nombre `go-tutorial` y configura su repositorio local para crear entornos con la versión más reciente de la plataforma Go.

- (Opcional) Ejecute de nuevo `eb init` para configurar un par de claves predeterminadas de forma que pueda usar SSH para conectarse a la instancia EC2 donde se ejecuta la aplicación.

```
~/eb-go$ eb init
Do you want to set up SSH for your instances?
(y/n): y
Select a keypair.
1) my-keypair
2) [ Create new KeyPair ]
```

Seleccione un par de claves si ya tiene uno o siga las instrucciones para crear uno. Si no ve el símbolo del sistema o más adelante necesita cambiar la configuración, ejecute `eb init -i`.

- Cree un entorno e implemente la aplicación en él con `eb create`. Elastic Beanstalk crea automáticamente un archivo zip para la aplicación y lo inicia en el puerto 5000.

```
~/eb-go$ eb create go-env
```

Elastic Beanstalk tarda unos cinco minutos en crear el entorno.

Paso 3: Ejecute la aplicación en Elastic Beanstalk

Cuando finalice el proceso de creación del entorno, abra su sitio web con `eb open`

```
~/eb-go$ eb open
```

¡Enhorabuena! ¡Ha implementado una aplicación Go con Elastic Beanstalk! Se abre una ventana del navegador con el nombre de dominio creado para su aplicación.

Paso 4: Limpiar

Puede cerrar su entorno cuando termine de trabajar con la aplicación. Elastic Beanstalk AWS cancela todos los recursos asociados a su entorno.

Para finalizar el entorno de Elastic Beanstalk con la CLI de EB, ejecute el siguiente comando.

```
~/eb-go$ eb terminate
```

AWS recursos para su aplicación

Acabas de crear una aplicación de instancia única. Sirve como una aplicación de muestra sencilla con una sola instancia EC2, por lo que no requiere balanceo de carga ni escalado automático. Para las aplicaciones de instancia única, Elastic Beanstalk crea los siguientes recursos: AWS

- EC2 instance (Instancia EC2): máquina virtual de Amazon EC2 configurada para ejecutar aplicaciones web en la plataforma que elija.

Cada plataforma ejecuta un conjunto distinto de software, archivos de configuración y scripts compatibles con una determinada versión de lenguaje, marco y contenedor web (o una combinación de ellos). La mayoría de las plataformas utilizan Apache o nginx como un proxy inverso que procesa el tráfico web delante de la aplicación web, reenvía las solicitudes a esta, administra los recursos estáticos y genera registros de acceso y errores.

- Instance security group (Grupo de seguridad de la instancia): grupo de seguridad de Amazon EC2 configurado para permitir el tráfico entrante en el puerto 80. Este recurso permite que el tráfico HTTP procedente del balanceador de carga llegue a la instancia EC2 en la que se ejecuta la aplicación web. De forma predeterminada, el tráfico no está permitido en otros puertos.
- Bucket de Amazon S3: ubicación de almacenamiento para el código fuente, los registros y otros artefactos que se crean al utilizar Elastic Beanstalk.
- CloudWatch Alarmas de Amazon: dos CloudWatch alarmas que monitorean la carga de las instancias de su entorno y se activan si la carga es demasiado alta o demasiado baja. Cuando se activa una alarma, en respuesta, el grupo de Auto Scaling aumenta o reduce los recursos.
- AWS CloudFormation pila: Elastic AWS CloudFormation Beanstalk se utiliza para lanzar los recursos de su entorno y propagar los cambios de configuración. Los recursos se definen en una plantilla que puede verse en la [consola de AWS CloudFormation](#).
- Nombre de dominio: nombre de dominio que direcciona el tráfico a la aplicación web con el formato *subdominio.región.elasticbeanstalk.com*.

Elastic Beanstalk administra todos estos recursos. Cuando termina su entorno, Elastic Beanstalk termina todos los recursos que este contiene.

Siguientes pasos

Una vez que disponga de un entorno que ejecute una aplicación, podrá implementar una nueva versión de la aplicación o una aplicación distinta en cualquier momento. La implementación de una nueva versión de la aplicación es una tarea muy rápida, ya que no se requiere aprovisionar ni reiniciar instancias EC2. También puede explorar su nuevo entorno con la consola de Elastic Beanstalk. Para ver los pasos detallados, consulte [Explore su entorno](#) en el capítulo Introducción de esta guía.

Después de que implemente una aplicación de ejemplo o dos y esté listo para empezar a desarrollar y poner en funcionamiento las aplicaciones de Go de forma local, consulte [Configuración del entorno de desarrollo de Go](#).

Implemente con la consola de Elastic Beanstalk

También puede usar la consola de Elastic Beanstalk para iniciar la aplicación de muestra. Para ver los pasos detallados, consulte [Crear una aplicación de ejemplo](#) en el capítulo Introducción de esta guía.

Configuración del entorno de desarrollo de Go

Configure un entorno de desarrollo de Go para probar la aplicación de forma local antes de implementarla en AWS Elastic Beanstalk. Este tema describe los pasos para configurar sus entorno de desarrollo, e incluye enlaces a páginas de instalación de herramientas útiles.

Para ver procedimientos de configuración y herramientas comunes que se utilizan en todos los lenguajes, consulte [Configuración de su máquina de desarrollo para su uso con Elastic Beanstalk](#).

Instalación de Go

Para ejecutar las aplicaciones de Go de forma local, instale Go. Si no necesita una versión específica, obtenga la versión más reciente que es compatible con Elastic Beanstalk. Consulte [Go](#) en el documento Plataformas de AWS Elastic Beanstalk para ver una lista de las versiones que se admiten actualmente.

Descárguese Go en <https://golang.org/doc/install>.

Instalación del SDK de AWS para Go

Si necesita administrar recursos de AWS en la aplicación, instale el SDK de AWS para Go con el siguiente comando.

```
$ go get github.com/aws/aws-sdk-go
```

Para obtener más información, consulte el [SDK de AWS para Go](#).

Uso de la plataforma Elastic Beanstalk Go

Puede utilizar AWS Elastic Beanstalk para ejecutar, crear y configurar aplicaciones basadas en Go. Existen dos formas de implementar aplicaciones sencillas de Go:

- Puede proporcionar un paquete de código fuente con un archivo en la raíz llamado `application.go` que contenga el paquete principal de la aplicación. Elastic Beanstalk crea el binario con el siguiente comando:

```
go build -o bin/application application.go
```

Después de compilar la aplicación, Elastic Beanstalk la inicia en el puerto 5000.

- Puede proporcionar un paquete de código fuente con un archivo binario llamado `application`. El archivo binario puede estar ubicado en la raíz o en el directorio `bin/` del paquete de código fuente. Si sitúa el archivo binario `application` en las dos ubicaciones, Elastic Beanstalk utiliza el archivo del directorio `bin/`.

Elastic Beanstalk lanza esta aplicación en el puerto 5000.

En ambos casos, con Go 1.11 o posterior, también puede proporcionar requisitos de módulo en un archivo llamado `go.mod`. Para obtener más información, consulte [Migración a módulos Go](#) en el blog Go.

Existen dos modos de implementar aplicaciones complejas de Go:

- Puede proporcionar un paquete de código fuente que incluya los archivos de código fuente de la aplicación junto con un archivo [Buildfile](#) y un archivo [Procfile](#). El archivo Buildfile contiene un comando para crear la aplicación y el archivo Procfile incluye instrucciones para ejecutar la aplicación.
- Puede proporcionar un paquete de código fuente que incluya los archivos binarios de la aplicación junto con un archivo Procfile. El archivo Procfile contiene instrucciones para ejecutar la aplicación.

La plataforma Go incluye un servidor proxy para servir recursos estáticos, reenviar el tráfico a la aplicación y comprimir las respuestas. Puede [ampliar o invalidar la configuración del proxy predeterminada](#) para escenarios avanzados.

Para obtener más información sobre las diversas formas en las que puede ampliar una plataforma Elastic Beanstalk basada en Linux, consulte [the section called “Ampliación de plataformas Linux”](#).

Configuración del entorno de Go

La configuración de la plataforma Go le permite ajustar el comportamiento de sus instancias de Amazon EC2. Puede editar la configuración de instancia de Amazon EC2 del entorno de Elastic Beanstalk mediante la consola de Elastic Beanstalk.

Utilice la consola de Elastic Beanstalk para habilitar la rotación de registros en Amazon S3 y configurar variables que la aplicación pueda leer desde el entorno.

Para configurar el entorno Go en la consola de Elastic Beanstalk

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Actualizaciones, supervisión y registro, seleccione Editar.

Log Options (Opciones de registro)

La sección Log Options tiene dos valores:

- Instance profile (Perfil de instancia): especifica el perfil de instancia que tiene permiso para obtener acceso al bucket de Amazon S3 asociado con la aplicación.
- Enable log file rotation to Amazon S3 (Habilitar la rotación de archivos de registro para Amazon S3): especifica si los archivos de registro de las instancias de Amazon EC2 de la aplicación se copian en el bucket de Amazon S3 asociado a la aplicación.

Archivos estáticos

Para mejorar el rendimiento, puede utilizar la sección Static files (Archivos estáticos) para configurar el servidor proxy para entregar archivos estáticos de servidor (por ejemplo, HTML o imágenes) desde un conjunto de directorios dentro de su aplicación web. Para cada directorio, se establece la ruta virtual para la asignación de directorios. Cuando el servidor proxy recibe una solicitud de un archivo en la ruta especificada, proporciona directamente el archivo en lugar de direccionar la solicitud a la aplicación.

Para obtener más información sobre la configuración de archivos estáticos mediante archivos de configuración o la consola de Elastic Beanstalk, consulte [the section called “Archivos estáticos”](#).

Propiedades del entorno

La sección Environment Properties (Propiedades de entorno) le permite especificar opciones de configuración del entorno en las instancias de Amazon EC2 que ejecutan la aplicación. Las propiedades del entorno se pasan como pares de clave-valor a la aplicación.

Dentro del entorno de Go que se ejecuta en Elastic Beanstalk, es posible acceder a las variables de entorno mediante la función `os.Getenv`. Por ejemplo, podría leer una propiedad denominada `API_ENDPOINT` en una variable con el siguiente código:

```
endpoint := os.Getenv("API_ENDPOINT")
```

Para obtener más información, consulte [Propiedades del entorno y otras opciones del software](#).

Espacios de nombres de configuración de Go

Puede usar un [archivo de configuración](#) para definir opciones de configuración y realizar otras tareas de configuración en las instancias durante las implementaciones. Las opciones de configuración se pueden definir a través del servicio de Elastic Beanstalk o la plataforma que utilice y están organizadas por espacios de nombres.

La plataforma Go no define ningún espacio de nombres específico de la plataforma. Puede configurar el proxy para que sirva archivos estáticos mediante el espacio de nombres `aws:elasticbeanstalk:environment:proxy:staticfiles`. Para obtener información detallada y un ejemplo, consulte [the section called “Archivos estáticos”](#).

Elastic Beanstalk cuenta con numerosas opciones de configuración para personalizar el entorno. Además de los archivos de configuración, también puede definir opciones en la consola,

configuraciones guardadas, la CLI de EB o la AWS CLI. Para obtener más información, consulte [Opciones de configuración](#).

Plataforma Go de la AMI de Amazon Linux (anterior a Amazon Linux 2)

Si su entorno Elastic Beanstalk Go utiliza una versión de la plataforma AMI de Amazon Linux (anterior a Amazon Linux 2), lea la información adicional de esta sección.

Notas

- La información de este tema solo se aplica a ramificaciones de plataforma basadas en la AMI de Amazon Linux (AL1). Las ramificaciones de la plataforma AL2023/AL2 son incompatibles con las versiones anteriores de la plataforma de la AMI de Amazon Linux (AL1) y requieren ajustes de configuración diferentes.
- El [18 de julio de 2022](#), Elastic Beanstalk estableció el estado de todas las ramificaciones de la plataforma basadas en la AMI de Amazon Linux (AL1) como retirado. Para obtener más información sobre la migración a una ramificación de la plataforma Amazon Linux 2023 actual y totalmente compatible, consulte [Migración de su aplicación de Linux de Elastic Beanstalk a Amazon Linux 2023 o Amazon Linux 2](#).

Ir a espacios de nombres de configuración: AMI de Amazon Linux (AL1)

Puede usar un [archivo de configuración](#) para definir opciones de configuración y realizar otras tareas de configuración en las instancias durante las implementaciones. Las opciones de configuración se pueden definir a través del servicio de Elastic Beanstalk o la plataforma que utilice y están organizadas por espacios de nombres.

Note

La información de este tema solo se aplica a ramificaciones de plataforma basadas en la AMI de Amazon Linux (AL1). Las ramificaciones de la plataforma AL2023/AL2 son incompatibles con las versiones anteriores de la plataforma de la AMI de Amazon Linux (AL1) y requieren ajustes de configuración diferentes.

La plataforma Go AMI de Amazon admite un espacio de nombres de configuración específico de la plataforma, además de los [espacios de nombre compatibles con todas las plataformas](#). El espacio

de nombres `aws:elasticbeanstalk:container:golang:staticfiles` le permite definir opciones que asignen las rutas de su aplicación web a carpetas del paquete de código fuente de la aplicación que incluyen contenido estático.

Por ejemplo, este [archivo de configuración](#) le indica el servidor proxy que proporcione los archivos en la carpeta `staticimages` de la ruta `/images`:

Example `.ebextensions/go-settings.config`

```
option_settings:
  aws:elasticbeanstalk:container:golang:staticfiles:
    /html: statichtml
    /images: staticimages
```

Elastic Beanstalk cuenta con numerosas opciones de configuración para personalizar el entorno. Además de los archivos de configuración, también puede definir opciones en la consola, configuraciones guardadas, la CLI de EB o la AWS CLI. Para obtener más información, consulte [Opciones de configuración](#).

Configuración del proceso de la aplicación con un Procfile

Si desea especificar un comando personalizado que inicie una aplicación de Go, incluya un archivo llamado `Procfile` en la raíz del paquete de código fuente.

Para obtener más información sobre cómo escribir y utilizar un `Procfile`, expanda la sección `Buildfile` y `Procfile` en [the section called “Ampliación de plataformas Linux”](#).

Example `Procfile`

```
web: bin/server
queue_process: bin/queue_processor
foo: bin/fooapp
```

Debe llamar a la aplicación web, principal y enumerarla como el primer comando en su `Procfile`. Elastic Beanstalk expone la aplicación web principal en la raíz de la URL entorno; por ejemplo, `http://my-go-env.elasticbeanstalk.com`.

Elastic Beanstalk también ejecuta todas las aplicaciones cuyo nombre no contiene el prefijo `web_`, aunque estas aplicaciones no están disponibles desde el exterior de la instancia.

Elastic Beanstalk espera que los procesos del archivo `Procfile` se ejecuten de forma continua. Elastic Beanstalk supervisa estas aplicaciones y reinicia todos los procesos que terminan. En los procesos de corta ejecución, utilice un comando [Buildfile](#).

Uso de un archivo Procfile en la AMI de Amazon Linux (anterior a Amazon Linux 2)

Si su entorno Elastic Beanstalk Go utiliza una versión de la plataforma AMI de Amazon Linux (anterior a Amazon Linux 2), lea la información adicional de esta sección.

Notas

- La información de este tema solo se aplica a ramificaciones de plataforma basadas en la AMI de Amazon Linux (AL1). Las ramificaciones de la plataforma AL2023/AL2 son incompatibles con las versiones anteriores de la plataforma de la AMI de Amazon Linux (AL1) y requieren ajustes de configuración diferentes.
- El [18 de julio de 2022](#), Elastic Beanstalk estableció el estado de todas las ramificaciones de la plataforma basadas en la AMI de Amazon Linux (AL1) como retirado. Para obtener más información sobre la migración a una ramificación de la plataforma Amazon Linux 2023 actual y totalmente compatible, consulte [Migración de su aplicación de Linux de Elastic Beanstalk a Amazon Linux 2023 o Amazon Linux 2](#).

Transferencia de puertos: AMI de Amazon Linux (AL1)

Note

La información de este tema solo se aplica a ramificaciones de plataforma basadas en la AMI de Amazon Linux (AL1). Las ramificaciones de la plataforma AL2023/AL2 son incompatibles con las versiones anteriores de la plataforma de la AMI de Amazon Linux (AL1) y requieren ajustes de configuración diferentes.

Elastic Beanstalk configura el proxy nginx para redirigir solicitudes a la aplicación en el número de puerto especificado en la PORT [propiedad del entorno](#) de la aplicación. La aplicación siempre debe usar ese puerto como escucha. Para obtener acceso a esta variable dentro de la aplicación, llame al método `os.Getenv("PORT")`.

Elastic Beanstalk utiliza el número de puerto especificado en la propiedad del entorno PORT como puerto de la primera aplicación de Procfile y después va a aumentando el número de puerto de las siguientes aplicaciones de Procfile en incrementos de 100. Si no se establece la propiedad del entorno PORT, Elastic Beanstalk utiliza 5000 para el puerto inicial.

En el ejemplo anterior, la propiedad de entorno PORT de la aplicación web es 5000, la variable de la aplicación queue_process es 5100 y la variable de la aplicación foo es 5200.

Puede especificar el puerto inicial utilizando la opción PORT con el espacio de nombres [aws:elasticbeanstalk:application:environment](#), tal y como se muestra en el ejemplo siguiente.

```
option_settings:
  - namespace: aws:elasticbeanstalk:application:environment
    option_name: PORT
    value: <first_port_number>
```

Para obtener más información sobre cómo configurar las propiedades de la aplicación, consulte [Opciones de configuración](#).

Compilación de ejecutable en el servidor con un Buildfile

Si desea especificar un comando personalizado de configuración y compilación para la aplicación de Go, incluya un archivo llamado Buildfile en la raíz del paquete de código fuente. El nombre de archivo distingue entre mayúsculas y minúsculas. Utilice el formato siguiente en el archivo Buildfile:

```
<process_name>: <command>
```

El comando del archivo Buildfile debe ajustarse a la siguiente expresión regular: `^[A-Za-z0-9_]+:\s*\.+\$`.

Elastic Beanstalk no monitoriza la aplicación que se ejecuta con un Buildfile. Utilice un archivo Buildfile con comandos que se ejecuten durante cortos periodos de tiempo y terminen después de completar las tareas. Para procesos de larga educación que no deben terminar, utilice en su lugar el archivo [Procfile](#).

En el siguiente ejemplo de Buildfile, build.sh es un script de shell que se encuentra en la raíz del paquete de código fuente:

```
make: ./build.sh
```

Todas las rutas de `Buildfile` son rutas relativas que hacen referencia a la raíz del paquete de código fuente. Si sabe con antelación dónde se van a situar los archivos en la instancia, puede incluir rutas absolutas en `Buildfile`.

Configuración del proxy inverso

Elastic Beanstalk utiliza nginx como proxy inverso para asignar su aplicación al balanceador de carga de Elastic Load Balancing en el puerto 80. Elastic Beanstalk proporciona una configuración de nginx predeterminada que puede ampliar o anular por completo con su propia configuración.

De forma predeterminada, Elastic Beanstalk configura el proxy nginx para reenviar las solicitudes a la aplicación en el puerto 5000. Puede invalidar el puerto predeterminado estableciendo la [propiedad del entorno](#) `PORT` en el puerto en el que desea que la aplicación principal esté a la escucha.

Note

El puerto en que su aplicación está a la escucha no afecta al puerto en el que el servidor nginx está a la escucha para recibir las solicitudes del balanceador de carga.

Configuración del servidor proxy en la versión de su plataforma

Todas las plataformas AL2023/AL2 son compatibles con una característica de configuración de proxy uniforme. Para obtener más información sobre la configuración del servidor proxy en las versiones de su plataforma que ejecutan AL2023/AL2, expanda la sección Configuración del proxy inverso de [the section called “Ampliación de plataformas Linux”](#).

Configuración del proxy en la AMI de Amazon Linux (anterior a Amazon Linux 2)

Notas

- La información de este tema solo se aplica a ramificaciones de plataforma basadas en la AMI de Amazon Linux (AL1). Las ramificaciones de la plataforma AL2023/AL2 son incompatibles con las versiones anteriores de la plataforma de la AMI de Amazon Linux (AL1) y requieren ajustes de configuración diferentes.
- El [18 de julio de 2022](#), Elastic Beanstalk estableció el estado de todas las ramificaciones de la plataforma basadas en la AMI de Amazon Linux (AL1) como retirado. Para obtener más información sobre la migración a una ramificación de la plataforma Amazon Linux

2023 actual y totalmente compatible, consulte [Migración de su aplicación de Linux de Elastic Beanstalk a Amazon Linux 2023 o Amazon Linux 2](#).

Si su entorno Elastic Beanstalk Go utiliza una versión de la plataforma AMI de Amazon Linux (anterior a Amazon Linux 2), lea la información de esta sección.

Ampliación y anulación de la configuración predeterminada del proxy: AMI de Amazon Linux (AL1)

Elastic Beanstalk utiliza nginx como proxy inverso para asignar la aplicación al balanceador de carga del puerto 80. Si desea proporcionar su propia configuración nginx, puede anular la configuración predeterminada proporcionada por Elastic Beanstalk agregando el archivo `.ebextensions/nginx/nginx.conf` en el paquete de código fuente. Si este archivo está presente, Elastic Beanstalk lo utiliza en lugar del archivo de configuración nginx predeterminado.

Si también desea incluir directivas en el bloque `nginx.conf http`, puede proporcionar otros archivos de configuración en el directorio `.ebextensions/nginx/conf.d/` del paquete de código fuente. Todos los archivos de este directorio debe tener la extensión `.conf`.

Para aprovechar la funcionalidad proporcionada por Elastic Beanstalk, como [Informes y monitorización de estado mejorados](#), las asignaciones automáticos de aplicaciones y los archivos estáticos, debe incluir la línea siguiente en el bloque `server` del archivo de configuración nginx:

```
include conf.d/elasticbeanstalk/*.conf;
```

Creación e implementación de aplicaciones Java en Elastic Beanstalk

AWS Elastic Beanstalk admite dos plataformas para aplicaciones Java.

- Tomcat: una plataforma basada en Apache Tomcat, un contenedor web de código abierto para aplicaciones que utilizan servlets de Java y JavaServer Pages (JSP) para servir solicitudes HTTP. Tomcat facilita el desarrollo de aplicaciones web proporcionando varios subprocesos, configuración de seguridad declarativa y amplia capacidad de personalización. Elastic Beanstalk tiene ramas de plataforma para cada una de las versiones principales actuales de Tomcat. Para obtener más información, consulte [La plataforma Tomcat](#).
- Java SE: plataforma para aplicaciones que no utilizan un contenedor web o utilizan otro distinto de Tomcat, como Jetty o GlassFish. Puede incluir cualquier archivo Java de la biblioteca (JAR)

usado por la aplicación en el paquete de código fuente que implementa en Elastic Beanstalk. Para obtener más información, consulte [La plataforma Java SE](#).

Las ramificaciones recientes de las plataformas Tomcat y Java SE se basan en Amazon Linux 2 y posterior, y utilizan Corretto, la distribución de Java SE de AWS. Los nombres de estas ramificaciones en las listas de plataformas incluyen la palabra Corretto en lugar de Java, por ejemplo, Corretto 11 with Tomcat 8.5.

Para obtener una lista de las versiones actuales de la plataforma, consulte [Tomcat](#) y [Java SE](#) en la guía Plataformas de AWS Elastic Beanstalk.

AWS proporciona varias herramientas para trabajar con Java y Elastic Beanstalk.

Independientemente de la ramificación de plataforma que elija, puede utilizar el [SDK de AWS para Java](#) para usar otros servicios de AWS desde dentro de la aplicación Java. El SDK de AWS para Java es un conjunto de bibliotecas que le permiten utilizar las API de AWS desde el código de su aplicación sin necesidad de escribir llamadas HTTP sin procesar desde cero.

Si utiliza el entorno de desarrollo integrado (IDE) de Eclipse para desarrollar su aplicación Java, también puede obtener [AWS Toolkit for Eclipse](#). El conjunto de herramientas Toolkit for Eclipse de AWS es un complemento de código abierto que le permite administrar los recursos de AWS, incluidas las aplicaciones y los entornos de Elastic Beanstalk, desde el IDE de Eclipse.

Si prefiere utilizar la línea de comandos, instale la [interfaz de línea de comandos de Elastic Beanstalk](#) (CLI de EB) y úsela para crear, monitorizar y administrar sus entornos de Elastic Beanstalk desde la línea de comandos. Si ejecuta varios entornos para su aplicación, la CLI de EB se integra con Git para que pueda asociar cada uno de los entornos con una ramificación de Git diferente.

En los temas de este capítulo se supone que tiene conocimientos de los entornos de Elastic Beanstalk. Si no ha usado Elastic Beanstalk antes, pruebe el [tutorial de introducción](#) para conocer los conceptos básicos.

Temas

- [Introducción a Java en Elastic Beanstalk](#)
- [Configuración del entorno de desarrollo de Java](#)
- [Uso de la plataforma Tomcat de Elastic Beanstalk](#)
- [Uso de la plataforma Java SE de Elastic Beanstalk](#)
- [Añadir una instancia de base de datos de Amazon RDS a un entorno de aplicaciones Java](#)
- [Uso del conjunto de herramientas AWS Toolkit for Eclipse](#)

- [Recursos](#)

Introducción a Java en Elastic Beanstalk

Para comenzar a utilizar aplicaciones Java en AWS Elastic Beanstalk, lo único que necesita es el [paquete de código fuente](#) de la aplicación que va a cargar como su primera versión de la aplicación y va a implementar en un entorno. Cuando crea un entorno, Elastic Beanstalk asigna todos los recursos de AWS necesarios para ejecutar una aplicación web escalable.

Lanzar un entorno con una aplicación Java de ejemplo

Elastic Beanstalk proporciona aplicaciones de ejemplo de una sola página para cada plataforma, así como ejemplos más complejos que ilustran el uso de recursos de AWS adicionales como Amazon RDS y características y API específicas de la plataforma o del lenguaje.

Los ejemplos de una sola página muestran el mismo código que obtiene cuando crea un entorno sin proporcionar su propio código fuente. Los ejemplos más complejos se encuentran en GitHub y puede que sea necesario compilarlos para implementarlos en un entorno de Elastic Beanstalk.

Muestras

Nom	Versiones compatibles	Tipo de entorr	Fuente	Descripción
Tomcat (página única)	Todas las ramas de plataforma Tomcat con Corretto	Servicio web Entorno de trabaj	tomcat.zi p	<p>Aplicación web Tomcat con una sola página (<code>index.jsp</code>) configurada para mostrarse en la raíz del sitio web.</p> <p>Para los entornos de trabajo, este ejemplo incluye un archivo <code>cron.yaml</code> que configura una tarea programada que llama a <code>scheduled.jsp</code> una vez por minuto. Cuando se llama <code>scheduled.jsp</code>, se escribe un archivo log en <code>/tmp/sample-app.log</code>. Por último, se incluye un archivo de configuración en <code>.ebextensions</code> que copia los registros de <code>/tmp/a</code></p>

Nombre	Versiones compatibles	Tipo de entorno	Fuente	Descripción
				<p>las ubicaciones que Elastic Beanstalk lee cuando solicita registros del entorno.</p> <p>Si habilita la integración de X-Ray en un entorno que ejecute este ejemplo, la aplicación muestra contenido adicional sobre X-Ray y proporciona una opción para generar información de depuración que puede ver en la consola de X-Ray.</p>
Corretto (página única)	Corretto 11 Corretto 8	Servicio web	corretto.zip	<p>Aplicación Corretto con archivos de configuración <code>Buildfile</code> y <code>Procfile</code>.</p> <p>Si habilita la integración de X-Ray en un entorno que ejecute este ejemplo, la aplicación muestra contenido adicional sobre X-Ray y proporciona una opción para generar información de depuración que puede ver en la consola de X-Ray.</p>

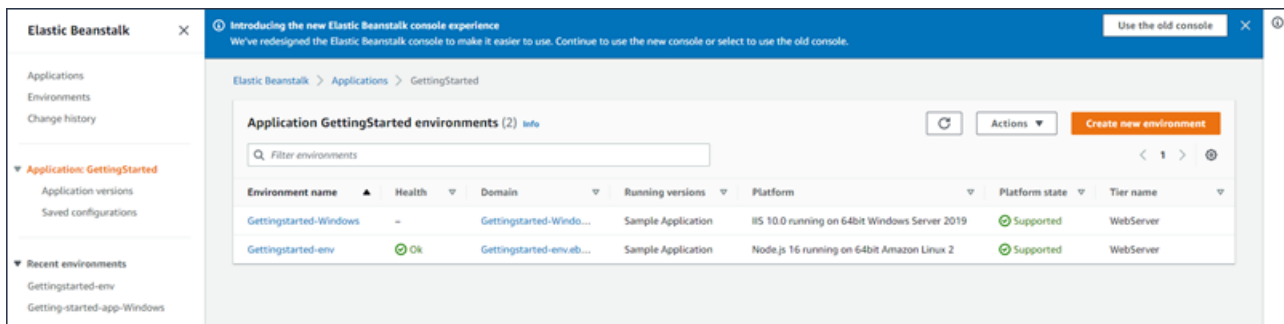
Nombre	Versiones compatibles	Tipo de entorno	Fuente	Descripción
Scorekeep	Java 8	Servicio web	Clonar el repositorio en GitHub.com	<p>Scorekeep es una API web RESTful que utiliza la plataforma Spring a fin de ofrecer una interfaz para la creación y administración de usuarios, sesiones y juegos. La API está empaquetada con una aplicación web de Angular 1.5 que utiliza la API a través de HTTP.</p> <p>La aplicación utiliza características de la plataforma de Java SE para descargar dependencias y compilarse en la instancia , lo que minimiza el tamaño del paquete de código fuente. La aplicación incluye también archivos de configuración nginx que invalidan la configuración predeterminada para servir la aplicación web front-end estáticamente en el puerto 80 a través del proxy y dirigir las solicitudes dirigidas a las rutas incluidas bajo /api a la API que se ejecuta en localhost:5000 .</p> <p>Scorekeep también incluye una rama xray que muestra cómo instrumentar una aplicación Java para usar con AWS X-Ray. Muestra la instrumentación de solicitudes HTTP entrantes con un filtro servlet, la instrumentación automática y manual de un cliente con un SDK de AWS, la configuración de registro y la instrumentación de solicitudes HTTP salientes y clientes SQL.</p>

Nombre	Versiones compatibles	Tipo de entorno	Fuente	Descripción
				<p>Consulte el archivo readme para obtener instrucciones o use el Tutorial de introducción de AWS X-Ray para probar la aplicación con X-Ray.</p>
Does it Have Snakes?	Tomcat 8 con Java 8	Servicio web	Clonar el repositorio en GitHub.com	<p>Does it Have Snakes? es una aplicación web de Tomcat que muestra el uso de archivos de configuración de Elastic Beanstalk, Amazon RDS, JDBC, PostgreSQL, Servlets, JSP, Simple Tag Support, Tag Files, Log4J, Bootstrap y Jackson.</p> <p>El código fuente de este proyecto incluye un script de compilación mínimo que compila los servlets y modelos en archivos de clases y empaqueta los archivos necesarios en un archivo web que puede implementar en un entorno de Elastic Beanstalk. Consulte el archivo readme del repositorio del proyecto para obtener instrucciones completas.</p>
Locust Load Generator	Java 8	Servicio web	Clonar el repositorio en GitHub.com	<p>Aplicación web que puede utilizar para realizar una prueba de carga de otra aplicación web que se ejecute en un entorno de Elastic Beanstalk diferente. Muestra el uso de los archivos Buildfile y Procfile, DynamoDB y Locust, una herramienta de pruebas de carga de código abierto.</p>

Descargue cualquiera de las aplicaciones de ejemplo e impleméntelas en Elastic Beanstalk siguiendo estos pasos:

Para lanzar un entorno con una aplicación de ejemplo (consola)

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Applications (Aplicaciones) y, a continuación, elija el nombre de una aplicación existente de la lista o [cree una](#).
3. En la página de información general de la aplicación, seleccione Create new environment (Crear un nuevo entorno).



Esto abre el asistente de Create environment (Creación de entorno). El asistente proporciona un conjunto de pasos para crear un entorno nuevo.

Step 1
Configure environment

Step 2
Configure service access

Step 3 - optional
Configure instance traffic and scaling

Step 4 - optional
Set up networking, database, and tags

Step 5 - optional
Configure updates, monitoring, and logging

Step 6
Review

Configure environment [Info](#)

Environment tier [Info](#)

Amazon Elastic Beanstalk has two types of environment tiers to support different types of web applications.

- Web server environment**
Run a website, web application, or web API that serves HTTP requests. [Learn more](#)
- Worker environment**
Run a worker application that processes long-running workloads on demand or performs tasks on a schedule. [Learn more](#)

Application information [Info](#)

Application name
GettingStarted
Maximum length of 100 characters.

▶ **Application tags (optional)**

Environment information [Info](#)

Choose the name, subdomain and description for your environment. These cannot be changed later.

Environment name
GettingStarted-env
Must be from 4 to 40 characters in length. The name can contain only letters, numbers, and hyphens. It can't start or end with a hyphen. This name must be unique within a region in your account.

Domain name
 .us-east-1.elasticbeanstalk.com

Environment description

Platform [Info](#)

Platform type

- Managed platform**
Platforms published and maintained by Amazon Elastic Beanstalk. [Learn more](#)
- Custom platform**
Platforms created and owned by you. This option is unavailable if you have no platforms.

Platform


Platform branch

Platform version

Application code [Info](#)


- Sample application**
- Existing version**
Application versions that you have uploaded.
- Upload your code**
Upload a source bundle from your computer or copy one from Amazon S3.

4. Para el [environmet tier](#) (nivel de entorno), elija Web server environment (Entorno del servidor web) o Worker environment (Entorno de trabajo). Una vez creado, no se puede cambiar la capa del entorno.

 Note

La [plataforma .NET en Windows Server](#) no admite la capa del entorno de trabajo.

5. En Plataforma, seleccione la plataforma y la ramificación de la plataforma que coincidan con el lenguaje utilizado por la aplicación.

 Note

Elastic Beanstalk admite varias [versiones](#) para la mayoría de las plataformas que se indican. De forma predeterminada, la consola selecciona la versión recomendada para la plataforma y la ramificación de la plataforma que elija. Si la aplicación requiere una versión diferente, puede seleccionarla aquí. Para obtener más información acerca de las versiones compatibles de la plataforma, consulte [the section called “Plataformas admitidas”](#).

6. En Application code (Código de aplicación), seleccione Sample application (Aplicación de muestra).
7. Para los Configuration presets (Modelos de configuración preestablecidos), elija Single instance (Instancia individual).
8. Elija Next (Siguiente).
9. Aparece la página Configurar acceso a servicio.

Configure service access [Info](#)

Service access
IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profiles allow Elastic Beanstalk to create and manage your environment. Both the IAM role and instance profile must be attached to IAM managed policies that contain the required permissions. [Learn more](#)

Service role

Create and use new service role
 Use an existing service role

Existing service roles
Choose an existing IAM role for Elastic Beanstalk to assume as a service role. The existing IAM role must have the required IAM managed policies.

aws-elasticbeanstalk-service-role

EC2 key pair
Select an EC2 key pair to securely log in to your EC2 instances. [Learn more](#)

Choose a key pair

EC2 instance profile
Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.

aws-elasticbeanstalk-ec2-role

10. Seleccione Crear un rol de servicio para Rol de servicio.

11. A continuación, nos centraremos en la lista desplegable Perfil de instancia de EC2. Los valores que aparecen en esta lista desplegable pueden variar, en función de si su cuenta ha creado previamente un entorno nuevo.

Seleccione una de las siguientes opciones, dependiendo de los valores que aparezcan en la lista.

- Si `aws-elasticbeanstalk-ec2-role` aparece en la lista desplegable, selecciónelo en la lista desplegable Perfil de instancia de EC2.
- Si aparece otro valor en la lista y es el perfil de instancia de EC2 predeterminado destinado a sus entornos, selecciónelo en la lista desplegable Perfil de instancia de EC2.
- Si en la lista desplegable Perfil de instancia de EC2 no aparecen valores para seleccionar, amplíe el procedimiento siguiente, Crear un rol de IAM para el perfil de instancia de EC2.

Siga los pasos de Crear un rol de IAM para el perfil de instancia de EC2 para crear un rol de IAM que pueda seleccionar posteriormente para el Perfil de instancia de EC2. Luego, vuelva a este paso.

Ahora que ha creado un rol de IAM y ha actualizado la lista, aparece como opción en la lista desplegable. Seleccione el rol de IAM que acaba de crear en la lista desplegable Perfil de instancia de EC2.

12. Seleccione Skip to review (Saltar a revisión) en la página Configure service access (Configurar el acceso al servicio).

Se seleccionarán los valores predeterminados para este paso y se omitirán los pasos opcionales.

13. En la página Review (Revisar), se muestra un resumen de todas sus elecciones.

Para personalizar aún más su entorno, elija Edit (Editar) junto al paso que incluye los elementos que desee configurar. Solo puede establecer las siguientes opciones durante la creación del entorno:

- Environment name
- Nombre de dominio
- Versión de la plataforma
- Procesador
- VPC
- datos y búsqueda

Puede modificar los siguientes ajustes después de crear el entorno, pero requieren que se aprovisionen nuevas instancias u otros recursos, lo que puede llevar algún tiempo:

- Tipo de instancia, volumen raíz, par de claves y rol de AWS Identity and Access Management (IAM)
- Base de datos interna de Amazon RDS
- Balanceador de carga

Para obtener más información sobre los ajustes disponibles, consulte [El asistente de creación de nuevo entorno](#).

14. Elija Submit (Enviar) en la parte inferior de la página para iniciar la creación del nuevo entorno.

Crear un rol de IAM para el perfil de instancia de EC2

Configure service access [Info](#)

Service access
IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profiles allow Elastic Beanstalk to create and manage your environment. Both the IAM role and instance profile must be attached to IAM managed policies that contain the required permissions. [Learn more](#)

Service role

Create and use new service role

Use an existing service role

Existing service roles
Choose an existing IAM role for Elastic Beanstalk to assume as a service role. The existing IAM role must have the required IAM managed policies.

aws-elasticbeanstalk-service-role

EC2 key pair
Select an EC2 key pair to securely log in to your EC2 instances. [Learn more](#)

Choose a key pair

EC2 instance profile
Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.

aws-elasticbeanstalk-ec2-role

Para crear un rol de IAM para la selección del perfil de instancia de EC2


1. Seleccione Ver detalles de permisos. Esta opción aparece en la lista desplegable Perfil de instancia de EC2.

Aparece una ventana modal titulada Ver permisos del perfil de instancia. En esta ventana se muestran los perfiles administrados que deberá asociar al nuevo perfil de instancia de EC2 que cree. También proporciona un enlace para ejecutar la consola de IAM.

2. Seleccione el enlace Consola de IAM que aparece en la parte superior de la ventana.
3. En el panel de navegación de la consola de IAM, elija Roles.
4. Elija Create role (Crear rol).
5. En Tipo de entidad de confianza, seleccione Servicio de AWS.
6. En Use case (Caso de uso), elija EC2.
7. Elija Next (Siguiente).
8. Asocie las políticas administradas adecuadas. Desplácese por la ventana modal Ver permisos del perfil de instancia para ver las políticas administradas. Las políticas también aparecen aquí:

- `AWSElasticBeanstalkWebTier`
- `AWSElasticBeanstalkWorkerTier`
- `AWSElasticBeanstalkMulticontainerDocker`

9. Elija Next (Siguiente).
10. Escriba un nombre para el rol.
11. (Opcional) Añada etiquetas al rol.
12. Elija Create role (Crear rol).
13. Vuelva a la ventana de la consola de Elastic Beanstalk que está abierta.
14. Cierre la ventana modal Ver permisos del perfil de instancia.

 Important

No cierre la página del navegador donde se muestra la consola de Elastic Beanstalk.

15. Seleccione



(actualizar), junto a la lista desplegable Perfil de instancia de EC2.

Esto actualiza la lista desplegable, para que el rol que acaba de crear aparezca en ella.

Pasos siguientes

Una vez que disponga de un entorno que ejecute una aplicación, podrá [implementar una nueva versión](#) de la aplicación o una aplicación totalmente diferente en cualquier momento. La implementación de una nueva versión de la aplicación es una tarea muy rápida, ya que no se requiere aprovisionar ni reiniciar instancias EC2.

Después de haber implementado una o dos aplicaciones de ejemplo y una vez que esté preparado para empezar a desarrollar y ejecutar aplicaciones Java localmente, consulte [la siguiente sección](#) para configurar un entorno de desarrollo de Java con todas las herramientas y bibliotecas que va a necesitar.

Configuración del entorno de desarrollo de Java

Configure el entorno de desarrollo de Java para probar la aplicación localmente antes de implementarla en AWS Elastic Beanstalk. En este tema, se explican los pasos de configuración del

entorno de desarrollo y se incluyen enlaces a páginas de instalación donde encontrará herramientas útiles.

Para ver procedimientos de configuración y herramientas comunes que se utilizan en todos los lenguajes, consulte [Configuración de su máquina de desarrollo](#).

Secciones

- [Instalación del kit de desarrollo de Java](#)
- [Instalación de un contenedor web](#)
- [Descarga de bibliotecas](#)
- [Instalación del SDK de AWS para Java](#)
- [Instalación de un IDE o un editor de texto](#)
- [Instalación del conjunto de herramientas AWS Toolkit for Eclipse](#)

Instalación del kit de desarrollo de Java

Instale el kit de desarrollo de Java (JDK). Si no tiene ninguna preferencia, obtenga la versión más reciente. Descargue JDK en oracle.com

JDK incluye el compilador de Java, que se puede utilizar para crear archivos de origen en los archivos de clases que se pueden ejecutar en un servidor web de Elastic Beanstalk.

Instalación de un contenedor web

Si aún no tiene otro marco o contenedor web, instale la versión apropiada de Tomcat:

- [Descargar Tomcat 8 \(requiere Java 7 o versiones posteriores\)](#)
- [Descargar Tomcat 7 \(requiere Java 6 o versiones posteriores\)](#)

Descarga de bibliotecas

Las plataformas Elastic Beanstalk contienen algunas bibliotecas de forma predeterminada. Descargue las bibliotecas que la aplicación va a utilizar y guárdelas en la carpeta de proyecto para implementar el paquete de código fuente de la aplicación.

Si ha instalado Tomcat localmente, puede copiar las bibliotecas de Servlet API y JavaServer Pages (JSP) API de la carpeta de instalación. Si la implementación se realiza con una versión de plataforma

de Tomcat, no tiene que incluir estos archivos en el paquete de código fuente, aunque deben estar en `classpath` para poder compilar las clases que los utilizan.

JUnit, Google Guava y Apache Commons cuentan con varias bibliotecas útiles. Visite sus páginas de inicio para obtener más información:

- [Download JUnit](#)
- [Download Google Guava](#)
- [Download Apache Commons](#)

Instalación del SDK de AWS para Java

Si necesita administrar recursos de AWS en la aplicación, instale el SDK de AWS para Java. Por ejemplo, con el AWS SDK for Java, puede utilizar Amazon DynamoDB (DynamoDB) para compartir los estados de sesión de las aplicaciones de Apache Tomcat en varios servidores web. Para obtener más información, consulte [Administración del estado de la sesión de Tomcat con Amazon DynamoDB](#) en la documentación del SDK de AWS para Java.

Visite la [página de inicio del SDK de AWS para Java](#) si desea obtener más información e instrucciones de instalación.

Instalación de un IDE o un editor de texto

Los entornos de desarrollo integrado (IDE) cuentan con una amplia gama de características que facilitan el desarrollo de aplicaciones. Si no ha utilizado un IDE para el desarrollo de Java, pruebe Eclipse e IntelliJ y determine cuál es el que mejor se adapta a sus necesidades.

- [Instalar Eclipse IDE for Java EE Developers](#)
- [Instalar IntelliJ](#)

Note

Un IDE podría agregar archivos a la carpeta de proyectos que tal vez no quiera confirmar en el control de código fuente. Para evitar confirmar estos archivos en el control de código fuente, utilice `.gitignore` o la característica equivalente de la herramienta de control de código fuente.

Si simplemente quiere comenzar a escribir el código y no necesita todas las características de un IDE, considere la posibilidad de [instalar Sublime Text](#).

Instalación del conjunto de herramientas AWS Toolkit for Eclipse

El [conjunto de herramientas AWS Toolkit for Eclipse](#) es un complemento de código abierto para el IDE de Eclipse para Java que facilita a los desarrolladores las tareas de desarrollo, depuración e implementación de aplicaciones Java a través de AWS. Visite la [página de inicio del conjunto de herramientas AWS Toolkit for Eclipse](#) para consultar las instrucciones de instalación.

Uso de la plataforma Tomcat de Elastic Beanstalk

Important

AWS Elastic Beanstalk instala Log4j desde los repositorios de paquetes predeterminados de Amazon Linux en sus plataformas Tomcat para Amazon Linux 1 y Amazon Linux 2. Las versiones de Log4j disponibles en los repositorios de Amazon Linux 1 y Amazon Linux 2 no se ven afectadas por [CVE-2021-44228](#) ni [CVE-2021-45046](#) en su configuración predeterminada.

Si ha cambiado la configuración del uso de log4j por parte de su aplicación o ha instalado versiones más recientes de log4j, le recomendamos que tome medidas para actualizar el código de la aplicación para mitigar este problema.

Por precaución, Elastic Beanstalk lanzó nuevas versiones de la plataforma que utilizan los repositorios de paquetes predeterminados de Amazon Linux más recientes, que incluyen el [JDK con hotpatches aplicados de Log4j](#), en el [lanzamiento de la plataforma de Amazon Linux del 21 de diciembre de 2021](#). Si ha personalizado la instalación de log4j como dependencia de la aplicación, le recomendamos que actualice a la versión más reciente de la plataforma Elastic Beanstalk para mitigar CVE-2021-44228 o CVE-2021-45046. También puede habilitar las actualizaciones administradas automatizadas como parte de las prácticas de actualización normales.

Para obtener más información sobre las actualizaciones de software relacionadas con la seguridad de Amazon Linux, consulte [Centro de seguridad de Amazon Linux](#).

La plataforma Tomcat de AWS Elastic Beanstalk es un conjunto de [versiones de plataformas](#) para aplicaciones web Java que se pueden ejecutar en un contenedor web Tomcat. Tomcat se ejecuta detrás de un servidor proxy nginx. Cada rama de plataforma corresponde a una versión principal de Tomcat, como Java 8 con Tomcat 8.

En la consola de Elastic Beanstalk hay opciones de configuración disponibles para [modificar la configuración de un entorno en ejecución](#). Para evitar perder la configuración del entorno cuando lo termina, puede usar las [configuraciones guardadas](#) para guardar la configuración y aplicarla posteriormente a otro entorno.

Para guardar la configuración en el código fuente, puede incluir [archivos de configuración](#). Los valores de configuración de los archivos de configuración se aplican cada vez que crea un entorno o que implementa la aplicación. También puede usar archivos de configuración para instalar paquetes, ejecutar scripts y llevar a cabo otras operaciones de personalización de instancias durante las implementaciones.

La plataforma Tomcat de Elastic Beanstalk incluye un proxy inverso que reenvía las solicitudes a su aplicación. Puede usar [opciones de configuración](#) para configurar el servidor proxy para servir recursos estáticos de una carpeta del código fuente con el fin de reducir la carga en su aplicación. Para escenarios avanzados, puede [incluir sus propios archivos .conf](#) en el paquete de código fuente para ampliar la configuración del proxy de Elastic Beanstalk o sobrescribirla completamente.

Note

Elastic Beanstalk admite [nginx](#) (el valor predeterminado) y [Apache HTTP Server](#) como servidores proxy en la plataforma Tomcat. Si su entorno Tomcat de Elastic Beanstalk utiliza una rama de plataforma de la AMI de Amazon Linux (anterior a Amazon Linux 2), también tiene la opción de utilizar [Apache HTTP Server versión 2.2](#). Apache (el más reciente) es el valor predeterminado en estas ramas de plataforma más antiguas.

El [18 de julio de 2022](#), Elastic Beanstalk estableció el estado de todas las ramificaciones de la plataforma basadas en la AMI de Amazon Linux (AL1) como retirado. Para obtener más información sobre la migración a una ramificación de la plataforma Amazon Linux 2023 actual y totalmente compatible, consulte [Migración de su aplicación de Linux de Elastic Beanstalk a Amazon Linux 2023 o Amazon Linux 2](#).

Debe empaquetar las aplicaciones Java en un archivo de aplicación web (WAR) con una estructura específica. Para obtener información sobre la estructura necesaria y su relación con la estructura de directorios de su proyecto, consulte [Estructuración de la carpeta de proyectos](#).

Para ejecutar varias aplicaciones en el mismo servidor web, puede [agrupar varios archivos WAR](#) en un solo paquete de código fuente. Cada aplicación de un paquete múltiple de código fuente WAR se ejecuta en la ruta raíz (ROOT.war se ejecuta en `myapp.elasticbeanstalk.com/`) o en una ruta

directamente debajo de ella (`app2.war` se ejecuta en `myapp.elasticbeanstalk.com/app2/`), según lo especifique el nombre del paquete WAR. En un único paquete de código fuente WAR, la aplicación siempre se ejecuta en la ruta raíz.

La configuración aplicada en la consola de Elastic Beanstalk anula la misma configuración en los archivos de configuración, si existe. Esto le permite tener la configuración predeterminada en los archivos de configuración y anularla con la configuración específica del entorno en la consola. Para obtener más información acerca de la prioridad y otros métodos para cambiar valores de configuración, consulte [Opciones de configuración](#).

Para obtener más información sobre las diversas formas en las que puede ampliar una plataforma Elastic Beanstalk basada en Linux, consulte [the section called “Ampliación de plataformas Linux”](#).

Temas

- [Configuración del entorno de Tomcat](#)
- [Espacios de nombres de configuración de Tomcat](#)
- [Agrupación de varios archivos WAR en entornos Tomcat](#)
- [Estructuración de la carpeta de proyectos](#)
- [Configuración del servidor proxy de su entorno Tomcat](#)

Configuración del entorno de Tomcat

La plataforma Tomcat de Elastic Beanstalk proporciona algunas opciones específicas de la plataforma, además de las opciones estándar que tienen todas las plataformas. Estas opciones le permiten configurar la máquina virtual Java (JVM) que se ejecuta en los servidores web del entorno y definir las propiedades del sistema que proporcionan las cadenas de configuración a su aplicación.

Puede utilizar la consola de Elastic Beanstalk para habilitar la rotación de registros en Amazon S3 y configurar variables que la aplicación pueda leer desde el entorno.

Para configurar el entorno Tomcat en la consola de Elastic Beanstalk

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Actualizaciones, supervisión y registro, seleccione Editar.

Opciones de contenedor

Puede especificar estas opciones específicas de la plataforma:

- Proxy server (Servidor proxy): el servidor proxy que utilizar en las instancias del entorno. De forma predeterminada, se utiliza nginx.

Opciones del contenedor de JVM

El tamaño de montón de la máquina virtual Java (JVM) determina la cantidad de objetos que su aplicación puede crear en memoria antes de que se produzca la [recopilación de elementos no utilizados](#). Puede modificar Initial JVM Heap Size (-Xms option) y Maximum JVM Heap Size (opción -Xmx). Un tamaño de montón inicial mayor permite que se creen más objetos antes de que se recopilen los elementos no utilizados, pero también implica que el recolector de elementos no utilizados tardará más tiempo en compactar el montón. El tamaño de montón máximo especifica la cantidad máxima de memoria que la JVM puede asignar al ampliar el montón durante un periodo de actividad intensa.

Note

La memoria disponible depende del tipo de instancia de Amazon EC2. Para obtener más información acerca de los tipos de instancias EC2 disponibles para su entorno de Elastic Beanstalk, consulte [Tipos de instancias](#) en la Guía del usuario de Amazon Elastic Compute Cloud para las instancias de Linux.

La generación permanente es una sección del montón de JVM que almacena definiciones de clases y metadatos asociados. Para modificar el tamaño de la generación permanente, especifique el nuevo tamaño en la opción Maximum JVM PermGen Size (-XX:MaxPermSize). Esta configuración solo se

aplica a Java 7 y versiones anteriores. Esta opción quedó obsoleta en JDK 8 y fue sustituida por la opción `MaxMetaspace` (`-XX:MaxMetaspaceSize`).

Important

JDK 17 eliminó la compatibilidad con la opción `-XX:MaxPermSize` de Java. El uso de esta opción con un entorno que se ejecuta en una ramificación de la plataforma de Elastic Beanstalk con Corretto 17 generará un error. Elastic Beanstalk publicó su primera ramificación de la plataforma que ejecuta Tomcat con Corretto 17 el [13 de julio de 2023](#). Para obtener más información, consulte los recursos siguientes.

- Sitio web de documentación de Oracle Java: [opciones de Java eliminadas](#)
- Sitio web de documentación de Oracle Java: sección de Metadatos de clase en [Otras consideraciones](#)

Para obtener más información sobre las plataformas de Elastic Beanstalk y sus componentes, consulte [Plataformas compatibles](#) en la guía de Plataformas de AWS Elastic Beanstalk.

Log Options (Opciones de registro)

La sección Log Options (Opciones de registro) tiene dos valores:

- Instance profile (Perfil de instancia): especifica el perfil de instancia que tiene permiso para obtener acceso al bucket de Amazon S3 asociado con la aplicación.
- Enable log file rotation to Amazon S3 (Habilitar la rotación de archivos de registro para Amazon S3): especifica si los archivos de registro de las instancias de Amazon EC2 de la aplicación se copian en el bucket de Amazon S3 asociado a la aplicación.

Archivos estáticos

Para mejorar el rendimiento, puede utilizar la sección Static files (Archivos estáticos) para configurar el servidor proxy para entregar archivos estáticos de servidor (por ejemplo, HTML o imágenes) desde un conjunto de directorios dentro de su aplicación web. Para cada directorio, se establece la ruta virtual para la asignación de directorios. Cuando el servidor proxy recibe una solicitud de un archivo en la ruta especificada, proporciona directamente el archivo en lugar de direccionar la solicitud a la aplicación.

Para obtener más información sobre la configuración de archivos estáticos mediante archivos de configuración o la consola de Elastic Beanstalk, consulte [the section called “Archivos estáticos”](#).

Propiedades del entorno

En la sección Environment Properties (Propiedades de entorno), puede especificar opciones de configuración del entorno en las instancias de Amazon EC2 que ejecutan su aplicación. Las propiedades del entorno se pasan como pares de clave-valor a la aplicación.

La plataforma Tomcat define una propiedad comodín denominada JDBC_CONNECTION_STRING para los entornos Tomcat para pasar una cadena de conexión a una base de datos externa.

Note

Si asocia una instancia de base de datos de RDS a su entorno, cree la cadena de conexión de JDBC dinámicamente desde las propiedades del entorno de Amazon Relational Database Service (Amazon RDS) proporcionadas por Elastic Beanstalk. Utilice JDBC_CONNECTION_STRING solo para las instancias de base de datos que no se aprovisionan con Elastic Beanstalk.

Para obtener más información sobre el uso de Amazon RDS con su aplicación Java, consulte [Añadir una instancia de base de datos de Amazon RDS a un entorno de aplicaciones Java](#).

Dentro del entorno Tomcat que se ejecuta en Elastic Beanstalk, las variables de entorno están accesibles mediante `System.getProperty()`. Por ejemplo, podría leer una propiedad denominada API_ENDPOINT en una variable con el siguiente código.

```
String endpoint = System.getProperty("API_ENDPOINT");
```

Para obtener más información, consulte [Propiedades del entorno y otras opciones del software](#).

Espacios de nombres de configuración de Tomcat

Puede usar un [archivo de configuración](#) para definir opciones de configuración y realizar otras tareas de configuración en las instancias durante las implementaciones. Las opciones de configuración se pueden definir a través del servicio de Elastic Beanstalk o la plataforma que utilice y están organizadas por espacios de nombres.

La plataforma Tomcat admite las opciones de los siguientes espacios de nombres, además de las [opciones admitidas para todos los entornos de Elastic Beanstalk](#):

- `aws:elasticbeanstalk:container:tomcat:jvmoptions`: modificar la configuración de JVM. Las opciones de este espacio de nombres se corresponden con las opciones de la consola de administración de la siguiente manera:
 - `Xms` – JVM command line options (opciones de la línea de comandos de JVM)
 - `JVM Options` – JVM command line options (opciones de la línea de comandos de JVM)
- `aws:elasticbeanstalk:environment:proxy`: elija el servidor proxy del entorno.

El siguiente archivo de configuración de ejemplo muestra el uso de las opciones de configuración específicas de Tomcat.

Example `.ebextensions/tomcat-settings.config`

```
option_settings:
  aws:elasticbeanstalk:container:tomcat:jvmoptions:
    Xms: 512m
    JVM Options: '-Xmn128m'
  aws:elasticbeanstalk:application:environment:
    API_ENDPOINT: mywebapi.zkpexsjtmd.us-west-2.elasticbeanstalk.com
  aws:elasticbeanstalk:environment:proxy:
    ProxyServer: apache
```

Elastic Beanstalk cuenta con numerosas opciones de configuración para personalizar el entorno. Además de los archivos de configuración, también puede definir opciones en la consola, configuraciones guardadas, la CLI de EB o la AWS CLI. Para obtener más información, consulte [Opciones de configuración](#).

Plataforma Tomcat de la AMI de Amazon Linux (anterior a Amazon Linux 2)

Si su entorno Tomcat de Elastic Beanstalk utiliza una versión de la plataforma de la AMI de Amazon Linux (anterior a Amazon Linux 2), lea la información adicional de esta sección.

Notas

- La información de este tema solo se aplica a ramificaciones de plataforma basadas en la AMI de Amazon Linux (AL1). Las ramificaciones de la plataforma AL2023/AL2 son incompatibles con las versiones anteriores de la plataforma de la AMI de Amazon Linux (AL1) y requieren ajustes de configuración diferentes.

- El [18 de julio de 2022](#), Elastic Beanstalk estableció el estado de todas las ramificaciones de la plataforma basadas en la AMI de Amazon Linux (AL1) como retirado. Para obtener más información sobre la migración a una ramificación de la plataforma Amazon Linux 2023 actual y totalmente compatible, consulte [Migración de su aplicación de Linux de Elastic Beanstalk a Amazon Linux 2023 o Amazon Linux 2](#).

Espacios de nombres de configuración de Tomcat: AMI de Amazon Linux (AL1)

La plataforma Tomcat de la AMI de Amazon Linux admite opciones adicionales en los siguientes espacios de nombres:

- `aws:elasticbeanstalk:container:tomcat:jvmoptions`: además de las opciones mencionadas anteriormente en esta página para este espacio de nombres, las versiones más antiguas de la plataforma de la AMI de Amazon Linux también admiten:
 - `XX:MaxPermSize` – Maximum JVM permanent generation size (Máximo de generación permanente de JVM)
- `aws:elasticbeanstalk:environment:proxy`: además de elegir el servidor proxy, configure también la compresión de respuesta.

El siguiente archivo de configuración de ejemplo muestra el uso de las opciones de configuración de los espacios de nombre del proxy.

Example `.ebextensions/tomcat-settings.config`

```
option_settings:  
  aws:elasticbeanstalk:environment:proxy:  
    GzipCompression: 'true'  
    ProxyServer: nginx
```

Incluir los archivos de configuración de Elastic Beanstalk: AMI de Amazon Linux (AL1)

Para implementar los archivos de configuración de `.ebextensions`, incluirlos en código fuente de la aplicación. Para una sola aplicación, añada `.ebextensions` a un archivo WAR comprimido ejecutando el siguiente comando:

Example

```
zip -ur your_application.war .ebextensions
```

Si se trata de una aplicación que necesite varios archivos WAR, consulte [Agrupación de varios archivos WAR en entornos Tomcat](#) para obtener información más detallada.

Agrupación de varios archivos WAR en entornos Tomcat

Si la aplicación web contiene varios componentes de aplicaciones web, puede simplificar las implementaciones y reducir los costos operativos ejecutando los componentes en un único entorno, en lugar de ejecutar un entorno diferente para cada componente. Esta estrategia resulta eficaz con aplicaciones ligeras que no requieren una gran cantidad de recursos y con entornos de desarrollo y pruebas.

Para implementar varias aplicaciones web en el entorno, combine los archivos WAR (archivo de aplicación web) de cada componente en un único [paquete de código fuente](#).

Para crear un paquete de código fuente de la aplicación que contenga varios archivos WAR, organice estos archivos utilizando la siguiente estructura.

```
MyApplication.zip
### .ebextensions
### .platform
### foo.war
### bar.war
### ROOT.war
```

Si implementa un paquete de código fuente que contiene varios archivos WAR en un entorno de AWS Elastic Beanstalk, el acceso a cada aplicación estará disponible en diferentes rutas fuera del nombre de dominio raíz. El ejemplo anterior contenía tres aplicaciones: `foo`, `bar` y `ROOT`. `ROOT.war` es un nombre de archivo especial que le indica a Elastic Beanstalk que ejecute esa aplicación en el dominio raíz, por lo que las tres aplicaciones estarán disponibles en `http://MyApplication.elasticbeanstalk.com/foo`, `http://MyApplication.elasticbeanstalk.com/bar` y `http://MyApplication.elasticbeanstalk.com`.

El paquete fuente puede incluir archivos WAR, una carpeta `.ebextensions` opcional y una carpeta `.platform` opcional. Para obtener información detallada sobre estas carpetas de configuración opcionales, consulte [the section called “Ampliación de plataformas Linux”](#).

Para lanzar un entorno (consola)

1. Abra la consola de Elastic Beanstalk con este enlace preconfigurado:
console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=tutorials&environmentType=LoadBalanced
2. En Platform (Plataforma), seleccione la plataforma y la ramificación de plataforma que coincidan con el idioma utilizado por su aplicación, o la plataforma Docker para aplicaciones basadas en contenedores.
3. En Application code (Código de aplicación), seleccione Upload your code (Cargar su propio código).
4. Elija Local file (Archivo local), seleccione Choose file (Seleccionar archivo) y abra el paquete de código fuente.
5. Elija Review and launch (Revisar y lanzar).
6. Revise la configuración disponible y, a continuación, elija Create app (Crear aplicación).

Para obtener más información sobre la creación de paquetes de código fuente, consulte [Cree el paquete de código fuente de la aplicación..](#)

Estructuración de la carpeta de proyectos

Para que todo funcione al realizar la implementación en un servidor Tomcat, los archivos de aplicación web (archivos WAR) compilados con Java Platform Enterprise Edition (Java EE) deben estar estructurados de acuerdo con ciertas [directrices](#). El directorio de proyectos no tiene que ajustarse a los mismos estándares, pero es recomendable estructurarlo de la misma manera para simplificar la compilación y el empaquetado. Estructurar la carpeta de proyectos igual que el contenido del archivo WAR también le ayuda a comprender cómo se relacionan los archivos y cómo se comportan en un servidor web.

En la siguiente jerarquía recomendada, el código fuente de la aplicación web se coloca en un directorio `src` para aislarlo del script de compilación y del archivo WAR que genera.

```
~/workspace/my-app/  
|-- build.sh           - Build script that compiles classes and creates a WAR  
|-- README.MD         - Readme file with information about your project, notes  
|-- ROOT.war          - Source bundle artifact created by build.sh  
`-- src                - Source code folder  
    |-- WEB-INF        - Folder for private supporting files  
    |   |-- classes    - Compiled classes
```



```
| |-- lib          - JAR libraries
| |-- tags        - Tag files
| |-- tlds        - Tag Library Descriptor files
| `-- web.xml     - Deployment Descriptor
|-- com          - Uncompiled classes
|-- css          - Style sheets
|-- images       - Image files
|-- js           - JavaScript files
`-- default.jsp  - JSP (JavaServer Pages) webpage
```

El contenido del archivo `src` coincide con lo que se va a empaquetar e implementar en el servidor, a excepción de la carpeta `com`. La carpeta `com` contiene las clases no compiladas (archivos `.java`). Deben compilarse y guardarse en el directorio `WEB-INF/classes` para que estén accesibles desde el código de la aplicación.

El directorio `WEB-INF` contiene código y configuraciones que no se proporcionan públicamente en el servidor web. Las demás carpetas de la raíz del directorio de código fuente (`css`, `images` y `js`) están disponibles públicamente en la ruta correspondiente del servidor web.

El siguiente ejemplo es idéntico al directorio de proyectos anterior, salvo que contiene más archivos y subdirectorios. En este ejemplo, el proyecto contiene etiquetas sencillas, un modelo y clases de compatibilidad, así como un archivo de Java Server Pages (JSP) para un recurso `record`. También contiene una hoja de estilos y JavaScript for [Bootstrap](#), un archivo JSP y una página para errores 404.

`WEB-INF/lib` contiene un archivo de Java Archive (JAR) con el controlador de Java Database Connectivity (JDBC) para PostgreSQL. `WEB-INF/classes` está vacío, ya que no se han compilado todavía los archivos de clase.

```
~/workspace/my-app/
|-- build.sh
|-- README.MD
|-- ROOT.war
`-- src
    |-- WEB-INF
    | |-- classes
    | |-- lib
    | | `-- postgresql-9.4-1201.jdbc4.jar
    | |-- tags
    | | `-- header.tag
    | |-- tlds
    | | `-- records.tld
```

```

|   `-- web.xml
|-- com
|   `-- myapp
|       |-- model
|       |   `-- Record.java
|       `-- web
|           `-- ListRecords.java
|-- css
|   |-- bootstrap.min.css
|   `-- myapp.css
|-- images
|   `-- myapp.png
|-- js
|   `-- bootstrap.min.js
|-- 404.jsp
|-- default.jsp
`-- records.jsp

```

Compilación de un archivo WAR con un script de shell

`build.sh` es un script de shell muy sencillo que compila las clases de Java, construye un archivo WAR y lo copia en el directorio `webapps` de Tomcat para las pruebas locales.

```

cd src
javac -d WEB-INF/classes com/myapp/model/Record.java
javac -classpath WEB-INF/lib/*:WEB-INF/classes -d WEB-INF/classes com/myapp/model/
Record.java
javac -classpath WEB-INF/lib/*:WEB-INF/classes -d WEB-INF/classes com/myapp/web/
ListRecords.java

jar -cvf ROOT.war *.jsp images css js WEB-INF
cp ROOT.war /Library/Tomcat/webapps
mv ROOT.war ../

```

Dentro del archivo WAR, encontrará la misma estructura que en el directorio `src` del ejemplo anterior, a excepción de la carpeta `src/com`. El comando `jar` crea automáticamente el archivo `META-INF/MANIFEST.MF`.

```

~/workspace/my-app/ROOT.war
|-- META-INF
|   `-- MANIFEST.MF
|-- WEB-INF

```

```
| |-- classes
| |   |-- com
| |     |-- myapp
| |       |-- model
| |         |-- Records.class
| |           |-- web
| |             |-- ListRecords.class
|-- lib
| |   |-- postgresql-9.4-1201.jdbc4.jar
|-- tags
| |   |-- header.tag
|-- tlds
| |   |-- records.tld
|-- web.xml
|-- css
| |   |-- bootstrap.min.css
| |   |-- myapp.css
|-- images
| |   |-- myapp.png
|-- js
| |   |-- bootstrap.min.js
|-- 404.jsp
|-- default.jsp
|-- records.jsp
```

Uso `.gitignore`

Para evitar que se validen los archivos de clases compiladas y los archivos WAR en su repositorio de Git o que aparezcan mensajes cuando ejecute comandos Git, agregue los tipos de archivo pertinentes en un archivo denominado `.gitignore` en la carpeta de su proyecto.

```
~/workspace/myapp/.gitignore
```

```
*.zip
*.class
```

Configuración del servidor proxy de su entorno Tomcat

La plataforma Tomcat utiliza [nginx](#) (el valor predeterminado) o [Apache HTTP Server](#) como proxy inverso para transmitir las solicitudes del puerto 80 de la instancia al contenedor web Tomcat que está a la escucha en el puerto 8080. Elastic Beanstalk proporciona una configuración de proxy predeterminada que puede ampliar o anular por completo con su propia configuración.

Configuración del servidor proxy en la versión de su plataforma

Todas las plataformas AL2023/AL2 son compatibles con una característica de configuración de proxy uniforme. Para obtener más información sobre la configuración del servidor proxy en las versiones de su plataforma que ejecutan AL2023/AL2, expanda la sección Configuración del proxy inverso de [the section called “Ampliación de plataformas Linux”](#).

Configuración del proxy en la plataforma Tomcat de la AMI de Amazon Linux (anterior a Amazon Linux 2)

Si su entorno Tomcat de Elastic Beanstalk utiliza una versión de la plataforma de la AMI de Amazon Linux (anterior a Amazon Linux 2), lea la información adicional de esta sección.

Notas

- La información de este tema solo se aplica a ramificaciones de plataforma basadas en la AMI de Amazon Linux (AL1). Las ramificaciones de la plataforma AL2023/AL2 son incompatibles con las versiones anteriores de la plataforma de la AMI de Amazon Linux (AL1) y requieren ajustes de configuración diferentes.
- El [18 de julio de 2022](#), Elastic Beanstalk estableció el estado de todas las ramificaciones de la plataforma basadas en la AMI de Amazon Linux (AL1) como retirado. Para obtener más información sobre la migración a una ramificación de la plataforma Amazon Linux 2023 actual y totalmente compatible, consulte [Migración de su aplicación de Linux de Elastic Beanstalk a Amazon Linux 2023 o Amazon Linux 2](#).

Elegir un servidor proxy para su entorno Tomcat: AMI de Amazon Linux (AL1)

Las versiones de la plataforma Tomcat basadas en la AMI de Amazon Linux (anterior a Amazon Linux 2) utilizan [Apache 2.4](#) para el proxy de forma predeterminada. Puede elegir usar [Apache 2.2](#) o [nginx](#) incluyendo un [archivo de configuración](#) en el código fuente. El siguiente ejemplo configura Elastic Beanstalk para usar nginx.

Example .ebextensions/nginx-proxy.config

```
option_settings:  
  aws:elasticbeanstalk:environment:proxy:  
    ProxyServer: nginx
```

Migración de Apache 2.2 a Apache 2.4: AMI de Amazon Linux (AL1)

Si la aplicación se desarrolló para [Apache 2.2](#), lea esta sección para obtener información sobre la migración a [Apache 2.4](#).

A partir de las configuraciones de la versión 3.0.0 de la plataforma Tomcat, que se publicaron con la [actualización de Java con la plataforma Tomcat el 24 de mayo de 2018](#), Apache 2.4 es el proxy predeterminado de la plataforma Tomcat. Los archivos `.conf` de Apache 2.4 son en su mayoría, pero no todos, compatibles con los de Apache 2.2. Elastic Beanstalk incluye archivos `.conf` predeterminados que funcionan correctamente con cada versión de Apache. Si la aplicación no personaliza la configuración de Apache, tal y como se explica en [Ampliación y anulación de la configuración predeterminada de Apache: AMI de Amazon Linux \(AL1\)](#), debería migrar a Apache 2.4 sin problemas.

Si la aplicación amplía o anula la configuración de Apache, es posible que tenga que realizar algunos cambios para migrar a Apache 2.4. Para obtener más información, consulte [Upgrading to 2.4 from 2.2](#) en el sitio web de The Apache Software Foundation. Como medida temporal, hasta que migre correctamente a Apache 2.4, puede utilizar Apache 2.2 con su aplicación; para ello, incluya el segmento [archivo de configuración](#) en el código fuente.

Example `.ebextensions/apache-legacy-proxy.config`

```
option_settings:
  aws:elasticbeanstalk:environment:proxy:
    ProxyServer: apache/2.2
```

Para obtener una solución rápida, también puede seleccionar el servidor proxy en la consola de Elastic Beanstalk.

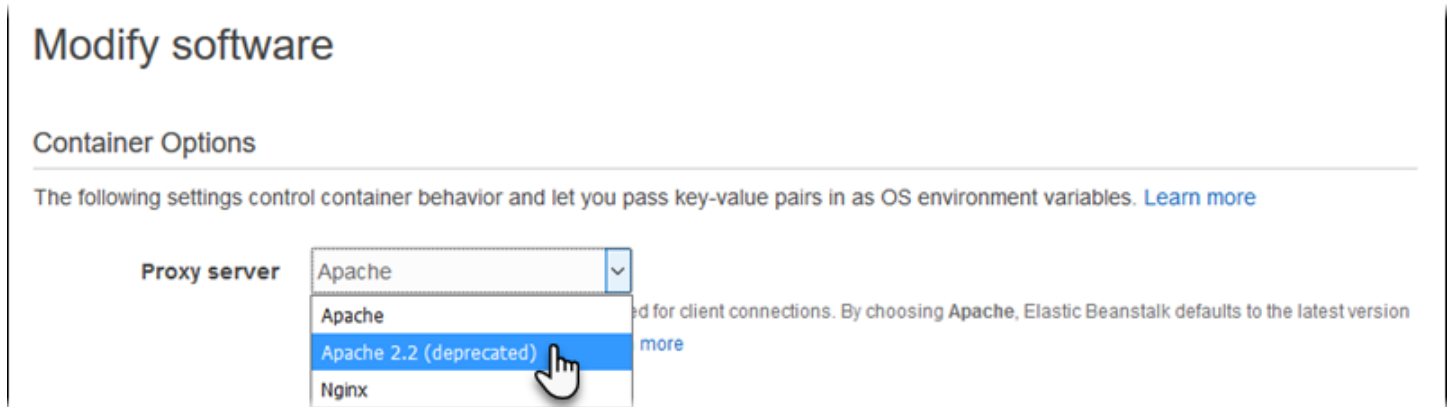
Para seleccionar el proxy en el entorno Tomcat en la consola de Elastic Beanstalk

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Actualizaciones, supervisión y registro, seleccione Editar.
5. En Proxy server (Servidor proxy), elija Apache 2.2 (deprecated).
6. Para guardar los cambios, elija Aplicar en la parte inferior de la página.



Ampliación y anulación de la configuración predeterminada de Apache: AMI de Amazon Linux (AL1)

Puede ampliar la configuración de Apache predeterminada de Elastic Beanstalk con archivos de configuración adicionales. Si lo prefiere, puede anular por completo la configuración de Apache predeterminada de Elastic Beanstalk.

Note

- Todas las plataformas de Amazon Linux 2 admiten una función de configuración de proxy uniforme. Para obtener más información sobre la configuración del servidor proxy en las versiones de la plataforma Tomcat que ejecutan Amazon Linux 2, expanda la sección Configuración de proxy inverso en [the section called “Ampliación de plataformas Linux”](#).
- Si va a migrar su aplicación de Elastic Beanstalk a una plataforma de Amazon Linux 2, asegúrese de leer también la información en [the section called “Migración a AL2023/AL2”](#).

Para ampliar la configuración de Apache predeterminada de Elastic Beanstalk, añada los archivos de configuración `.conf` a una carpeta con el nombre `.ebextensions/httpd/conf.d` en el paquete de código fuente de su aplicación. La configuración de Apache de Elastic Beanstalk incluye archivos `.conf` en esta carpeta automáticamente.

```
~/workspace/my-app/
```

```
|-- .ebextensions
|  -- httpd
|    -- conf.d
|      -- myconf.conf
|      -- ssl.conf
|-- index.jsp
```

Por ejemplo, la siguiente configuración de Apache 2.4 agrega un agente de escucha en el puerto 5000.

Example `.ebextensions/httpd/conf.d/port5000.conf`

```
listen 5000
<VirtualHost *:5000>
  <Proxy *>
    Require all granted
  </Proxy>
  ProxyPass / http://localhost:8080/ retry=0
  ProxyPassReverse / http://localhost:8080/
  ProxyPreserveHost on

  ErrorLog /var/log/httpd/elasticbeanstalk-error_log
</VirtualHost>
```


Para anular por completo la configuración de Apache predeterminada de Elastic Beanstalk, incluya una configuración en el paquete de código fuente en `.ebextensions/httpd/conf/httpd.conf`.

```
~/workspace/my-app/
|-- .ebextensions
|  `-- httpd
|    `-- conf
|      `-- httpd.conf
`-- index.jsp
```

Si anula la configuración de Apache de Elastic Beanstalk, agregue las siguientes líneas a su `httpd.conf` para extraer las configuraciones de Elastic Beanstalk de [Informes y monitorización de estado mejorados](#), la compresión de respuesta y los archivos estáticos.

```
IncludeOptional conf.d/*.conf
IncludeOptional conf.d/elasticbeanstalk/*.conf
```

Si su entorno utiliza Apache 2.2 como proxy, reemplace las directivas `IncludeOptional` por `Include`. Para obtener más información acerca del comportamiento de estas dos directivas en las dos versiones de Apache, consulte [Include en Apache 2.4](#), [IncludeOptional en Apache 2.4](#) e [Include en Apache 2.2](#).

 Note

Para anular el agente de escucha predeterminado en el puerto 80, incluya un archivo denominado `00_application.conf` en `.ebextensions/httpd/conf.d/elasticbeanstalk/` para sobrescribir la configuración de Elastic Beanstalk.

Para consultar un ejemplo funcional, eche un vistazo al archivo de configuración predeterminado de Elastic Beanstalk en `/etc/httpd/conf/httpd.conf` en una instancia de su entorno. Todos los archivos de la carpeta `.ebextensions/httpd` del paquete de código fuente se copian en `/etc/httpd` durante las implementaciones.

Ampliación de la configuración de nginx predeterminada: AMI de Amazon Linux (AL1)

Para ampliar la configuración predeterminada de nginx de Elastic Beanstalk, añada los archivos de configuración `.conf` a una carpeta con el nombre `.ebextensions/nginx/conf.d/` en el paquete de código fuente de su aplicación. La configuración de nginx de Elastic Beanstalk incluye archivos `.conf` en esta carpeta automáticamente.

```
~/workspace/my-app/  
|-- .ebextensions  
|   |-- nginx  
|       |-- conf.d  
|           |-- elasticbeanstalk  
|               |-- my-server-conf.conf  
|               |-- my-http-conf.conf  
|-- index.jsp
```

Los archivos con la extensión `.conf` en la carpeta `conf.d` se incluyen en el bloque `http` de la configuración predeterminada. Los archivos de la carpeta `conf.d/elasticbeanstalk` se incluyen en el bloque `server` dentro del bloque `http`.

Para anular completamente la configuración predeterminada de nginx de Elastic Beanstalk, incluya una configuración en el paquete de código fuente de `.ebextensions/nginx/nginx.conf`.


```
~/workspace/my-app/  
|-- .ebextensions  
|   |-- nginx  
|       |-- nginx.conf  
|-- index.jsp
```

Notas

- Si anula la configuración de nginx de Elastic Beanstalk, añada la siguiente línea al bloque `server` de su configuración para extraer las configuraciones de Elastic Beanstalk del agente de escucha del puerto 80, la compresión de respuesta y los archivos estáticos.

```
include conf.d/elasticbeanstalk/*.conf;
```

- Para anular el agente de escucha predeterminado en el puerto 80, incluya un archivo denominado `00_application.conf` en `.ebextensions/nginx/conf.d/elasticbeanstalk/` para sobrescribir la configuración de Elastic Beanstalk.
- Incluya también la línea siguiente en el bloque `http` de su configuración para extraer las configuraciones de Elastic Beanstalk para [Informes y monitorización de estado mejorados](#) y los registros.

```
include conf.d/*.conf;
```

Para consultar un ejemplo funcional, eche un vistazo al archivo de configuración predeterminado de Elastic Beanstalk en `/etc/nginx/nginx.conf` en una instancia de su entorno. Todos los archivos de la carpeta `.ebextensions/nginx` del paquete de código fuente se copian en `/etc/nginx` durante las implementaciones.

Uso de la plataforma Java SE de Elastic Beanstalk

La plataforma Java SE de AWS Elastic Beanstalk es un conjunto de [versiones de plataformas](#) para aplicaciones web Java que se pueden ejecutar por sí solas a partir de un archivo JAR compilado. Puede compilar su aplicación localmente o cargar el código fuente con un script de compilación para compilarlo en la instancia. Las versiones de la plataforma Java SE se agrupan en ramas de plataforma, cada una de las cuales corresponde a una versión principal de Java, por ejemplo Java 8 y Java 7.

Note

Elastic Beanstalk no analiza el archivo JAR de su aplicación. Mantenga los archivos que necesita Elastic Beanstalk fuera del archivo JAR. Por ejemplo, incluya el archivo `cron.yaml` de un [entorno de trabajo](#) en la raíz del paquete de código fuente de la aplicación junto al archivo JAR.

En la consola de Elastic Beanstalk hay opciones de configuración disponibles para [modificar la configuración de un entorno en ejecución](#). Para evitar perder la configuración del entorno cuando lo termina, puede usar las [configuraciones guardadas](#) para guardar la configuración y aplicarla posteriormente a otro entorno.

Para guardar la configuración en el código fuente, puede incluir [archivos de configuración](#). Los valores de configuración de los archivos de configuración se aplican cada vez que crea un entorno o que implementa la aplicación. También puede usar archivos de configuración para instalar paquetes, ejecutar scripts y llevar a cabo otras operaciones de personalización de instancias durante las implementaciones.

La plataforma Java SE de Elastic Beanstalk incluye un servidor [nginx](#) que actúa como proxy inverso, que sirve contenido estático almacenado en caché y pasa solicitudes a su aplicación. La plataforma proporciona también opciones para configurar el servidor proxy y atender recursos estáticos de una carpeta del código fuente con el fin de reducir la carga en su aplicación. Para escenarios avanzados, puede [incluir sus propios archivos .conf](#) en el paquete de código fuente para ampliar la configuración del proxy de Elastic Beanstalk o sobrescribirla completamente.

Si solo proporciona un único archivo JAR para el código fuente de la aplicación (por sí mismo, no dentro de un paquete de código fuente), Elastic Beanstalk cambia el nombre de su archivo JAR a `application.jar` y, a continuación, lo ejecuta usando `java -jar application.jar`. Para configurar los procesos que se ejecutan en las instancias de servidor en su entorno, incluya un [Procfile](#) opcional en su paquete de código fuente. El `Procfile` es necesario si tiene más de un JAR en la raíz del paquete de código fuente, o bien si desea personalizar el comando Java para establecer opciones de JVM.

Le recomendamos que siempre proporcione un paquete `Procfile` en el origen junto con su aplicación. De esta forma, puede controlar con precisión qué procesos ejecuta Elastic Beanstalk para su aplicación y qué argumentos reciben estos procesos.

Para compilar clases Java y ejecutar otros comandos de compilación en las instancias EC2 de su entorno en el momento de la implementación, incluya un [Buildfile](#) en el paquete de código fuente de su aplicación. Un `Buildfile` le permite implementar el código fuente tal y como está y compilarlo en el servidor en lugar de compilar los archivos JAR localmente. La plataforma Java SE incluye herramientas de compilación comunes que le permiten compilar en el servidor.

Para obtener más información sobre las diversas formas en las que puede ampliar una plataforma Elastic Beanstalk basada en Linux, consulte [the section called “Ampliación de plataformas Linux”](#).

Configuración del entorno de Java SE

La configuración de la plataforma Java SE le permite ajustar el comportamiento de sus instancias de Amazon EC2. Puede editar la configuración de instancia de Amazon EC2 del entorno Elastic Beanstalk mediante la consola de Elastic Beanstalk.

Utilice la consola de Elastic Beanstalk para habilitar la rotación de registros en Amazon S3 y configurar variables que la aplicación pueda leer desde el entorno.

Para configurar el entorno Java SE en la consola de Elastic Beanstalk

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Actualizaciones, supervisión y registro, seleccione Editar.

Log Options (Opciones de registro)

La sección Log Options tiene dos valores:

- Instance profile (Perfil de instancia): especifica el perfil de instancia que tiene permiso para obtener acceso al bucket de Amazon S3 asociado con la aplicación.

- **Enable log file rotation to Amazon S3 (Habilitar la rotación de archivos de registro para Amazon S3):** especifica si los archivos de registro de las instancias de Amazon EC2 de la aplicación se copian en el bucket de Amazon S3 asociado a la aplicación.

Archivos estáticos

Para mejorar el rendimiento, puede utilizar la sección Static files (Archivos estáticos) para configurar el servidor proxy para entregar archivos estáticos de servidor (por ejemplo, HTML o imágenes) desde un conjunto de directorios dentro de su aplicación web. Para cada directorio, se establece la ruta virtual para la asignación de directorios. Cuando el servidor proxy recibe una solicitud de un archivo en la ruta especificada, proporciona directamente el archivo en lugar de direccionar la solicitud a la aplicación.

Para obtener más información sobre la configuración de archivos estáticos mediante archivos de configuración o la consola de Elastic Beanstalk, consulte [the section called “Archivos estáticos”](#).

Propiedades del entorno

La sección Environment Properties (Propiedades de entorno) le permite especificar opciones de configuración del entorno en las instancias de Amazon EC2 que ejecutan la aplicación. Las propiedades del entorno se pasan como pares de clave-valor a la aplicación.

Dentro del entorno de Java SE que se ejecuta en Elastic Beanstalk, es posible acceder a las variables de entorno mediante `System.getenv()`. Por ejemplo, podría leer una propiedad denominada `API_ENDPOINT` en una variable con el siguiente código:

```
String endpoint = System.getenv("API_ENDPOINT");
```

Para obtener más información, consulte [Propiedades del entorno y otras opciones del software](#).

Espacios de nombres de configuración de Java SE

Puede usar un [archivo de configuración](#) para definir opciones de configuración y realizar otras tareas de configuración en las instancias durante las implementaciones. Las opciones de configuración se pueden definir a través del servicio de Elastic Beanstalk o la plataforma que utilice y están organizadas por espacios de nombres.

La plataforma Java SE no define ningún espacio de nombres específico de la plataforma. Puede configurar el proxy para que sirva archivos estáticos mediante el espacio de nombres

`aws:elasticbeanstalk:environment:proxy:staticfiles`. Para obtener información detallada y un ejemplo, consulte [the section called “Archivos estáticos”](#).

Elastic Beanstalk cuenta con numerosas opciones de configuración para personalizar el entorno. Además de los archivos de configuración, también puede definir opciones en la consola, configuraciones guardadas, la CLI de EB o la AWS CLI. Para obtener más información, consulte [Opciones de configuración](#).

Plataforma Java SE de la AMI de Amazon Linux (anterior a Amazon Linux 2)

Si su entorno Java SE de Elastic Beanstalk utiliza una versión de la plataforma de la AMI de Amazon Linux (anterior a Amazon Linux 2), lea la información adicional de esta sección.

Notas

- La información de este tema solo se aplica a ramificaciones de plataforma basadas en la AMI de Amazon Linux (AL1). Las ramificaciones de la plataforma AL2023/AL2 son incompatibles con las versiones anteriores de la plataforma de la AMI de Amazon Linux (AL1) y requieren ajustes de configuración diferentes.
- El [18 de julio de 2022](#), Elastic Beanstalk estableció el estado de todas las ramificaciones de la plataforma basadas en la AMI de Amazon Linux (AL1) como retirado. Para obtener más información sobre la migración a una ramificación de la plataforma Amazon Linux 2023 actual y totalmente compatible, consulte [Migración de su aplicación de Linux de Elastic Beanstalk a Amazon Linux 2023 o Amazon Linux 2](#).

Espacios de nombres de configuración de Java SE: AMI de Amazon Linux (AL1)

Puede usar un [archivo de configuración](#) para definir opciones de configuración y realizar otras tareas de configuración en las instancias durante las implementaciones. Las opciones de configuración se pueden definir a través del servicio de Elastic Beanstalk o la plataforma que utilice y están organizadas por espacios de nombres.

La plataforma Java SE admite un espacio de nombres de configuración específico de la plataforma, además de los [espacios de nombres compatibles con todas las plataformas](#). El espacio de nombres `aws:elasticbeanstalk:container:java:staticfiles` le permite definir opciones que asignen las rutas de su aplicación web a carpetas del paquete de código fuente de la aplicación que incluyen contenido estático.

Por ejemplo, este fragmento de [option_settings](#) define dos opciones en el espacio de nombres de archivos estáticos. La primera asigna la ruta `/public` a una carpeta con el nombre `public` y la segunda asigna la ruta `/images` a una carpeta con el nombre `img`:

```
option_settings:
  aws:elasticbeanstalk:container:java:staticfiles:
    /html: statichtml
    /images: staticimages
```

Las carpetas que se asignan a través de este espacio de nombres deben ser carpetas reales situadas en la raíz del paquete de código fuente. No puede asignar una ruta a una carpeta en un archivo JAR.

Elastic Beanstalk cuenta con numerosas opciones de configuración para personalizar el entorno. Además de los archivos de configuración, también puede definir opciones en la consola, configuraciones guardadas, la CLI de EB o la AWS CLI. Para obtener más información, consulte [Opciones de configuración](#).

Compilación de archivos JAR en el servidor con un archivo Buildfile

Puede compilar los archivos de clases y los archivos JAR de su aplicación en las instancias EC2 de su entorno invocando un comando de compilación desde un archivo `Buildfile` del paquete de código fuente.

Los comandos de un `Buildfile` solo se ejecutan una vez y deben terminar cuando se completan, mientras que se espera que los comandos de un [Procfile](#) se ejecuten durante el ciclo de vida de la aplicación y se reinicien si terminan. Para ejecutar los archivos JAR de su aplicación, utilice un `Procfile`.

Para obtener más información sobre la ubicación y la sintaxis de un `Buildfile`, expanda la sección `Buildfile and Procfile (Buildfile y Procfile)` en [the section called “Ampliación de plataformas Linux”](#).

El siguiente ejemplo de `Buildfile` ejecuta Apache Maven para compilar una aplicación web a partir de código fuente. Consulte los [ejemplos de aplicaciones web Java](#) para ver una aplicación de ejemplo que utilice esta característica.

Example Buildfile

```
build: mvn assembly:assembly -DdescriptorId=jar-with-dependencies
```

La plataforma Java SE incluye las siguientes herramientas de compilación, que puede invocar desde un script de compilación:

- `javac`: compilador de Java
- `ant`: Apache Ant
- `mvn`: Apache Maven
- `gradle`: Gradle

Configuración del proceso de la aplicación con un Procfile

Si tiene más de un archivo JAR en la raíz del paquete de código fuente de su aplicación, debe incluir un archivo `Procfile` que indique a Elastic Beanstalk qué archivos JAR debe ejecutar. También puede incluir un archivo `Procfile` para una sola aplicación JAR para configurar la máquina virtual Java (JVM) que ejecuta su aplicación.

Le recomendamos que siempre proporcione un paquete `Procfile` en el origen junto con su aplicación. De esta forma, puede controlar con precisión qué procesos ejecuta Elastic Beanstalk para su aplicación y qué argumentos reciben estos procesos.

Para obtener más información sobre cómo escribir y utilizar un `Procfile`, expanda la sección `Buildfile` y `Procfile` en [the section called “Ampliación de plataformas Linux”](#).

Example Procfile

```
web: java -Xms256m -jar server.jar
cache: java -jar mycache.jar
web_foo: java -jar other.jar
```

El comando que ejecuta el archivo JAR principal de su aplicación debe llamarse `web` y debe ser el primer comando incluido en el archivo `Procfile`. El servidor `nginx` reenvía todas las solicitudes HTTP que recibe del balanceador de carga de su entorno a esta aplicación.

Elastic Beanstalk asume que todas las entradas del archivo `Procfile` deben ejecutarse en todo momento y reinicia automáticamente todas las aplicaciones definidas en el archivo `Procfile` que termina. Para ejecutar comandos que terminan y no deben reiniciarse, utilice un [Buildfile](#).

Uso de un archivo Procfile en la AMI de Amazon Linux (anterior a Amazon Linux 2)

Si su entorno Java SE de Elastic Beanstalk utiliza una versión de la plataforma de la AMI de Amazon Linux (anterior a Amazon Linux 2), lea la información adicional de esta sección.

Notas

- La información de este tema solo se aplica a ramificaciones de plataforma basadas en la AMI de Amazon Linux (AL1). Las ramificaciones de la plataforma AL2023/AL2 son incompatibles con las versiones anteriores de la plataforma de la AMI de Amazon Linux (AL1) y requieren ajustes de configuración diferentes.
- El [18 de julio de 2022](#), Elastic Beanstalk estableció el estado de todas las ramificaciones de la plataforma basadas en la AMI de Amazon Linux (AL1) como retirado. Para obtener más información sobre la migración a una ramificación de la plataforma Amazon Linux 2023 actual y totalmente compatible, consulte [Migración de su aplicación de Linux de Elastic Beanstalk a Amazon Linux 2023 o Amazon Linux 2](#).

Transferencia de puertos: AMI de Amazon Linux (AL1)

De forma predeterminada, Elastic Beanstalk configura el proxy nginx para reenviar las solicitudes a la aplicación en el puerto 5000. Puede invalidar el puerto predeterminado estableciendo la [propiedad del entorno](#) PORT en el puerto en el que desea que la aplicación principal esté a la escucha.

Si utiliza un Procfile para ejecutar varias aplicaciones, Elastic Beanstalk en las versiones de la plataforma de la AMI de Amazon Linux espera que cada aplicación adicional esté a la escucha en un puerto 100 mayor que el anterior. Elastic Beanstalk establece la variable PORT accesible desde cada aplicación en el puerto en el que espera que se ejecute la aplicación. Puede tener acceso a esta variable en el código de su aplicación llamando a `System.getenv("PORT")`.

En el Procfile de ejemplo anterior, la aplicación web está a la escucha en el puerto 5000, cache en el puerto 5100 y web_foo en el puerto 5200. web configura su puerto de escucha leyendo la variable PORT y añade 100 a dicho número para determinar en qué puerto cache está a la escucha para poder enviarle solicitudes.

Configuración del proxy inverso

Elastic Beanstalk utiliza [nginx](#) como proxy inverso para asignar su aplicación al balanceador de carga de Elastic Load Balancing en el puerto 80. Elastic Beanstalk proporciona una configuración de nginx predeterminada que puede ampliar o anular por completo con su propia configuración.

De forma predeterminada, Elastic Beanstalk configura el proxy nginx para reenviar las solicitudes a la aplicación en el puerto 5000. Puede invalidar el puerto predeterminado estableciendo la [propiedad del entorno](#) `PORT` en el puerto en el que desea que la aplicación principal esté a la escucha.

Note

El puerto en que su aplicación está a la escucha no afecta al puerto en el que el servidor nginx está a la escucha para recibir las solicitudes del balanceador de carga.

Configuración del servidor proxy en la versión de su plataforma

Todas las plataformas AL2023/AL2 son compatibles con una característica de configuración de proxy uniforme. Para obtener más información sobre la configuración del servidor proxy en las versiones de su plataforma que ejecutan AL2023/AL2, expanda la sección Configuración del proxy inverso de [the section called “Ampliación de plataformas Linux”](#).

Configuración del proxy en la AMI de Amazon Linux (anterior a Amazon Linux 2)

Si su entorno Java SE de Elastic Beanstalk utiliza una versión de la plataforma de la AMI de Amazon Linux (anterior a Amazon Linux 2), lea la información adicional de esta sección.

Notas

- La información de este tema solo se aplica a ramificaciones de plataforma basadas en la AMI de Amazon Linux (AL1). Las ramificaciones de la plataforma AL2023/AL2 son incompatibles con las versiones anteriores de la plataforma de la AMI de Amazon Linux (AL1) y requieren ajustes de configuración diferentes.
- El [18 de julio de 2022](#), Elastic Beanstalk estableció el estado de todas las ramificaciones de la plataforma basadas en la AMI de Amazon Linux (AL1) como retirado. Para obtener más información sobre la migración a una ramificación de la plataforma Amazon Linux

2023 actual y totalmente compatible, consulte [Migración de su aplicación de Linux de Elastic Beanstalk a Amazon Linux 2023 o Amazon Linux 2](#).

Ampliación y anulación de la configuración predeterminada del proxy: AMI de Amazon Linux (AL1)

Para ampliar la configuración predeterminada de nginx de Elastic Beanstalk, añada los archivos de configuración `.conf` a una carpeta con el nombre `.ebextensions/nginx/conf.d/` en el paquete de código fuente de su aplicación. La configuración de nginx de Elastic Beanstalk incluye archivos `.conf` en esta carpeta automáticamente.

```
~/workspace/my-app/  
|-- .ebextensions  
|   |-- nginx  
|       |-- conf.d  
|           |-- myconf.conf  
|-- web.jar
```

Para anular por completo la configuración de nginx predeterminada de Elastic Beanstalk, incluya una configuración en el paquete de código fuente en `.ebextensions/nginx/nginx.conf`:

```
~/workspace/my-app/  
|-- .ebextensions  
|   |-- nginx  
|       |-- nginx.conf  
|-- web.jar
```

Si anula la configuración de nginx de Elastic Beanstalk, agregue la siguiente línea a su `nginx.conf` para extraer las configuraciones de Elastic Beanstalk de [Informes y monitorización de estado mejorados](#), los mapeos automáticos de la aplicación y los archivos estáticos.

```
include conf.d/elasticbeanstalk/*.conf;
```

La siguiente configuración de ejemplo de la [aplicación de ejemplo Scorekeep](#) sustituye la configuración predeterminada de Elastic Beanstalk para servir una aplicación web estática desde el subdirectorio `public` de `/var/app/current`, donde la plataforma Java SE copia el código fuente de la aplicación. La ubicación `/api` reenvía el tráfico a las rutas en `/api/` a la aplicación Sprint que escucha en el puerto 5000. La aplicación web sirve el resto del tráfico en la ruta raíz.

Example

```
user                nginx;
error_log           /var/log/nginx/error.log warn;
pid                /var/run/nginx.pid;
worker_processes   auto;
worker_rlimit_nofile 33282;

events {
    worker_connections 1024;
}

http {
    include          /etc/nginx/mime.types;
    default_type     application/octet-stream;

    log_format main '$remote_addr - $remote_user [$time_local] "$request" '
                   '$status $body_bytes_sent "$http_referer" '
                   '"$http_user_agent" "$http_x_forwarded_for"';

    include          conf.d/*.conf;

    map $http_upgrade $connection_upgrade {
        default      "upgrade";
    }

    server {
        listen        80 default_server;
        root /var/app/current/public;

        location / {
            }git pull

        location /api {
            proxy_pass          http://127.0.0.1:5000;
            proxy_http_version  1.1;

            proxy_set_header    Connection      $connection_upgrade;
            proxy_set_header     Upgrade        $http_upgrade;
            proxy_set_header     Host          $host;
            proxy_set_header     X-Real-IP     $remote_addr;
            proxy_set_header     X-Forwarded-For $proxy_add_x_forwarded_for;
        }
    }
}
```

```
access_log    /var/log/nginx/access.log main;

client_header_timeout 60;
client_body_timeout   60;
keepalive_timeout    60;
gzip                  off;
gzip_comp_level       4;

# Include the Elastic Beanstalk generated locations
include conf.d/elasticbeanstalk/01_static.conf;
include conf.d/elasticbeanstalk/healthd.conf;
}
}
```

Añadir una instancia de base de datos de Amazon RDS a un entorno de aplicaciones Java

Puede utilizar una instancia de base de datos de Amazon Relational Database Service (Amazon RDS) para almacenar los datos recopilados y modificados por la aplicación. La base de datos se puede adjuntar al entorno y administrarse con Elastic Beanstalk, o bien, crearse y administrarse de forma externa.

Si es la primera vez que utiliza Amazon RDS, añada una instancia de base de datos a un entorno de prueba con la consola de Elastic Beanstalk y verifique que su aplicación puede conectarse a ella.

Para agregar una instancia de base de datos al entorno

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Database (Base de datos), seleccione Edit (Editar).
5. Seleccione un motor de base de datos y escriba el nombre de usuario y la contraseña.

6. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

Se tarda alrededor de 10 minutos en añadir una instancia de base de datos. Cuando se haya completado la actualización del entorno, el nombre de host de la instancia de base de datos y otra información de conexión estarán disponibles para su aplicación mediante las siguientes propiedades de entorno

Nombre de la propiedad	Descripción	Valor de la propiedad
RDS_HOSTNAME	El nombre de host de la instancia de base de datos.	En la pestaña Connectivity & security (Conectividad y seguridad) de la consola de Amazon RDS: Endpoint (Punto de enlace).
RDS_PORT	El puerto en donde la instancia de base de datos acepta las conexiones. El valor predeterminado varía dependiendo del motor de base de datos.	En la pestaña Connectivity & security (Conectividad y seguridad) de la consola de Amazon RDS: Port (Puerto).
RDS_DB_NAME	El nombre de la base de dato, ebdb .	En la pestaña Configuration (Configuración) de la consola de Amazon RDS: DB Name (Nombre de la base de datos).
RDS_USERNAME	El nombre de usuario que ha configurado para la base de datos.	En la pestaña Configuration (Configuración) de la consola de Amazon RDS: Master username (Nombre de usuario maestro).
RDS_PASSWORD	La contraseña que ha configurado para la base de datos.	No está disponible como referencia en la consola de Amazon RDS.

Para obtener más información sobre cómo configurar una instancia de base de datos interna, consulte [Añadir una base de datos al entorno de Elastic Beanstalk](#). Para obtener instrucciones sobre cómo configurar una base de datos externa para usarla con Elastic Beanstalk, consulte [Uso de Elastic Beanstalk con Amazon RDS](#).

Para conectarse a la base de datos, añada el archivo JAR del controlador correspondiente a su aplicación, cargue la clase del controlador en el código y cree un objeto de conexión con las propiedades de entorno proporcionadas por Elastic Beanstalk.

Secciones

- [Descargar el controlador JDBC](#)
- [Conexión a una base de datos \(plataformas Java SE\)](#)
- [Conexión a una base de datos \(plataformas Tomcat\)](#)
- [Resolución de problemas de conexión a la base de datos](#)

Descargar el controlador JDBC

Necesitará el archivo JAR del controlador JDBC para el motor de base de datos que elija. Guarde el archivo JAR en su código fuente e inclúyalo en su classpath cuando compile la clase que crea las conexiones a la base de datos.

Puede encontrar el controlador más reciente para su motor de base de datos en las siguientes ubicaciones:

- MySQL: [MySQL Connector/J](#)
- Oracle SE-1: [controlador JDBC de Oracle](#)
- Postgres: [controlador JDBC de Postgres](#)
- SQL Server: [controlador JDBC de Microsoft](#)

Para utilizar el controlador JDBC, llame a `Class.forName()` para cargarlo antes de crear la conexión con `DriverManager.getConnection()` en su código.

JDBC utiliza una cadena de conexión con el siguiente formato:

```
jdbc:driver://hostname:port/dbName?user=userName&password=password
```

Puede recuperar el nombre de host, el puerto, el nombre de la base de datos, el nombre de usuario y la contraseña desde las variables de entorno que Elastic Beanstalk proporciona a la aplicación. El nombre del controlador es específico del tipo de base de datos y de la versión del controlador. A continuación se muestran algunos nombres de controlador de ejemplo:

- `mysql` para MySQL
- `postgresql` para PostgreSQL
- `oracle:thin` para Oracle Thin
- `oracle:oci` para Oracle OCI
- `oracle:oci8` para Oracle OCI 8
- `oracle:kprb` para Oracle KPRB
- `sqlserver` para SQL Server

Conexión a una base de datos (plataformas Java SE)

En un entorno de Java SE, use `System.getenv()` para leer las variables de conexión del entorno. El siguiente código de ejemplo muestra una clase que crea una conexión a una base de datos PostgreSQL.

```
private static Connection getRemoteConnection() {
    if (System.getenv("RDS_HOSTNAME") != null) {
        try {
            Class.forName("org.postgresql.Driver");
            String dbName = System.getenv("RDS_DB_NAME");
            String userName = System.getenv("RDS_USERNAME");
            String password = System.getenv("RDS_PASSWORD");
            String hostname = System.getenv("RDS_HOSTNAME");
            String port = System.getenv("RDS_PORT");
            String jdbcUrl = "jdbc:postgresql://" + hostname + ":" + port + "/" + dbName + "?
user=" + userName + "&password=" + password;
            logger.trace("Getting remote connection with connection string from environment
variables.");
            Connection con = DriverManager.getConnection(jdbcUrl);
            logger.info("Remote connection successful.");
            return con;
        }
        catch (ClassNotFoundException e) { logger.warn(e.toString());}
        catch (SQLException e) { logger.warn(e.toString());}
    }
}
```

```
    return null;
}
```

Conexión a una base de datos (plataformas Tomcat)

En un entorno Tomcat, las propiedades de entorno se proporcionan como propiedades del sistema accesibles con `System.getProperty()`.

El siguiente código de ejemplo muestra una clase que crea una conexión a una base de datos PostgreSQL.

```
private static Connection getRemoteConnection() {
    if (System.getProperty("RDS_HOSTNAME") != null) {
        try {
            Class.forName("org.postgresql.Driver");
            String dbName = System.getProperty("RDS_DB_NAME");
            String userName = System.getProperty("RDS_USERNAME");
            String password = System.getProperty("RDS_PASSWORD");
            String hostname = System.getProperty("RDS_HOSTNAME");
            String port = System.getProperty("RDS_PORT");
            String jdbcUrl = "jdbc:postgresql://" + hostname + ":" + port + "/" + dbName + "?
user=" + userName + "&password=" + password;
            logger.trace("Getting remote connection with connection string from environment
variables.");
            Connection con = DriverManager.getConnection(jdbcUrl);
            logger.info("Remote connection successful.");
            return con;
        }
        catch (ClassNotFoundException e) { logger.warn(e.toString());}
        catch (SQLException e) { logger.warn(e.toString());}
    }
    return null;
}
```

Si tiene problemas para obtener una conexión o ejecutar instrucciones SQL, pruebe a colocar el siguiente código en un archivo JSP. Este código establece una conexión con una instancia de base de datos, crea una tabla y escribe en ella.

```
<%@ page import="java.sql.*" %>
<%
    // Read RDS connection information from the environment
    String dbName = System.getProperty("RDS_DB_NAME");
```



```
String userName = System.getProperty("RDS_USERNAME");
String password = System.getProperty("RDS_PASSWORD");
String hostname = System.getProperty("RDS_HOSTNAME");
String port = System.getProperty("RDS_PORT");
String jdbcUrl = "jdbc:mysql://" + hostname + ":" +
    port + "/" + dbName + "?user=" + userName + "&password=" + password;

// Load the JDBC driver
try {
    System.out.println("Loading driver...");
    Class.forName("com.mysql.jdbc.Driver");
    System.out.println("Driver loaded!");
} catch (ClassNotFoundException e) {
    throw new RuntimeException("Cannot find the driver in the classpath!", e);
}

Connection conn = null;
Statement setupStatement = null;
Statement readStatement = null;
ResultSet resultSet = null;
String results = "";
int numresults = 0;
String statement = null;

try {
    // Create connection to RDS DB instance
    conn = DriverManager.getConnection(jdbcUrl);

    // Create a table and write two rows
    setupStatement = conn.createStatement();
    String createTable = "CREATE TABLE Beanstalk (Resource char(50));";
    String insertRow1 = "INSERT INTO Beanstalk (Resource) VALUES ('EC2 Instance');";
    String insertRow2 = "INSERT INTO Beanstalk (Resource) VALUES ('RDS Instance');";

    setupStatement.addBatch(createTable);
    setupStatement.addBatch(insertRow1);
    setupStatement.addBatch(insertRow2);
    setupStatement.executeBatch();
    setupStatement.close();

} catch (SQLException ex) {
    // Handle any errors
    System.out.println("SQLException: " + ex.getMessage());
    System.out.println("SQLState: " + ex.getSQLState());
}
```

```
        System.out.println("VendorError: " + ex.getErrorCode());
    } finally {
        System.out.println("Closing the connection.");
        if (conn != null) try { conn.close(); } catch (SQLException ignore) {}
    }

    try {
        conn = DriverManager.getConnection(jdbcUrl);

        readStatement = conn.createStatement();
        resultSet = readStatement.executeQuery("SELECT Resource FROM Beanstalk;");

        resultSet.first();
        results = resultSet.getString("Resource");
        resultSet.next();
        results += ", " + resultSet.getString("Resource");

        resultSet.close();
        readStatement.close();
        conn.close();

    } catch (SQLException ex) {
        // Handle any errors
        System.out.println("SQLException: " + ex.getMessage());
        System.out.println("SQLState: " + ex.getSQLState());
        System.out.println("VendorError: " + ex.getErrorCode());
    } finally {
        System.out.println("Closing the connection.");
        if (conn != null) try { conn.close(); } catch (SQLException ignore) {}
    }
    %>
```

Para mostrar los resultados, coloque el siguiente código en el cuerpo de la parte HTML del archivo JSP.

```
<p>Established connection to RDS. Read first two rows: <%= results %></p>
```

Resolución de problemas de conexión a la base de datos

Si surgen problemas al conectarse a una base de datos desde la aplicación, examine el archivo log del contenedor web y la base de datos.

Consulta de logs

Puede ver todos los registros de su entorno de Elastic Beanstalk desde Eclipse. Si no abrió la vista del Explorador de AWS, elija la flecha situada junto al ícono naranja de AWS en la barra de herramientas y, a continuación, elija Mostrar la vista del explorador de AWS. Expanda AWS Elastic Beanstalk y el nombre de su entorno y, a continuación, abra el menú contextual (haga clic con el botón derecho) del servidor. Elija Open in WTP Server Editor (Abrir en el editor del servidor de WTP).

Elija la pestaña Log (Registro) de la vista Server (Servidor) para ver los registros acumulados de su entorno. Para abrir los últimos registros, seleccione el botón Refresh (Actualizar) en la esquina superior derecha de la página.

Desplácese hacia abajo para localizar los logs de Tomcat en `/var/log/tomcat7/catalina.out`. Si ha cargado la página web desde nuestro ejemplo anterior varias veces, es posible que vea lo siguiente.

```
-----  
/var/log/tomcat7/catalina.out  
-----  
INFO: Server startup in 9285 ms  
Loading driver...  
Driver loaded!  
SQLException: Table 'Beanstalk' already exists  
SQLState: 42S01  
VendorError: 1050  
Closing the connection.  
Closing the connection.
```

Toda la información que la aplicación web envía a la salida estándar aparece en el log del contenedor web. En el ejemplo anterior, la aplicación intenta crear la tabla cada vez que se carga la página. Esto produce una excepción SQL en cada carga de página después de la primera.


Esto es aceptable porque se trata de un ejemplo. Pero en las aplicaciones reales mantenga las definiciones de base de datos en objetos del esquema, realice las transacciones desde las clases del modelo y coordine las solicitudes con servlets de controlador.

Conexión a una instancia de base de datos de RDS

Puede conectarse directamente a la instancia de base de datos de RDS en su entorno de Elastic Beanstalk usando la aplicación cliente de MySQL.


En primer lugar, abra el grupo de seguridad en su instancia de base de datos de RDS para permitir el tráfico desde su equipo.

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

 Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Database (Base de datos), seleccione Edit (Editar).
5. Junto a Endpoint (Punto de enlace), elija el enlace de la consola de Amazon RDS.
6. En la página de detalles de la instancia de RDS Dashboard (Panel RDS), bajo Security and Network (Seguridad y red), elija el grupo de seguridad que empieza por rds- situado junto a Security Groups (Grupos de seguridad).

 Note

La base de datos podría tener varias entradas etiquetadas como Security Groups (Grupos de seguridad). Utilice la primera, que comienza con awseb, solo si tiene una cuenta anterior que no dispone de una [Amazon Virtual Private Cloud](#) (Amazon VPC) predeterminada.

7. En Security group details (Detalles del grupo de seguridad), elija la pestaña Inbound (Entrada) y, a continuación, elija Edit (Editar).
8. Añada una regla para MySQL (puerto 3306) que permita el tráfico procedente de la dirección IP, especificada en formato CIDR.
9. Seleccione Save (Guardar). Los cambios surtirán efecto de inmediato.

Consulte de nuevo los datos de configuración de Elastic Beanstalk para el entorno y anote el punto de enlace. Utilizará el nombre de dominio para conectarse a la instancia de base de datos de RDS.

Instale el cliente de MySQL e inicie una conexión con la base de datos en el puerto 3306. En Windows, instale MySQL Workbench desde la página de inicio de MySQL y siga las instrucciones.

En Linux, instale el cliente de MySQL con el administrador de paquetes de su distribución. El siguiente ejemplo funciona en Ubuntu y otros derivados de Debian.

```
// Install MySQL client
$ sudo apt-get install mysql-client-5.5
...
// Connect to database
$ mysql -h aas839jo2vwhwb.cnubrrfwfka8.us-west-2.rds.amazonaws.com -u username -
ppassword ebdb
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 117
Server version: 5.5.40-log Source distribution
...
```

Una vez que se haya conectado, puede ejecutar comandos SQL para ver el estado de la base de datos, si se crearon las tablas y las filas y otra información.

```
mysql> SELECT Resource from Beanstalk;
+-----+
| Resource      |
+-----+
| EC2 Instance |
| RDS Instance |
+-----+
2 rows in set (0.01 sec)
```

Uso del conjunto de herramientas AWS Toolkit for Eclipse

El conjunto de herramientas de AWS Toolkit for Eclipse integra características de administración de AWS Elastic Beanstalk con el entorno de desarrollo de Tomcat para facilitar la creación del entorno así como su configuración y la implementación de código. El conjunto de herramientas es compatible con numerosas cuentas de AWS, permite administrar los entornos existentes y permite conectarse directamente a las instancias del entorno para la resolución de problemas.

Note

El conjunto de herramientas AWS Toolkit for Eclipse solamente es compatible con proyectos que utilizan la plataforma Java con Tomcat, pero no la plataforma Java SE.

Para obtener más información sobre los requisitos previos y la instalación de el conjunto de herramientas AWS Toolkit for Eclipse, visite <https://aws.amazon.com/eclipse>. También puede ver el video [Using AWS Elastic Beanstalk with the AWS Toolkit for Eclipse](#). Este tema contiene también información útil sobre herramientas, temas con procedimientos y otros recursos para los desarrolladores de Java.

Importación de entornos existentes en Eclipse

Puede importar a Eclipse los entornos existentes creados en la consola de administración de AWS.

Para importar entornos existentes, expanda el nodo AWS Elastic Beanstalk y haga doble clic en un entorno en AWS Explorer (Explorador de AWS) dentro de Eclipse. Ahora puede implementar las aplicaciones de Elastic Beanstalk en este entorno.

Administración de entornos de aplicaciones de Elastic Beanstalk

Temas

- [Modificación de la configuración del entorno](#)
- [Cambiar el tipo de entorno](#)
- [Configuración de instancias de servidor EC2 con AWS Toolkit for Eclipse](#)
- [Configuración de Elastic Load Balancing con AWS Toolkit for Eclipse](#)
- [Configuración de Auto Scaling con AWS Toolkit for Eclipse](#)
- [Configuración de notificaciones con AWS Toolkit for Eclipse](#)
- [Configuración de contenedores de Java con AWS Toolkit for Eclipse](#)
- [Configuración de las propiedades del sistema con el conjunto de herramientas AWS Toolkit for Eclipse](#)

Con el AWS Toolkit for Eclipse, puede modificar el aprovisionamiento y la configuración de los recursos de AWS que se utilizan en sus entornos de aplicaciones. Para obtener más información sobre la administración de sus entornos de aplicaciones a través de la consola de administración de AWS, consulte [Administración de entornos](#). En esta sección, se abordan las configuraciones

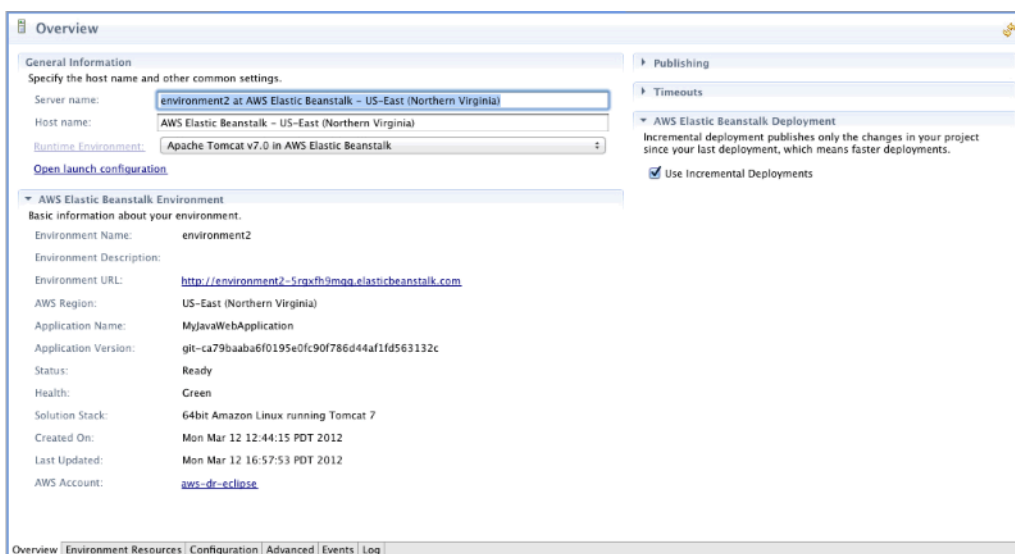
específicas del servicio que se pueden editar en AWS Toolkit for Eclipse durante la configuración del entorno de aplicaciones. Para obtener más información sobre AWS Toolkit for Eclipse, consulte la [Guía de introducción de AWS Toolkit for Eclipse](#).

Modificación de la configuración del entorno

Cuando implementa su aplicación, Elastic Beanstalk configura una serie de servicios informáticos en la nube de AWS. Puede controlar la forma en que se configura cada uno de estos servicios a través del conjunto de herramientas AWS Toolkit for Eclipse.

Para editar la configuración del entorno de una aplicación

1. Si Eclipse no muestra la vista de AWS Explorer, en el menú, elija Window (Ventana), Show View (Mostrar vista), AWS Explorer (Explorador de AWS). Expanda el nodo Elastic Beanstalk y el nodo de la aplicación.
2. En AWS Explorer (Explorador de AWS), haga doble clic en su entorno de Elastic Beanstalk.
3. En la parte inferior del panel, haga clic en la pestaña Configuration (Configuración).



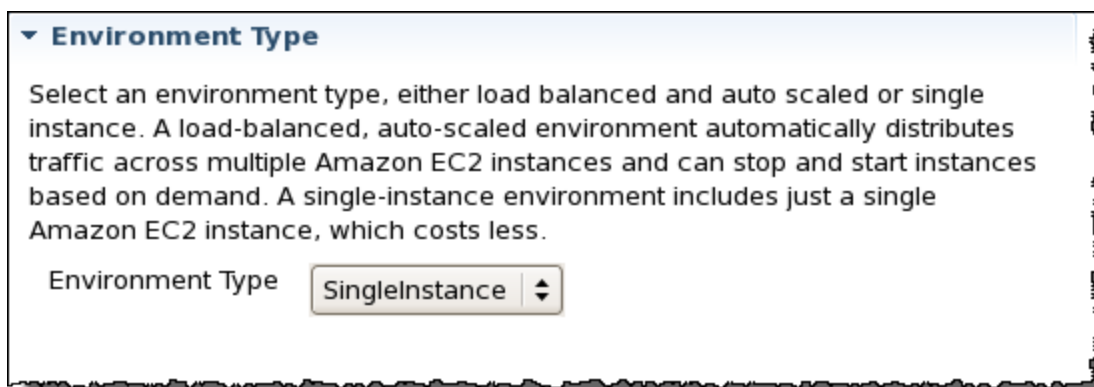
Ahora, podrá configurar las opciones de los elementos siguientes:

- Instancias de servidor EC2
- Balanceador de carga
- Autoescalado
- Notificaciones
- Tipos de entornos

- Propiedades del entorno

Cambiar el tipo de entorno

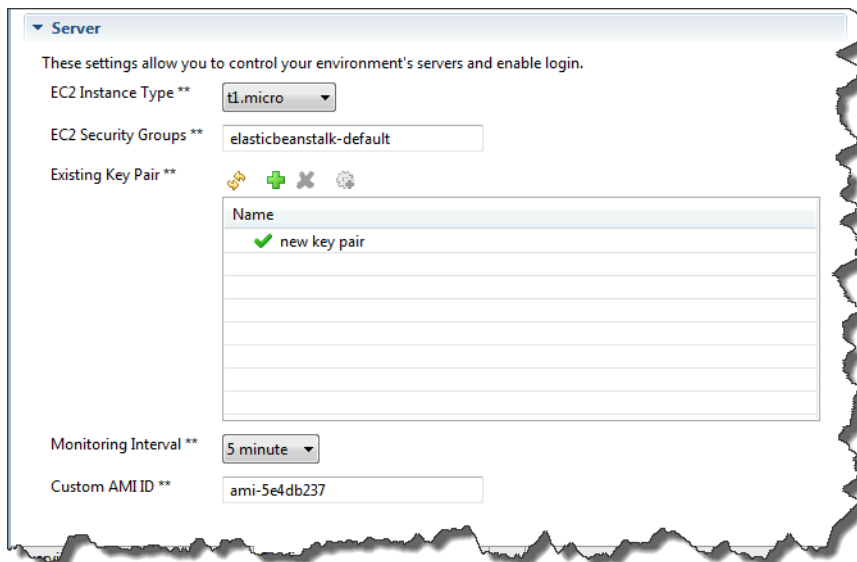
En AWS Toolkit for Eclipse, en la sección Environment Type (Tipo de entorno) en la pestaña Configuration (Configuración) del entorno, seleccione un entorno Load balanced, auto scaled (Equilibrador de carga, autoescalado) o Single instance (Instancia individual), en función de los requisitos de la aplicación que implemente. Si la aplicación requiere escalabilidad, seleccione Load balanced, auto scaled (Carga balanceada, escalado automático). Si se trata de una aplicación sencilla con poco tráfico, seleccione Single instance (Instancia única). Para obtener más información, consulte [Tipos de entornos](#).



Configuración de instancias de servidor EC2 con AWS Toolkit for Eclipse

Amazon Elastic Compute Cloud (EC2) es un servicio web para lanzar y administrar las instancias de servidor de los centros de datos de Amazon. Puede utilizar instancias de servidor Amazon EC2 en cualquier momento, durante el tiempo que las necesite y con cualquier finalidad legal. Las instancias están disponibles en diferentes tamaños y configuraciones. Para obtener más información, vaya a la [página del producto de Amazon EC2](#).

En Server (Servidor), en la pestaña Configuration (Configuración) del entorno de Toolkit for Eclipse, puede editar la configuración de la instancia de Amazon EC2 del entorno de Elastic Beanstalk.



The screenshot shows the 'Server' configuration section in the AWS Elastic Beanstalk console. It includes the following fields and options:

- EC2 Instance Type ****: A dropdown menu set to 't1.micro'.
- EC2 Security Groups ****: A text input field containing 'elasticbeanstalk-default'.
- Existing Key Pair ****: A section with icons for creating a new key pair (green plus), deleting an existing one (red X), and refreshing the list (circular arrow). Below it is a table with one entry: 'new key pair' with a green checkmark in the first column.
- Monitoring Interval ****: A dropdown menu set to '5 minute'.
- Custom AMI ID ****: A text input field containing 'ami-5e4db237'.

Tipos de instancias de Amazon EC2

Instance type (Tipo de instancia) muestra los tipos de instancias disponibles para la aplicación de Elastic Beanstalk. Cambie el tipo de instancia para seleccionar un servidor con las características más adecuadas para su aplicación (entre ellas, el tamaño de la memoria y la potencia de la CPU). Por ejemplo, las aplicaciones de operaciones intensivas de ejecución prolongada pueden necesitar más memoria o CPU.

Para obtener más información acerca de los tipos de instancias de Amazon EC2 disponibles con la aplicación de Elastic Beanstalk, consulte [Tipos de instancias](#) en la Guía del usuario de Amazon Elastic Compute Cloud.

Grupos de seguridad de Amazon EC2

Puede controlar el acceso a la aplicación de Elastic Beanstalk mediante un grupo de seguridad de Amazon EC2. Los grupos de seguridad definen las reglas del firewall para las instancias. Estas reglas especifican qué tráfico de la red de entrada debe entregarse en la instancia. El resto del tráfico de entrada se descartará. Puede modificar las reglas de un grupo en cualquier momento. Las nuevas reglas se aplicarán automáticamente a todas las instancias que estén en ejecución y a las que se lancen en el futuro.

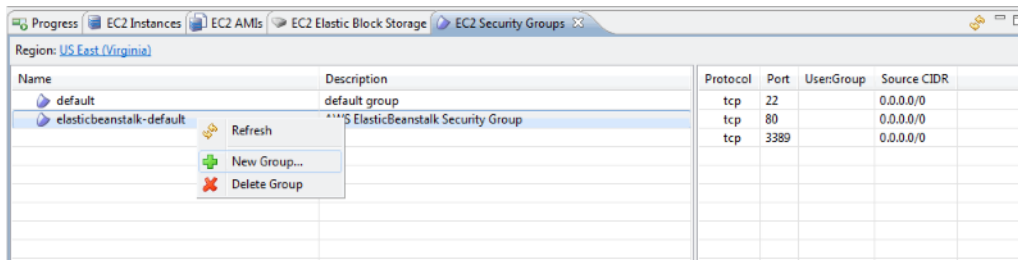
Para configurar los grupos de seguridad de Amazon EC2 puede usar la consola de administración de AWS o el AWS Toolkit for Eclipse. Puede especificar qué grupos de seguridad de Amazon EC2 controlan el acceso a la aplicación de Elastic Beanstalk especificando los nombres de uno o varios grupos de seguridad de Amazon EC2 (delimitados por comas) en el cuadro EC2 Security Groups (Grupos de seguridad de EC2).

Note

Si ejecuta la aplicación utilizando un tipo de contenedor heredado y desea habilitar las comprobaciones de estado en la aplicación, asegúrese de que el puerto 80 (HTTP) esté accesible desde 0.0.0.0/0 como el intervalo de CIDR de origen. Para obtener más información sobre las comprobaciones de estado, consulte [Comprobaciones de estado](#). Para comprobar si está usando un tipo de contenedor heredado, consulte [the section called “¿Por qué algunas versiones de la plataforma están marcadas como heredadas?”](#)

Para crear un grupo de seguridad con AWS Toolkit for Eclipse

1. En AWS Toolkit for Eclipse haga clic en la pestaña AWS Explorer (Explorador de AWS). Expanda el nodo Amazon EC2 y haga doble clic en Security Groups (Grupos de seguridad).
2. Haga clic con el botón derecho en cualquier lugar de la tabla de la izquierda y seleccione New Group (Grupo nuevo).



3. En el cuadro de diálogo Security Group (Grupo de seguridad), escriba un nombre y una descripción para el grupo de seguridad y haga clic en OK (Aceptar).

Para obtener más información sobre los grupos de seguridad de Amazon EC2, consulte [Uso de grupos de seguridad](#) en la Guía del usuario de Amazon Elastic Compute Cloud.

Pares de claves de Amazon EC2

Puede iniciar sesión de forma segura en las instancias de Amazon EC2 aprovisionadas por la aplicación de Elastic Beanstalk con un par de claves de Amazon EC2.

Important

Debe crear un par de claves de Amazon EC2 y configurar las instancias de Amazon EC2 aprovisionadas por Elastic Beanstalk de forma que utilicen el par de claves de Amazon EC2 para poder obtener acceso a las instancias de Amazon EC2 aprovisionadas por Elastic

Beanstalk. Puede crear el par de claves con el Publish to Beanstalk Wizard (Asistente de Publicación en Beanstalk) del conjunto de herramientas AWS Toolkit for Eclipse cuando implemente su aplicación en Elastic Beanstalk. Si lo desea, también puede configurar el par de claves de Amazon EC2 con la [consola de administración de AWS](#). Si desea obtener instrucciones acerca de cómo crear un par de claves para Amazon EC2, consulte la [Guía de introducción de Amazon Elastic Compute Cloud](#).

Para obtener más información sobre los pares de claves de Amazon EC2, consulte [Uso de credenciales de Amazon EC2](#) en la Guía del usuario de Amazon Elastic Compute Cloud. Para obtener más información acerca de cómo conectarse a las instancias de Amazon EC2, vaya a [Conexión a instancias](#) y [Conexión a una instancia de Linux/UNIX desde Windows con PuTTY](#) en la Guía del usuario de Amazon Elastic Compute Cloud.

Métricas de CloudWatch

De manera predeterminada, solo están habilitadas las métricas básicas de Amazon CloudWatch. Estas devuelven datos en periodos de cinco minutos. Puede habilitar métricas de CloudWatch de un minuto pormenorizada al seleccionar 1 minute (1 minuto) en el Monitoring Interval (Intervalo de monitoreo) de la sección Server (Servidor) en la pestaña Configuration (Configuración) del entorno en el AWS Toolkit for Eclipse.

Note

Se pueden aplicar gastos por las métricas en intervalos de un minuto del servicio Amazon CloudWatch. Consulte [Amazon CloudWatch](#) para obtener más información.

ID de AMI personalizada

Puede anular la AMI predeterminada utilizada en las instancias de Amazon EC2 y sustituirla por la suya personalizada al ingresar el identificador de la AMI personalizada en el cuadro Custom AMI ID (ID de AMI personalizada) de la sección Server (Servidor) en la pestaña Configuration (Configuración) del entorno en el AWS Toolkit for Eclipse.

Important

El uso de una AMI propia es una tarea avanzada que debe llevarse a cabo con precaución. Si necesita una AMI personalizada, le recomendamos que comience con la AMI

predeterminada de Elastic Beanstalk y después la modifique. Para darlas por buenas, Elastic Beanstalk espera que las instancias de Amazon EC2 satisfagan una serie de requisitos, entre ellos disponer de un administrador host en ejecución. Si no se cumplen, es posible que el entorno no funcione correctamente.

Configuración de Elastic Load Balancing con AWS Toolkit for Eclipse

Elastic Load Balancing es un servicio web de Amazon que mejora la disponibilidad y escalabilidad de la aplicación. Con Elastic Load Balancing, puede distribuir las cargas de la aplicación entre dos o más instancias de Amazon EC2. Elastic Load Balancing mejora la disponibilidad a través de la redundancia y admite el aumento del tráfico en la aplicación.

Elastic Load Balancing distribuye y balancea automáticamente el tráfico entrante de la aplicación entre todas las instancias de servidor EC2 que están en ejecución. Además, el servicio facilita la incorporación de nuevas instancias cuando es necesario aumentar la capacidad de la aplicación.

Cuando se implementa una aplicación, Elastic Beanstalk aprovisiona automáticamente Elastic Load Balancing. En Load Balancing (Balanceo de carga), en la pestaña Configuration (Configuración) del entorno de Toolkit for Eclipse, puede editar la configuración de balanceo de carga del entorno de Elastic Beanstalk.

Load Balancing

These settings allow you to control the behavior of your environment's load balancer.

HTTP Port: 80

HTTPS Port: OFF

SSL Certificate Id:

EC2 Instance Health Check

These settings allow you to configure how AWS Elastic Beanstalk determines whether an EC2 instance is healthy or not.

Application Health Check URL:

Health Check Interval (seconds):

Health Check Timeout (seconds):

Healthy Check Count Threshold:

Unhealthy Check Count Threshold:

Sessions

These settings allow you to control how your load balancer handles session cookies.

Enable Session Stickiness

Cookie Expiration Period (seconds):

En las siguientes secciones se describen los parámetros de Elastic Load Balancing que puede configurar para la aplicación.

Puertos

El balanceador de carga provisionado para atender las solicitudes de la aplicación de Elastic Beanstalk envía solicitudes a las instancias de Amazon EC2 donde se ejecuta la aplicación. El equilibrador de carga provisionado puede atender las solicitudes de los puertos HTTP y HTTPS y direccionarlas a las instancias de Amazon EC2 de la aplicación de AWS Elastic Beanstalk. De forma predeterminada, el balanceador de carga atiende las solicitudes en el puerto HTTP. Al menos uno de los puertos (HTTP o HTTPS) debe estar activo.



⚠ Important

Asegúrese de que el puerto especificado no esté bloqueado; de lo contrario, los usuarios no podrán conectarse a la aplicación de Elastic Beanstalk.

Control del puerto HTTP

Para desactivar el puerto HTTP, seleccione OFF en HTTP Puerto Listener. Para activar el puerto HTTP, seleccione un puerto HTTP (por ejemplo, 80).

📘 Note

Para tener acceso al entorno usando un puerto diferente del predeterminado 80, por ejemplo, el puerto 8080, añada un agente de escucha al balanceador de carga existente y configúrelo para que escuche en dicho puerto.

Por ejemplo, si usa la [AWS CLI para los equilibradores de carga clásicos](#), escriba el siguiente comando, reemplazando `LOAD_BALANCER_NAME` por el nombre de su equilibrador de carga para Elastic Beanstalk.

```
aws elb create-load-balancer-listeners --load-balancer-name LOAD_BALANCER_NAME
--listeners "Protocol=HTTP, LoadBalancerPort=8080, InstanceProtocol=HTTP,
InstancePort=80"
```

Por ejemplo, si usa la [AWS CLI para los equilibradores de carga de aplicaciones](#), escriba el siguiente comando, reemplazando `LOAD_BALANCER_ARN` por el ARN de su equilibrador de carga para Elastic Beanstalk.

```
aws elbv2 create-listener --load-balancer-arn LOAD_BALANCER_ARN --protocol HTTP  
--port 8080
```

Si quiere que Elastic Beanstalk supervise el entorno, no elimine el agente de escucha del puerto 80.

Control del puerto HTTPS

Elastic Load Balancing admite el protocolo HTTPS/TLS para permitir el cifrado del tráfico en las conexiones cliente con el balanceador de carga. Las conexiones entre el balanceador de carga y las instancias EC2 se establecen utilizando texto sin formato. De forma predeterminada, el puerto HTTPS está desactivado.

Para activar el puerto HTTPS

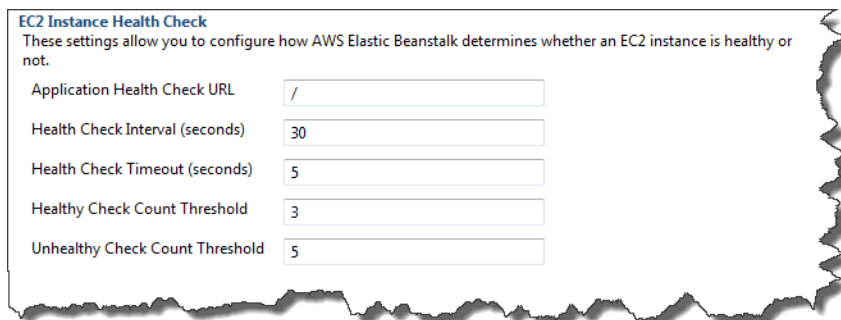
1. Cree un certificado nuevo con AWS Certificate Manager (ACM) o cargue un certificado y una clave en AWS Identity and Access Management (IAM). Para obtener más información sobre cómo solicitar un certificado de ACM, consulte [Solicitar un certificado](#) en la Guía del usuario de AWS Certificate Manager. Para obtener más información sobre la importación de certificados de terceros en ACM, consulte [Importar certificados](#) en la Guía del usuario de AWS Certificate Manager. Si ACM no está [disponible en su región de AWS](#), use AWS Identity and Access Management (IAM) para cargar un certificado de terceros. Los servicios de ACM e IAM guardan el certificado y proporcionan un nombre de recurso de Amazon (ARN) para el certificado SSL. Para obtener más información sobre la creación y la carga de certificados en IAM, consulte [Uso de certificados de servidor](#) en la Guía del usuario de IAM.
2. Especifique el puerto HTTPS seleccionando un puerto en la lista desplegable de HTTPS Listener Port.
3. En el cuadro de texto SSL Certificate ID (ID de certificado SSL), escriba el nombre de recurso de Amazon (ARN) del certificado SSL. Por ejemplo,
arn:aws:iam::123456789012:server-certificate/abc/certs/build o
arn:aws:acm:us-

east-2:123456789012:certificate/12345678-12ab-34cd-56ef-12345678. Utilice el certificado SSL que ha creado y cargado en el paso 1.

Para desactivar el puerto HTTPS, seleccione OFF en HTTPS Listener Port (HTTP Puerto Listener).

Comprobaciones de estado

Puede administrar la configuración de la comprobación de estado a través de la sección EC2 Instance Health Check (Comprobación de estado de instancia EC2) del panel Load Balancing (Balanceo de carga).



EC2 Instance Health Check
These settings allow you to configure how AWS Elastic Beanstalk determines whether an EC2 instance is healthy or not.

Application Health Check URL	/
Health Check Interval (seconds)	30
Health Check Timeout (seconds)	5
Healthy Check Count Threshold	3
Unhealthy Check Count Threshold	5

En la siguiente lista se describen los parámetros de comprobación de estado que puede definir para la aplicación.

- Para determinar el estado de la instancia, Elastic Beanstalk busca un código de respuesta 200 en una dirección URL que consulta. De forma predeterminada, Elastic Beanstalk busca contenedores no heredados en TCP:80 y contenedores heredados en HTTP:80. Puede anular para que coincida con un recurso existente en la aplicación (por ejemplo, **/myapp/index.jsp**) utilizando el cuadro Application Health Check URL (URL de comprobación de estado de aplicación). Si anula la URL predeterminada, Elastic Beanstalk utilizará HTTP para consultar el recurso. Para comprobar si está usando un tipo de contenedor heredado, consulte [the section called “¿Por qué algunas versiones de la plataforma están marcadas como heredadas?”](#)
- En Health Check Interval (seconds) (Intervalo de comprobación de estado [segundos]), especifique el número de segundos que deben transcurrir entre las comprobaciones de estado de las instancias de Amazon EC2 de la aplicación.
- En Health Check Timeout (Tiempo de espera de la comprobación de estado), especifique el número de segundos que Elastic Load Balancing va a esperar para recibir una respuesta antes de considerar que la instancia no responde.
- Utilice los cuadros Healthy Check Count Threshold (Umbral de recuento de comprobación de estado) y Unhealthy Check Count Threshold (Umbral de recuento de comprobación incorrecto)

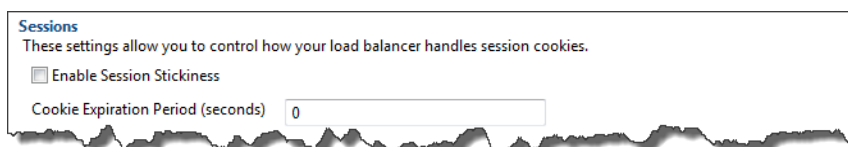
para especificar el número de sondeos de URL consecutivos que deben realizarse correcta o incorrectamente antes de que Elastic Load Balancing cambie el estado de la instancia. Por ejemplo, si se especifica el valor 5 en el cuadro de texto Unhealthy Check Count Threshold (Umbral de recuento de comprobación incorrecto), la dirección URL devolvería un mensaje de error o un mensaje de tiempo de espera agotado cinco veces seguidas antes de que Elastic Load Balancing considerara que no se ha superado la comprobación de estado.

Sesiones

De forma predeterminada, un balanceador de carga direcciona cada solicitud por separado a la instancia de servidor con menor carga. Por su parte, una sesión sticky vincula la sesión de un usuario con una instancia de servidor específica para que todas las solicitudes que provengan de ese usuario durante la sesión se envíen a la misma instancia de servidor.

Elastic Beanstalk utiliza cookies HTTP generadas por el balanceador de carga cuando las sesiones sticky están habilitadas en una aplicación. El balanceador de carga utiliza una cookie especial generada por el balanceador de carga para hacer un seguimiento de la instancia de aplicación de cada solicitud. Cuando el balanceador de carga recibe una solicitud, primero comprueba si esta cookie está presente en la solicitud. Si lo está, la solicitud se envía a la instancia de aplicación especificada en la cookie. Si no encuentra ninguna cookie, el balanceador de carga elige una instancia de aplicación en función del algoritmo de balanceo de carga existente. Para vincular las solicitudes posteriores del mismo usuario a esa instancia de aplicación, se inserta una cookie en la respuesta. La configuración de la política define el vencimiento de la cookie, que establece el periodo de validez de cada cookie.

En Load Balancer (Balanceador de carga), en la sección Sessions (Sesiones), especifique si el balanceador de carga de la aplicación permite o no las sesiones sticky y la duración de cada cookie.



Para obtener más información sobre Elastic Load Balancing, consulte la [Guía para desarrolladores de Elastic Load Balancing](#).

Configuración de Auto Scaling con AWS Toolkit for Eclipse

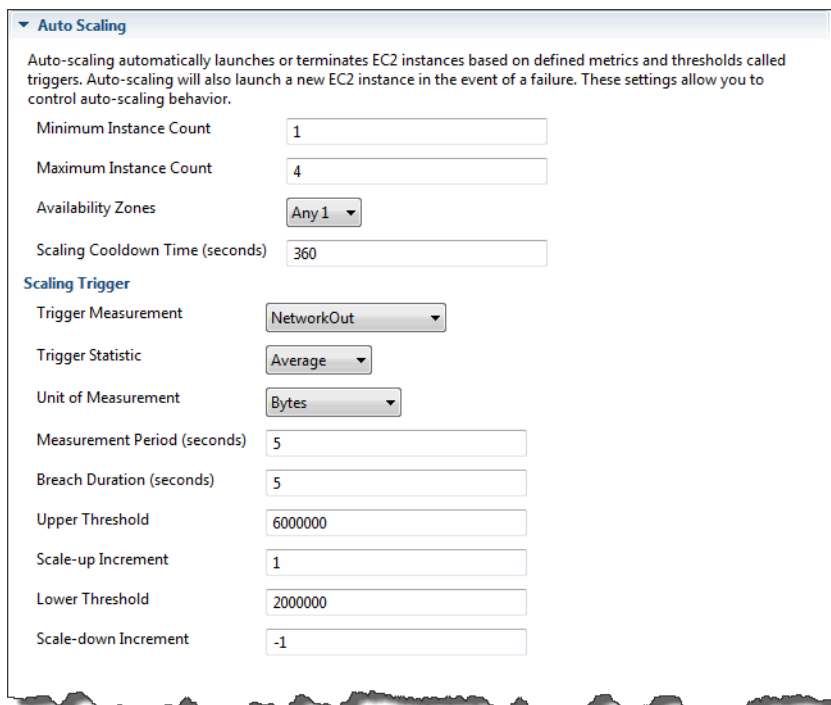
Amazon EC2 Auto Scaling es un servicio web de Amazon diseñado para lanzar o terminar automáticamente instancias de Amazon EC2 en función de disparadores definidos por el usuario.

Los usuarios pueden configurar grupos de Auto Scaling y asociar disparadores a esos grupos para escalar automáticamente los recursos informáticos en función de determinadas métricas, como el uso de ancho de banda o la utilización de CPU. Amazon EC2 Auto Scaling trabaja con Amazon CloudWatch para recuperar las métricas de las instancias de servidor en las que se ejecuta la aplicación.

Amazon EC2 Auto Scaling le permite tomar un grupo de instancias de Amazon EC2 y definir diversos parámetros para hacer que el número de instancias aumente o se reduzca automáticamente. Amazon EC2 Auto Scaling puede añadir o quitar instancias de Amazon EC2 de ese grupo, lo que le ayudará a administrar eficazmente los cambios de tráfico de la aplicación.

Amazon EC2 Auto Scaling también monitoriza el estado de cada instancia de Amazon EC2 que lanza. Si una instancia termina de forma inesperada, Amazon EC2 Auto Scaling lo detecta y lanza una instancia de sustitución. Esta función le permite mantener un número fijo de instancias de Amazon EC2 automáticamente.

Elastic Beanstalk aprovisiona Amazon EC2 Auto Scaling a su aplicación. En Auto Scaling, en la pestaña Configuration (Configuración) de su entorno de Toolkit for Eclipse, puede editar la configuración de Auto Scaling del entorno de Elastic Beanstalk.



Auto-scaling automatically launches or terminates EC2 instances based on defined metrics and thresholds called triggers. Auto-scaling will also launch a new EC2 instance in the event of a failure. These settings allow you to control auto-scaling behavior.

Minimum Instance Count	<input type="text" value="1"/>
Maximum Instance Count	<input type="text" value="4"/>
Availability Zones	<input type="text" value="Any 1"/>
Scaling Cooldown Time (seconds)	<input type="text" value="360"/>
Scaling Trigger	
Trigger Measurement	<input type="text" value="NetworkOut"/>
Trigger Statistic	<input type="text" value="Average"/>
Unit of Measurement	<input type="text" value="Bytes"/>
Measurement Period (seconds)	<input type="text" value="5"/>
Breach Duration (seconds)	<input type="text" value="5"/>
Upper Threshold	<input type="text" value="6000000"/>
Scale-up Increment	<input type="text" value="1"/>
Lower Threshold	<input type="text" value="2000000"/>
Scale-down Increment	<input type="text" value="-1"/>

En las secciones siguientes, se explica cómo se configuran los parámetros de Auto Scaling para la aplicación.

Configuración de lanzamiento

Puede editar la configuración de lanzamiento para controlar la forma en que la aplicación de Elastic Beanstalk aprovisiona los recursos de Amazon EC2 Auto Scaling.

Utilice las opciones Minimum Instance Count (Número mínimo de instancias) y Maximum Instance Count (Número máximo de instancias) para especificar el tamaño mínimo y máximo del grupo de Auto Scaling que la aplicación de Elastic Beanstalk utiliza.

A screenshot of the Elastic Beanstalk launch configuration settings. The settings are displayed in a white box with a torn paper effect on the right side. The settings are: Minimum Instance Count (input field with value 1), Maximum Instance Count (input field with value 4), Availability Zones (dropdown menu with value Any 1), and Scaling Cooldown Time (seconds) (input field with value 360).

Note

Para mantener un número fijo de instancias de Amazon EC2, establezca los cuadros de texto Minimum Instance Count (Número mínimo de instancias) y Maximum Instance Count (Número máximo de instancias) en el mismo valor.

En Availability Zones (Zonas de disponibilidad), especifique el número de zonas de disponibilidad que quiere usar con las instancias de Amazon EC2. Es importante definir este número si quiere crear aplicaciones que sean tolerantes a errores: Si una zona de disponibilidad deja de funcionar, las instancias seguirán ejecutándose en las otras.

Note

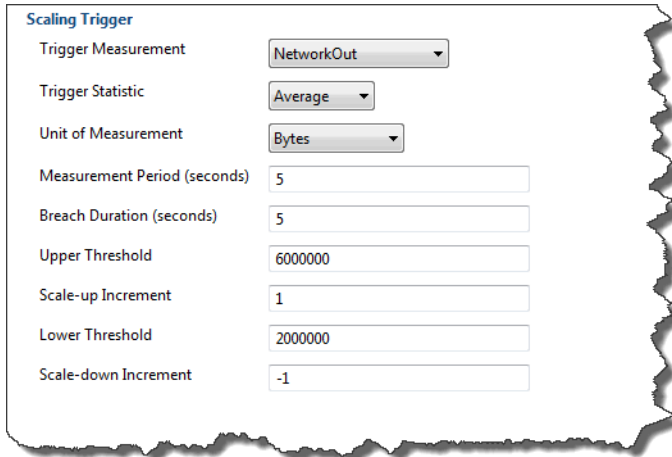
En la actualidad, no es posible especificar en qué zona de disponibilidad se encontrará la instancia.

Desencadenadores

Un disparador es un mecanismo de Amazon EC2 Auto Scaling que se define para indicar al sistema cuándo debe aumentar (escalado ascendente) y reducir (escalado descendente) el número de instancias. Puede configurar disparadores para activar cualquiera de las métricas publicadas en

Amazon CloudWatch, como el uso de la CPU, y determinar si se han cumplido las condiciones especificadas. Cuando se supera el umbral superior o inferior de la métrica durante el período de tiempo especificado, el disparador lanza un proceso de larga ejecución llamado actividad de escalado.

Puede definir un desencadenador de escalado para su aplicación de Elastic Beanstalk con el conjunto de herramientas AWS Toolkit for Eclipse.



Field	Value
Trigger Measurement	NetworkOut
Trigger Statistic	Average
Unit of Measurement	Bytes
Measurement Period (seconds)	5
Breach Duration (seconds)	5
Upper Threshold	6000000
Scale-up Increment	1
Lower Threshold	2000000
Scale-down Increment	-1

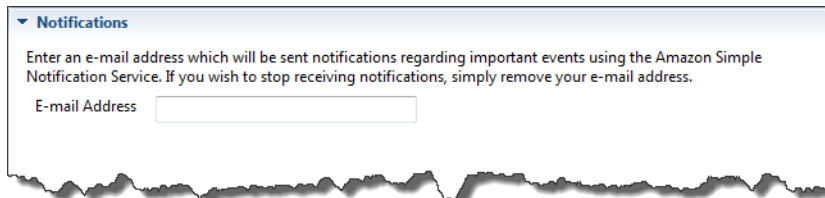
En Toolkit for Eclipse, puede configurar la siguiente lista de parámetros de disparadores en la sección Scaling Trigger (Disparador de escalado) de la pestaña Configuration (Configuración) del entorno.

- En Trigger Measurement (Medida desencadenadora), especifique la métrica del disparador.
- En Trigger Statistic (Estadística de disparador), especifique qué estadística utilizará el disparador: **Minimum, Maximum, Sum** o **Average**.
- En Unit of Measurement (Unidad de medición), especifique las unidades de medida del disparador.
- En Measurement Period (Período de medición), especifique con qué frecuencia Amazon CloudWatch mide las métricas del disparador. En Breach Duration (Duración de la interrupción), especifique el período de tiempo que una métrica puede sobrepasar el límite definido (según lo especificado en Upper Threshold (Umbral superior) y Lower Threshold (Umbral inferior)) antes de que se active el disparador.
- En Scale-up Increment (Incremento de escalado vertical) y Scale-down Increment (Incremento de escalado horizontal), especifique el número de instancias de Amazon EC2 que se van a agregar o quitar al efectuar una actividad de escalado.

Para obtener más información sobre Amazon EC2 Auto Scaling, consulte la sección Auto Scaling de Amazon EC2 en la [Guía de documentación de Amazon Elastic Compute Cloud](#).

Configuración de notificaciones con AWS Toolkit for Eclipse

Elastic Beanstalk utiliza Amazon Simple Notification Service (Amazon SNS) para enviarle notificaciones sobre eventos importantes que afectan a la aplicación. Para habilitar las notificaciones de Amazon SNS, solo tiene que especificar la dirección de correo electrónico en el cuadro de texto Email Address (Dirección de correo electrónico) situado bajo Notifications (Notificaciones) en la pestaña Configuration (Configuración) del entorno de Toolkit for Eclipse. Para desactivar las notificaciones de Amazon SNS, quite la dirección de correo electrónico del cuadro de texto.



Configuración de contenedores de Java con AWS Toolkit for Eclipse

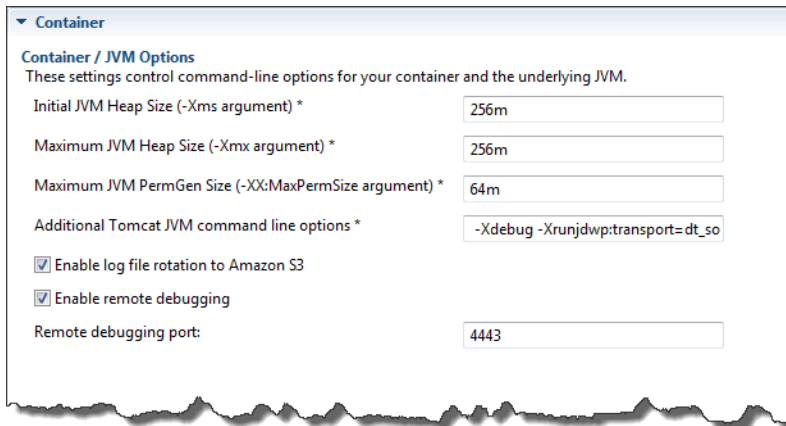
El panel Container/JVM Options (Opciones de contenedor/JVM) le permite ajustar el comportamiento de la máquina virtual de Java en las instancias de Amazon EC2 y habilitar o deshabilitar la rotación de registros de Amazon S3. Puede utilizar AWS Toolkit for Eclipse para configurar la información del contenedor. Para obtener más información sobre las opciones disponibles en los entornos de Tomcat, consulte [the section called “Configuración del entorno de Tomcat”](#).

Note

Puede modificar la configuración sin tiempo de inactividad intercambiando el CNAME de los entornos. Para obtener más información, consulte [Implementaciones Blue/Green \(azul/verde\) con Elastic Beanstalk](#).

Para obtener acceso al panel Container/JVM Options (Opciones de contenedor/JVM) de su aplicación de Elastic Beanstalk

1. Si Eclipse no muestra la vista de AWS Explorer, en el menú, elija Window (Ventana), Show View (Mostrar vista), AWS Explorer (Explorador de AWS). Expanda el nodo Elastic Beanstalk y el nodo de la aplicación.
2. En AWS Explorer (Explorador de AWS), haga doble clic en su entorno de Elastic Beanstalk.
3. En la parte inferior del panel, haga clic en la pestaña Configuration (Configuración).
4. En Container (Contenedor), puede configurar las opciones del contenedor.



Depuración remota

Para probar la aplicación en remoto, puede ejecutarla en modo de depuración.

Para habilitar la depuración remota

1. Seleccione Enable remote debugging (Habilitar depuración remota).
2. En Remote debugging port (Puerto de depuración remota), especifique el número de puerto que desea usar con la depuración remota

La opción Additional Tomcat JVM command line options (Opciones de línea de comandos adicionales de Tomcat JVM) se rellena automáticamente.


Para iniciar la depuración remota

1. En el menú de AWS Toolkit for Eclipse, elija Window (Ventana), Show View (Mostrar vista), Other (Otro).
2. Expanda la carpeta Server (Servidor) y elija Servers (Servidores). Seleccione OK (Aceptar).
3. En el panel Servers (Servidores), haga clic con el botón derecho del servidor en el que se ejecuta la aplicación y seleccione Restart in Debug (Reiniciar en depuración).


Configuración de las propiedades del sistema con el conjunto de herramientas AWS Toolkit for Eclipse

En el siguiente ejemplo, se establece la propiedad del sistema JDBC_CONNECTION_STRING en el conjunto de herramientas AWS Toolkit for Eclipse. Después de configurar esta propiedad, esta pasa

a estar disponible en la aplicación de Elastic Beanstalk como una propiedad del sistema llamada `JDBC_CONNECTION_STRING`.

 Note

El conjunto de herramientas AWS Toolkit for Eclipse aún no permite modificar la configuración, incluidas las propiedades del sistema, de los entornos en una VPC. A menos que tenga una cuenta antigua que utilice EC2 Classic, deberá usar la consola de administración de AWS (que se describe en la siguiente sección) o la [CLI de EB](#).

 Note

Las opciones de configuración del entorno pueden contener cualquier carácter ASCII imprimible, excepto el acento grave (' , ASCII 96), y no pueden superar los 200 caracteres.

Para establecer las propiedades del sistema de la aplicación de Elastic Beanstalk

1. Si Eclipse no muestra la vista de AWS Explorer, en el menú elija Window (Ventana), Show View (Mostrar vista), Other (Otro). Expanda AWS Toolkit (Conjunto de herramientas de AWS) y, a continuación, elija AWS Explorer (Explorador de AWS).
2. En el panel AWS Explorer (Explorador de AWS), expanda Elastic Beanstalk así como el nodo de su aplicación y, a continuación, haga doble clic en el entorno de Elastic Beanstalk.
3. En la parte inferior del panel del entorno, haga clic en la pestaña Advanced (Avanzado).
4. En `aws:elasticbeanstalk:application:environment`, haga clic en `JDBC_CONNECTION_STRING` y escriba una cadena de conexión. Por ejemplo, la siguiente cadena de conexión de JDBC se conectaría a una instancia de base de datos de MySQL en el puerto 3306 del host local con el nombre de usuario `me` y la contraseña `mypassword`:

```
jdbc:mysql://localhost:3306/mydatabase?user=me&password=mypassword
```

Esta cadena de conexión estaría accesible para la aplicación de Elastic Beanstalk como una propiedad del sistema llamada `JDBC_CONNECTION_STRING`.

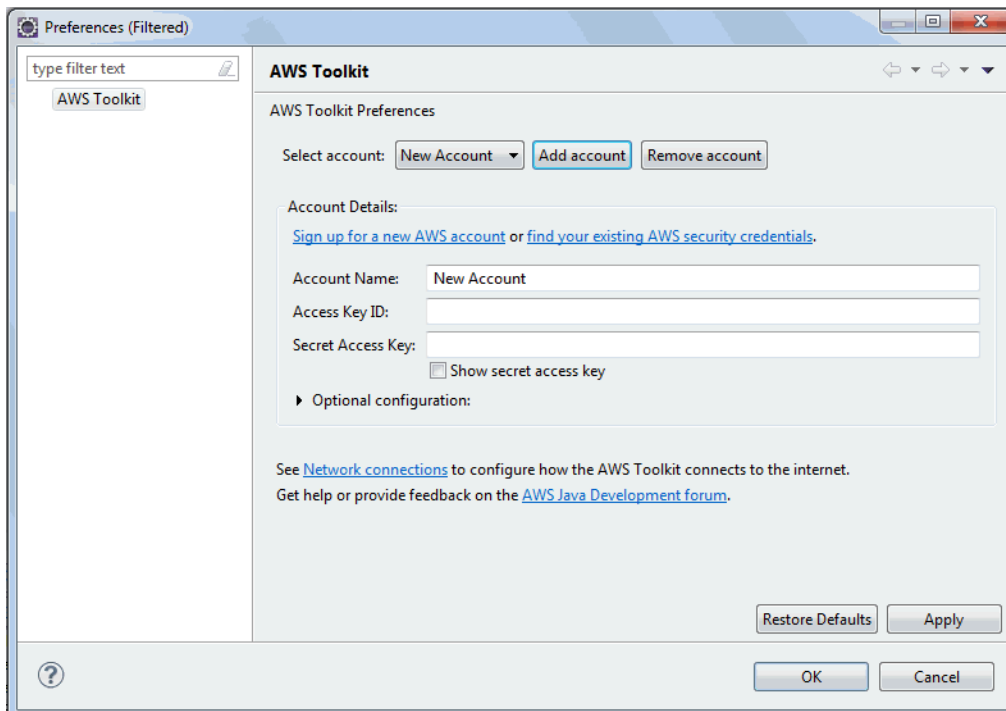
5. Presione Ctrl+C en el teclado o seleccione File (Archivo), Save (Guardar) para guardar los cambios realizados en la configuración del entorno. Los cambios tardan aproximadamente un minuto en reflejarse.

Administración de varias cuentas de AWS

Es posible que quiera configurar diversas cuentas de AWS para realizar diferentes tareas, como pruebas, ensayo y producción. Puede utilizar el conjunto de herramientas AWS Toolkit for Eclipse para agregar, editar y eliminar las cuentas con facilidad.

Para agregar una cuenta de AWS con el conjunto de herramientas AWS Toolkit for Eclipse

1. En Eclipse, asegúrese de que la barra de herramientas está visible. En la barra de herramientas, haga clic en la flecha situada junto al icono de AWS y seleccione Preferences (Preferencias).
2. Haga clic en Add account (Agregar cuenta).



3. En el cuadro de texto Account Name (Nombre de la cuenta, especifique el nombre que se va mostrar para la cuenta.
4. En el cuadro de texto Access Key ID (ID de clave de acceso), escriba su ID de clave de acceso de AWS.
5. En el cuadro de texto Secret Access Key (Clave de acceso secreta), escriba su clave secreta de AWS.

Para acceder a la API, necesita un ID de clave de acceso y una clave de acceso secreta. Utilice las claves de acceso de usuario de IAM en lugar de las claves de acceso de Usuario raíz de la cuenta de AWS. Para obtener más información sobre la creación de claves de acceso, consulte [Administración de las claves de acceso de los usuarios de IAM](#) en la Guía del usuario de IAM

6. Haga clic en OK (Aceptar).

Si desea utilizar una cuenta distinta para implementar una aplicación en Elastic Beanstalk

1. En la barra de herramientas de Eclipse, haga clic en la flecha situada junto al icono de AWS y seleccione Preferences (Preferencias).
2. En Default Account (Cuenta predeterminada), seleccione la cuenta que desea utilizar para implementar aplicaciones en Elastic Beanstalk.
3. Haga clic en OK (Aceptar).
4. En el panel Project Explorer (Explorador de proyectos), haga clic con el botón derecho en la aplicación que desea implementar y seleccione Amazon Web Services > Deploy to Elastic Beanstalk (Implementar en Elastic Beanstalk).

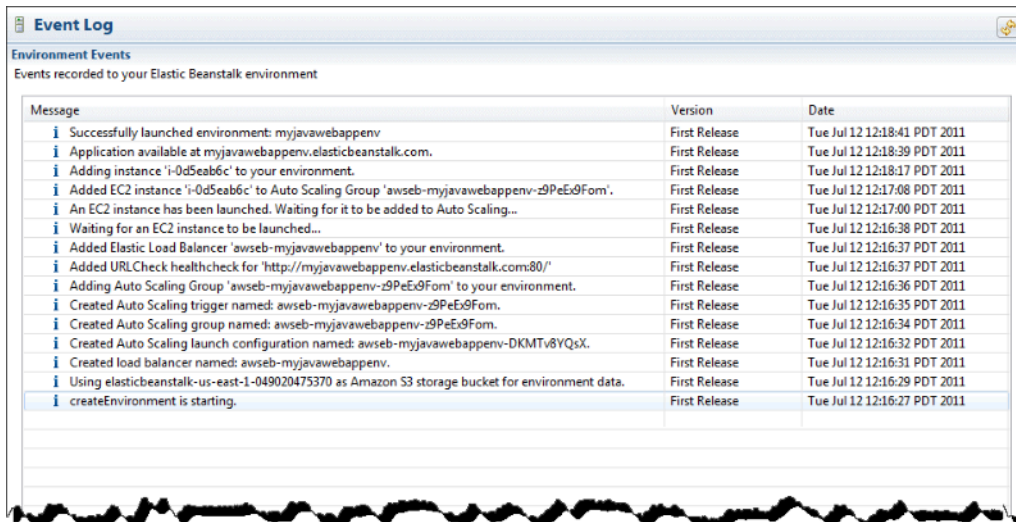
Consulta de eventos de

Puede utilizar el conjunto de herramientas AWS Toolkit for Eclipse para obtener acceso a los eventos y notificaciones relacionados con su aplicación.

Para ver los eventos de la aplicación

1. Si Eclipse no muestra la vista de AWS Explorer (Explorador de AWS), en el menú, haga clic en Window (Ventana), Show View (Mostrar vista), AWS Explorer (Explorador de AWS). Expanda el nodo Elastic Beanstalk y el nodo de la aplicación.
2. En AWS Explorer (Explorador de AWS), haga doble clic en su entorno de Elastic Beanstalk.
3. En la parte inferior del panel, haga clic en la pestaña Events (Eventos).

Aparece una lista con los eventos de todos los entornos de la aplicación.



The screenshot shows the 'Event Log' window in AWS Elastic Beanstalk. It displays a list of environment events for the environment 'myjavawebappenv'. The events are recorded to the environment and include messages, versions, and dates. The events are as follows:

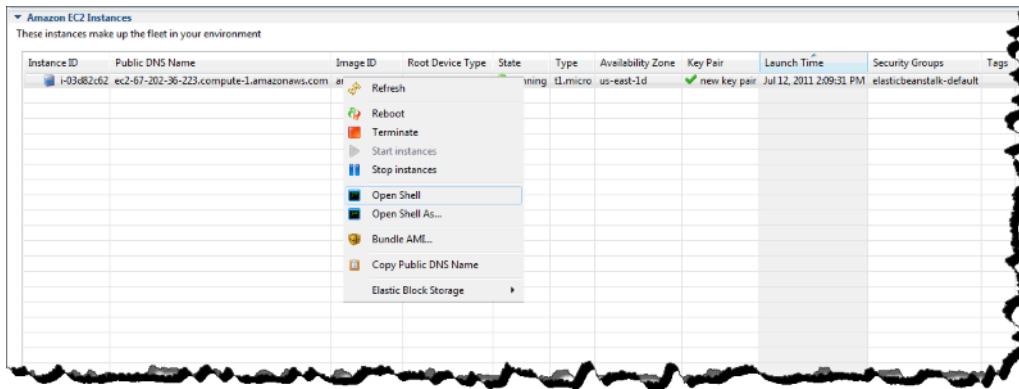
Message	Version	Date
Successfully launched environment: myjavawebappenv	First Release	Tue Jul 12 12:18:41 PDT 2011
Application available at myjavawebappenv.elasticbeanstalk.com.	First Release	Tue Jul 12 12:18:39 PDT 2011
Adding instance 'i-0d5eab6c' to your environment.	First Release	Tue Jul 12 12:18:17 PDT 2011
Added EC2 instance 'i-0d5eab6c' to Auto Scaling Group 'awseb-myjavawebappenv-z9PeE9Fom'.	First Release	Tue Jul 12 12:17:08 PDT 2011
An EC2 instance has been launched. Waiting for it to be added to Auto Scaling...	First Release	Tue Jul 12 12:17:00 PDT 2011
Waiting for an EC2 instance to be launched...	First Release	Tue Jul 12 12:16:38 PDT 2011
Added Elastic Load Balancer 'awseb-myjavawebappenv' to your environment.	First Release	Tue Jul 12 12:16:37 PDT 2011
Added URLCheck healthcheck for 'http://myjavawebappenv.elasticbeanstalk.com:80/'	First Release	Tue Jul 12 12:16:37 PDT 2011
Adding Auto Scaling Group 'awseb-myjavawebappenv-z9PeE9Fom' to your environment.	First Release	Tue Jul 12 12:16:36 PDT 2011
Created Auto Scaling trigger named: awseb-myjavawebappenv-z9PeE9Fom.	First Release	Tue Jul 12 12:16:35 PDT 2011
Created Auto Scaling group named: awseb-myjavawebappenv-z9PeE9Fom.	First Release	Tue Jul 12 12:16:34 PDT 2011
Created Auto Scaling launch configuration named: awseb-myjavawebappenv-DKMTv8YQsX.	First Release	Tue Jul 12 12:16:32 PDT 2011
Created load balancer named: awseb-myjavawebappenv.	First Release	Tue Jul 12 12:16:31 PDT 2011
Using elasticbeanstalk-us-east-1-049020475370 as Amazon S3 storage bucket for environment data.	First Release	Tue Jul 12 12:16:29 PDT 2011
createEnvironment is starting.	First Release	Tue Jul 12 12:16:27 PDT 2011

Listado y conexión a instancias de servidor

Puede ver una lista con las instancias de Amazon EC2 que están en ejecución en el entorno de su aplicación de Elastic Beanstalk a través del conjunto de herramientas AWS Toolkit for Eclipse o de la consola de administración de AWS. Puede conectarse a estas instancias utilizando Secure Shell (SSH). Para obtener información sobre cómo mostrar las instancias del servidor y conectarse a ellas a través de la consola de administración de AWS, consulte [Listado y conexión a instancias de servidor](#). En la siguiente sección, se detalla un procedimiento para consultar las instancias de servidor y conectarse a ellas con el conjunto de herramientas AWS Toolkit for Eclipse.

Para ver las instancias de Amazon EC2 de un entorno y conectarse a ellas

1. En el conjunto de herramientas AWS Toolkit for Eclipse, haga clic en AWS Explorer (Explorador de AWS). Expanda el nodo Amazon EC2 y haga doble clic en Instances (Instancias).
2. En la ventana Amazon EC2 Instances, en la columna Instance ID (ID de instancia), haga clic con el botón derecho en el valor de Instance ID (ID de instancia) de la instancia de Amazon EC2 que está en ejecución en el balanceador de carga de la aplicación. A continuación, haga clic en Open Shell (Abrir Shell).



Eclipse abre automáticamente el cliente SSH y establece la conexión con la instancia EC2.

Para obtener más información sobre cómo conectarse a una instancia de Amazon EC2, consulte la [Guía de introducción de Amazon Elastic Compute Cloud](#).

Terminación de un entorno

Para evitar que se apliquen cargos por recursos de AWS que no utiliza, termine el entorno en ejecución con el conjunto de herramientas AWS Toolkit for Eclipse. Para obtener más información sobre la terminación del entorno, consulte [Terminación de un entorno de Elastic Beanstalk](#).

Para terminar un entorno

1. En el conjunto de herramientas AWS Toolkit for Eclipse, haga clic en el panel AWS Explorer (Explorador de AWS). Expanda el nodo Elastic Beanstalk.
2. Expanda la aplicación de Elastic Beanstalk y haga clic con el botón derecho en el entorno de Elastic Beanstalk.
3. Haga clic en Terminate Environment (Terminar entorno). Elastic Beanstalk tarda unos minutos en terminar los recursos de AWS que se ejecutan en el entorno.

Recursos

Existen varios lugares a los que puede acudir para obtener ayuda adicional con el desarrollo de las aplicaciones Java.

Recurso	Descripción
El foro de desarrollo de Java de AWS	Publique sus preguntas y obtenga comentarios.

Recurso	Descripción
Centro de desarrolladores de Java	Un único lugar para obtener código de ejemplo, documentación, herramientas y recursos adicionales.

Trabajar con .NET Core en Linux

Consulte el .NET on AWS Developer Center

¿Has pasado por nuestro Centro de desarrolladores de .Net? Es nuestra ventanilla única para todo lo relacionado con .NET AWS.

Para obtener más información, consulte el [.NET on AWS Developer Center](#).

AWS Elastic Beanstalk para .NET core en Linux facilita la implementación, la administración y el escalado de sus aplicaciones web mediante Amazon Web Services. En este capítulo se proporcionan instrucciones para implementar la aplicación web .NET Core en Elastic Beanstalk en los entornos de Amazon Linux. Puede implementar la aplicación en solo unos minutos mediante la interfaz de línea de comandos de Elastic Beanstalk (EB CLI) o la consola de Elastic Beanstalk.

Siga los pasos de [QuickStart para .NET Core en Linux](#) para crear e implementar una nueva aplicación web de ASP.NET Core con la CLI de EB.

Temas

- [QuickStart: Implemente una aplicación .NET Core para Linux en Elastic Beanstalk](#)
- [Configuración del entorno de desarrollo .NET Core de Linux](#)
- [.NET Core en la plataforma Linux](#)
- [AWS Toolkit for Visual Studio: trabajar con .Net Core](#)
- [Migración de .NET en plataformas de Windows Server a la plataforma .NET Core en Linux](#)

QuickStart: Implemente una aplicación .NET Core para Linux en Elastic Beanstalk

Este QuickStart tutorial explica el proceso de creación de una aplicación .NET Core para Linux e implementarla en un AWS Elastic Beanstalk entorno.

Note

Este QuickStart tutorial está destinado a fines de demostración. No utilice la aplicación creada en este tutorial para el tráfico de producción.

Secciones

- [Tu AWS cuenta](#)
- [Requisitos previos](#)
- [Paso 1: Cree una aplicación.NET Core para Linux](#)
- [Paso 2: Ejecute la aplicación localmente](#)
- [Paso 3: Implemente su aplicación.NET Core en Linux con la CLI de EB](#)
- [Paso 4: Ejecute la aplicación en Elastic Beanstalk](#)
- [Paso 5: Eliminar](#)
- [AWS recursos para su aplicación](#)
- [Sigüientes pasos](#)
- [Implemente con la consola de Elastic Beanstalk](#)

Tu AWS cuenta

Si aún no eres AWS cliente, debes crear una AWS cuenta. El registro le permite acceder a Elastic Beanstalk AWS y a otros servicios que necesite.

Si ya tiene una AWS cuenta, puede pasar a [Requisitos previos](#)

Creará una AWS cuenta

Inscríbese en una Cuenta de AWS

Si no tiene una Cuenta de AWS, complete los siguientes pasos para crearlo.

Para suscribirse a una Cuenta de AWS

1. Abra <https://portal.aws.amazon.com/billing/signup>.
2. Siga las instrucciones que se le indiquen.

Parte del procedimiento de registro consiste en recibir una llamada telefónica e indicar un código de verificación en el teclado del teléfono.

Cuando te registras en una Cuenta de AWS, Usuario raíz de la cuenta de AWS se crea una. El usuario raíz tendrá acceso a todos los Servicios de AWS y recursos de esa cuenta. Como práctica recomendada de seguridad, asigne acceso administrativo a un usuario y utilice únicamente el usuario raíz para realizar [tareas que requieren acceso de usuario raíz](#).

AWS te envía un correo electrónico de confirmación una vez finalizado el proceso de registro. Puede ver la actividad de la cuenta y administrar la cuenta en cualquier momento entrando en <https://aws.amazon.com/> y seleccionando Mi cuenta.

Creación de un usuario con acceso administrativo

Después de crear un usuario administrativo Cuenta de AWS, asegúrelo Usuario raíz de la cuenta de AWS AWS IAM Identity Center, habilite y cree un usuario administrativo para no usar el usuario root en las tareas diarias.

Proteja su Usuario raíz de la cuenta de AWS

1. Inicie sesión [AWS Management Console](#) como propietario de la cuenta seleccionando el usuario root e introduciendo su dirección de Cuenta de AWS correo electrónico. En la siguiente página, escriba su contraseña.

Para obtener ayuda para iniciar sesión con el usuario raíz, consulte [Iniciar sesión como usuario raíz](#) en la Guía del usuario de AWS Sign-In .

2. Active la autenticación multifactor (MFA) para el usuario raíz.

Para obtener instrucciones, consulte [Habilitar un dispositivo MFA virtual para el usuario Cuenta de AWS raíz \(consola\)](#) en la Guía del usuario de IAM.

Creación de un usuario con acceso administrativo

1. Activar IAM Identity Center.

Consulte las instrucciones en [Activar AWS IAM Identity Center](#) en la Guía del usuario de AWS IAM Identity Center .

2. En IAM Identity Center, conceda acceso administrativo a un usuario.

Para ver un tutorial sobre su uso Directorio de IAM Identity Center como fuente de identidad, consulte [Configurar el acceso de los usuarios con la configuración predeterminada Directorio de IAM Identity Center en la](#) Guía del AWS IAM Identity Center usuario.

Iniciar sesión como usuario con acceso de administrador

- Para iniciar sesión con el usuario de IAM Identity Center, utilice la URL de inicio de sesión que se envió a la dirección de correo electrónico cuando creó el usuario de IAM Identity Center.

Para obtener ayuda para iniciar sesión con un usuario del Centro de identidades de IAM, consulte [Iniciar sesión en el portal de AWS acceso](#) en la Guía del AWS Sign-In usuario.

Concesión de acceso a usuarios adicionales

1. En IAM Identity Center, cree un conjunto de permisos que siga la práctica recomendada de aplicar permisos de privilegios mínimos.

Para conocer las instrucciones, consulte [Create a permission set](#) en la Guía del usuario de AWS IAM Identity Center .

2. Asigne usuarios a un grupo y, a continuación, asigne el acceso de inicio de sesión único al grupo.

Para conocer las instrucciones, consulte [Add groups](#) en la Guía del usuario de AWS IAM Identity Center .

Requisitos previos

Para seguir los procedimientos de esta guía, necesitará un shell o un terminal de línea de comando donde pueda ejecutar los comandos. Los comandos aparecen en listas y van precedidos del símbolo del sistema (\$) y del nombre del directorio actual, si es aplicable.

```
~/eb-project$ this is a command  
this is output
```

En Linux y macOS, puede utilizar el administrador de shell y paquetes preferido. En Windows, puede [instalar el subsistema de Windows para Linux](#) y obtener una versión de Ubuntu y Bash integrada en Windows.

CLI DE EB

En este tutorial también se utiliza la interfaz de línea de comandos de Elastic Beanstalk (CLI de EB). Para obtener detalles sobre la instalación y configuración de la CLI de EB, consulte [Instalación de la CLI de EB](#) y [Configuración de la CLI de EB](#).

.NET Core en Linux

[Si no tiene el SDK de .NET instalado en su máquina local, puede instalarlo siguiendo el enlace Descargar .NET que aparece en el sitio web de documentación de .NET.](#)

Compruebe la instalación del SDK de .NET ejecutando el siguiente comando.

```
~$ dotnet --info
```

Paso 1: Cree una aplicación .NET Core para Linux

Cree el directorio del proyecto.

```
~$ mkdir eb-dotnetcore
~$ cd eb-dotnetcore
```

A continuación, cree un ejemplo de aplicación Hello World ejecutando los siguientes comandos.

```
~/eb-dotnetcore$ dotnet new web --name HelloElasticBeanstalk
~/eb-dotnetcore$ cd HelloElasticBeanstalk
```

Paso 2: Ejecute la aplicación localmente

Ejecute el siguiente comando para ejecutar la aplicación localmente.

```
~/eb-dotnetcore/HelloElasticBeasntalk$ dotnet run
```

El resultado debería tener un aspecto similar al siguiente texto.

```
Building...
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: https://localhost:7294
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5052
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
```

```
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
```

Note

El dotnet comando selecciona un puerto al azar al ejecutar la aplicación localmente. En este ejemplo, el puerto es 5052. Al implementar la aplicación en el entorno de Elastic Beanstalk, la aplicación se ejecutará en el puerto 5000.

Introduzca la dirección URL `http://localhost:port` en su navegador web. Para este ejemplo específico, el comando es `http://localhost:5052`. El navegador web debería mostrar «¡Hola mundo!».

Paso 3: Implemente su aplicación.NET Core en Linux con la CLI de EB

Ejecute los siguientes comandos para crear un entorno de Elastic Beanstalk para esta aplicación.

Para crear un entorno e implementar su aplicación.NET Core en Linux

1. Compila y publica tu aplicación en una carpeta para desplegarla en el entorno de Elastic Beanstalk que vas a crear.

```
~$ cd eb-dotnetcore/HelloElasticBeanstalk
~/eb-dotnetcore/HelloElasticBeanstalk$ dotnet publish -o site
```

2. Navegue hasta el directorio del sitio en el que acaba de publicar la aplicación.

```
~/eb-dotnetcore/HelloElasticBeanstalk$ cd site
```

3. Inicialice el repositorio de la CLI de EB con el comando `eb init`.

Tenga en cuenta los siguientes detalles relacionados con la versión de la rama de la plataforma que especifique en el comando:

- Sustituya `x.y.z` el siguiente comando por la versión más reciente de la rama de plataforma .NET 6 en AL2023.
- Para encontrar la versión más reciente de la rama de plataforma, consulte [.NET Core en plataformas compatibles con Linux](#) en la guía de AWS Elastic Beanstalk plataformas.

- Un ejemplo del nombre de una pila de soluciones que incluye el número de versión es `64bit-amazon-linux-2023-v3.1.1-running-.net-6`. En este ejemplo, la versión en rama es la 3.1.1.

```
~eb-dotnetcore/HelloElasticBeanstalk/site$ eb init -p 64bit-amazon-linux-2023-  
vx.y.z-running-.net-6 dotnetcore-tutorial --region us-east-2  
Application dotnetcore-tutorial has been created.
```

Este comando crea una aplicación con un nombre `dotnetcore-tutorial` y configura el repositorio local para crear entornos con la versión de plataforma .NET Core para Linux especificada en el comando.

4. (Opcional) Ejecute de nuevo `eb init` para configurar un par de claves predeterminadas de forma que pueda usar SSH para conectarse a la instancia EC2 donde se ejecuta la aplicación.

```
~eb-dotnetcore/HelloElasticBeanstalk/site$ eb init  
Do you want to set up SSH for your instances?  
(y/n): y  
Select a keypair.  
1) my-keypair  
2) [ Create new KeyPair ]
```

Seleccione un par de claves si ya tiene uno o siga las instrucciones para crear uno. Si no ve el símbolo del sistema o más adelante necesita cambiar la configuración, ejecute `eb init -i`.

5. Cree un entorno e implemente la aplicación en él con `eb create`. Elastic Beanstalk crea automáticamente un archivo zip para la aplicación y lo inicia en el puerto 5000.

a

```
~eb-dotnetcore/HelloElasticBeanstalk/site$ eb create dotnet-tutorial
```

Elastic Beanstalk tarda unos cinco minutos en crear el entorno.

Paso 4: Ejecute la aplicación en Elastic Beanstalk

Cuando finalice el proceso de creación del entorno, abra su sitio web con. `eb open`

```
~eb-dotnetcore/HelloElasticBeanstalk/site$ eb open
```

¡Enhorabuena! ¡Ha implementado una aplicación .NET Core en Linux con Elastic Beanstalk! Se abre una ventana del navegador con el nombre de dominio creado para su aplicación.

Paso 5: Eliminar

Puede cerrar su entorno cuando termine de trabajar con la aplicación. Elastic Beanstalk AWS cancela todos los recursos asociados a su entorno.

Para finalizar el entorno de Elastic Beanstalk con la CLI de EB, ejecute el siguiente comando.

```
~eb-dotnetcore/HelloElasticBeanstalk/site$ eb terminate
```

AWS recursos para su aplicación

Acabas de crear una aplicación de instancia única. Sirve como una aplicación de muestra sencilla con una sola instancia EC2, por lo que no requiere balanceo de carga ni escalado automático. Para las aplicaciones de instancia única, Elastic Beanstalk crea los siguientes recursos: AWS

- **EC2 instance (Instancia EC2):** máquina virtual de Amazon EC2 configurada para ejecutar aplicaciones web en la plataforma que elija.

Cada plataforma ejecuta un conjunto distinto de software, archivos de configuración y scripts compatibles con una determinada versión de lenguaje, marco y contenedor web (o una combinación de ellos). La mayoría de las plataformas utilizan Apache o nginx como un proxy inverso que procesa el tráfico web delante de la aplicación web, reenvía las solicitudes a esta, administra los recursos estáticos y genera registros de acceso y errores.

- **Instance security group (Grupo de seguridad de la instancia):** grupo de seguridad de Amazon EC2 configurado para permitir el tráfico entrante en el puerto 80. Este recurso permite que el tráfico HTTP procedente del balanceador de carga llegue a la instancia EC2 en la que se ejecuta la aplicación web. De forma predeterminada, el tráfico no está permitido en otros puertos.
- **Bucket de Amazon S3:** ubicación de almacenamiento para el código fuente, los registros y otros artefactos que se crean al utilizar Elastic Beanstalk.
- **CloudWatch Alarmas de Amazon:** dos CloudWatch alarmas que monitorean la carga de las instancias de su entorno y se activan si la carga es demasiado alta o demasiado baja. Cuando se activa una alarma, en respuesta, el grupo de Auto Scaling aumenta o reduce los recursos.

- **AWS CloudFormation pila:** Elastic AWS CloudFormation Beanstalk se utiliza para lanzar los recursos de su entorno y propagar los cambios de configuración. Los recursos se definen en una plantilla que puede verse en la [consola de AWS CloudFormation](#).
- **Nombre de dominio:** nombre de dominio que direcciona el tráfico a la aplicación web con el formato *subdominio.región*.elasticbeanstalk.com.

Elastic Beanstalk administra todos estos recursos. Cuando termina su entorno, Elastic Beanstalk termina todos los recursos que este contiene.

Siguientes pasos

Una vez que disponga de un entorno que ejecute una aplicación, podrá implementar una nueva versión de la aplicación o una aplicación distinta en cualquier momento. La implementación de una nueva versión de la aplicación es una tarea muy rápida, ya que no se requiere aprovisionar ni reiniciar instancias EC2. También puede explorar su nuevo entorno con la consola de Elastic Beanstalk. Para ver los pasos detallados, consulte [Explore su entorno](#) en el capítulo Introducción de esta guía.

Una vez que haya implementado una o dos aplicaciones de muestra y esté listo para empezar a desarrollar y ejecutar aplicaciones de .NET Core en Linux de forma local, consulte [Configuración del entorno de desarrollo .NET Core de Linux](#).

Implemente con la consola de Elastic Beanstalk

También puede usar la consola de Elastic Beanstalk para iniciar la aplicación de muestra. Para ver los pasos detallados, consulte [Crear una aplicación de ejemplo](#) en el capítulo Introducción de esta guía.

Configuración del entorno de desarrollo .NET Core de Linux

Configure un entorno de desarrollo de .NET para probar la aplicación de forma local antes de implementarla en AWS Elastic Beanstalk. En este tema, se explican los pasos de configuración del entorno de desarrollo y se incluyen enlaces a páginas de instalación donde encontrará herramientas útiles.

Para ver procedimientos de configuración y herramientas comunes que se utilizan en todos los lenguajes, consulte [Configuración de su máquina de desarrollo para su uso con Elastic Beanstalk](#).

Secciones

- [Instalación del SDK de .NET Core](#)
- [Instalación de un IDE](#)
- [Instalación de AWS Toolkit for Visual Studio](#)

Instalación del SDK de .NET Core

Puede usar el SDK de .NET Core para desarrollar aplicaciones que se ejecutan en Linux.

Consulte la [página de descargas de .NET](#) para descargar e instalar el SDK de .NET Core.

Instalación de un IDE

Los entornos de desarrollo integrado (IDE) proporcionan un intervalo de características que facilitan el desarrollo de aplicaciones. Si no ha usado un IDE para el desarrollo en .NET, pruebe Visual Studio Community para empezar.

Consulte la página [Visual Studio Community](#) para descargar e instalar Visual Studio Community.

Instalación de AWS Toolkit for Visual Studio

[AWS Toolkit for Visual Studio](#) es un complemento de código abierto para el IDE de Visual Studio que facilita a los desarrolladores el desarrollo, la depuración y la implementación de aplicaciones .NET mediante AWS. Consulte la [página principal de Toolkit for Visual Studio](#) para consultar las instrucciones de instalación.

.NET Core en la plataforma Linux

La plataforma AWS Elastic Beanstalk .NET Core en Linux es un conjunto de [versiones de plataforma](#) para aplicaciones .NET Core que se ejecutan en el sistema operativo Linux.

Para obtener más información sobre las diversas formas en las que puede ampliar una plataforma Elastic Beanstalk basada en Linux, consulte [the section called “Ampliación de plataformas Linux”](#). A continuación, se muestran algunas consideraciones específicas de la plataforma.

Introducción a la plataforma .NET Core en Linux

Servidor proxy

Las plataforma .NET Core de Elastic Beanstalk incluye un proxy inverso que reenvía las solicitudes a su aplicación. De forma predeterminada, Elastic Beanstalk utiliza [nginx](#) como servidor proxy. Puede

elegir no utilizar ningún servidor proxy y configurar [Kestrel](#) como su servidor web. Kestrel se incluye de forma predeterminada en las plantillas de proyecto ASP.NET Core.

Estructura de la aplicación

Puede publicar aplicaciones dependientes del tiempo de ejecución que utilicen el tiempo de ejecución de .NET Core proporcionado por Elastic Beanstalk. También puede publicar aplicaciones autónomas que incluyan el tiempo de ejecución de .NET Core y las dependencias de la aplicación en el paquete de código fuente. Para obtener más información, consulte [the section called “Agrupación de aplicaciones”](#).

Configuración de la plataforma

Para configurar los procesos que se ejecutan en las instancias de servidor en su entorno, incluya un [Profile](#) opcional en su paquete de código fuente. Se requiere un Profile si tiene más de una aplicación en su paquete de código fuente.

Le recomendamos que proporcione siempre un Profile en el paquete de código fuente con su aplicación. De esta forma, puede controlar con precisión qué procesos ejecuta Elastic Beanstalk para su aplicación.

En la consola de Elastic Beanstalk hay opciones de configuración disponibles para [modificar la configuración de un entorno en ejecución](#). Para evitar perder la configuración del entorno cuando lo termina, puede usar las [configuraciones guardadas](#) para guardar la configuración y aplicarla posteriormente a otro entorno.

Para guardar la configuración en el código fuente, puede incluir [archivos de configuración](#). Los valores de configuración de los archivos de configuración se aplican cada vez que crea un entorno o que implementa la aplicación. También puede usar archivos de configuración para instalar paquetes, ejecutar scripts y llevar a cabo otras operaciones de personalización de instancias durante las implementaciones.

La configuración aplicada en la consola de Elastic Beanstalk anula la misma configuración en los archivos de configuración, si existe. Esto le permite tener la configuración predeterminada en los archivos de configuración y anularla con la configuración específica del entorno en la consola. Para obtener más información acerca de la prioridad y otros métodos para cambiar valores de configuración, consulte [Opciones de configuración](#).

Configuración de su entorno de .NET Core en Linux

La configuración de la plataforma .NET Core en Linux le permite ajustar el comportamiento de las instancias de Amazon EC2. Puede editar la configuración de instancia de Amazon EC2 del entorno de Elastic Beanstalk mediante la consola de Elastic Beanstalk.

Utilice la consola de Elastic Beanstalk para habilitar la rotación de registros en Amazon S3 y configurar variables que la aplicación pueda leer desde el entorno.

Para configurar su entorno de .NET Core en Linux mediante la consola de Elastic Beanstalk

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Actualizaciones, supervisión y registro, seleccione Editar.

Log Options (Opciones de registro)

La sección Log Options (Opciones de registro) tiene dos valores:

- Instance profile (Perfil de instancia): especifica el perfil de instancia que tiene permiso para obtener acceso al bucket de Amazon S3 asociado con la aplicación.
- Enable log file rotation to Amazon S3 (Habilitar la rotación de archivos de registro para Amazon S3): especifica si los archivos de registro de las instancias de Amazon EC2 de la aplicación se copian en el bucket de Amazon S3 asociado a la aplicación.

Propiedades del entorno

La sección Environment Properties (Propiedades del entorno) le permite especificar opciones de configuración del entorno en las instancias Amazon EC2 que ejecutan su aplicación. Las propiedades del entorno se pasan como pares de clave-valor a la aplicación.

Dentro del entorno de .NET Core en Linux que se ejecuta en Elastic Beanstalk, es posible acceder a las variables de entorno mediante `Environment.GetEnvironmentVariable("variable-name")`. Por ejemplo, podría leer una propiedad denominada `API_ENDPOINT` en una variable con el siguiente código.

```
string endpoint = Environment.GetEnvironmentVariable("API_ENDPOINT");
```

Para obtener más información, consulte [Propiedades del entorno y otras opciones del software](#).

.NET Core en la configuración de espacio de nombres de Linux

Puede usar un [archivo de configuración](#) para definir opciones de configuración y realizar otras tareas de configuración en las instancias durante las implementaciones. Las opciones de configuración se pueden definir a través del servicio de Elastic Beanstalk o la plataforma que utilice y están organizadas por espacios de nombres.

La plataforma .NET Core en Linux admite opciones en el siguiente espacio de nombres, además de las [opciones admitidas para todos los entornos Elastic Beanstalk](#):

- `aws:elasticbeanstalk:environment:proxy`: elija utilizar `nginx` o ningún servidor proxy. Los valores válidos son `nginx` o `none`.

En el siguiente archivo de configuración de ejemplo se muestra el uso de las opciones de configuración específicas de .NET Core en Linux.

Example `.ebextensions/proxy-settings.config`

```
option_settings:  
  aws:elasticbeanstalk:environment:proxy:  
    ProxyServer: none
```

Elastic Beanstalk cuenta con numerosas opciones de configuración para personalizar el entorno. Además de los archivos de configuración, también puede definir opciones en la consola, configuraciones guardadas, la CLI de EB o la AWS CLI. Para obtener más información, consulte [Opciones de configuración](#).

Agrupación de aplicaciones para la plataforma .NET Core en Linux

Puede ejecutar aplicaciones .NET Core que dependen del tiempo de ejecución y autónomas en AWS Elastic Beanstalk.

Una aplicación dependiente del tiempo de ejecución utiliza un tiempo de ejecución .NET Core que Elastic Beanstalk proporciona para ejecutar la aplicación. Elastic Beanstalk utiliza el archivo `runtimeconfig.json` del paquete de origen para determinar el tiempo de ejecución que se va a utilizar para la aplicación. Elastic Beanstalk elige el último tiempo de ejecución compatible disponible para su aplicación.

Una aplicación autónoma incluye el tiempo de ejecución de .NET Core, su aplicación y sus dependencias. Para utilizar una versión del tiempo de ejecución de .NET Core que Elastic Beanstalk no incluye en sus plataformas, proporcione una aplicación autónoma.

Ejemplos

Puede compilar aplicaciones autónomas y dependientes del tiempo de ejecución con el comando `dotnet publish`. Para obtener más información sobre la publicación de aplicaciones de .NET Core, consulte [Información general sobre la publicación de aplicaciones de .NET Core](#) en la documentación de .NET Core.

La estructura de archivos de ejemplo que se muestra a continuación, define una sola aplicación que utiliza un tiempo de ejecución de .NET Core que Elastic Beanstalk proporciona.

```
### appsettings.Development.json
### appsettings.json
### dotnetcoreapp.deps.json
### dotnetcoreapp.dll
### dotnetcoreapp.pdb
### dotnetcoreapp.runtimeconfig.json
### web.config
### Procfile
### .ebextensions
### .platform
```

Puede incluir varias aplicaciones en su paquete de código fuente. En el siguiente ejemplo se definen dos aplicaciones para ejecutarse en el mismo servidor web. Para ejecutar varias aplicaciones, debe incluir un [Procfile](#) en el paquete de código fuente. Para obtener una aplicación de ejemplo completa, consulte [dotnet-core-linux-multiple-apps.zip](#).

```
### DotnetMultipleApp1
# ### Amazon.Extensions.Configuration.SystemsManager.dll
# ### appsettings.Development.json
# ### appsettings.json
```



```
# ### AWSSDK.Core.dll
# ### AWSSDK.Extensions.NETCore.Setup.dll
# ### AWSSDK.SimpleSystemsManagement.dll
# ### DotnetMultipleApp1.deps.json
# ### DotnetMultipleApp1.dll
# ### DotnetMultipleApp1.pdb
# ### DotnetMultipleApp1.runtimeconfig.json
# ### Microsoft.Extensions.PlatformAbstractions.dll
# ### Newtonsoft.Json.dll
# ### web.config
### DotnetMultipleApp2
# ### Amazon.Extensions.Configuration.SystemsManager.dll
# ### appsettings.Development.json
# ### appsettings.json
# ### AWSSDK.Core.dll
# ### AWSSDK.Extensions.NETCore.Setup.dll
# ### AWSSDK.SimpleSystemsManagement.dll
# ### DotnetMultipleApp2.deps.json
# ### DotnetMultipleApp2.dll
# ### DotnetMultipleApp2.pdb
# ### DotnetMultipleApp2.runtimeconfig.json
# ### Microsoft.Extensions.PlatformAbstractions.dll
# ### Newtonsoft.Json.dll
# ### web.config
### Procfile
### .ebextensions
### .platform
```

Uso de un archivo Procfile para configurar su entorno de .NET Core en Linux

Para ejecutar varias aplicaciones en el mismo servidor web, debe incluir un Procfile en el paquete de código fuente que indique a Elastic Beanstalk qué aplicaciones debe ejecutar.

Le recomendamos que proporcione siempre un Procfile en el paquete de código fuente con su aplicación. De esta forma, puede controlar con precisión qué procesos ejecuta Elastic Beanstalk para su aplicación y qué argumentos reciben estos procesos.

En el ejemplo siguiente se utiliza un Procfile para especificar dos aplicaciones de Elastic Beanstalk para ejecutar en el mismo servidor web.

Example Procfile

```
web: dotnet ./dotnet-core-app1/dotnetcoreapp1.dll
```

```
web2: dotnet ./dotnet-core-app2/dotnetcoreapp2.dll
```

Para obtener más información sobre cómo escribir y utilizar un Procfile, expanda la sección Buildfile y Procfile en [the section called “Ampliación de plataformas Linux”](#).

Configuración del servidor proxy para su entorno de .NET Core en Linux

AWS Elastic Beanstalk utiliza [nginx](#) como proxy inverso para transmitir solicitudes a la aplicación. Elastic Beanstalk proporciona una configuración de nginx predeterminada que puede ampliar o anular por completo con su propia configuración.

De forma predeterminada, Elastic Beanstalk configura el proxy nginx para reenviar las solicitudes a la aplicación en el puerto 5000. Puede invalidar el puerto predeterminado estableciendo la [propiedad del entorno](#) PORT en el puerto en el que desea que la aplicación principal esté a la escucha.

Note

El puerto en que su aplicación está a la escucha no afecta al puerto en el que el servidor nginx está a la escucha para recibir las solicitudes del equilibrador de carga.

Configuración del servidor proxy en la versión de su plataforma

Todas las plataformas AL2023/AL2 son compatibles con una característica de configuración de proxy uniforme. Para obtener más información sobre la configuración del servidor proxy en las versiones de su plataforma que ejecutan AL2023/AL2, expanda la sección Configuración del proxy inverso de [the section called “Ampliación de plataformas Linux”](#).

El siguiente archivo de configuración de ejemplo amplía la configuración nginx de su entorno. La configuración dirige solicitudes a `/api` a una segunda aplicación web que escucha en el puerto 5200 del servidor web. De forma predeterminada, Elastic Beanstalk reenvía las solicitudes a una sola aplicación que escucha en el puerto 5000.

Example `01_custom.conf`

```
location /api {
    proxy_pass          http://127.0.0.1:5200;
    proxy_http_version 1.1;
```

```
proxy_set_header Upgrade $http_upgrade;
proxy_set_header Connection $http_connection;
proxy_set_header Host $host;
proxy_cache_bypass $http_upgrade;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header X-Forwarded-Proto $scheme;
}
```

AWS Toolkit for Visual Studio: trabajar con .Net Core

AWS Toolkit for Visual Studio es un complemento para el IDE de Visual Studio. Con el kit de herramientas, puede implementar y administrar aplicaciones en Elastic Beanstalk mientras trabaja en su entorno de Visual Studio.

En este tema se muestra cómo puede realizar las siguientes tareas mediante AWS Toolkit for Visual Studio:

- Cree una aplicación web ASP.NET Core con una plantilla de Visual Studio.
- Cree un entorno Elastic Beanstalk Amazon Linux.
- Implementar la aplicación web ASP.NET Core en el nuevo entorno de Amazon Linux.

En este tema también se explora cómo puede utilizar AWS Toolkit for Visual Studio para administrar los entornos de aplicaciones de Elastic Beanstalk y monitorear el estado de la aplicación.

Secciones

- [Requisitos previos](#)
- [Crear un nuevo proyecto de aplicación](#)
- [Cree un entorno Elastic Beanstalk e implemente su aplicación](#)
- [Terminación de un entorno](#)
- [Administración de entornos de aplicaciones de Elastic Beanstalk](#)
- [Monitorización del estado de mantenimiento de la aplicación](#)

Requisitos previos

Antes de empezar este tutorial, necesitará instalar AWS Toolkit for Visual Studio. Para obtener instrucciones, consulte [Instalación de AWS Toolkit for Visual Studio](#).

Si no ha utilizado nunca este conjunto de herramientas, lo primero que tendrá que hacer después de instalarlo es registrar sus credenciales de AWS con el conjunto de herramientas. Para obtener más información al respecto, consulte [Proporcionar credenciales de AWS](#).

Crear un nuevo proyecto de aplicación

Si no tiene un proyecto de aplicación.NET Core en Visual Studio, puede crear uno fácilmente utilizando una de las plantillas de proyecto de Visual Studio.

Para crear un nuevo proyecto de aplicación web ASP.NET

1. En Visual Studio, desde el menú File (Archivo), elija New (Nuevo) y, a continuación, elija Project (Proyecto).
2. En el cuadro de diálogo Create a new project (Crear un nuevo proyecto), seleccione C#, seleccione Linux, y, a continuación, seleccione Nube.
3. En la lista de plantillas de proyecto que aparece, seleccione ASP.NET Core Web Application (Aplicación web de ASP.NET Core), y, a continuación, seleccione Next (Siguiente).

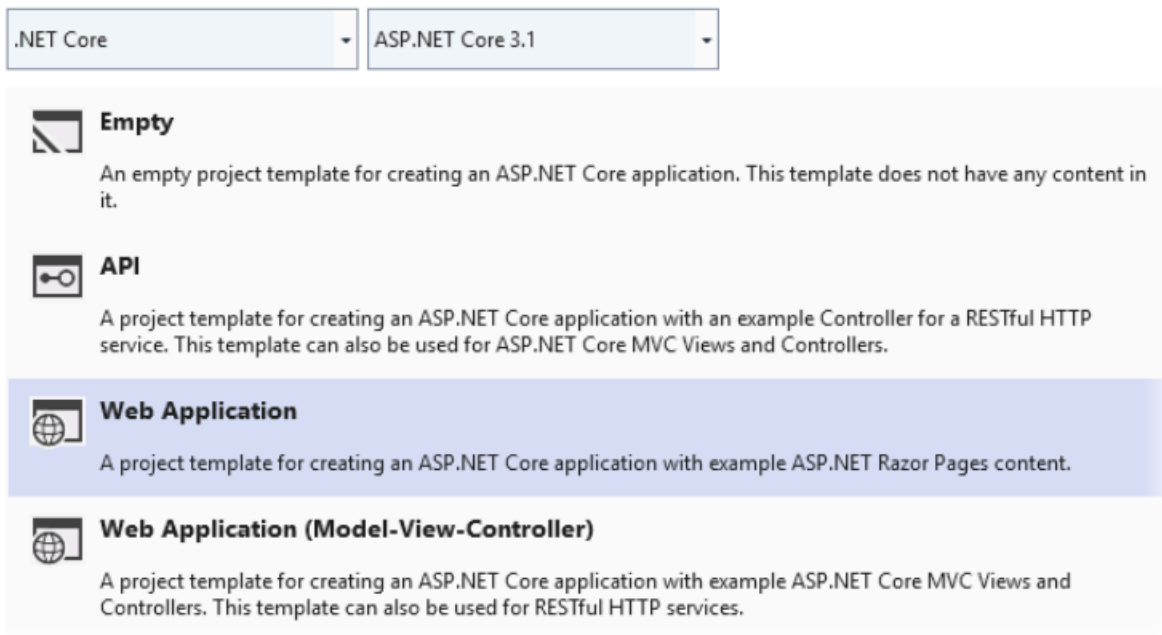
Note

Si no ve ASP.NET Core Web Application (Aplicación web de ASP.NET Core) en las plantillas de proyecto, puede instalarlo en Visual Studio.

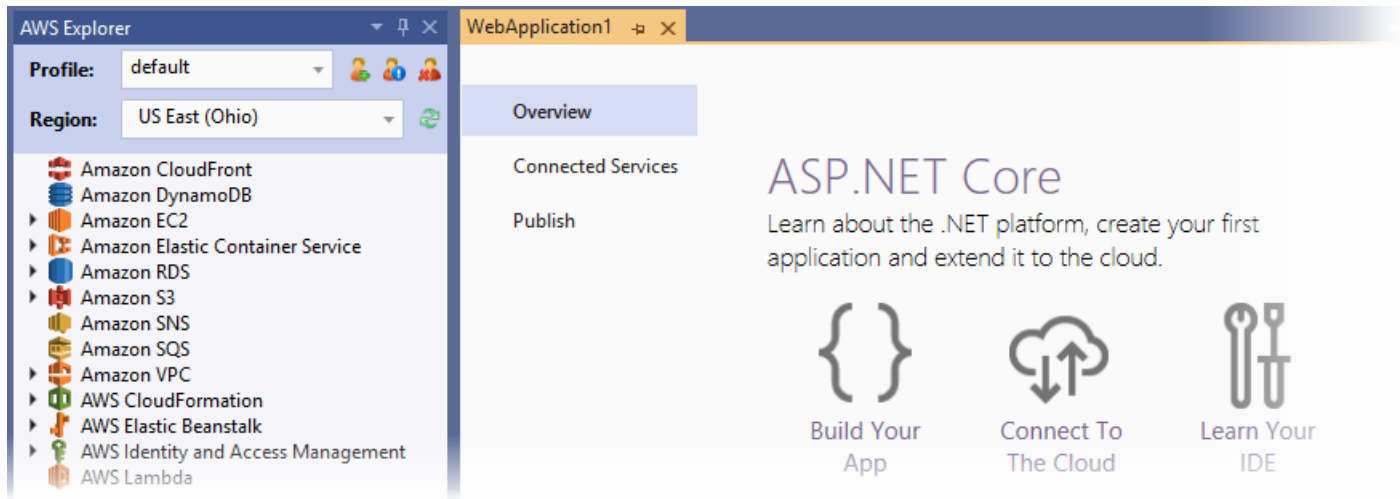
1. Desplácese hasta la parte inferior de la lista de plantillas y seleccione el enlace Install more tools and features (Instalar más herramientas y características) que se encuentra debajo de la lista de plantillas.
 2. Si se le pide que permita que la aplicación de Visual Studio realice cambios en el dispositivo, seleccione Yes (Sí).
 3. Elija la pestaña Workloads (Cargas de trabajo) y, a continuación, seleccione ASP.NET and web development. (ASP.NET y desarrollo web).
 4. Seleccione el botón Modify (Modificar). El Visual Studio Installer (Instalador de Visual Studio) instala la plantilla del proyecto.
 5. Cuando termine la instalación, salga del panel para volver a donde lo dejó en Visual Studio.
4. En el cuadro de diálogo Configure your new project (Configurar el nuevo proyecto), escriba un Project name (Nombre del proyecto). El Solution name (Nombre de la solución) adopta de forma predeterminada el nombre del proyecto. A continuación, elija Create (Crear).

5. En el cuadro de diálogo Create a new ASP.NET Core web application (Crear una nueva aplicación web de ASP.NET Core), seleccione .NET Core y, a continuación, seleccione ASP.NET Core 3.1. En la lista de tipos de aplicación mostrada, seleccione Web Application (Aplicación web) y después seleccione el botón Create (Crear).

Create a new ASP.NET Core web application



Visual Studio muestra el cuadro de diálogo Creating Project (Creando el proyecto) mientras se crea la aplicación. Después de que Visual Studio complete la generación de la aplicación, se muestra un panel con el nombre de la aplicación.

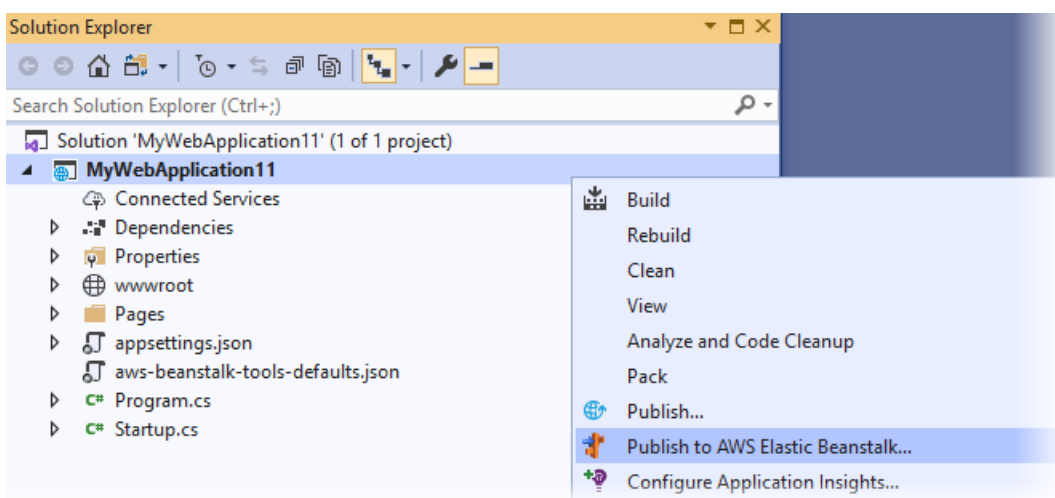


Cree un entorno Elastic Beanstalk e implemente su aplicación

En esta sección se describe cómo crear un entorno de Elastic Beanstalk para la aplicación e implementar la aplicación en ese entorno.

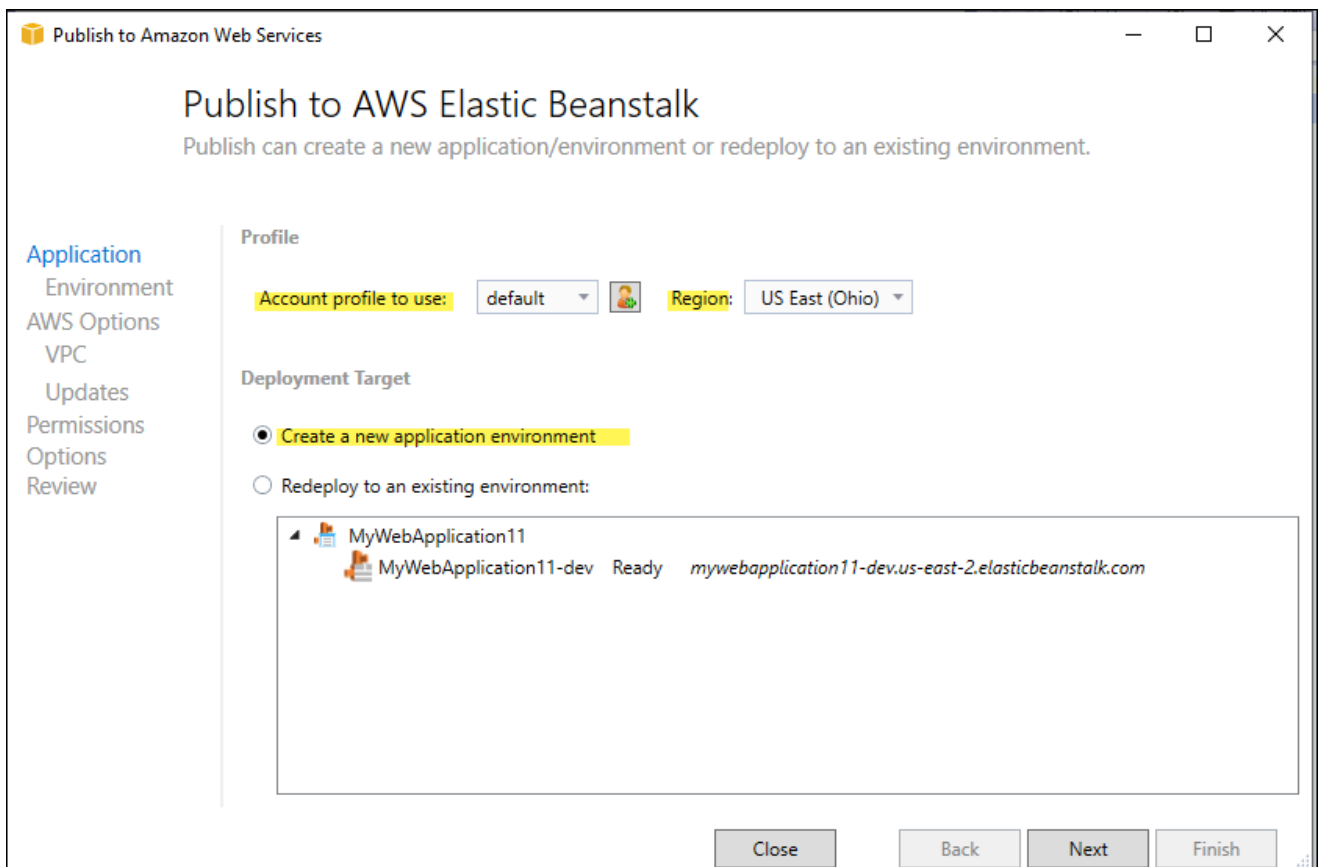
Para crear un nuevo entorno e implementar la aplicación

1. En Visual Studio, seleccione View (Ver), y, a continuación, Solution Explorer (Explorador de soluciones).
2. En Solution Explorer (Explorador de soluciones), abra el menú contextual (haga clic con el botón derecho) de la aplicación y, a continuación, seleccione Publish to (Publicar en)AWS Elastic Beanstalk.




3. En el asistente Publicar en AWS Elastic Beanstalk, ingrese la información de la cuenta.

- a. En Account profile to use (Perfil de cuenta que se va a usar), seleccione su cuenta predeterminada o elija el icono Add another account (Agregar otra cuenta) para especificar la información de una cuenta nueva.
- b. En Region (Región), seleccione la región en la que desea implementar la aplicación. Para obtener información acerca de las regiones de AWS disponibles, consulte [Puntos de enlace y cuotas de AWS Elastic Beanstalk](#) en la Referencia general de AWS. Si selecciona una región que no es compatible con Elastic Beanstalk, la opción para implementar en Elastic Beanstalk no estará disponible.
- c. Seleccione Create a new application environment (Crear un nuevo entorno de aplicación), y, a continuación, elija Next (Siguiente).



4. En el cuadro de diálogo Application Environment (Entorno de aplicación), escriba los datos del nuevo entorno de aplicación.
5. En el siguiente cuadro de diálogo de opciones de AWS, defina las opciones de Amazon EC2 y otras opciones relacionadas con AWS para la aplicación implementada.
 - a. En Container type (Tipo de contenedor), seleccione 64bit Amazon Linux 2 v <n.n.n> running .NET Core (Amazon Linux 2 v de 64 bits <n.n.n.> ejecutando .NET Core).

 Note

Le recomendamos que seleccione la versión actual de la plataforma de Linux. Esta versión contiene las correcciones de errores y seguridad más recientes que se incluyen en nuestra Amazon Machine Image (AMI) más reciente.

- b. En Instance Type (Tipo de instancia), seleccione t2.micro. (Al elegir un tipo de microinstancia se minimiza el costo asociado con la ejecución de la instancia).
- c. En Key pair (Par de claves), seleccione Create new key pair (Crear nuevo par de claves). Escriba un nombre para el nuevo par de claves y, a continuación, elija OK (Aceptar). (En este ejemplo, usamos **myuseastkeypair**). El par de claves permite el acceso de Escritorio remoto a las instancias de Amazon EC2. Para obtener más información sobre los pares de claves de Amazon EC2, consulte [Uso de credenciales](#) en la Guía del usuario de Amazon Elastic Compute Cloud.
- d. Si se trata de una aplicación sencilla con poco tráfico, seleccione Single instance environment (Entorno de una sola instancia). Para obtener más información, consultar [Tipos de entornos](#)
- e. Seleccione Siguiente.

AWS
Set Amazon EC2 and other AWS-related options for the deployed application.

Amazon EC2 Launch Configuration

Container type *: 64bit Amazon Linux 2 v2.0.2 running .NET Core

Instance type *: t2.micro Key pair *: myuseastkeypair

Use custom AMI:

Use non-default VPC Single instance environment Enable Rolling Deployments

Load balancer type

Choose the type of load balancer to use when not creating a single instance environment.

Application
An Application Load Balancer makes routing decisions at the application layer (HTTP/HTTPS).

Relational Database Access

Select the Amazon RDS security groups to be modified to permit access from the EC2 instance(s) hosting your application.

Close Back Next Finish

Para obtener más información sobre las opciones de AWS que no se utilizan en este ejemplo, tenga en cuenta las siguientes páginas:

- Para Use custom AMI (Usar AMI personalizada), consulte [Uso de una imagen de máquina de Amazon \(AMI\) personalizada](#).
- Si no selecciona Single instance environment (Entorno de una sola instancia), tiene que elegir un valor en Load balance type (Tipo de balanceo de carga). Para obtener más información, consulte [Balanceador de carga del entorno de Elastic Beanstalk](#).
- Elastic Beanstalk utiliza la configuración predeterminada de [Amazon VPC](#) (Amazon Virtual Private Cloud) si no seleccionó Use non-default VPC (Usar VPC no predeterminada). Para obtener más información, consulte [Uso de Elastic Beanstalk con Amazon VPC](#).
- Al elegir la opción Enable Rolling Deployments (Habilitar implementaciones acumuladas) se divide una implementación en lotes para evitar posibles tiempos de inactividad durante las implementaciones. Para obtener más información, consulte [Implementación de aplicaciones en entornos Elastic Beanstalk](#).

- Al elegir la opción Relational Database Access (Acceso a base de datos relacional) se conecta el entorno de Elastic Beanstalk a una base de datos de Amazon RDS creada previamente con grupos de seguridad de base de datos de Amazon RDS. Para obtener más información, consulte [Control de acceso con grupos de seguridad](#) en la Guía de usuario de Amazon RDS.
6. Seleccione Next (Siguiente) en el cuadro de diálogo Permissions (Permisos).
 7. Seleccione Next (Siguiente) en el cuadro de diálogo Applications Options (Opciones de aplicaciones).
 8. Revise las opciones de implementación. Después de comprobar que la configuración es correcta, seleccione Deploy (Implementar).

Su aplicación web ASP.NET Core se exporta como un archivo de implementación web. A continuación, este archivo se carga en Amazon S3 y se registra como una nueva versión de la aplicación con Elastic Beanstalk. La característica de implementación de Elastic Beanstalk monitorea el entorno hasta que esté disponible con el código que acaba de implementar. La pestaña Status (Estado) correspondiente a su entorno se encuentra en la pestaña Env:<nombre del entorno>. Cuando el estado cambie a Environment is healthy (El entorno está en buen estado), puede seleccionar la dirección URL para lanzar la aplicación web.

MyWebApplication11 Env: MyWebApplication11-dev

Apply Changes Connect to Instance Restart App Environment Refresh

URL: <http://mywebapplication11-dev.us-east-2.elasticbeanstalk.com/>

Application: MyWebApplication11 Running Version: v20200721020104

Container Type: 64bit Amazon Linux 2 v1.0.0 running .NET Core Created: 7/21/2020 2:54:23 AM

Status: **Environment is healthy** Updated: 7/21/2020 2:56:54 AM

Events	Filter:
Monitoring	
Resources	
AWS X-Ray	
Server	
Notifications	
Container	
Advanced	
Logs	

Event Time	Event Type	Version Label	Event Details
7/21/2020 2:56:54 AM	INFO		Successfully launched environment: MyWebApplication11
7/21/2020 2:56:54 AM	INFO		Application available at mywebapplication11-dev.us-east-
7/21/2020 2:56:45 AM	INFO	v20200721020104	Added EC2 instance 'i-00c5680f13fc6f089' to Auto Scaling
7/21/2020 2:56:45 AM	INFO	v20200721020104	Environment health has been set to GREEN
7/21/2020 2:56:45 AM	INFO	v20200721020104	Adding instance 'i-00c5680f13fc6f089' to your environme
7/21/2020 2:56:18 AM	INFO		Waiting for EC2 instances to launch. This may take a few r
7/21/2020 2:54:58 AM	INFO		Created EIP: 3.14.235.39
7/21/2020 2:54:42 AM	INFO		Created security group named: awseb-e-ffi5z3mn6m-stac
7/21/2020 2:54:24 AM	INFO		Using elasticbeanstalk-us-east-2-164656829171 as Amaz
7/21/2020 2:54:23 AM	INFO		createEnvironment is starting.

Terminación de un entorno

Con la finalidad de evitar que se apliquen cargos por recursos de AWS no utilizados, puede utilizar AWS Toolkit for Visual Studio para terminar un entorno en ejecución.

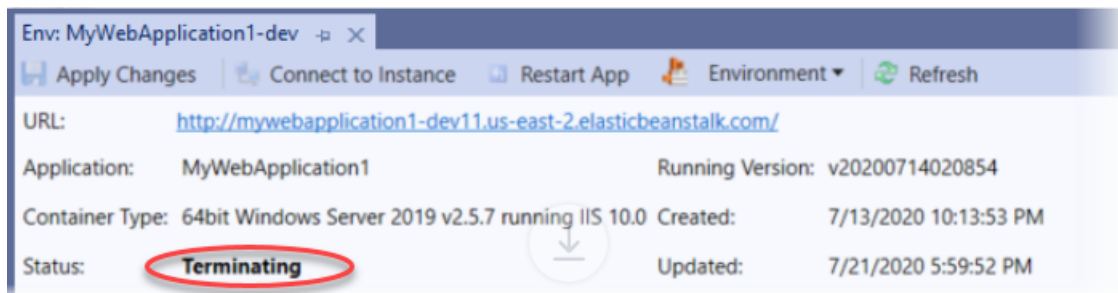
Note

Siempre podrá lanzar más adelante un nuevo entorno con la misma versión.

Para terminar un entorno

1. Expanda el nodo Elastic Beanstalk y el nodo de aplicación. En el AWS Explorer (Explorador de AWS), abra el menú contextual (haga clic con el botón derecho) del entorno de aplicación y seleccione **Terminate Environment** (Terminar el entorno).
2. Cuando se le pregunte, Seleccione **Yes (Sí)** para confirmar que desea terminar el entorno. Elastic Beanstalk tarda unos minutos en terminar los recursos de AWS que se ejecutan en el entorno.

El valor del campo **Status** (Estado) de su entorno en la pestaña **Env: <nombre del entorno>** cambia a **Terminating** (Terminando) y finalmente a **Terminated** (Terminado).



Note

Cuando termine su entorno, el CNAME asociado con el entorno terminado quedará a disposición de cualquier persona que desee utilizarlo.

Administración de entornos de aplicaciones de Elastic Beanstalk

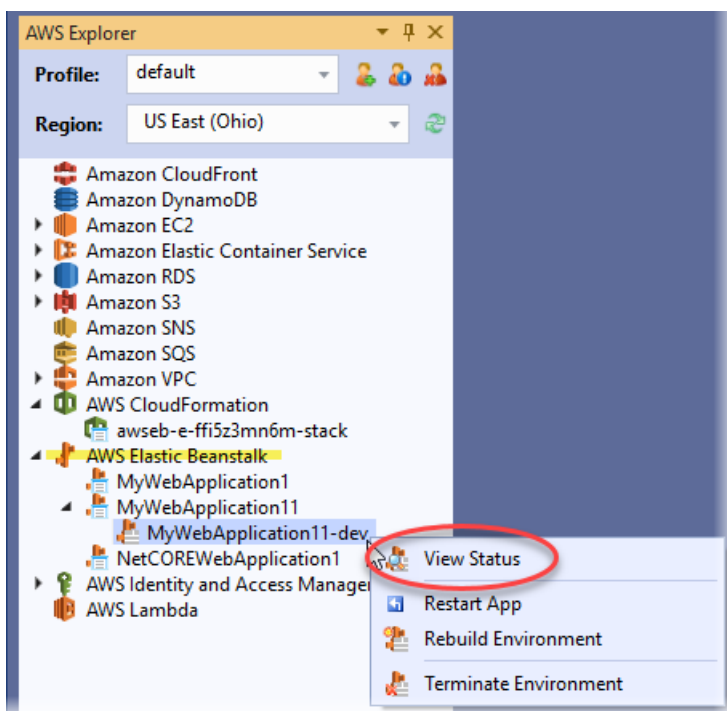
Con AWS Toolkit for Visual Studio y la Management Console de AWS, puede cambiar el aprovisionamiento y la configuración de los recursos de AWS que se utilizan en los entornos de aplicaciones. Para obtener más información sobre la administración de sus entornos de aplicaciones a través de la Management Console de AWS, consulte [Administración de entornos](#). En esta sección se describe la configuración de servicios específica que puede editar en AWS Toolkit for Visual Studio como parte de la configuración del entorno de aplicaciones.

Modificación de las opciones de configuración del entorno

Cuando se implementa la aplicación, Elastic Beanstalk configura varios servicios informáticos conectados en la nube de AWS. Puede controlar cómo se configuran estos servicios individuales mediante AWS Toolkit for Visual Studio.

Para editar la configuración del entorno de una aplicación

1. Vaya a Visual Studio y en el menú File (Archivo), elija AWS Explorer (Explorador de AWS).
2. Expanda el nodo Elastic Beanstalk y el nodo de la aplicación. Abra el menú contextual (haga clic con el botón derecho) de su entorno de aplicación y seleccione View Status (Ver estado).



Ahora, podrá configurar las opciones de los elementos siguientes:

- AWS X-Ray
- Servidor
- Balanceador de carga (solo se aplica a entornos con varias instancias)
- Auto Scaling (solo se aplica a entornos con varias instancias)
- Notificaciones
- Contenedor
- Opciones de configuración avanzadas

Configuración de AWS X-Ray mediante AWS Toolkit for Visual Studio

AWS X-Ray proporciona funciones de rastreo de solicitudes, recopilación de excepciones y creación de perfiles. Con el panel de AWS X-Ray, puede habilitar o desactivar X-Ray para su aplicación. A fin de obtener más información sobre X-Ray, consulte la [Guía para desarrolladores de AWS X-Ray](#).

AWS X-Ray is a service that collects data about requests that your application serves, and provides tools you can use to view, filter, and gain insights into that data to identify issues and opportunities for optimization. For any traced request to your application, you can see detailed information not only about the request and response, but also about calls that your application makes to downstream AWS resources, microservices, databases and HTTP web APIs.

▼ Scorekeep AWS ElasticBeanstalk Environment			
Scorekeep	200	39.0 ms	✓
DynamoDB	200	5.0 ms	✓
DynamoDB	200	5.0 ms	✓
DynamoDB	200	5.0 ms	✓
DynamoDB	200	5.0 ms	✓
# GameModel saveGame	-	14.0 ms	✓
DynamoDB	200	5.0 ms	✓
DynamoDB	200	5.0 ms	✓

Enable AWS X-Ray true

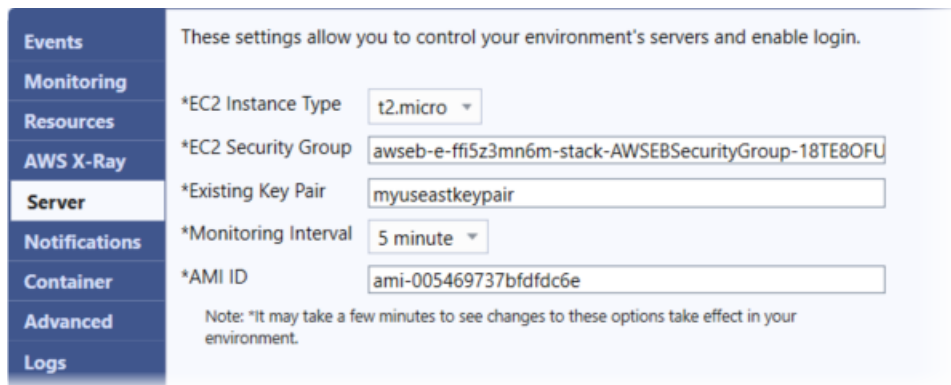
To see your application's service map and traces visit the [AWS X-Ray Console](#).

To learn how to instrument your .NET application visit the [AWS X-Ray SDK for .NET GitHub repository](#).

Configuración de instancias EC2 con AWS Toolkit for Visual Studio

Puede utilizar Amazon Elastic Compute Cloud (Amazon EC2) para lanzar y administrar instancias de servidor en los centros de datos de Amazon. Puede utilizar instancias de servidor Amazon EC2 en cualquier momento, durante el tiempo que las necesite y con cualquier finalidad legal. Las instancias están disponibles en diferentes tamaños y configuraciones. Para obtener más información, consulte [Amazon EC2](#).

Puede editar la configuración de la instancia de Amazon EC2 con la pestaña Server (Servidor) en la pestaña del entorno de la aplicación en AWS Toolkit for Visual Studio.



The screenshot shows the 'Server' configuration page in the AWS Elastic Beanstalk console. On the left is a navigation menu with options: Events, Monitoring, Resources, AWS X-Ray, Server (highlighted), Notifications, Container, Advanced, and Logs. The main content area has a header: 'These settings allow you to control your environment's servers and enable login.' Below this are several configuration fields:

- *EC2 Instance Type: t2.micro
- *EC2 Security Group: awseb-e-ffi5z3mn6m-stack-AWSEBSecurityGroup-18TE8OFU
- *Existing Key Pair: myuseastkeypair
- *Monitoring Interval: 5 minute
- *AMI ID: ami-005469737bfdfdc6e

A note at the bottom states: 'Note: *It may take a few minutes to see changes to these options take effect in your environment.'

Tipos de instancias de Amazon EC2

Instance type (Tipo de instancia) muestra los tipos de instancias disponibles para la aplicación de Elastic Beanstalk. Cambie el tipo de instancia para seleccionar un servidor con las características más adecuadas para su aplicación (entre ellas, el tamaño de la memoria y la potencia de la CPU). Por ejemplo, las aplicaciones de operaciones intensivas de ejecución prolongada pueden necesitar más memoria o CPU.

Para obtener más información acerca de los tipos de instancias de Amazon EC2 disponibles con la aplicación de Elastic Beanstalk, consulte [Tipos de instancias](#) en la Guía del usuario de Amazon Elastic Compute Cloud.

Grupos de seguridad de Amazon EC2

Puede controlar el acceso a la aplicación de Elastic Beanstalk mediante un grupo de seguridad de Amazon EC2. Los grupos de seguridad definen las reglas del firewall para las instancias. Estas reglas especifican qué tráfico procedente de la red se debe entregar a la instancia. Se descarta el resto del tráfico entrante. Puede modificar las reglas de un grupo en cualquier momento. Las nuevas reglas se aplicarán automáticamente a todas las instancias que estén en ejecución y a las que se lancen en el futuro.

Puede especificar qué grupos de seguridad de Amazon EC2 controlan el acceso a la aplicación de Elastic Beanstalk. Para ello, ingrese los nombres de grupos de seguridad específicos de Amazon EC2 (separe varios grupos de seguridad con comas) en el cuadro de texto EC2 Security Groups (Grupos de seguridad de EC2). Puede hacerlo con la Management Console de AWS o con AWS Toolkit for Visual Studio.

Para crear un grupo de seguridad con el AWS Toolkit for Visual Studio

1. Vaya a Visual Studio, en AWS Explorer (Explorador de AWS) expanda el nodo de Amazon EC2 y, a continuación, seleccione Security Groups (Grupos de seguridad).
2. Seleccione Create Security Group (Crear grupo de seguridad) y escriba un nombre y una descripción para el grupo de seguridad.
3. Seleccione OK (Aceptar).

Para obtener más información sobre los grupos de seguridad de Amazon EC2, consulte [Uso de grupos de seguridad](#) en la Guía del usuario de Amazon Elastic Compute Cloud.

Pares de claves de Amazon EC2

Puede iniciar sesión de forma segura en las instancias de Amazon EC2 aprovisionadas por la aplicación de Elastic Beanstalk con un par de claves de Amazon EC2.

Important

Debe crear un par de claves de Amazon EC2 y configurar las instancias de Amazon EC2 aprovisionadas por Elastic Beanstalk para poder acceder a estas instancias. Puede crear el par de claves con el asistente Publish to AWS (Publicar en AWS) en el AWS Toolkit for Visual Studio cuando implemente la aplicación en Elastic Beanstalk. Si desea crear pares de claves adicionales con el conjunto de herramientas, siga los pasos que se indican a continuación. Si lo desea, también puede configurar el par de claves de Amazon EC2 con la [Management Console de AWS](#). Si desea obtener instrucciones acerca de cómo crear un par de claves para Amazon EC2, consulte la [Guía de introducción de Amazon Elastic Compute Cloud](#).

El cuadro de texto Existing Key Pair (Par de claves existente) le permite especificar el nombre de un par de claves de Amazon EC2 que puede utilizar para iniciar sesión de forma segura en las instancias de Amazon EC2 que ejecutan la aplicación de Elastic Beanstalk.

Para especificar el nombre de un par de claves de Amazon EC2

1. Expanda el nodo Amazon EC2 y seleccione Key Pairs (Pares de claves).
2. Seleccione Create Key Pair (Crear par de claves) y escriba el nombre del par de claves.
3. Seleccione OK (Aceptar).

Para obtener más información sobre los pares de claves de Amazon EC2, consulte [Uso de credenciales de Amazon EC2](#) en la Guía del usuario de Amazon Elastic Compute Cloud. Para obtener más información acerca de cómo conectarse a instancias Amazon EC2, consulte

Intervalo de monitorización

De manera predeterminada, solo están habilitadas las métricas básicas de Amazon CloudWatch. Estas devuelven datos en periodos de cinco minutos. Puede habilitar métricas de CloudWatch de un minuto pormenorizada al seleccionar 1 minute (1 minuto) en el Monitoring Interval (Intervalo de monitoreo) de la sección Server (Servidor) en la pestaña Configuration (Configuración) del entorno en el AWS Toolkit for Eclipse.

Note

Se pueden aplicar gastos por las métricas en intervalos de un minuto del servicio Amazon CloudWatch. Consulte [Amazon CloudWatch](#) para obtener más información.

ID de AMI personalizada

Puede anular la AMI predeterminada utilizada en las instancias de Amazon EC2 y sustituirla por la suya personalizada al ingresar el identificador de la AMI personalizada en el cuadro Custom AMI ID (ID de AMI personalizada) de la sección Server (Servidor) en la pestaña Configuration (Configuración) del entorno en el AWS Toolkit for Eclipse.

Important

El uso de una AMI propia es una tarea avanzada que debe llevarse a cabo con precaución. Si necesita una AMI personalizada, le recomendamos que comience con la AMI predeterminada de Elastic Beanstalk y después la modifique. Para darlas por buenas, Elastic Beanstalk espera que las instancias de Amazon EC2 satisfagan una serie de requisitos, entre ellos disponer de un administrador host en ejecución. Si no se cumplen, es posible que el entorno no funcione correctamente.

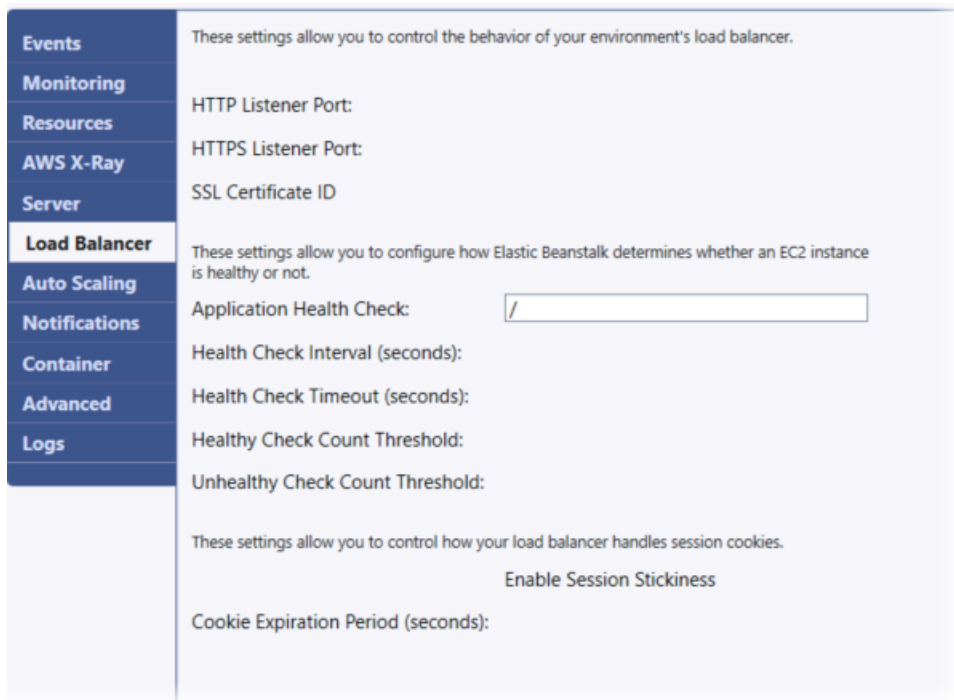
Configuración de Elastic Load Balancing mediante AWS Toolkit for Visual Studio

Elastic Load Balancing es un servicio web de Amazon que le ayuda a mejorar la disponibilidad y escalabilidad de la aplicación. Este servicio le permite distribuir fácilmente las cargas de la aplicación

entre dos o más instancias de Amazon EC2. Elastic Load Balancing mejora la disponibilidad a través de la redundancia adicional proporcionada y admite el aumento del tráfico en la aplicación.

Con Elastic Load Balancing, puede distribuir y equilibrar automáticamente el tráfico entrante de aplicaciones entre todas las instancias en ejecución. También puede agregar nuevas instancias fácilmente cuando se requiera aumentar la capacidad de la aplicación.

Cuando se implementa una aplicación, Elastic Beanstalk aprovisiona automáticamente Elastic Load Balancing. Puede editar la configuración de las instancias de Amazon EC2 del entorno de Elastic Beanstalk con la pestaña Load Balancer (Equilibrador de carga) incluida en la pestaña del entorno de la aplicación en el AWS Toolkit for Visual Studio.



En las siguientes secciones se describen los parámetros de Elastic Load Balancing que puede configurar para la aplicación.

Puertos

El balanceador de carga aprovisionado para atender las solicitudes de la aplicación de Elastic Beanstalk envía solicitudes a las instancias de Amazon EC2 donde se ejecuta la aplicación. El equilibrador de carga aprovisionado puede atender las solicitudes de los puertos HTTP y HTTPS y direccionarlas a las instancias de Amazon EC2 de la aplicación de AWS Elastic Beanstalk. De forma predeterminada, el balanceador de carga atiende las solicitudes en el puerto HTTP. Para que esto funcione, al menos uno de los puertos (HTTP o HTTPS) debe estar activo.



⚠ Important

Asegúrese de que el puerto especificado no está bloqueado; de lo contrario, no podrá conectarse a la aplicación de Elastic Beanstalk.

Control del puerto HTTP

Para desactivar el puerto HTTP, seleccione OFF en HTTP Listener Port (HTTP Puerto Listener).
Para activar el puerto HTTP, seleccione un puerto HTTP (por ejemplo, 80) en la lista.

📘 Note

Para tener acceso al entorno usando un puerto diferente del predeterminado 80, por ejemplo, el puerto 8080, añada un agente de escucha al balanceador de carga existente y configúrelo para que escuche en dicho puerto.

Por ejemplo, si usa la [AWS CLI para los equilibradores de carga clásicos](#), escriba el siguiente comando, reemplazando `LOAD_BALANCER_NAME` por el nombre de su equilibrador de carga para Elastic Beanstalk.

```
aws elb create-load-balancer-listeners --load-balancer-name LOAD_BALANCER_NAME
--listeners "Protocol=HTTP, LoadBalancerPort=8080, InstanceProtocol=HTTP,
InstancePort=80"
```

Por ejemplo, si usa la [AWS CLI para los equilibradores de carga de aplicaciones](#), escriba el siguiente comando, reemplazando `LOAD_BALANCER_ARN` por el ARN de su equilibrador de carga para Elastic Beanstalk.

```
aws elbv2 create-listener --load-balancer-arn LOAD_BALANCER_ARN --protocol HTTP
--port 8080
```

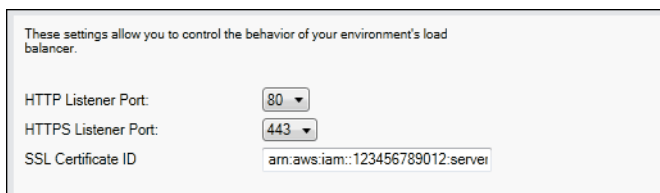
Si quiere que Elastic Beanstalk supervise el entorno, no elimine el agente de escucha del puerto 80.

Control del puerto HTTPS

Elastic Load Balancing admite el protocolo HTTPS/TLS para permitir el cifrado del tráfico en las conexiones cliente con el balanceador de carga. Las conexiones entre el balanceador de carga y las instancias EC2 usan cifrado de texto sin formato. De forma predeterminada, el puerto HTTPS está desactivado.

Para activar el puerto HTTPS

1. Cree un certificado nuevo con AWS Certificate Manager (ACM) o cargue un certificado y una clave en AWS Identity and Access Management (IAM). Para obtener más información sobre cómo solicitar un certificado de ACM, consulte [Solicitar un certificado](#) en la Guía del usuario de AWS Certificate Manager. Para obtener más información sobre la importación de certificados de terceros en ACM, consulte [Importar certificados](#) en la Guía del usuario de AWS Certificate Manager. Si ACM no está [disponible en su región](#), use AWS Identity and Access Management (IAM) para cargar un certificado de terceros. Los servicios de ACM e IAM guardan el certificado y proporcionan un nombre de recurso de Amazon (ARN) para el certificado SSL. Para obtener más información sobre la creación y la carga de certificados en IAM, consulte [Uso de certificados de servidor](#) en la Guía del usuario de IAM.
2. Especifique el puerto HTTPS seleccionando un puerto en HTTPS Listener Port (HTTP Puerto Listener).



The screenshot shows a configuration panel for an Elastic Load Balancing listener. It contains three settings:

- HTTP Listener Port: 80
- HTTPS Listener Port: 443
- SSL Certificate ID: arn:aws:iam::123456789012:server

3. En SSL Certificate ID (ID de certificado de SSL), escriba el nombre de recurso de Amazon (ARN) de su certificado SSL. Por ejemplo, **arn:aws:iam::123456789012:server-certificate/abc/certs/build** o **arn:aws:acm:us-east-2:123456789012:certificate/12345678-12ab-34cd-56ef-12345678**. Utilice el certificado SSL que ha creado o cargado en el paso 1.

Para desactivar el puerto HTTPS, seleccione OFF en HTTPS Listener Port (HTTP Puerto Listener).

Comprobaciones de estado

La definición de comprobación de estado incluye una URL que se utiliza para las consultas del estado de las instancias. De forma predeterminada, Elastic Beanstalk usa contenedores no heredados en TCP:80 y contenedores heredados en HTTP:80. Puede invalidar la URL predeterminada para que se ajuste a un recurso existente de la aplicación (por ejemplo, `/myapp/default.aspx`) escribiéndolo en el cuadro Application Health Check URL (URL de comprobación de estado de aplicación). Si anula la URL predeterminada, Elastic Beanstalk utilizará HTTP para consultar el recurso. Para comprobar si está usando un tipo de contenedor heredado, consulte [the section called “¿Por qué algunas versiones de la plataforma están marcadas como heredadas?”](#)

Puede administrar la configuración de la comprobación de estado a través de la sección EC2 Instance Health Check (Comprobación de estado de instancia EC2) del panel Load Balancing (Balanceo de carga).

These settings allow you to configure how Elastic Beanstalk determines whether an EC2 instance is healthy or not.		
Application Health Check:	<input type="text" value="/"/>	
Health Check Interval (seconds):	<input type="text" value="30"/>	(5 - 300)
Health Check Timeout (seconds):	<input type="text" value="5"/>	(2 - 60)
Healthy Check Count Threshold:	<input type="text" value="3"/>	(2 - 10)
Unhealthy Check Count Threshold:	<input type="text" value="5"/>	(2 - 10)

La definición de comprobación de estado incluye una URL que se utiliza para las consultas del estado de las instancias. Invalide la URL predeterminada para que coincida con un recurso existente en la aplicación (por ejemplo, `/myapp/index.jsp`) escribiéndolo en el cuadro Application Health Check URL (URL de comprobación de estado de aplicación).

En la siguiente lista se describen los parámetros de comprobación de estado que puede definir para la aplicación.

- En Health Check Interval (seconds) (Intervalo de comprobación de estado [segundos]), especifique el número de segundos que Elastic Load Balancing espera entre las comprobaciones de estado de las instancias de Amazon EC2 de su aplicación.
- En Health Check Timeout (seconds) (Tiempo de espera de comprobación de estado [segundos]), especifique el número de segundos que Elastic Load Balancing espera para recibir una respuesta antes de considerar que la instancia no responde.
- En Healthy Check Count Threshold (Umbral de recuento de estado correcto) y Unhealthy Check Count Threshold (Umbral de recuento de estado incorrecto), especifique el número de sondeos de URL consecutivos que deben realizarse correcta o incorrectamente antes de que Elastic Load Balancing cambie el estado de la instancia. Por ejemplo, si se especifica **5** en Unhealthy Check

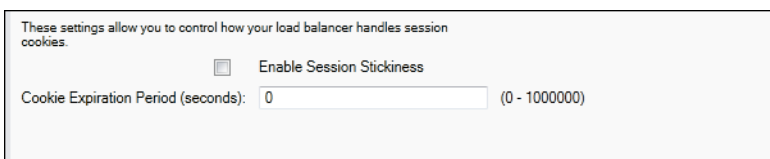
Count Threshold (Umbral de recuento de comprobación incorrecto) significa que la dirección URL devuelve un mensaje de error o un mensaje de tiempo de espera agotado cinco veces seguidas antes de que Elastic Load Balancing considere que no se ha producido un error en la comprobación de estado.

Sesiones

De forma predeterminada, un balanceador de carga direcciona cada solicitud por separado a la instancia de servidor con menor carga. Por su parte, una sesión sticky vincula la sesión de un usuario con una instancia de servidor específica para que todas las solicitudes que provengan de ese usuario durante la sesión se envíen a la misma instancia de servidor.

Elastic Beanstalk utiliza cookies HTTP generadas por el balanceador de carga cuando las sesiones sticky están habilitadas en una aplicación. El balanceador de carga utiliza una cookie especial generada por el balanceador de carga para hacer un seguimiento de la instancia de aplicación de cada solicitud. Cuando el balanceador de carga recibe una solicitud, primero comprueba si esta cookie está presente en la solicitud. Si está presente, la solicitud se envía a la instancia de la aplicación especificada en la cookie. Si no hay ninguna cookie, el balanceador de carga elige una instancia de aplicación en función del algoritmo de balanceo de carga existente. Para vincular las solicitudes posteriores del mismo usuario a esa instancia de aplicación, se inserta una cookie en la respuesta. La configuración de la política define el vencimiento de la cookie, que establece el periodo de validez de cada cookie.

Puede usar la sección Sessions (Sesiones) de la pestaña Load Balancer (Balanceador de carga) para especificar si el balanceador de carga de la aplicación permite persistencia de sesión.



The screenshot shows a configuration panel for session cookies. At the top, it reads: "These settings allow you to control how your load balancer handles session cookies." Below this, there is a checkbox labeled "Enable Session Stickiness" which is currently unchecked. Underneath, there is a text input field for "Cookie Expiration Period (seconds)" with the value "0" entered. To the right of the input field, the range "(0 - 1000000)" is displayed.

Para obtener más información sobre Elastic Load Balancing, consulte la [Guía para desarrolladores de Elastic Load Balancing](#).

Configuración de Auto Scaling mediante AWS Toolkit for Visual Studio

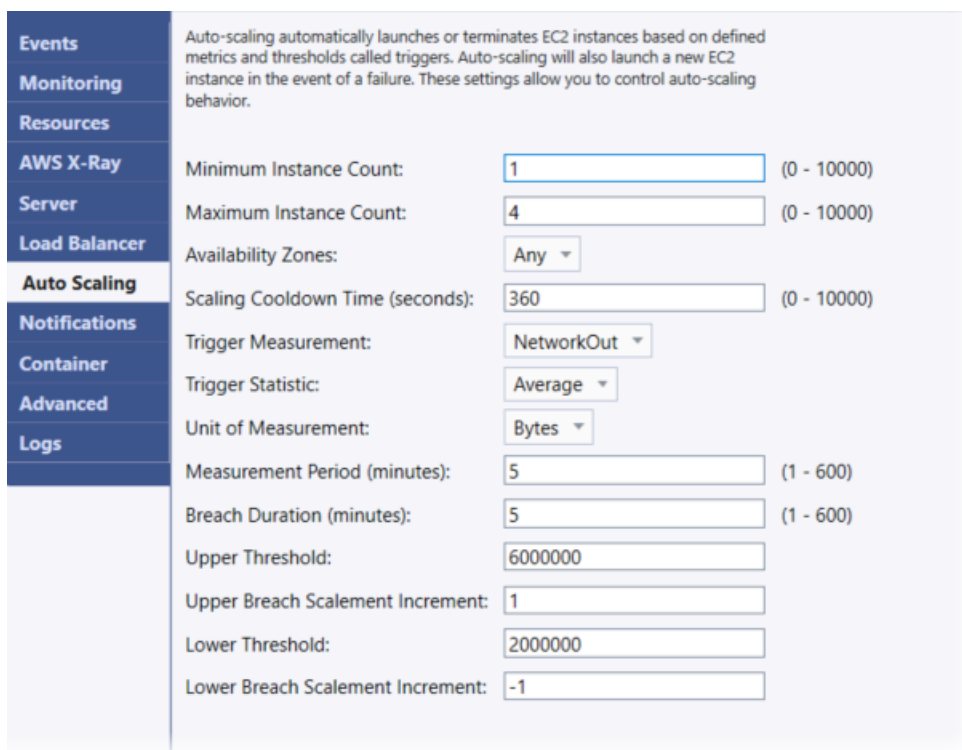
Amazon EC2 Auto Scaling es un servicio web de Amazon diseñado para lanzar o terminar automáticamente instancias de Amazon EC2 en función de disparadores definidos por el usuario. Puede configurar grupos de Auto Scaling y asociar disparadores a esos grupos para escalar automáticamente los recursos informáticos en función de determinadas métricas, como el uso de

ancho de banda o la utilización de CPU. Amazon EC2 Auto Scaling trabaja con Amazon CloudWatch para recuperar las métricas de las instancias de servidor en las que se ejecuta la aplicación.

Amazon EC2 Auto Scaling le permite tomar un grupo de instancias de Amazon EC2 y definir diversos parámetros para hacer que el número de instancias aumente o se reduzca automáticamente. Amazon EC2 Auto Scaling puede añadir o quitar instancias de Amazon EC2 de ese grupo, lo que le ayudará a administrar eficazmente los cambios de tráfico de la aplicación.

Amazon EC2 Auto Scaling también monitoriza el estado de cada instancia de Amazon EC2 que lanza. Si una instancia termina de forma inesperada, Amazon EC2 Auto Scaling lo detecta y lanza una instancia de sustitución. Esta función le permite mantener un número fijo de instancias de Amazon EC2 automáticamente.

Elastic Beanstalk aprovisiona Amazon EC2 Auto Scaling a su aplicación. Puede editar la configuración de las instancias de Amazon EC2 del entorno de Elastic Beanstalk en la pestaña Auto Scaling incluida en la pestaña del entorno de la aplicación en AWS Toolkit for Visual Studio.



Events
Monitoring
Resources
AWS X-Ray
Server
Load Balancer
Auto Scaling
Notifications
Container
Advanced
Logs

Auto-scaling automatically launches or terminates EC2 instances based on defined metrics and thresholds called triggers. Auto-scaling will also launch a new EC2 instance in the event of a failure. These settings allow you to control auto-scaling behavior.

Minimum Instance Count:	<input type="text" value="1"/>	(0 - 10000)
Maximum Instance Count:	<input type="text" value="4"/>	(0 - 10000)
Availability Zones:	<input type="text" value="Any"/>	
Scaling Cooldown Time (seconds):	<input type="text" value="360"/>	(0 - 10000)
Trigger Measurement:	<input type="text" value="NetworkOut"/>	
Trigger Statistic:	<input type="text" value="Average"/>	
Unit of Measurement:	<input type="text" value="Bytes"/>	
Measurement Period (minutes):	<input type="text" value="5"/>	(1 - 600)
Breach Duration (minutes):	<input type="text" value="5"/>	(1 - 600)
Upper Threshold:	<input type="text" value="6000000"/>	
Upper Breach Scalement Increment:	<input type="text" value="1"/>	
Lower Threshold:	<input type="text" value="2000000"/>	
Lower Breach Scalement Increment:	<input type="text" value="-1"/>	

En la siguiente sección se explica cómo se configuran los parámetros de Auto Scaling para la aplicación.

Lanzar la configuración

Puede editar la configuración de lanzamiento para controlar la forma en que la aplicación de Elastic Beanstalk aprovisiona los recursos de Amazon EC2 Auto Scaling.

Los cuadros Minimum Instance Count (Número mínimo de instancias) y Maximum Instance Count (Número máximo de instancias) le permiten especificar el tamaño mínimo y máximo del grupo de Auto Scaling que la aplicación de Elastic Beanstalk utiliza.

Auto-scaling automatically launches or terminates EC2 instances based on defined metrics and thresholds called triggers. Auto-scaling will also launch a new EC2 instance in the event of a failure. These settings allow you to control auto-scaling behavior.

Minimum Instance Count:	<input type="text" value="1"/>	(1 - 10000)
Maximum Instance Count:	<input type="text" value="4"/>	(1 - 10000)
Availability Zones:	<input type="text" value="Any"/>	
Scaling Cooldown Time (seconds):	<input type="text" value="360"/>	(0 - 10000)

Note

Para mantener un número fijo de instancias de Amazon EC2, establezca los cuadros Minimum Instance Count (Número mínimo de instancias) y Maximum Instance Count (Número máximo de instancias) en el mismo valor.

El cuadro Availability Zones (Zonas de disponibilidad) le permite especificar el número de zonas de disponibilidad en las que desea que se lancen las instancias de Amazon EC2. Es importante definir este número si desea crear aplicaciones tolerantes a errores. Si una zona de disponibilidad deja de funcionar, las instancias se ejecutarán en las otras zonas de disponibilidad.

Note

En la actualidad, no es posible especificar en qué zona de disponibilidad se encontrará la instancia.

Desencadenadores

Un desencadenador es un mecanismo de Amazon EC2 Auto Scaling que se define para indicar al sistema cuándo debe aumentar (escalado ascendente) o reducir (escalado descendente) el

número de instancias. Puede configurar los disparadores para que se activen a raíz de cualquiera de las métricas publicadas en Amazon CloudWatch, como el uso de la CPU, y determinar si se han cumplido las condiciones especificadas. Cuando se supera el umbral superior o inferior de las condiciones que ha especificado para la métrica durante el periodo de tiempo especificado, el disparador lanza un proceso de larga ejecución llamado actividad de escalado.

Puede definir un desencadenador de escalado para su aplicación de Elastic Beanstalk utilizando AWS Toolkit for Visual Studio.

Trigger Measurement:	<input type="text" value="NetworkOut"/>	
Trigger Statistic:	<input type="text" value="Average"/>	
Unit of Measurement:	<input type="text" value="Bytes"/>	
Measurement Period (minutes):	<input type="text" value="5"/>	(1 - 600)
Breach Duration (minutes):	<input type="text" value="5"/>	(1 - 600)
Upper Threshold:	<input type="text" value="6000000"/>	(0 - 20000000)
Upper Breach Scalement Increment:	<input type="text" value="1"/>	
Lower Threshold:	<input type="text" value="2000000"/>	(0 - 20000000)
Lower Breach Scalement Increment:	<input type="text" value="-1"/>	

Los disparadores de Amazon EC2 Auto Scaling funcionan observando una métrica de Amazon CloudWatch específica para una instancia. Las métricas incluyen el uso de la CPU, el tráfico de red y la actividad del disco. Use la opción Trigger Measurement (Medición de desencadenador) para seleccionar una métrica para el desencadenador.

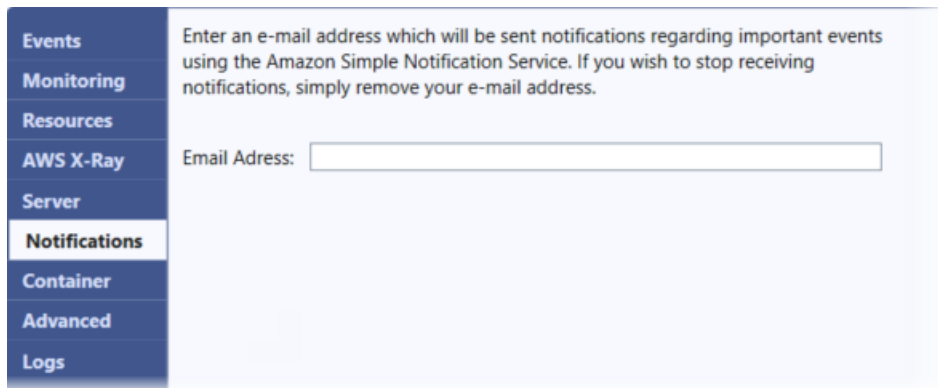
En la siguiente lista se describen los parámetros de desencadenador que puede configurar en la Management Console de AWS.

- Puede especificar qué estadística debe usar el disparador. Puede seleccionar Minimum (Mínimo), Maximum (Máximo), Sum (Suma) o Average (Media) para Trigger Statistic (Estadística de desencadenador).
- En Unit of Measurement (Unidad de medición), especifique la unidad de medida del disparador.
- El valor del cuadro Measurement Period (Periodo de medición) especifica la frecuencia con la que Amazon CloudWatch calcula las métricas del disparador. La Breach Duration (Duración de la interrupción) es la cantidad de tiempo que una métrica puede sobrepasar el límite definido (según lo especificado en Upper Threshold (Umbral superior) y Lower Threshold (Umbral inferior)) antes de que se active el desencadenador.
- En Upper breach scale increment (Incremento de escalado de infracción superior) y Lower breach scale increment (Incremento de escalado de infracción inferior), especifique el número de instancias de Amazon EC2 que se van a agregar o quitar al efectuar una actividad de escalado.

Para obtener más información sobre Amazon EC2 Auto Scaling, consulte la sección Auto Scaling de Amazon EC2 en la [Guía de documentación de Amazon Elastic Compute Cloud](#).

Configuración de notificaciones con AWS Toolkit for Visual Studio

Elastic Beanstalk utiliza Amazon Simple Notification Service (Amazon SNS) para enviarle notificaciones sobre eventos importantes que afectan a la aplicación. Para habilitar las notificaciones de Amazon SNS, ingrese su dirección de correo electrónico en el cuadro Email Address (Dirección de correo electrónico). Para desactivar estas notificaciones, quite la dirección de correo electrónico del cuadro.

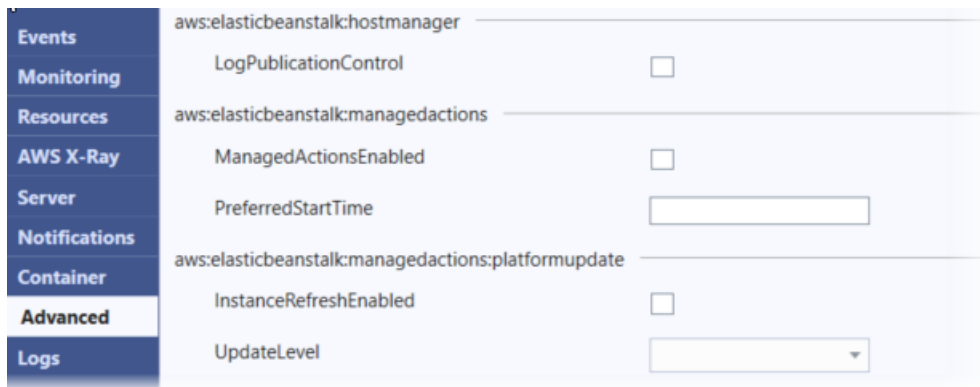


The screenshot shows a configuration panel for Notifications. On the left is a vertical navigation menu with the following items: Events, Monitoring, Resources, AWS X-Ray, Server, Notifications (highlighted), Container, Advanced, and Logs. The main content area contains the following text: "Enter an e-mail address which will be sent notifications regarding important events using the Amazon Simple Notification Service. If you wish to stop receiving notifications, simply remove your e-mail address." Below this text is a text input field labeled "Email Address:".

Configuración de opciones del entorno adicionales con AWS Toolkit for Visual Studio

Elastic Beanstalk define un gran número de opciones de configuración que puede utilizar para configurar el comportamiento del entorno y los recursos que contiene. Las opciones de configuración se organizan en espacios de nombres como `aws:autoscaling:asg`. Cada espacio de nombres define opciones para el grupo de Auto Scaling de un entorno. El panel Advanced (Opciones avanzadas) muestra los espacios de nombres de las opciones de configuración en orden alfabético. Puede actualizarlos una vez creado el entorno.

Para obtener una lista completa de espacios de nombres y opciones, incluidos los valores predeterminados y admitidos para cada uno, consulte [Opciones generales para todos los entornos y Historial de plataformas de .NET Core en Linux](#).

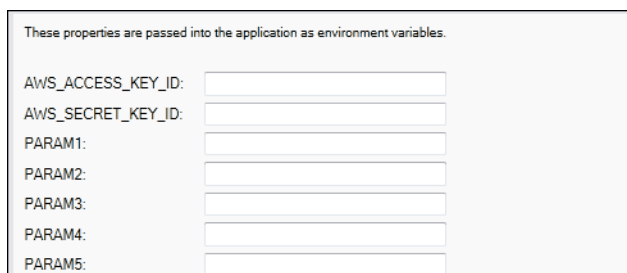


The screenshot shows the configuration page for Elastic Beanstalk managed actions. On the left is a navigation menu with options: Events, Monitoring, Resources, AWS X-Ray, Server, Notifications, Container, Advanced, and Logs. The main area is divided into sections by horizontal lines:

- aws:elasticbeanstalk:hostmanager**
 - LogPublicationControl:
- aws:elasticbeanstalk:managedactions**
 - ManagedActionsEnabled:
 - PreferredStartTime:
- aws:elasticbeanstalk:managedactions:platformupdate**
 - InstanceRefreshEnabled:
 - UpdateLevel:

Configuración de contenedores de .NET Core con AWS Toolkit for Visual Studio

El panel Container (Contenedor) le permite especificar variables de entorno que puede leer desde el código de la aplicación.



The screenshot shows a configuration panel titled "These properties are passed into the application as environment variables." It contains the following fields:

- AWS_ACCESS_KEY_ID:
- AWS_SECRET_KEY_ID:
- PARAM1:
- PARAM2:
- PARAM3:
- PARAM4:
- PARAM5:

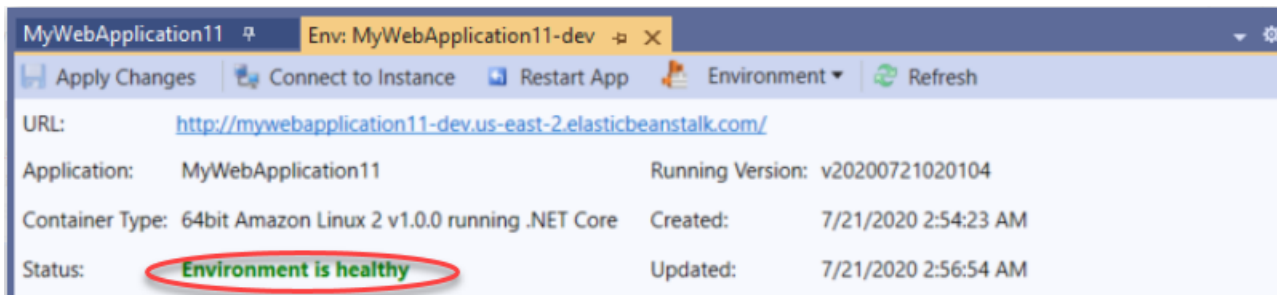
Monitorización del estado de mantenimiento de la aplicación

Es importante saber que su sitio web de producción está disponible y responde a las solicitudes. Elastic Beanstalk ofrece funciones que le ayudan a supervisar la capacidad de respuesta de su aplicación. Monitorea las estadísticas sobre la aplicación y le avisa cuando se superan los umbrales.

Para obtener información sobre la monitorización de estado proporcionada por Elastic Beanstalk, consulte [Informes de estado básicos](#).

Para obtener acceso a la información sobre el funcionamiento de su aplicación, puede utilizar el AWS Toolkit for Visual Studio o la Management Console de AWS.

En el conjunto de herramientas, se muestra el estado del entorno y la aplicación en el campo Status (Estado).



Para monitorear el estado de la aplicación

1. En AWS Toolkit for Visual Studio, vaya a AWS Explorer (Explorador de AWS), expanda el nodo Elastic Beanstalk y, a continuación, el nodo de la aplicación.
2. Abra el menú contextual (haga clic con el botón derecho) de su entorno de aplicación y seleccione View Status (Ver estado).
3. En la pestaña del entorno de la aplicación, seleccione Monitoring (Monitorización).

El panel Monitoring (Monitorización) contiene un conjunto de gráficos donde se muestra el uso que hace de los recursos un entorno de aplicación concreto.



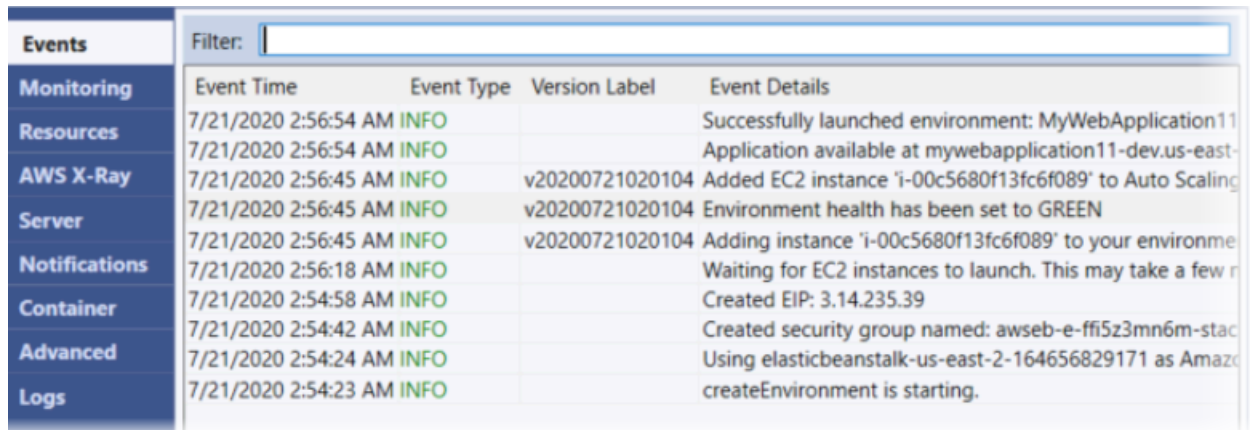
Note

De forma predeterminada, el intervalo de tiempo está establecido en la última hora. Para modificar este valor, en la lista Time Range (Intervalo de tiempo), seleccione un intervalo de tiempo diferente.

Para ver los eventos asociados a su aplicación, puede utilizar el AWS Toolkit for Visual Studio o Management Console de AWS.

Para ver los eventos de la aplicación

1. Vaya a AWS Toolkit for Visual Studio y en AWS Explorer (Explorador de AWS), expanda el nodo Elastic Beanstalk y el nodo de la aplicación.
2. Abra el menú contextual (haga clic con el botón derecho) de su entorno de aplicación y seleccione View Status (Ver estado).
3. En la pestaña del entorno de la aplicación, seleccione Events (Eventos).



Events	Filter:
Monitoring	Event Time Event Type Version Label Event Details
Resources	7/21/2020 2:56:54 AM INFO Successfully launched environment: MyWebApplication11
AWS X-Ray	7/21/2020 2:56:54 AM INFO Application available at mywebapplication11-dev.us-east-
Server	7/21/2020 2:56:45 AM INFO v20200721020104 Added EC2 instance 'i-00c5680f13fc6f089' to Auto Scaling
Notifications	7/21/2020 2:56:45 AM INFO v20200721020104 Environment health has been set to GREEN
Container	7/21/2020 2:56:45 AM INFO v20200721020104 Adding instance 'i-00c5680f13fc6f089' to your environme
Advanced	7/21/2020 2:56:18 AM INFO Waiting for EC2 instances to launch. This may take a few r
Logs	7/21/2020 2:54:58 AM INFO Created EIP: 3.14.235.39
	7/21/2020 2:54:42 AM INFO Created security group named: awseb-e-ffi5z3mn6m-stac
	7/21/2020 2:54:24 AM INFO Using elasticbeanstalk-us-east-2-164656829171 as Amaz
	7/21/2020 2:54:23 AM INFO createEnvironment is starting.

Migración de .NET en plataformas de Windows Server a la plataforma .NET Core en Linux

Puede migrar aplicaciones que se ejecutan en [.NET en plataformas Windows Server](#) a las plataformas NET Core de Linux. A continuación, se muestran algunas consideraciones al migrar de Windows a plataformas Linux.

Consideraciones para migrar a la plataforma .NET Core en Linux

Area	Cambios e información
Configuración de aplicaciones	En las plataformas Windows, se utiliza un manifiesto de implementación para especificar las aplicaciones que se ejecutan en el entorno. Las plataformas .NET Core de Linux utilizan un Profile para especificar las aplicaciones que se ejecutan en las instancias de su entorno. Para obtener detalles sobre la agrupación de aplicaciones, consulte the section called "Agrupación de aplicaciones" .
Servidor proxy	En las plataformas Windows, se utiliza IIS como servidor proxy de la aplicación. El .NET Core en plataformas Linux incluye nginx como proxy inverso de forma

Area	Cambios e información
	predeterminada. Puede elegir no utilizar ningún servidor proxy y utilizar Kestrel como servidor web de su aplicación. Para obtener más información, consulte the section called “Servidor proxy” .
Dirección amiento	En las plataformas Windows, se utiliza IIS en el código de aplicación y se incluye un manifiesto de implementación para configurar la ruta de IIS. Para la plataforma .NET Core en Linux, se utiliza enrutamiento ASP .NET Core en el código de la aplicación y se actualiza la configuración nginx de su entorno. Para obtener más información, consulte the section called “Servidor proxy” .
Registros	Las plataformas Linux y Windows transmiten registros diferentes. Para obtener información, consulte the section called “Cómo Elastic Beanstalk configura CloudWatch Logs” .

Creación e implementación de aplicaciones .NET Windows en Elastic Beanstalk

Consulte el .NET on AWS Developer Center

¿Has visitado nuestro Centro de desarrolladores de .Net? Es nuestra ventanilla única para todo lo relacionado con .NET AWS.

Para obtener más información, consulte el [.NET on AWS Developer Center](#).

AWS Elastic Beanstalk para .NET facilita la implementación, la administración y el escalado de las aplicaciones web de ASP.NET y .NET Core que utilizan Amazon Web Services. En este capítulo se proporcionan instrucciones para crear, probar, implementar y volver a implementar la aplicación web de Windows en Elastic Beanstalk. Puede implementar la aplicación en solo unos minutos mediante la interfaz de línea de comandos de Elastic Beanstalk (EB CLI) o la consola de Elastic Beanstalk.

En este capítulo se proporcionan los siguientes tutoriales:

- [QuickStart para .NET Core en Windows](#)
- [Implementación de una aplicación ASP.NET Core](#)

Si necesita ayuda con el desarrollo de aplicaciones de Windows.NET Core, puede acudir a varios lugares:

- [Foro de desarrollo de .NET](#): publique sus preguntas y reciba comentarios.
- [.NET Developer Center](#): ventanilla única donde encontrará ejemplos de código, documentación, herramientas y recursos adicionales.
- [AWS Documentación del SDK para .NET](#): obtenga información sobre la configuración del SDK y la ejecución de ejemplos de código, las características del SDK e información detallada sobre las operaciones de la API para el SDK.

Note

Esta plataforma no admite las siguientes características de Elastic Beanstalk:

- Entornos de trabajo. Para obtener más información, consulte [Entornos de trabajo de Elastic Beanstalk](#).
- Registros de paquete. Para obtener más información, consulte [Consulta de los registros de instancias](#).

Temas

- [Ramas de plataforma Windows 2012 de Elastic Beanstalk retiradas y compatibilidad con TLS 1.2](#)
- [QuickStart: Implemente una aplicación .NET Core para Windows en Elastic Beanstalk](#)
- [Tutorial: Implementación de una aplicación ASP.NET Core con Elastic Beanstalk](#)
- [Configuración del entorno de desarrollo de .NET](#)
- [Uso de la plataforma .NET de Elastic Beanstalk](#)
- [Añadir una instancia de base de datos de Amazon RDS a un entorno de aplicaciones .NET](#)
- [El AWS Toolkit for Visual Studio](#)
- [Migración de la aplicación .NET local a Elastic Beanstalk](#)

Ramas de plataforma Windows 2012 de Elastic Beanstalk retiradas y compatibilidad con TLS 1.2

En este tema se proporcionan recomendaciones si sus aplicaciones se están ejecutando actualmente en las ramas retiradas de la plataforma Windows Server 2012 R2. También aborda la obsolescencia de la compatibilidad con las versiones de los protocolos TLS 1.0 y 1.1 en los puntos finales de nuestra API de AWS servicio y en las ramas de plataforma afectadas.

Se retiraron las ramificaciones de la plataforma Windows Server 2012 R2

Elastic Beanstalk retiró las [sucursales de la plataforma Windows Server 2012 R2 el 4 de diciembre de 2023](#) y convirtió las AMI asociadas a esas plataformas en privadas el 10 de abril de 2024. Esta acción impide el lanzamiento de instancias en los entornos de Windows Server 2012 que utilizan la AMI de Beanstalk predeterminada.

Si tiene algún entorno que se ejecute en sucursales retiradas de la plataforma Windows, le recomendamos que lo migre a una de las siguientes plataformas de Windows Server, que son actuales y totalmente compatibles:

- Windows Server 2022 con IIS 10.0 versión 2.x
- Windows Server 2019 con IIS 10.0 versión 2.x

Para obtener información completa sobre la migración, consulte [Migración desde versiones principales anteriores de la plataforma Windows Server](#).

Para obtener más información acerca de la obsolescencia de la plataforma, consulte [Política de compatibilidad con plataformas Elastic Beanstalk](#).

Note

Si no puede migrar a estas plataformas totalmente compatibles, le recomendamos que utilice las AMI personalizadas creadas con las AMI Core de Windows Server 2012 R2 o Windows Server 2012 R2 como imagen base, si aún no lo ha hecho. Para obtener instrucciones detalladas, consulte [Conservar el acceso a una Imagen de máquina de Amazon \(AMI\) de una plataforma retirada](#). Póngase en contacto con el [AWS Support Center](#) si necesita acceso temporal a una AMI mientras realiza uno de estos pasos de migración.

Compatibilidad con TLS 1.2

El 31 de diciembre de 2023, AWS comenzó a aplicar plenamente la TLS 1.2 en todos los puntos finales de la AWS API. Esta acción eliminó la posibilidad de usar las versiones 1.0 y 1.1 de TLS con todas las API. Esta información se comunicó originalmente el [28 de junio de 2022](#). Para evitar el riesgo de que la disponibilidad se vea afectada, actualice los entornos que ejecuten las versiones de plataforma identificadas aquí a una versión más reciente lo antes posible, si aún no lo ha hecho.

Impacto potencial

Las versiones de las plataformas Elastic Beanstalk que ejecutan TLS v1.1 o versiones anteriores se ven afectadas. Este cambio afecta a las acciones del entorno, que incluyen, entre otras, las siguientes: implementaciones de configuración, implementaciones de aplicaciones, escalado automático, lanzamiento de un nuevo entorno, rotación de registros, informes de estado mejorados y publicación de registros de aplicaciones en el bucket de Amazon S3 asociado a sus aplicaciones.

Versiones de la plataforma Windows afectadas

Se recomienda a los clientes que tengan entornos de Elastic Beanstalk en la siguiente versión de la plataforma que actualicen cada uno de sus entornos correspondientes a la versión 2.8.3 o posterior de la plataforma Windows, lanzada el [18 de febrero de 2022](#).

- Windows Server 2019: versión de la plataforma 2.8.2 o versiones anteriores

Se recomienda a los clientes que tengan entornos de Elastic Beanstalk en las siguientes versiones de plataforma que actualicen cada uno de sus entornos correspondientes a la versión 2.10.7 o posterior de la plataforma Windows, lanzada el [28 de diciembre de 2022](#).

- Windows Server 2016: versión de la plataforma 2.10.6 o versiones anteriores
- Windows Server 2012: todas las versiones de plataforma; esta plataforma se retiró el [4 de diciembre de 2023](#)
- Windows Server 2008: todas las versiones de la plataforma (esta plataforma se retiró el [28 de octubre de 2019](#))

Para obtener una lista de estas versiones de la plataforma de Windows Server, consulte [Plataformas admitidas](#) en la guía Plataformas de AWS Elastic Beanstalk .

Para obtener más información y conocer las prácticas recomendadas sobre la actualización del entorno, consulte [Actualización de la versión de la plataforma del entorno de Elastic Beanstalk](#).

QuickStart: Implemente una aplicación.NET Core para Windows en Elastic Beanstalk

Este QuickStart tutorial explica el proceso de creación de una aplicación.NET Core para Windows e implementarla en un AWS Elastic Beanstalk entorno.

Note

Este QuickStart tutorial está destinado a fines de demostración. No utilice la aplicación creada en este tutorial para el tráfico de producción.

Secciones

- [Tu AWS cuenta](#)
- [Requisitos previos](#)
- [Paso 1: Cree una aplicación.NET Core para Windows](#)
- [Paso 2: ejecuta tu aplicación localmente](#)
- [Paso 3: Implemente su aplicación.NET Core en Windows con la CLI de EB](#)
- [Paso 4: Ejecute la aplicación en Elastic Beanstalk](#)
- [Paso 5: Eliminar](#)
- [AWS recursos para su aplicación](#)
- [Siguiendo pasos](#)
- [Implemente con la consola de Elastic Beanstalk](#)

Tu AWS cuenta

Si aún no eres AWS cliente, debes crear una AWS cuenta. El registro le permite acceder a Elastic Beanstalk AWS y a otros servicios que necesite.

Si ya tiene una AWS cuenta, puede pasar a [Requisitos previos](#)

Crea una AWS cuenta

Inscríbese en una Cuenta de AWS

Si no tiene una Cuenta de AWS, complete los siguientes pasos para crearlo.

Para suscribirte a una Cuenta de AWS

1. Abra <https://portal.aws.amazon.com/billing/signup>.
2. Siga las instrucciones que se le indiquen.

Parte del procedimiento de registro consiste en recibir una llamada telefónica e indicar un código de verificación en el teclado del teléfono.

Cuando te registras en una Cuenta de AWS, Usuario raíz de la cuenta de AWS se crea una. El usuario raíz tendrá acceso a todos los Servicios de AWS y recursos de esa cuenta. Como práctica recomendada de seguridad, asigne acceso administrativo a un usuario y utilice únicamente el usuario raíz para realizar [tareas que requieren acceso de usuario raíz](#).

AWS te envía un correo electrónico de confirmación una vez finalizado el proceso de registro. Puede ver la actividad de la cuenta y administrar la cuenta en cualquier momento entrando en <https://aws.amazon.com/> y seleccionando Mi cuenta.

Creación de un usuario con acceso administrativo

Después de crear un usuario administrativo Cuenta de AWS, asegúrelo Usuario raíz de la cuenta de AWS AWS IAM Identity Center, habilite y cree un usuario administrativo para no usar el usuario root en las tareas diarias.

Proteja su Usuario raíz de la cuenta de AWS

1. Inicie sesión [AWS Management Console](#) como propietario de la cuenta seleccionando el usuario root e introduciendo su dirección de Cuenta de AWS correo electrónico. En la siguiente página, escriba su contraseña.

Para obtener ayuda para iniciar sesión con el usuario raíz, consulte [Iniciar sesión como usuario raíz](#) en la Guía del usuario de AWS Sign-In .

2. Active la autenticación multifactor (MFA) para el usuario raíz.

Para obtener instrucciones, consulte [Habilitar un dispositivo MFA virtual para el usuario Cuenta de AWS raíz \(consola\)](#) en la Guía del usuario de IAM.

Creación de un usuario con acceso administrativo

1. Activar IAM Identity Center.

Consulte las instrucciones en [Activar AWS IAM Identity Center](#) en la Guía del usuario de AWS IAM Identity Center .

2. En IAM Identity Center, conceda acceso administrativo a un usuario.

Para ver un tutorial sobre su uso Directorio de IAM Identity Center como fuente de identidad, consulte [Configurar el acceso de los usuarios con la configuración predeterminada Directorio de IAM Identity Center en la](#) Guía del AWS IAM Identity Center usuario.

Iniciar sesión como usuario con acceso de administrador

- Para iniciar sesión con el usuario de IAM Identity Center, utilice la URL de inicio de sesión que se envió a la dirección de correo electrónico cuando creó el usuario de IAM Identity Center.

Para obtener ayuda para iniciar sesión con un usuario del Centro de identidades de IAM, consulte [Iniciar sesión en el portal de AWS acceso](#) en la Guía del AWS Sign-In usuario.

Concesión de acceso a usuarios adicionales

1. En IAM Identity Center, cree un conjunto de permisos que siga la práctica recomendada de aplicar permisos de privilegios mínimos.

Para conocer las instrucciones, consulte [Create a permission set](#) en la Guía del usuario de AWS IAM Identity Center .

2. Asigne usuarios a un grupo y, a continuación, asigne el acceso de inicio de sesión único al grupo.

Para conocer las instrucciones, consulte [Add groups](#) en la Guía del usuario de AWS IAM Identity Center .

Requisitos previos

Para seguir los procedimientos de esta guía, necesitará un shell o un terminal de línea de comando donde pueda ejecutar los comandos. Los comandos se muestran en los listados precedidos de un símbolo de aviso (>) y del nombre del directorio actual, si procede.

```
C:\eb-project> this is a command  
this is output
```

CLI DE EB

En este tutorial también se utiliza la interfaz de línea de comandos de Elastic Beanstalk (CLI de EB). Para obtener detalles sobre la instalación y configuración de la CLI de EB, consulte [Instalación de la CLI de EB](#) y [Configuración de la CLI de EB](#).

.NET Core en Windows

Si no tiene el SDK de .NET instalado en su máquina local, puede instalarlo siguiendo el enlace [Descargar .NET](#) que aparece en el sitio web de [documentación de .NET](#).

Compruebe la instalación del SDK de .NET ejecutando el siguiente comando.

```
C:\> dotnet --info
```

Paso 1: Cree una aplicación .NET Core para Windows

Cree el directorio del proyecto.

```
C:\> mkdir eb-dotnetcore  
C:\> cd eb-dotnetcore
```

A continuación, cree un ejemplo de aplicación de servicio web RESTful de Hello World ejecutando los siguientes comandos.

```
C:\eb-dotnetcore> dotnet new web --name HelloElasticBeanstalk  
C:\eb-dotnetcore> cd HelloElasticBeanstalk
```

Paso 2: ejecuta tu aplicación localmente

Ejecute el siguiente comando para ejecutar la aplicación localmente.

```
C:\eb-dotnetcore\HelloElasticBeasntalk> dotnet run
```

El resultado debería tener un aspecto similar al siguiente texto.

```
info: Microsoft.Hosting.Lifetime[14]  
      Now listening on: https://localhost:7222  
info: Microsoft.Hosting.Lifetime[14]  
      Now listening on: http://localhost:5228  
info: Microsoft.Hosting.Lifetime[0]
```

```

Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: C:\Users\Administrator\eb-dotnetcore\HelloElasticBeanstalk

```

Note

El dotnet comando selecciona un puerto al azar al ejecutar la aplicación localmente. En este ejemplo, el puerto es 5228. Al implementar la aplicación en el entorno de Elastic Beanstalk, la aplicación se ejecutará en el puerto 5000.

Introduzca la dirección URL `http://localhost:port` en su navegador web. Para este ejemplo específico, el comando es `http://localhost:5228`. El navegador web debería mostrar «¡Hola mundo!».

Paso 3: Implemente su aplicación.NET Core en Windows con la CLI de EB

Ejecute los siguientes comandos para crear un entorno de Elastic Beanstalk para esta aplicación.

Para crear un entorno e implementar su aplicación.NET Core en Windows

1. Ejecute los siguientes comandos en el HelloElasticBeanstalk directorio para publicar y comprimir la aplicación.

```

C:\eb-dotnetcore\HelloElasticBeasntalk> dotnet publish -o site
C:\eb-dotnetcore\HelloElasticBeasntalk> cd site
C:\eb-dotnetcore\HelloElasticBeasntalk\site> Compress-Archive -Path * -
DestinationPath ../site.zip
C:\eb-dotnetcore\HelloElasticBeasntalk\site> cd ..

```

2. Cree un nuevo archivo en la carpeta HelloElasticBeanstalk llamada `aws-windows-deployment-manifest.json` con el siguiente contenido:

```

{
  "manifestVersion": 1,
  "deployments": {
    "aspNetCoreWeb": [
      {

```

```
        "name": "test-dotnet-core",
        "parameters": {
            "appBundle": "site.zip",
            "iisPath": "/",
            "iisWebSite": "Default Web Site"
        }
    }
}
```

3. Inicialice el repositorio de la CLI de EB con el comando `eb init`.

```
C:\eb-dotnetcore\HelloElasticBeasntalk> eb init -p iis dotnet-windows-server-tutorial --region us-east-2
```

Este comando crea una aplicación llamada `dotnet-windows-server-tutorial` y configura su repositorio local para crear entornos con la última versión de la plataforma de servidor Windows.

4. Cree un entorno e implemente la aplicación en él con `eb create`. Elastic Beanstalk crea automáticamente un archivo zip para la aplicación y lo inicia en el puerto 5000.

```
C:\eb-dotnetcore\HelloElasticBeasntalk> eb create dotnet-windows-server-env
```

Elastic Beanstalk tarda unos cinco minutos en crear el entorno.

Paso 4: Ejecute la aplicación en Elastic Beanstalk

Cuando finalice el proceso de creación del entorno, abra su sitio web con `eb open`

```
C:\eb-dotnetcore\HelloElasticBeasntalk> eb open
```

¡Enhorabuena! ¡Ha implementado una aplicación .NET Core en Windows con Elastic Beanstalk! Se abre una ventana del navegador con el nombre de dominio creado para su aplicación.

Paso 5: Eliminar

Puede cerrar su entorno cuando termine de trabajar con la aplicación. Elastic Beanstalk AWS cancela todos los recursos asociados a su entorno.

Para finalizar el entorno de Elastic Beanstalk con la CLI de EB, ejecute el siguiente comando.

```
C:\eb-dotnetcore\HelloElasticBeasntalk> eb terminate
```

AWS recursos para su aplicación

Acabas de crear una aplicación de instancia única. Sirve como una aplicación de muestra sencilla con una sola instancia EC2, por lo que no requiere balanceo de carga ni escalado automático. Para las aplicaciones de instancia única, Elastic Beanstalk crea los siguientes recursos: AWS

- EC2 instance (Instancia EC2): máquina virtual de Amazon EC2 configurada para ejecutar aplicaciones web en la plataforma que elija.

Cada plataforma ejecuta un conjunto distinto de software, archivos de configuración y scripts compatibles con una determinada versión de lenguaje, marco y contenedor web (o una combinación de ellos). La mayoría de las plataformas utilizan Apache o nginx como un proxy inverso que procesa el tráfico web delante de la aplicación web, reenvía las solicitudes a esta, administra los recursos estáticos y genera registros de acceso y errores.

- Instance security group (Grupo de seguridad de la instancia): grupo de seguridad de Amazon EC2 configurado para permitir el tráfico entrante en el puerto 80. Este recurso permite que el tráfico HTTP procedente del balanceador de carga llegue a la instancia EC2 en la que se ejecuta la aplicación web. De forma predeterminada, el tráfico no está permitido en otros puertos.
- Bucket de Amazon S3: ubicación de almacenamiento para el código fuente, los registros y otros artefactos que se crean al utilizar Elastic Beanstalk.
- CloudWatch Alarmas de Amazon: dos CloudWatch alarmas que monitorean la carga de las instancias de su entorno y se activan si la carga es demasiado alta o demasiado baja. Cuando se activa una alarma, en respuesta, el grupo de Auto Scaling aumenta o reduce los recursos.
- AWS CloudFormation pila: Elastic AWS CloudFormation Beanstalk se utiliza para lanzar los recursos de su entorno y propagar los cambios de configuración. Los recursos se definen en una plantilla que puede verse en la [consola de AWS CloudFormation](#).
- Nombre de dominio: nombre de dominio que direcciona el tráfico a la aplicación web con el formato *subdominio.región.elasticbeanstalk.com*.

Elastic Beanstalk administra todos estos recursos. Cuando termina su entorno, Elastic Beanstalk termina todos los recursos que este contiene.

Siguientes pasos

Una vez que disponga de un entorno que ejecute una aplicación, podrá implementar una nueva versión de la aplicación o una aplicación distinta en cualquier momento. La implementación de una nueva versión de la aplicación es una tarea muy rápida, ya que no se requiere aprovisionar ni reiniciar instancias EC2. También puede explorar su nuevo entorno con la consola de Elastic Beanstalk. Para ver los pasos detallados, consulte [Explore su entorno](#) en el capítulo Introducción de esta guía.

Una vez que haya implementado una o dos aplicaciones de ejemplo y esté listo para empezar a desarrollar y ejecutar localmente aplicaciones de .NET Core en Windows, consulte [Configuración del entorno de desarrollo de .NET](#)

Implemente con la consola de Elastic Beanstalk

También puede usar la consola de Elastic Beanstalk para iniciar la aplicación de muestra. Para ver los pasos detallados, consulte [Crear una aplicación de ejemplo](#) en el capítulo Introducción de esta guía.

Tutorial: Implementación de una aplicación ASP.NET Core con Elastic Beanstalk

En este tutorial, explicará el proceso de creación de una nueva aplicación de ASP.NET Core e implementarla en ella. AWS Elastic Beanstalk

En primer lugar, usará la herramienta de línea de comandos `dotnet` del SDK de .NET Core para generar una aplicación básica de línea de comandos de .NET Core, instalar las dependencias, compilar el código y ejecutar las aplicaciones localmente. A continuación, creará la clase `Program.cs` predeterminada y añadirá una clase `Startup.cs` de ASP.NET y archivos de configuración para crear una aplicación que sirva solicitudes HTTP con ASP.NET e IIS.

Finalmente, Elastic Beanstalk usa un [manifiesto de implementación](#) para configurar las implementaciones de aplicaciones de .NET Core, aplicaciones personalizadas y varias aplicaciones de .NET Core o MSBuild en un único servidor. Para implementar una aplicación de .NET Core en un entorno de Windows Server, tendrá que añadir un archivo del sitio a un paquete de código fuente de la aplicación con un manifiesto de implementación. El comando `dotnet publish` genera clases compiladas y dependencias que puede empaquetar con un archivo `web.config` para crear un archivo del sitio. El manifiesto de implementación indica a Elastic Beanstalk la ruta en la que debe

ejecutarse el sitio y se puede usar para configurar grupos de aplicación y ejecutar varias aplicaciones en diferentes rutas.

[El código fuente está disponible aquí: .zip dotnet-core-windows-tutorial](#)

Secciones

- [Requisitos previos](#)
- [Generar un proyecto de .NET Core](#)
- [Lance un entorno de Elastic Beanstalk](#)
- [Actualizar el código fuente](#)
- [Implementar la aplicación](#)
- [Limpieza](#)
- [Pasos siguientes](#)

Requisitos previos

Este tutorial utiliza el SDK de .NET Core para generar una aplicación básica de .NET Core, ejecutarla localmente y crear un paquete implementable.

Requisitos

- .NET Core (x64) 1.0.1, 2.0.0 o posterior

Para instalar el SDK de .NET Core

1. Descargue el instalador desde microsoft.com/net/core. Elija Windows. Elija Download .NET SDK.
2. Ejecute el instalador y siga las instrucciones.

Este tutorial utiliza una herramienta ZIP de línea de comandos para crea un paquete de código fuente que puede implementar en Elastic Beanstalk. Para utilizar el comando zip en Windows, puede instalar UnxUtils, una colección ligera de herramientas útiles de línea de comandos como zip y ls. Otra opción es [usar el Explorador de Windows](#) u otra herramienta ZIP para crear archivos de paquetes de código fuente.

Para instalar UnxUtils

1. Descargar [UnxUtils](#).

2. Extraiga el archivo en un directorio local. Por ejemplo, `C:\Program Files (x86)`.
3. Agregue la ruta de los binarios en la variable de usuario Windows PATH. Por ejemplo, `C:\Program Files (x86)\UnxUtils\usr\local\wbin`.
 - a. Pulse la tecla de Windows y, a continuación, escriba **environment variables**.
 - b. Elija Edit environment variables for your account (Editar las variables de entorno de esta cuenta).
 - c. Elija PATH y, a continuación, Edit (Editar).
 - d. Añada rutas al campo Variable value (Valor de la variable), separadas por punto y coma. Por ejemplo: **`C:\item1\path;C:\item2\path`**
 - e. Elija OK (Aceptar) dos veces para aplicar la nueva configuración.
 - f. Cierre las ventanas del símbolo del sistema en ejecución y, a continuación, vuelva a abrir una ventana del símbolo del sistema.
4. Abra una nueva ventana del símbolo del sistema y ejecute el comando `zip` comando para comprobar que funciona.

```
> zip -h
Copyright (C) 1990-1999 Info-ZIP
Type 'zip "-L"' for software license.
...
```

Generar un proyecto de .NET Core

Utilice la herramienta de línea de comandos `dotnet` para generar un nuevo proyecto C# .NET Core y ejecutarlo localmente. La aplicación de .NET Core predeterminada es una herramienta de línea de comandos que imprime `Hello World!` y termina.

Para generar un nuevo proyecto de .NET Core

1. Abra una nueva ventana de símbolo del sistema y desplácese hasta la carpeta de su usuario.

```
> cd %USERPROFILE%
```

2. Utilice el comando `dotnet new` para generar un nuevo proyecto de .NET Core.

```
C:\Users\username> dotnet new console -o dotnet-core-tutorial
Content generation time: 65.0152 ms
```

```
The template "Console Application" created successfully.  
C:\Users\username> cd dotnet-core-tutorial
```

3. Utilice el comando `dotnet restore` para instalar las dependencias.

```
C:\Users\username\dotnet-core-tutorial> dotnet restore  
Restoring packages for C:\Users\username\dotnet-core-tutorial\dotnet-core-  
tutorial.csproj...  
Generating MSBuild file C:\Users\username\dotnet-core-tutorial\obj\dotnet-core-  
tutorial.csproj.nuget.g.props.  
Generating MSBuild file C:\Users\username\dotnet-core-tutorial\obj\dotnet-core-  
tutorial.csproj.nuget.g.targets.  
Writing lock file to disk. Path: C:\Users\username\dotnet-core-tutorial\obj  
\project.assets.json  
Restore completed in 1.25 sec for C:\Users\username\dotnet-core-tutorial\dotnet-  
core-tutorial.csproj.  
  
NuGet Config files used:  
  C:\Users\username\AppData\Roaming\NuGet\NuGet.Config  
  C:\Program Files (x86)\NuGet\Config\Microsoft.VisualStudio.Offline.config  
Feeds used:  
  https://api.nuget.org/v3/index.json  
  C:\Program Files (x86)\Microsoft SDKs\NuGetPackages\
```

4. Utilice el comando `dotnet run` para compilar y ejecutar la aplicación localmente.

```
C:\Users\username\dotnet-core-tutorial> dotnet run  
Hello World!
```

Lance un entorno de Elastic Beanstalk

Utilice la consola de Elastic Beanstalk para lanzar un entorno de Elastic Beanstalk. En este ejemplo, lanzará con una plataforma.NET. Después de lanzar y configurar el entorno, puede implementar nuevo código fuente en cualquier momento.

Para lanzar un entorno (consola)

1. [Abra la consola de Elastic Beanstalk mediante este enlace preconfigurado:](https://console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=Tutorials&EnvironmentType=LoadBalanced)

2. En Platform (Plataforma), seleccione la plataforma y la ramificación de plataforma que coincidan con el idioma utilizado por la aplicación.
3. En Application code (Código de aplicación), seleccione Sample application (Aplicación de muestra).
4. Elija Review and launch (Revisar y lanzar).
5. Revise las opciones disponibles. Elige la opción disponible que quieras usar y, cuando estés listo, elige Create app (Crear aplicación).

Se tarda alrededor de diez minutos en crear el entorno. Durante este tiempo, puede actualizar el código fuente.

Actualizar el código fuente

Cambie la aplicación predeterminada a aplicación web que utiliza ASP.NET e IIS.

- ASP.NET es la plataforma de sitios web de .NET.
- IIS es el servidor web que ejecuta la aplicación en las instancias de Amazon EC2 de su entorno de Elastic Beanstalk.

Los ejemplos de código fuente que se muestran a [dotnet-core-tutorial-sourcecontinuación están disponibles aquí: .zip](#)

Note

El siguiente procedimiento muestra cómo convertir el código del proyecto en aplicación web. Para simplificar el proceso, puede generar el proyecto como aplicación web desde el principio. En la sección anterior [Generar un proyecto de .NET Core](#), modifique el comando del paso `dotnet new` con el siguiente comando.

```
C:\Users\username> dotnet new web -o dotnet-core-tutorial -n WindowsSampleApp
```

Para añadir compatibilidad con ASP.NET e IIS al código

1. Copie `Program.cs` en su directorio de aplicaciones para que se ejecute como un constructor de hosts web.

Example c:\users\username\dotnet-core-tutorial\ Program.cs

```
namespace Microsoft.AspNetCore.Hosting;
using WindowsSampleApp;

public static class Program
{
    public static void Main(string[] args)
    {
        CreateWebHostBuilder(args).Build().Run();
    }

    public static IWebHostBuilder CreateWebHostBuilder(string[] args) =>
        WebHost.CreateDefaultBuilder(args).UseStartup<Startup>();
}
```

2. Añada Startup.cs para ejecutar un sitio web de ASP.NET.

Example c:\users\username\dotnet-core-tutorial\ Startup.cs

```
namespace WindowsSampleApp
{
    public class Startup
    {
        public void Configure(IApplicationBuilder app)
        {
            app.UseRouting();
            app.UseEndpoints(endpoints =>
            {
                endpoints.MapGet("/", () => "Hello World from Elastic Beanstalk");
            });
        }
    }
}
```

3. Añada WindowsSampleApp.csproj que incluye middleware de IIS e incluye el archivo web.config desde el resultado de dotnet publish.

Note

El siguiente ejemplo se ha desarrollado con .NET Core Runtime 2.2.1. Es posible que tenga que modificar los valores de los atributos TargetFramework o Version de

los elementos `PackageReference` para que coincidan con la versión de .NET Core Runtime que utiliza en sus proyectos personalizados.

Example `c:\users\username\dotnet-core-tutorial\WindowsSampleApp.csproj`

```
<Project Sdk="Microsoft.NET.Sdk.Web">

  <PropertyGroup>
    <TargetFramework>net6.0</TargetFramework>
    <RollForward>LatestMajor</RollForward>
    <Nullable>enable</Nullable>
    <ImplicitUsings>enable</ImplicitUsings>
    <RootNamespace>WindowsSampleApp</RootNamespace>
  </PropertyGroup>

</Project>
```

A continuación, instale las nuevas dependencias y ejecute el sitio web de ASP.NET localmente.

Para ejecutar el sitio web localmente

1. Utilice el comando `dotnet restore` para instalar las dependencias.
2. Utilice el comando `dotnet run` para compilar y ejecutar la aplicación localmente.
3. Abra localhost:5000 para ver el sitio.

Para ejecutar la aplicación en un servidor web, debe agrupar el código fuente compilado con un archivo de configuración `web.config` y las dependencias del runtime. La herramienta `dotnet` proporciona un comando `publish` que recopila estos archivos en un directorio en función de la configuración de `dotnet-core-tutorial.csproj`.

Para crear el sitio web


- Utilice el comando `dotnet publish` para enviar el código compilado y las dependencias a una carpeta con el nombre `site`.

```
C:\users\username\dotnet-core-tutorial> dotnet publish -o site
```

Para implementar la aplicación en Elastic Beanstalk, empaquete el archivo del sitio con un [manifiesto de implementación](#). Esto indica a Elastic Beanstalk cómo ejecutarla.

Para crear un paquete de código fuente

1. Añada los archivos de la carpeta a un archivo ZIP.

 Note

Si utiliza una utilidad ZIP diferente, asegúrese de añadir todos los archivos a la carpeta raíz del archivo ZIP resultante. Esto es necesario para una implementación correcta de la aplicación en su entorno de Elastic Beanstalk.

```
C:\users\username\dotnet-core-tutorial> cd site
C:\users\username\dotnet-core-tutorial\site> zip ../site.zip *
  adding: dotnet-core-tutorial.deps.json (164 bytes security) (deflated 84%)
  adding: dotnet-core-tutorial.dll (164 bytes security) (deflated 59%)
  adding: dotnet-core-tutorial.pdb (164 bytes security) (deflated 28%)
  adding: dotnet-core-tutorial.runtimeconfig.json (164 bytes security) (deflated
26%)
  adding: Microsoft.AspNetCore.Authentication.Abstractions.dll (164 bytes security)
(deflated 49%)
  adding: Microsoft.AspNetCore.Authentication.Core.dll (164 bytes security)
(deflated 57%)
  adding: Microsoft.AspNetCore.Connections.Abstractions.dll (164 bytes security)
(deflated 51%)
  adding: Microsoft.AspNetCore.Hosting.Abstractions.dll (164 bytes security)
(deflated 49%)
  adding: Microsoft.AspNetCore.Hosting.dll (164 bytes security) (deflated 60%)
  adding: Microsoft.AspNetCore.Hosting.Server.Abstractions.dll (164 bytes security)
(deflated 44%)
  adding: Microsoft.AspNetCore.Http.Abstractions.dll (164 bytes security) (deflated
54%)
  adding: Microsoft.AspNetCore.Http.dll (164 bytes security) (deflated 55%)
  adding: Microsoft.AspNetCore.Http.Extensions.dll (164 bytes security) (deflated
50%)
  adding: Microsoft.AspNetCore.Http.Features.dll (164 bytes security) (deflated
50%)
  adding: Microsoft.AspNetCore.HttpOverrides.dll (164 bytes security) (deflated
49%)
```

```
adding: Microsoft.AspNetCore.Server.IISIntegration.dll (164 bytes security)
(deflated 46%)
adding: Microsoft.AspNetCore.Server.Kestrel.Core.dll (164 bytes security)
(deflated 63%)
adding: Microsoft.AspNetCore.Server.Kestrel.dll (164 bytes security) (deflated
46%)
adding: Microsoft.AspNetCore.Server.Kestrel.Https.dll (164 bytes security)
(deflated 44%)
adding: Microsoft.AspNetCore.Server.Kestrel.Transport.Abstractions.dll (164 bytes
security) (deflated 56%)
adding: Microsoft.AspNetCore.Server.Kestrel.Transport.Sockets.dll (164 bytes
security) (deflated 51%)
adding: Microsoft.AspNetCore.WebUtilities.dll (164 bytes security) (deflated 55%)
adding: Microsoft.Extensions.Configuration.Abstractions.dll (164 bytes security)
(deflated 48%)
adding: Microsoft.Extensions.Configuration.Binder.dll (164 bytes security)
(deflated 47%)
adding: Microsoft.Extensions.Configuration.dll (164 bytes security) (deflated
46%)
adding: Microsoft.Extensions.Configuration.EnvironmentVariables.dll (164 bytes
security) (deflated 46%)
adding: Microsoft.Extensions.Configuration.FileExtensions.dll (164 bytes
security) (deflated 47%)
adding: Microsoft.Extensions.DependencyInjection.Abstractions.dll (164 bytes
security) (deflated 54%)
adding: Microsoft.Extensions.DependencyInjection.dll (164 bytes security)
(deflated 53%)
adding: Microsoft.Extensions.FileProviders.Abstractions.dll (164 bytes security)
(deflated 46%)
adding: Microsoft.Extensions.FileProviders.Physical.dll (164 bytes security)
(deflated 47%)
adding: Microsoft.Extensions.FileSystemGlobbing.dll (164 bytes security)
(deflated 49%)
adding: Microsoft.Extensions.Hosting.Abstractions.dll (164 bytes security)
(deflated 47%)
adding: Microsoft.Extensions.Logging.Abstractions.dll (164 bytes security)
(deflated 54%)
adding: Microsoft.Extensions.Logging.dll (164 bytes security) (deflated 48%)
adding: Microsoft.Extensions.ObjectPool.dll (164 bytes security) (deflated 45%)
adding: Microsoft.Extensions.Options.dll (164 bytes security) (deflated 53%)
adding: Microsoft.Extensions.Primitives.dll (164 bytes security) (deflated 50%)
adding: Microsoft.Net.Http.Headers.dll (164 bytes security) (deflated 53%)
adding: System.IO.Pipelines.dll (164 bytes security) (deflated 50%)
```



```
adding: System.Runtime.CompilerServices.Unsafe.dll (164 bytes security) (deflated 43%)
adding: System.Text.Encodings.Web.dll (164 bytes security) (deflated 57%)
adding: web.config (164 bytes security) (deflated 39%)
C:\users\username\dotnet-core-tutorial\site> cd ../
```

2. Añada un manifiesto de implementación que apunte al archivo del sitio.

Example c:\users\username\ .json dotnet-core-tutorial aws-windows-deployment-manifest

```
{
  "manifestVersion": 1,
  "deployments": {
    "aspNetCoreWeb": [
      {
        "name": "test-dotnet-core",
        "parameters": {
          "appBundle": "site.zip",
          "iisPath": "/",
          "iisWebSite": "Default Web Site"
        }
      }
    ]
  }
}
```

3. Utilice el comando zip para crear un paquete de código fuente denominado dotnet-core-tutorial.zip.

```
C:\users\username\dotnet-core-tutorial> zip dotnet-core-tutorial.zip site.zip aws-
windows-deployment-manifest.json
adding: site.zip (164 bytes security) (stored 0%)
adding: aws-windows-deployment-manifest.json (164 bytes security) (deflated 50%)
```

Implementar la aplicación

Implemente el paquete de código fuente en el entorno de Elastic Beanstalk creado.

[Puede descargar el paquete fuente aquí: .zip dotnet-core-tutorial-bundle](#)

Para implementar un paquete de código fuente

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En la página de información general del entorno, elija Upload and deploy (Cargar e implementar).
4. Utilice el cuadro de diálogo en pantalla para cargar el paquete de código fuente.
5. Elija Deploy (Implementar).
6. Cuando se complete la implementación, seleccione la URL del para abrir el sitio web en una nueva pestaña.

La aplicación simplemente escribe `Hello from ASP.NET Core!` en la respuesta y termina de ejecutarse.

A screenshot of a terminal or browser output window with a decorative border. The text inside reads "Hello from ASP.NET Core!".

El lanzamiento de un entorno crea los siguientes recursos:

- Instancia de EC2: una máquina virtual de Amazon Elastic Compute Cloud (Amazon EC2) configurada para ejecutar aplicaciones web en la plataforma que elija.

Cada plataforma ejecuta un conjunto específico de software, archivos de configuración y scripts compatibles con una determinada versión de lenguaje, marco de trabajo y contenedor web (o una combinación de estos). La mayoría de las plataformas utilizan Apache o nginx como un proxy inverso que se sitúa delante de la aplicación web, reenvía las solicitudes a esta, administra los recursos estáticos y genera registros de acceso y errores.

- Instance security group (Grupo de seguridad de la instancia): grupo de seguridad de Amazon EC2 configurado para permitir el tráfico entrante en el puerto 80. Este recurso permite que el tráfico HTTP procedente del balanceador de carga llegue a la instancia EC2 en la que se ejecuta la aplicación web. De forma predeterminada, el tráfico no está permitido en otros puertos.

- **Balanceador de carga:** balanceador de carga de Elastic Load Balancing configurado para distribuir solicitudes a las instancias que se ejecutan en la aplicación. Los balanceadores de carga también permiten que las instancias no estén expuestas directamente a Internet.
- **Grupo de seguridad del balanceador de carga:** grupo de seguridad de Amazon EC2 configurado para permitir el tráfico entrante en el puerto 80. Este recurso permite que el tráfico HTTP procedente de Internet llegue al balanceador de carga. De forma predeterminada, el tráfico no está permitido en otros puertos.
- **Grupo de Auto Scaling:** grupo de Auto Scaling configurado para reemplazar una instancia si termina o deja de estar disponible.
- **Bucket de Amazon S3:** ubicación de almacenamiento para el código fuente, los registros y otros artefactos que se crean al utilizar Elastic Beanstalk.
- **CloudWatch Alarmas de Amazon:** dos CloudWatch alarmas que monitorean la carga de las instancias de su entorno y que se activan si la carga es demasiado alta o demasiado baja. Cuando se activa una alarma, en respuesta, el grupo de Auto Scaling aumenta o reduce los recursos.
- **AWS CloudFormation pila:** Elastic AWS CloudFormation Beanstalk se utiliza para lanzar los recursos de su entorno y propagar los cambios de configuración. Los recursos se definen en una plantilla que puede verse en la [consola de AWS CloudFormation](#).
- **Nombre de dominio:** nombre de dominio que direcciona el tráfico a la aplicación web con el formato *subdominio.región.elasticbeanstalk.com*.

Note

Para aumentar la seguridad de las aplicaciones de Elastic Beanstalk, el dominio elasticbeanstalk.com está registrado en la [lista de sufijos públicos \(PSL\)](#). Para mayor seguridad, se recomienda que utilice cookies con un prefijo `__Host-` en caso de que necesite configurar cookies confidenciales en el nombre de dominio predeterminado de sus aplicaciones de Elastic Beanstalk. Esta práctica le ayudará a defender su dominio contra intentos de falsificación de solicitudes entre sitios (CSRF). Para obtener más información, consulte la página de [configuración de cookies](#) en la red de desarrolladores de Mozilla.

Todos estos recursos los administra Elastic Beanstalk. Cuando termina su entorno, Elastic Beanstalk termina todos los recursos que este contiene.

Note

El bucket de Amazon S3 que crea Elastic Beanstalk se comparte entre los entornos y no se elimina durante la terminación del entorno. Para obtener más información, consulte [Usar Elastic Beanstalk con Amazon S3](#).

Limpieza

Cuando termine de trabajar con Elastic Beanstalk, puede terminar su entorno. [Elastic Beanstalk AWS cancela todos los recursos asociados al entorno, como las instancias de Amazon EC2, las instancias de bases de datos, los balanceadores de carga, los grupos de seguridad y las alarmas.](#)

Para cerrar el entorno de Elastic Beanstalk desde la consola

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. Elija Actions (Acciones) y, a continuación, Terminate Environment (Terminar el entorno).
4. Utilice el cuadro de diálogo en pantalla para confirmar la terminación del entorno.

Con Elastic Beanstalk, puede crear fácilmente un nuevo entorno para su aplicación en cualquier momento.

Pasos siguientes

A medida que siga desarrollando la aplicación, es probable que desee administrar los entornos e implementar su aplicación sin tener que crear manualmente un archivo .zip y cargarlo en la consola de Elastic Beanstalk. La interfaz de [línea de comandos de Elastic Beanstalk](#) (EB CLI easy-to-use) proporciona comandos para crear, configurar e implementar aplicaciones en entornos de Elastic Beanstalk desde la línea de comandos.

Si usa Visual Studio para desarrollar su aplicación, también puede usarlo AWS Toolkit for Visual Studio para implementar cambios, administrar sus entornos de Elastic Beanstalk y administrar otros recursos. Para obtener más información, consulte [El AWS Toolkit for Visual Studio](#).

Para el desarrollo y las pruebas, tal vez desee utilizar la funcionalidad de Elastic Beanstalk para añadir una instancia de base de datos administrada directamente a su entorno. Para obtener instrucciones acerca de cómo configurar una base de datos dentro del entorno, consulte [Añadir una base de datos al entorno de Elastic Beanstalk](#).

Por último, si pretende utilizar su aplicación en un entorno de producción, [configure un nombre de dominio personalizado](#) para su entorno y [habilite HTTPS](#) para las conexiones seguras.

Configuración del entorno de desarrollo de .NET

Configure el entorno de desarrollo de .NET para probar la aplicación localmente antes de implementarla en AWS Elastic Beanstalk. En este tema, se explican los pasos de configuración del entorno de desarrollo y se incluyen enlaces a páginas de instalación donde encontrará herramientas útiles.

Para ver procedimientos de configuración y herramientas comunes que se utilizan en todos los lenguajes, consulte [Configuración de su máquina de desarrollo para su uso con Elastic Beanstalk](#).

Secciones

- [Instalación de un IDE](#)
- [Instalación de AWS Toolkit for Visual Studio](#)

Si necesita administrar recursos de AWS en la aplicación, instale el SDK de AWS para .NET. Por ejemplo, puede utilizar Amazon S3 para almacenar y recuperar datos.

Con el SDK de AWS para .NET, puede ponerse en marcha en cuestión de minutos con un único paquete descargable que incluye plantillas de proyecto de Visual Studio, la biblioteca AWS .NET, ejemplos de código en C# y documentación. Se proporcionan ejemplos prácticos en C# sobre el uso de las bibliotecas para crear aplicaciones. Se proporcionan también tutoriales de vídeo online y documentación de referencia le que ayudarán a aprender a usar las bibliotecas y los ejemplos de código.

Para obtener más información e instrucciones de instalación, visite la [página de inicio del SDK de AWS para .NET](#).

Instalación de un IDE

Los entornos de desarrollo integrado (IDE) cuentan con una amplia gama de características que facilitan el desarrollo de aplicaciones. Si no ha usado un IDE para el desarrollo en .NET, pruebe Visual Studio Community para empezar.

Visite la página [Visual Studio Community](#) para descargar e instalar Visual Studio Community.

Instalación de AWS Toolkit for Visual Studio

[AWS Toolkit for Visual Studio](#) es un complemento de código abierto para el IDE de Visual Studio que facilita a los desarrolladores las tareas de desarrollo, depuración e implementación de aplicaciones .NET mediante AWS. Visite la [página de inicio de Toolkit for Visual Studio](#) para consultar las instrucciones de instalación.

Uso de la plataforma .NET de Elastic Beanstalk

AWS Elastic Beanstalk admite varias plataformas para diferentes versiones del marco de programación de .NET y Windows Server. Consulte [.NET en Windows Server con IIS](#) en el documento Plataformas de AWS Elastic Beanstalk para ver una lista completa.

Elastic Beanstalk proporciona [opciones de configuración](#) que se pueden usar para personalizar el software que se ejecuta en las instancias EC2 del entorno de Elastic Beanstalk. También puede configurar las variables de entorno que necesita la aplicación, habilitar la rotación de registros en Amazon S3 y establecer la configuración del marco .NET.

En la consola de Elastic Beanstalk hay opciones de configuración disponibles para [modificar la configuración de un entorno en ejecución](#). Para evitar perder la configuración del entorno cuando lo termina, puede usar las [configuraciones guardadas](#) para guardar la configuración y aplicarla posteriormente a otro entorno.

Para guardar la configuración en el código fuente, puede incluir [archivos de configuración](#). Los valores de configuración de los archivos de configuración se aplican cada vez que crea un entorno o que implementa la aplicación. También puede usar archivos de configuración para instalar paquetes, ejecutar scripts y llevar a cabo otras operaciones de personalización de instancias durante las implementaciones.

La configuración aplicada en la consola de Elastic Beanstalk anula la misma configuración en los archivos de configuración, si existe. Esto le permite tener la configuración predeterminada en los archivos de configuración y anularla con la configuración específica del entorno en la consola.

Para obtener más información acerca de la prioridad y otros métodos para cambiar valores de configuración, consulte [Opciones de configuración](#).

Configuración del entorno de .NET en la consola de Elastic Beanstalk

Puede utilizar la consola de Elastic Beanstalk para habilitar la rotación de registros en Amazon S3, configurar variables que la aplicación pueda leer desde el entorno y cambiar la configuración del marco .NET.

Para configurar el entorno de .NET en la consola de Elastic Beanstalk

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Actualizaciones, supervisión y registro, seleccione Editar.

Opciones de contenedor

- Target .NET runtime (Tiempo de ejecución de .NET de destino): establezca en 2.0 para ejecutar CLR v2.
- Enable 32-bit applications (Habilitar aplicaciones de 32 bits): establezca en True para ejecutar aplicaciones de 32 bits.

Log Options (Opciones de registro)

La sección Log Options tiene dos valores:

- Instance profile (Perfil de instancia): especifica el perfil de instancia que tiene permiso para obtener acceso al bucket de Amazon S3 asociado con la aplicación.
- Enable log file rotation to Amazon S3 (Habilitar la rotación de archivos de registro para Amazon S3): especifica si los archivos de registro de las instancias de Amazon EC2 de la aplicación se copian en el bucket de Amazon S3 asociado a la aplicación.

Propiedades del entorno

La sección Environment Properties (Propiedades de entorno) le permite especificar opciones de configuración del entorno en las instancias de Amazon EC2 que ejecutan la aplicación. Estos ajustes se pasan como pares clave-valor a la aplicación. Utilice `System.EnvironmentVariable` para leerlos. Pueden existir claves idénticas en ambos `web.config` y como propiedades de entorno. Utilice el espacio de nombres `System.Configuration` para leer valores de `web.config`.

```
NameValueCollection appConfig = ConfigurationManager.AppSettings;  
string endpoint = appConfig["API_ENDPOINT"];
```

Para obtener más información, consulte [Propiedades del entorno y otras opciones del software](#).

El espacio de nombres aws:elasticbeanstalk:container:dotnet:apppool

Puede usar un [archivo de configuración](#) para definir opciones de configuración y realizar otras tareas de configuración en las instancias durante las implementaciones. Las opciones de configuración se pueden definir a través del servicio de Elastic Beanstalk o la plataforma que utilice y están organizadas por espacios de nombres.

La plataforma .NET define opciones en el espacio de nombres `aws:elasticbeanstalk:container:dotnet:apppool` que puede utilizar para configurar el tiempo de ejecución de .NET.

En el siguiente ejemplo, el archivo de configuración muestra los ajustes de cada una de las opciones disponibles en este espacio de nombres:

Example .ebextensions/dotnet-settings.config

```
option_settings:  
  aws:elasticbeanstalk:container:dotnet:apppool:  
    Target Runtime: 2.0  
    Enable 32-bit Applications: True
```

Elastic Beanstalk cuenta con numerosas opciones de configuración para personalizar el entorno. Además de los archivos de configuración, también puede definir opciones en la consola, configuraciones guardadas, la CLI de EB o la AWS CLI. Para obtener más información, consulte [Opciones de configuración](#).

Migración entre las versiones principales de la plataforma Windows Server de Elastic Beanstalk

AWS Elastic Beanstalk ha tenido varias versiones principales de su plataforma Windows Server. En esta página se describen las principales mejoras de cada versión principal y lo que se ha de tener en cuenta antes de migrar a una versión posterior.

La plataforma Windows Server se encuentra actualmente en la versión 2 (v2). Si la aplicación utiliza una versión de la plataforma de Windows Server anterior a la versión 2, le recomendamos que migre a dicha versión.

Novedades en las versiones principales de la plataforma Windows Server

Plataforma Windows Server versión 2

La versión 2 (v2) de la plataforma Windows Server de Elastic Beanstalk se [publicó en febrero de 2019](#). La versión 2 acerca el comportamiento de la plataforma Windows Server al de las plataformas basadas en Linux de Elastic Beanstalk de varias maneras importantes. La versión 2 es totalmente compatible con la versión 1, lo que facilita la migración desde la versión 1.

La plataforma Windows Server ahora es compatible con lo siguiente:

- Control de versiones: cada lanzamiento obtiene un nuevo número de versión y puede consultar las versiones anteriores (que aún estén disponibles) al crear y administrar los entornos.
- Estado mejorado: consulte [Informes y monitorización de estado mejorados](#) para obtener más información.
- Implementaciones inmutables y continuas con un lote adicional: para ver más información sobre las políticas de implementación, consulte [Implementación de aplicaciones en entornos Elastic Beanstalk](#).
- Actualizaciones inmutables: para obtener más información acerca de los tipos de actualización, consulte [Cambios de configuración](#).
- Actualizaciones de plataformas administradas: para obtener más información, consulte [Actualizaciones de plataforma administradas](#).

Note

Las nuevas características de implementación y actualización dependen del estado mejorado. Habilite el estado mejorado para utilizarlas. Para obtener más información, consulte [Habilitación de informes de estado mejorado de Elastic Beanstalk](#).

Plataforma Windows Server versión 1

La versión 1.0.0 (v1) de la plataforma Windows Server de Elastic Beanstalk se publicó en octubre de 2015. Esta versión cambia el orden en que Elastic Beanstalk procesa los comandos en [archivos de configuración](#) durante la creación y las actualizaciones del entorno.

Las versiones anteriores de la plataforma no tienen número de versión en el nombre de la pila de soluciones:

- Windows Server 2012 R2 de 64 bits con IIS 8.5
- Windows Server Core 2012 R2 de 64 bits con IIS 8.5
- Windows Server 2012 de 64 bits con IIS 8
- Windows Server 2008 R2 de 64 bits con IIS 7.5

En las versiones anteriores, el orden de procesamiento de los archivos de configuración no era coherente. Durante la creación del entorno, `Container Commands` se ejecutaba después de que el código fuente de la aplicación se implementaba en IIS. Durante las implementaciones en entornos en ejecución, los comandos de contenedor se ejecutaban antes de que se implementara la nueva versión. En las operaciones de escalado vertical, los archivos de configuración no se procesaban.

Además, IIS se iniciaba antes de que los comandos de contenedor se ejecutaran. Este comportamiento ha hecho que algunos clientes implementen soluciones en los comandos de contenedor, que detienen el servidor de IIS antes de que los comandos se ejecuten y lo inician de nuevo cuando se han completado.

La versión 1 corrige esta incoherencia y acerca el comportamiento de la plataforma Windows Server al de las plataformas basadas en Linux de Elastic Beanstalk. En la plataforma de la versión 1, Elastic Beanstalk siempre ejecuta los comandos de contenedor antes de iniciar el servidor de IIS.

Las pilas de la solución de la plataforma de la v1 tienen v1 detrás de la versión de Windows Server:

- Windows Server 2012 R2 v1.1.0 de 64 bits con IIS 8.5
- Windows Server Core 2012 R2 v1.1.0 de 64 bits con IIS 8.5
- Windows Server 2012 v1.1.0 de 64 bits con IIS 8
- Windows Server 2008 R2 v1.1.0 de 64 bits con IIS 7.5

Asimismo, la plataforma de la v1 extrae el contenido del paquete de código fuente de la aplicación en `C:\staging\` antes de ejecutar los comandos de contenedor. Una vez que los comandos de contenedor se han completado, el contenido de esta carpeta se comprime en un archivo .zip y se implementa en IIS. Este flujo de trabajo le permite modificar el contenido del paquete de código fuente de la aplicación utilizando comandos o un script antes de la implementación.

Migración desde versiones principales anteriores de la plataforma Windows Server

Consulte en esta sección las consideraciones de migración antes de actualizar el entorno. Para actualizar la plataforma del entorno a una versión más reciente, consulte [Actualización de la versión de la plataforma del entorno de Elastic Beanstalk](#).

De V1 a V2

La plataforma de Windows Server v2 no es compatible con .NET Core 1.x y 2.0. Si va a migrar su aplicación desde Windows Server v1 hasta v2 y la aplicación utiliza una de estas versiones de .NET Core, actualice la aplicación a una versión de .NET Core compatible con v2. Para ver una lista de versiones compatibles, consulte [.NET en Windows Server con IIS](#) en Plataformas de AWS Elastic Beanstalk .

Si la aplicación utiliza una imagen de máquina de Amazon (Amazon Machine Image, AMI) personalizada, cree una nuevo AMI personalizada basada en una AMI de la plataforma de Windows Server v2. Para obtener más información, consulte [Uso de una imagen de máquina de Amazon \(AMI\) personalizada](#).

Note

Las características de implementación y actualización que son nuevas en Windows Server v2 dependen del estado mejorado. Al migrar un entorno a v2, el estado mejorado está deshabilitado. Habilítelo para utilizar estas características. Para obtener más información, consulte [Habilitación de informes de estado mejorado de Elastic Beanstalk](#).

Antes de la versión 1

Además de las cuestiones relativas a la migración desde v1, si va a migrar la aplicación desde una pila de soluciones de Windows Server anterior a la v1 y actualmente utiliza comandos de contenedor, quite los comandos que haya agregado para evitar incoherencias de procesamiento al migrar a una versión más reciente. A partir de la v1, está garantizado que los comandos de contenedores se van a ejecutar por completo antes de que el código fuente de la aplicación que se implementa y antes de que IIS se inicie. Esto le permite realizar cambios en el código fuente en `C:\staging` y modificar los archivos de configuración de IIS durante este paso sin problemas.

Por ejemplo, puede utilizar el AWS CLI para descargar un archivo DLL a la fuente de la aplicación desde Amazon S3:

```
.ebextensions\copy-dll.config
```

```
container_commands:
  copy-dll:
    command: aws s3 cp s3://DOC-EXAMPLE-BUCKET/dlls/large-dll.dll .\lib\
```

Para obtener más información sobre cómo usar archivos de configuración, consulte [Personalización avanzada de entornos con archivos de configuración \(.ebextensions\)](#).

Ejecutar múltiples aplicaciones y aplicaciones centrales ASP.NET con un manifiesto de implementación

Puede utilizar un manifiesto de implementación para indicarle a Elastic Beanstalk cómo se va a implementar la aplicación. Al utilizar este método, no es necesario utilizar `MSDeploy` para generar un paquete de código fuente para una única aplicación ASP.NET que se ejecuta en la ruta raíz del sitio web. En su lugar, puede utilizar un archivo de manifiesto para ejecutar varias aplicaciones en rutas diferentes. O bien, puede indicarle a Elastic Beanstalk que implemente y ejecute la aplicación con ASP.NET Core. También puede utilizar un manifiesto de implementación para configurar un grupo de aplicaciones en el que ejecutará las aplicaciones.

Los manifiestos de implementación añaden compatibilidad con las [aplicaciones de .NET Core](#) en Elastic Beanstalk. Puede implementar una aplicación .NET Framework sin un manifiesto de implementación. Sin embargo, las aplicaciones .NET Core requieren un manifiesto de implementación para ejecutarse en Elastic Beanstalk. Si utiliza un manifiesto de implementación, debe crear un archivo del sitio para cada aplicación y empaquetar todos ellos después en un segundo ZIP que contenga el manifiesto de implementación.

Los manifiestos de implementación también brindan la posibilidad de [ejecutar varias aplicaciones en diferentes rutas](#). Un manifiesto de implementación define una serie de objetivos de implementación, cada uno de ellos con un archivo del sitio y una ruta en la que IIS debe ejecutarse. Por ejemplo, puede ejecutar una API web en la ruta `/api` para que atienda las solicitudes asincrónicas y una aplicación web en la ruta raíz que utilice la API.

También puede utilizar un manifiesto de implementación para [ejecutar varias aplicaciones mediante grupos de aplicaciones en IIS o Kestrel](#). Puede configurar un grupo de aplicaciones para reiniciar las aplicaciones periódicamente, ejecutar aplicaciones de 32 bits o utilizar una versión específica del entorno de ejecución de .NET Framework.

Para una personalización completa, puede [escribir sus propios scripts de implementación](#) en Windows PowerShell e indicar a Elastic Beanstalk qué scripts debe ejecutar para instalar, desinstalar y reiniciar la aplicación.

El manifiesto de implementación y las características relacionadas requieren una plataforma de Windows Server [versión 1.2.0 o posterior](#).

Secciones

- [Aplicaciones de .NET Core](#)
- [Ejecutar múltiples aplicaciones](#)
- [Configurar grupos de aplicaciones](#)
- [Definir implementaciones personalizadas](#)

Aplicaciones de .NET Core

Puede utilizar un manifiesto de implementación para ejecutar aplicaciones .NET Core en Elastic Beanstalk. .NET Core es una versión multiplataforma de .NET que incluye una herramienta de línea de comandos (`dotnet`). Puede utilizarla para generar una aplicación, ejecutarla localmente y prepararla para su publicación.

Note

Visite [Tutorial: Implementación de una aplicación ASP.NET Core con Elastic Beanstalk](#) para ver un tutorial y una aplicación de ejemplo donde se usa un manifiesto de implementación para ejecutar una aplicación de .NET Core en Elastic Beanstalk.

Para ejecutar una aplicación de .NET Core en Elastic Beanstalk, puede ejecutar `dotnet publish` y empaquetar la salida en un archivo ZIP, sin incluir los directorios que contiene. Coloque el archivo del sitio en un paquete de código fuente con un manifiesto de implementación que tenga un destino de implementación de tipo `aspNetCoreWeb`.

El siguiente manifiesto de implementación ejecuta una aplicación de .NET Core desde un archivo del sitio llamado `dotnet-core-app.zip` situado en la ruta raíz.

Example `aws-windows-deployment-manifest.json`: .NET core

```
{
  "manifestVersion": 1,
  "deployments": {
    "aspNetCoreWeb": [
      {
        "name": "my-dotnet-core-app",
        "parameters": {
          "archive": "dotnet-core-app.zip",
          "iisPath": "/"
        }
      }
    ]
  }
}
```

Empaquete el manifiesto y el archivo del sitio en un ZIP para crear un paquete de código fuente.

Example `dotnet-core-bundle.zip`

```
.
|-- aws-windows-deployment-manifest.json
`-- dotnet-core-app.zip
```

El archivo del sitio contiene el código compilado de la aplicación, las dependencias y el archivo `web.config`.

Example `dotnet-core-app.zip`

```
.
|-- Microsoft.AspNetCore.Hosting.Abstractions.dll
|-- Microsoft.AspNetCore.Hosting.Server.Abstractions.dll
|-- Microsoft.AspNetCore.Hosting.dll
```

```
|-- Microsoft.AspNetCore.Http.Abstractions.dll
|-- Microsoft.AspNetCore.Http.Extensions.dll
|-- Microsoft.AspNetCore.Http.Features.dll
|-- Microsoft.AspNetCore.Http.dll
|-- Microsoft.AspNetCore.HttpOverrides.dll
|-- Microsoft.AspNetCore.Server.IISIntegration.dll
|-- Microsoft.AspNetCore.Server.Kestrel.dll
|-- Microsoft.AspNetCore.WebUtilities.dll
|-- Microsoft.Extensions.Configuration.Abstractions.dll
|-- Microsoft.Extensions.Configuration.EnvironmentVariables.dll
|-- Microsoft.Extensions.Configuration.dll
|-- Microsoft.Extensions.DependencyInjection.Abstractions.dll
|-- Microsoft.Extensions.DependencyInjection.dll
|-- Microsoft.Extensions.FileProviders.Abstractions.dll
|-- Microsoft.Extensions.FileProviders.Physical.dll
|-- Microsoft.Extensions.FileSystemGlobbing.dll
|-- Microsoft.Extensions.Logging.Abstractions.dll
|-- Microsoft.Extensions.Logging.dll
|-- Microsoft.Extensions.ObjectPool.dll
|-- Microsoft.Extensions.Options.dll
|-- Microsoft.Extensions.PlatformAbstractions.dll
|-- Microsoft.Extensions.Primitives.dll
|-- Microsoft.Net.Http.Headers.dll
|-- System.Diagnostics.Contracts.dll
|-- System.Net.WebSockets.dll
|-- System.Text.Encodings.Web.dll
|-- dotnet-core-app.deps.json
|-- dotnet-core-app.dll
|-- dotnet-core-app.pdb
|-- dotnet-core-app.runtimeconfig.json
`-- web.config
```

Consulte [el tutorial](#) para ver un ejemplo completo.

Ejecutar múltiples aplicaciones

Para ejecutar varias aplicaciones con un manifiesto de implementación, debe definir diferentes objetivos de implementación.

El manifiesto de implementación siguiente configura dos aplicaciones .NET Core. La `WebApiSampleApp` aplicación implementa una API web sencilla y atiende solicitudes asíncronas en la ruta `/api`. La aplicación `DotNetSampleApp` es una aplicación web que atiende solicitudes en la ruta raíz.

Example aws-windows-deployment-manifest.json: múltiples aplicaciones

```
{
  "manifestVersion": 1,
  "deployments": {
    "aspNetCoreWeb": [
      {
        "name": "WebAPISample",
        "parameters": {
          "appBundle": "WebApiSampleApp.zip",
          "iisPath": "/api"
        }
      },
      {
        "name": "DotNetSample",
        "parameters": {
          "appBundle": "DotNetSampleApp.zip",
          "iisPath": "/"
        }
      }
    ]
  }
}
```

Aquí hay disponible una aplicación de ejemplo con varias aplicaciones:

- [Paquete de código fuente desplegable: -v2.zip dotnet-multiapp-sample-bundle](#)
- Código fuente: [-v2.zip dotnet-multiapp-sample-source](#)

Configurar grupos de aplicaciones

Puede admitir varias aplicaciones en su entorno Windows. Hay dos enfoques disponibles:

- Puede utilizar el modelo de out-of-process alojamiento con el servidor web Kestrel. Con este modelo, puede configurar varias aplicaciones para que se ejecuten en un grupo de aplicaciones.
- Puede utilizar el modelo de alojamiento en proceso. Con este modelo, se utilizan varios grupos de aplicaciones para ejecutar varias aplicaciones con una sola aplicación en cada grupo. Si utiliza el servidor IIS y necesita ejecutar varias aplicaciones, debe utilizar este enfoque.

Si desea configurar Kestrel para que ejecute varias aplicaciones en un grupo de aplicaciones, agregue `hostingModel="OutOfProcess"` en el archivo `web.config`. Considere los siguientes ejemplos:

Example `web.config`: para el modelo de alojamiento de Kestrel out-of-process

```
<configuration>
<location path="." inheritInChildApplications="false">
<system.webServer>
<handlers>
<add
  name="aspNetCore"
  path="*" verb="*"
  modules="AspNetCoreModuleV2"
  resourceType="Unspecified" />
</handlers>
<aspNetCore
  processPath="dotnet"
  arguments=".\CoreWebApp-5-0.dll"
  stdoutLogEnabled="false"
  stdoutLogFile=".\logs\stdout"
  hostingModel="OutOfProcess" />
</system.webServer>
</location>
</configuration>
```

Example `aws-windows-deployment-manifest.json`: múltiples aplicaciones

```
{
  "manifestVersion": 1,
  "deployments": {"msDeploy": [
    {"name": "Web-app1",
      "parameters": {"archive": "site1.zip",
        "iisPath": "/"
      }
    },
    {"name": "Web-app2",
      "parameters": {"archive": "site2.zip",
        "iisPath": "/app2"
      }
    }
  ]
}
```

```
}
```

IIS no admite varias aplicaciones en un grupo de aplicaciones porque utiliza el modelo de alojamiento en proceso. Por lo tanto, debe configurar varias aplicaciones mediante la asignación de cada aplicación a un grupo de aplicaciones. En otras palabras, asigne solo una aplicación a un grupo de aplicaciones.

Puede configurar IIS para que use diferentes grupos de aplicaciones en el archivo `aws-windows-deployment-manifest.json`. Realice las siguientes actualizaciones al consultar el siguiente archivo de ejemplo:

- Agregue una sección `iisConfig` que incluya una subsección llamada `appPools`.
- En el bloque `appPools`, enumere los grupos de aplicaciones.
- En la sección `deployments`, defina una sección `parameters` para cada aplicación.
- Para cada aplicación, la sección `parameters` especifica un archivo, una ruta de acceso para ejecutarlo y una `appPool` en la que se ejecutará.

El siguiente manifiesto de implementación configura dos grupos de aplicaciones que reinician su aplicación cada 10 minutos. También adjuntan sus aplicaciones a una aplicación web .NET Framework que se ejecuta en la ruta especificada.

Example `aws-windows-deployment-manifest.json`: una aplicación por grupo de aplicaciones

```
{
  "manifestVersion": 1,
  "iisConfig": {"appPools": [
    {"name": "MyFirstPool",
      "recycling": {"regularTimeInterval": 10}
    },
    {"name": "MySecondPool",
      "recycling": {"regularTimeInterval": 10}
    }
  ]
},
  "deployments": {"msDeploy": [
    {"name": "Web-app1",
      "parameters": {
        "archive": "site1.zip",
        "iisPath": "/"
      }
    }
  ]
}
```

```
        "appPool": "MyFirstPool"
      }
    },
    {"name": "Web-app2",
     "parameters": {
       "archive": "site2.zip",
       "iisPath": "/app2",
       "appPool": "MySecondPool"
     }
    }
  ]
}
```

Definir implementaciones personalizadas

Para tener aún más control, puede personalizar completamente la implementación de una aplicación y definir una implementación personalizada.

El siguiente manifiesto de implementación indica a Elastic Beanstalk que ejecute un script `install` denominado `siteInstall.ps1`. Este script instala el sitio web durante el inicio y las implementaciones de la instancia. Además, el manifiesto de implementación también indica a Elastic Beanstalk que `uninstall` ejecute un script antes de instalar una nueva versión durante una implementación `restart` y un script para reiniciar la aplicación [si selecciona Reiniciar App Server en la consola de administración](#). AWS

Example `aws-windows-deployment-manifest.json`: implementación personalizada

```
{
  "manifestVersion": 1,
  "deployments": {
    "custom": [
      {
        "name": "Custom site",
        "scripts": {
          "install": {
            "file": "siteInstall.ps1"
          },
          "restart": {
            "file": "siteRestart.ps1"
          },
          "uninstall": {
```

```
        "file": "siteUninstall.ps1"  
      }  
    }  
  ]  
}  
}
```

Incluya los artefactos necesarios para ejecutar la aplicación en el paquete de código fuente con el manifiesto y los scripts.

Example Custom-site-bundle: .zip

```
.  
|-- aws-windows-deployment-manifest.json  
|-- siteInstall.ps1  
|-- siteRestart.ps1  
|-- siteUninstall.ps1  
`-- site-contents.zip
```

Añadir una instancia de base de datos de Amazon RDS a un entorno de aplicaciones .NET

Puede utilizar una instancia de base de datos de Amazon Relational Database Service (Amazon RDS) para almacenar datos recopilados y modificados por su aplicación. La base de datos se puede adjuntar al entorno y administrarse con Elastic Beanstalk, o bien, crearse como desacoplada y administrarse de forma externa a través de otro servicio. En este tema se proporcionan instrucciones para crear un Amazon RDS mediante la consola de Elastic Beanstalk. La base de datos se acoplará al entorno y se administrará con Elastic Beanstalk. Para obtener más información sobre el uso de Elastic Beanstalk con Amazon RDS, consulte [Añadir una base de datos al entorno de Elastic Beanstalk](#).

Secciones

- [Añadir una instancia de base de datos al entorno](#)
- [Descargar un controlador](#)
- [Conexión a una base de datos](#)

Añadir una instancia de base de datos al entorno

Para agregar una instancia de base de datos al entorno

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Database (Base de datos), seleccione Edit (Editar).
5. Seleccione un motor de base de datos y escriba el nombre de usuario y la contraseña.
6. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

Se tarda alrededor de 10 minutos en añadir una instancia de base de datos. Cuando se haya completado la actualización del entorno, el nombre de host de la instancia de base de datos y otra información de conexión estarán disponibles para su aplicación mediante las siguientes propiedades de entorno

Nombre de la propiedad	Descripción	Valor de la propiedad
RDS_HOSTNAME	El nombre de host de la instancia de base de datos.	En la pestaña Connectivity & security (Conectividad y seguridad) de la consola de Amazon RDS: Endpoint (Punto de enlace).
RDS_PORT	El puerto en donde la instancia de base de datos acepta las conexiones. El valor predeterminado varía	En la pestaña Connectivity & security (Conectividad y seguridad) de la consola de Amazon RDS: Port (Puerto).

Nombre de la propiedad	Descripción	Valor de la propiedad
	dependiendo del motor de base de datos.	
RDS_DB_NAME	El nombre de la base de dato, ebdb .	En la pestaña Configuration (Configuración) de la consola de Amazon RDS: DB Name (Nombre de la base de datos).
RDS_USERNAME	El nombre de usuario que ha configurado para la base de datos.	En la pestaña Configuration (Configuración) de la consola de Amazon RDS: Master username (Nombre de usuario maestro).
RDS_PASSWORD	La contraseña que ha configurado para la base de datos.	No está disponible como referencia en la consola de Amazon RDS.

Para obtener más información sobre la configuración de una instancia de bases de datos acopladas a un entorno de Elastic Beanstalk, consulte [Añadir una base de datos al entorno de Elastic Beanstalk](#).

Descargar un controlador

Descargue e instale el paquete EntityFramework y un controlador de base de datos para el entorno de desarrollo con NuGet.

Proveedores de bases de datos de Common Entity Framework para .NET

- SQL Server – `Microsoft.EntityFrameworkCore.SqlServer`
- MySQL – `Pomelo.EntityFrameworkCore.MySql`
- PostgreSQL – `Npgsql.EntityFrameworkCore.PostgreSQL`

Conexión a una base de datos

Elastic Beanstalk proporciona información de conexión para instancias de base de datos asociadas en propiedades de entorno. Use `ConfigurationManager.AppSettings` para leer las propiedades y configurar una conexión con la base de datos.

Example `Helpers.cs`: método de cadena de conexión

```
using System;
using System.Collections.Generic;
using System.Configuration;
using System.Linq;
using System.Web;

namespace MVC5App.Models
{
    public class Helpers
    {
        public static string GetRDSConnectionString()
        {
            var appConfig = ConfigurationManager.AppSettings;

            string dbname = appConfig["RDS_DB_NAME"];

            if (string.IsNullOrEmpty(dbname)) return null;

            string username = appConfig["RDS_USERNAME"];
            string password = appConfig["RDS_PASSWORD"];
            string hostname = appConfig["RDS_HOSTNAME"];
            string port = appConfig["RDS_PORT"];

            return "Data Source=" + hostname + ";Initial Catalog=" + dbname + ";User ID=" +
                username + ";Password=" + password + ";";
        }
    }
}
```

Utilice la cadena de conexión para inicializar el contexto de la base de datos.

Example `DBContext.cs`

```
using System.Data.Entity;
using System.Security.Claims;
```

```
using System.Threading.Tasks;
using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.EntityFramework;

namespace MVC5App.Models
{
    public class RDSContext : DbContext
    {
        public RDSContext()
            : base(GetRDSConnectionString())
        {
        }

        public static RDSContext Create()
        {
            return new RDSContext();
        }
    }
}
```

El AWS Toolkit for Visual Studio

Visual Studio dispone de plantillas para diferentes lenguajes de programación y diferentes tipos de aplicaciones. Puede empezar a trabajar con cualquiera de estas plantillas. El kit de herramientas AWS Toolkit for Visual Studio también cuenta con tres plantillas de proyecto que ponen en marcha el desarrollo de la aplicación: AWS Console Project, AWS Web Project y AWS Empty Project. En este ejemplo, va a crear una nueva aplicación web ASP.NET.

Para crear un nuevo proyecto de aplicación web ASP.NET

1. En Visual Studio, en el menú File (Archivo), haga clic en New (Nuevo) y en Project (Proyecto).
2. En el cuadro de diálogo New Project (Nuevo proyecto), haga clic en Installed Templates (Plantillas instaladas), en Visual C# y en Web. Haga clic en ASP.NET Empty Web Application (Aplicación web vacía de ASP.NET), escriba un nombre de proyecto y haga clic en OK (Aceptar).

Para ejecutar un proyecto

Aplique alguna de las siguientes acciones:

1. Pulse F5.

2. Seleccione Start Debugging (Iniciar depuración) en el menú Debug (Depurar).

Haga una prueba local

Con Visual Studio, resulta muy fácil probar la aplicación localmente. Para probar o ejecutar aplicaciones web ASP.NET, necesita un servidor web. Visual Studio dispone de varias opciones, como Internet Information Services (IIS), IIS Express o el servidor de desarrollo de Visual Studio integrado. Para obtener más información sobre cada una de estas opciones y decidir cuál es la mejor para usted, consulte [Web Servers in Visual Studio for ASP.NET Web Projects](#).

Cree un entorno de Elastic Beanstalk

Después de probar la aplicación, ya puede implementarla en Elastic Beanstalk.

Note

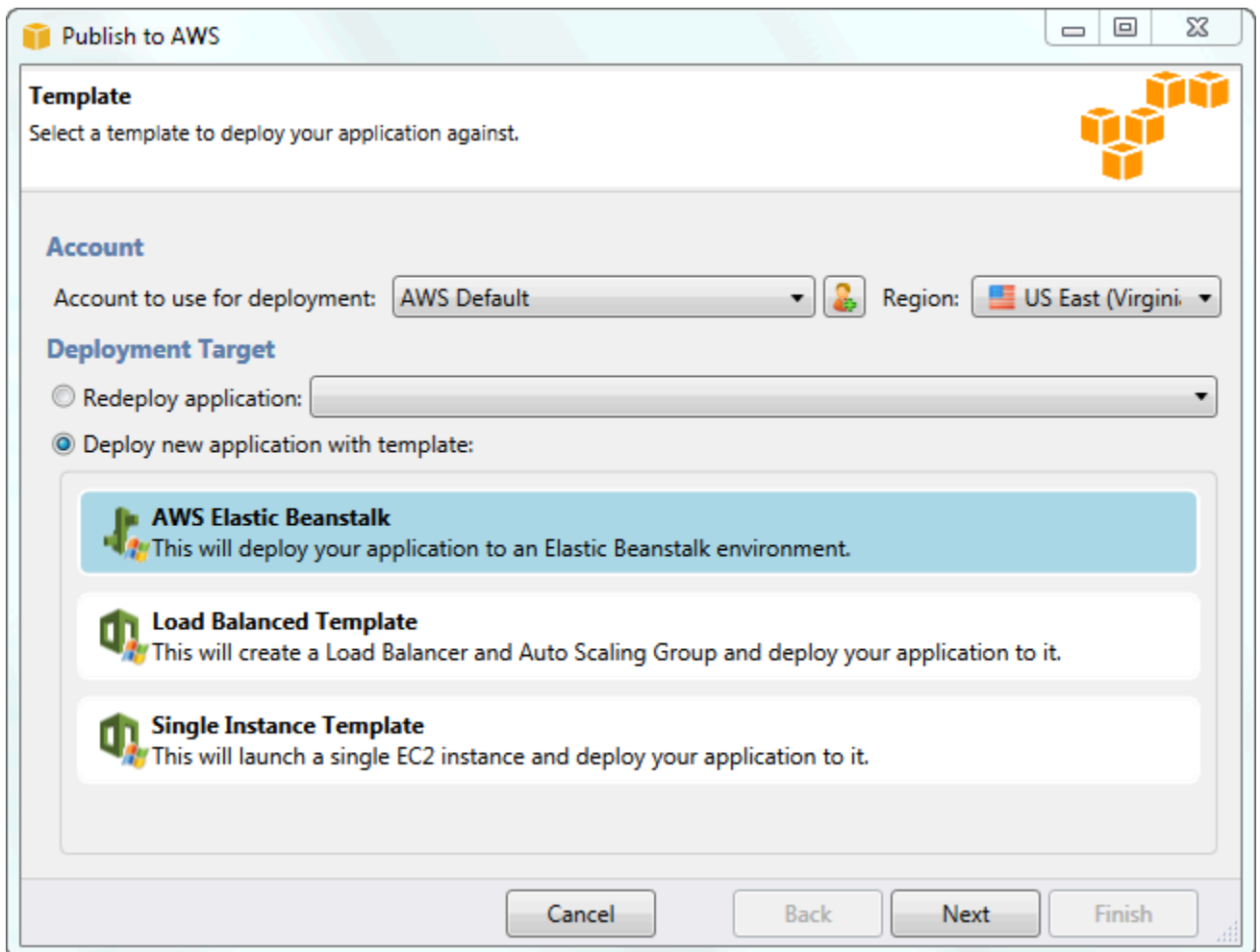
El [archivo de configuración](#) debe formar parte del proyecto que se va a incluir en el archivo. En lugar de incluir los archivos de configuración en el proyecto, puede utilizar Visual Studio para implementar todos los archivos en la carpeta del proyecto. En el Solution Explorer (Explorador de soluciones), haga clic con el botón derecho en el nombre de proyecto y, a continuación, haga clic en Properties (Propiedades). Haga clic en la pestaña Package/Publish Web (Empaquetar/publicar web). En la sección Items to deploy (Elementos que se van a implementar), seleccione All Files in the Project Folder (Todos los archivos de la carpeta de proyecto) en la lista desplegable.

Para implementar su aplicación en Elastic Beanstalk mediante el Toolkit for Visual Studio de AWS.

1. En Solution Explorer, haga clic con el botón derecho en la aplicación y, a continuación, seleccione Publicar en AWS.
2. En el asistente Publicar en AWS, ingrese la información de la cuenta.
 - a. En Cuenta de AWS que se utilizará para la implementación, seleccione su cuenta u Otro para ingresar los datos de una nueva cuenta.
 - b. En Region (Región), seleccione la región en la que desea implementar la aplicación. Para obtener información acerca de las regiones de AWS disponibles, consulte [Puntos de enlace y cuotas de AWS Elastic Beanstalk](#) en la Referencia general de AWS. Si selecciona una

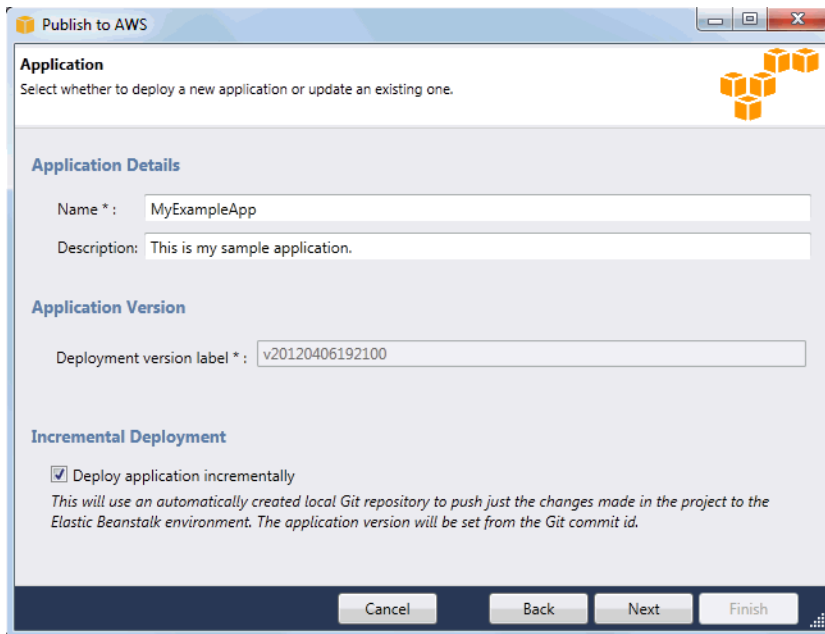
región que no es compatible con Elastic Beanstalk, la opción para implementar en Elastic Beanstalk dejará de estar disponible.

- c. Haga clic en Deploy new application with template (Implementar nueva aplicación con plantilla) y seleccione Elastic Beanstalk. Haga clic en Next (Siguiente).



3. En la página Application (Aplicación), especifique los detalles de la aplicación
 - a. En Name (Nombre), escriba el nombre de la aplicación.
 - b. En Description, escriba una descripción de la aplicación. Este paso es opcional.
 - c. La etiqueta de versión de la aplicación aparece automáticamente en Deployment version label (Etiqueta de versión de implementación).
 - d. Seleccione Deploy application incrementally (Implementar aplicación incrementalmente) para implementar únicamente los archivos modificados. Las implementaciones incrementales son más rápidas, ya que solo se actualizan los archivos que cambiaron, y no todos los archivos. Si selecciona esta opción, se definirá una versión de la aplicación a partir del ID de la confirmación de Git. Si decide no implementar la aplicación de manera

incremental, puede actualizar la etiqueta de versión en el cuadro Deployment version label (Etiqueta de versión de implementación).



- e. Haga clic en Next (Siguiente).
4. En la página Environment (Entorno), especifique los detalles del entorno.
 - a. Seleccione Create a new environment for this application (Crear un nuevo entorno para esta aplicación).
 - b. En Name (Nombre), escriba un nombre para su entorno.
 - c. En Description (Descripción), describa el entorno. Este paso es opcional.
 - d. Seleccione el tipo de entorno que desee en Type (Tipo).

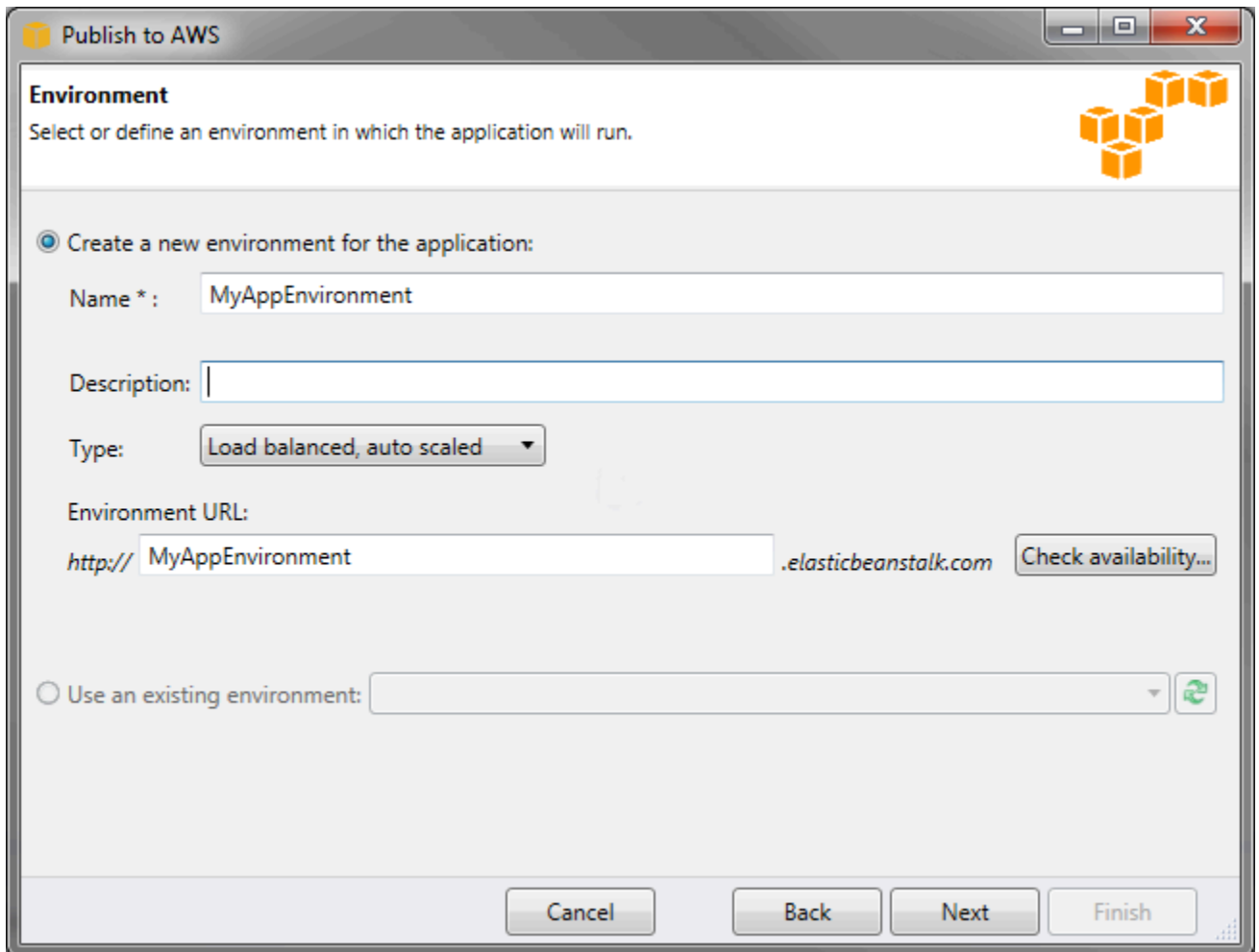
Puede seleccionar Load balanced, auto scaled (Carga balanceada, autoescalado) o Single instance (Instancia individual). Para obtener más información, consulte [Tipos de entornos](#).

Note

En los entornos con una única instancia, no se aplican el balanceador de carga, el Auto Scaling y la URL de comprobación de estado.

- e. La URL del entorno aparecerá automáticamente en el cuadro Environment URL (URL de entorno) cuando mueva el cursor a ese cuadro.

- f. Haga clic en Check availability (Comprobar disponibilidad) para asegurarse de que la URL está disponible.



Publish to AWS

Environment
Select or define an environment in which the application will run.

Create a new environment for the application:

Name * : MyAppEnvironment

Description:

Type: Load balanced, auto scaled

Environment URL:
http:// MyAppEnvironment .elasticbeanstalk.com

Use an existing environment:

- g. Haga clic en Siguiente.

5. En la página AWS Options (Opciones de AWS), configure otras opciones y la información de seguridad de la implementación.
- En Container Type (Tipo de contenedor), seleccione 64bit Windows Server 2012 running IIS 8 (Windows Server 2012 de 64 bits con IIS 8) o 64bit Windows Server 2008 running IIS 7.5 (Windows Server 2008 de 64 bits con IIS 7.5).
 - En Instance Type (Tipo de instancia), seleccione Micro.
 - En Key pair (Par de claves), seleccione Create new key pair (Crear nuevo par de claves). Escriba un nombre para el nuevo par de claves (en este ejemplo, utilizaremos **myuswestkeypair**) y haga clic en OK (Aceptar). El par de claves permite el acceso de Escritorio remoto a las instancias de Amazon EC2. Para obtener más información sobre los

pares de claves de Amazon EC2, consulte [Uso de credenciales](#) en la Guía del usuario de Amazon Elastic Compute Cloud.

- d. Seleccione un perfil de instancia.

Si no tiene un perfil de instancia, seleccione Create a default instance profile (Crear un perfil de instancia predeterminado). Para obtener información sobre el uso de perfiles de instancia con Elastic Beanstalk, consulte [Administración de perfiles de instancia de Elastic Beanstalk](#).

- e. Si tiene una VPC personalizada que quiere utilizar con el entorno, haga clic en Launch into VPC (Lanzar en VPC). Puede configurar los datos de la VPC en la página siguiente. Para obtener más información sobre Amazon VPC, consulte [Amazon Virtual Private Cloud \(Amazon VPC\)](#). Para ver una lista de los tipos de contenedores no heredados admitidos, consulte [the section called “¿Por qué algunas versiones de la plataforma están marcadas como heredadas?”](#)

Publish to AWS

AWS Options
Set Amazon EC2 options for the deployed application.

Amazon EC2

Container type *: 64bit Windows Server 2012 running IIS 8

Use custom AMI:

Instance type *: Micro Key pair *: myuswestkeypair

Launch Configuration

IAM Role: Use the default role (aws-elasticbeanstalk-ec2-role)

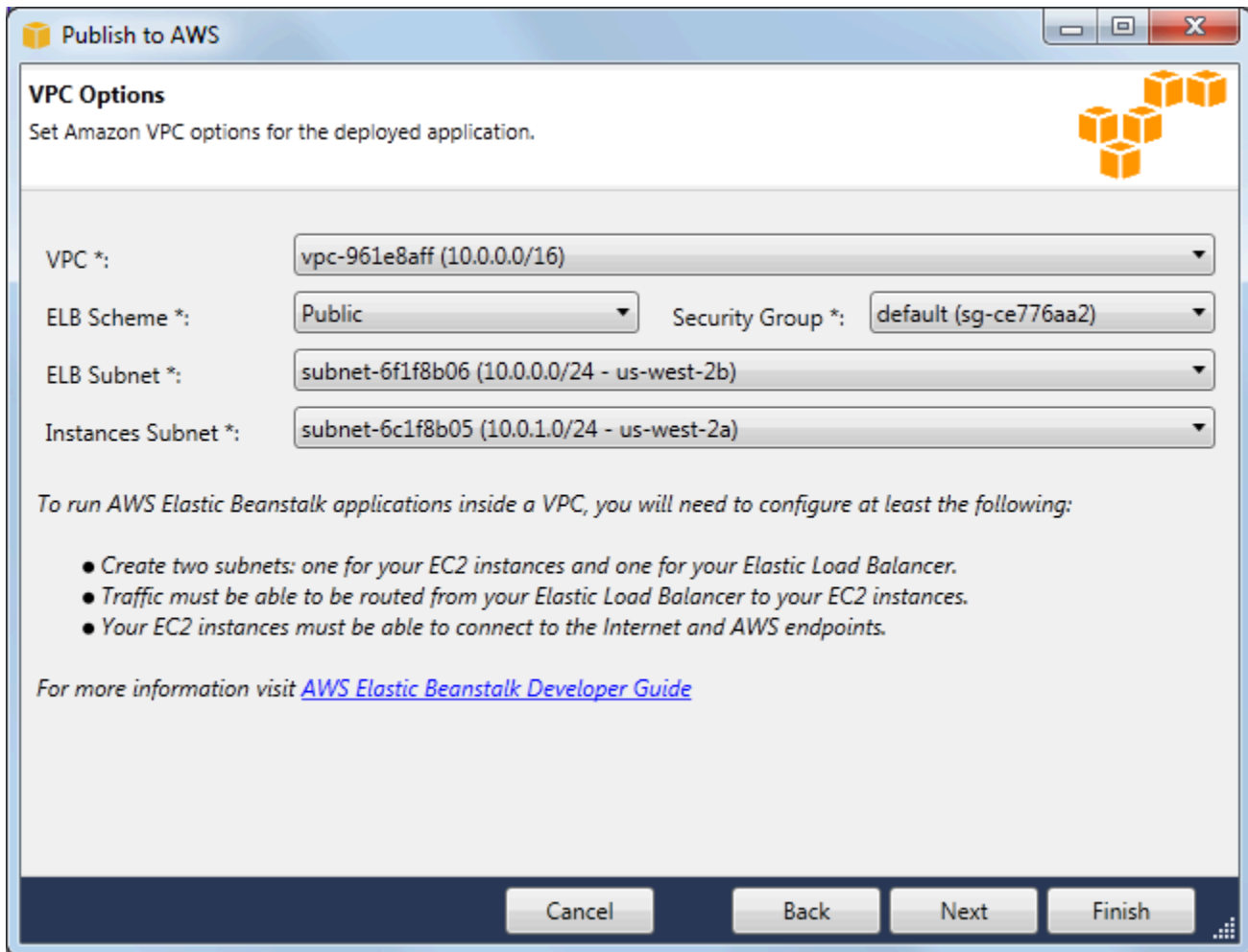
If you choose not to use the default role, you must grant the relevant permissions to Elastic Beanstalk. See [AWS Elastic Beanstalk Developer Guide](#) for more details.

Launch into VPC *If you elect to launch instances in a VPC, the next page will enable you to customize the VPC settings.*

Cancel Back Next Finish

- f. Haga clic en Siguiente.

6. Si decidió lanzar el entorno dentro de una VPC, aparecerá la página VPC Options (Opciones de VPC); de lo contrario, aparecerá la página Additional Options (Opciones adicionales). Aquí podrá configurar las opciones de la VPC.



Publish to AWS

VPC Options

Set Amazon VPC options for the deployed application.

VPC *: vpc-961e8aff (10.0.0.0/16)

ELB Scheme *: Public Security Group *: default (sg-ce776aa2)

ELB Subnet *: subnet-6f1f8b06 (10.0.0.0/24 - us-west-2b)

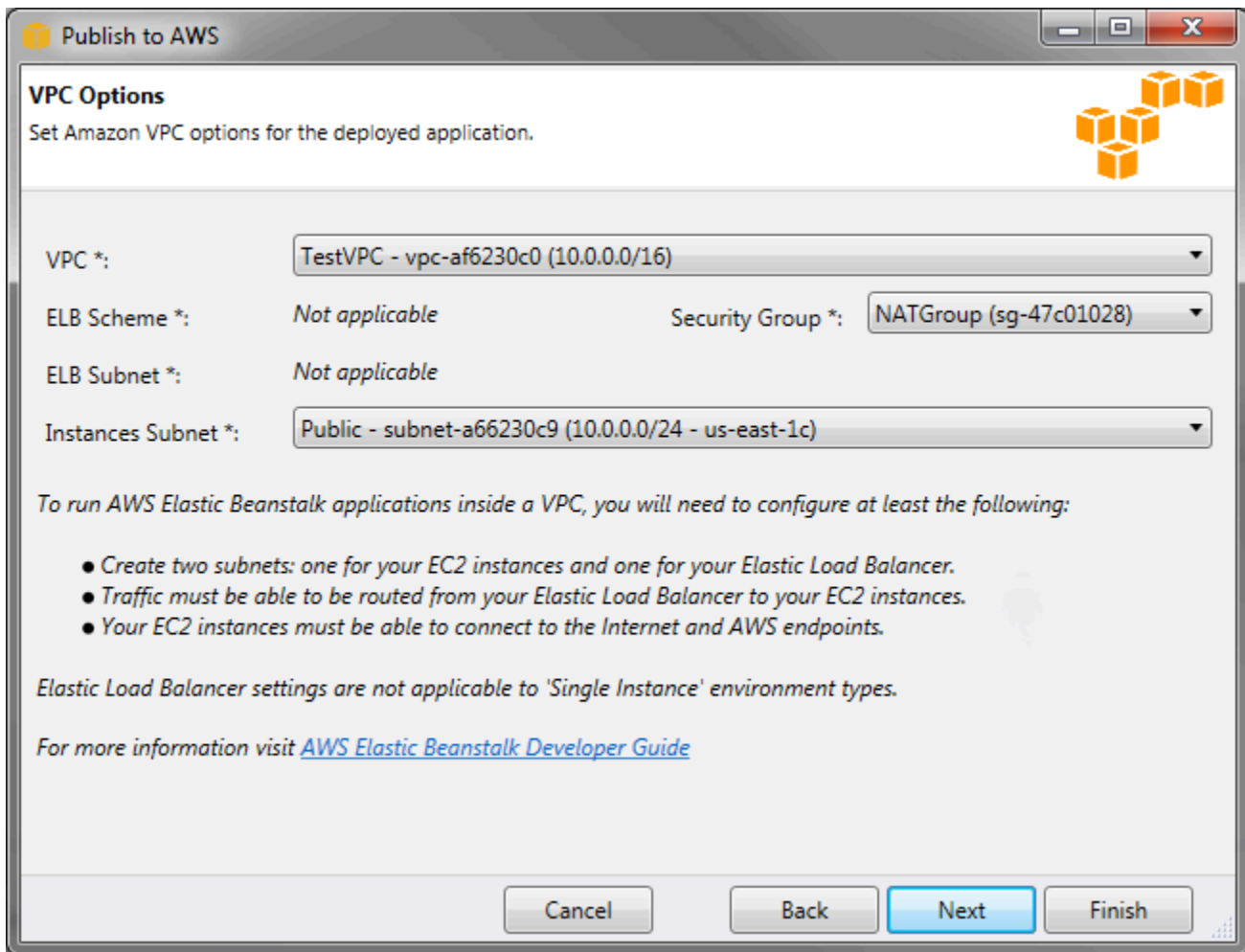
Instances Subnet *: subnet-6c1f8b05 (10.0.1.0/24 - us-west-2a)

To run AWS Elastic Beanstalk applications inside a VPC, you will need to configure at least the following:

- Create two subnets: one for your EC2 instances and one for your Elastic Load Balancer.
- Traffic must be able to be routed from your Elastic Load Balancer to your EC2 instances.
- Your EC2 instances must be able to connect to the Internet and AWS endpoints.

For more information visit [AWS Elastic Beanstalk Developer Guide](#)

Cancel Back Next Finish



Publish to AWS

VPC Options
Set Amazon VPC options for the deployed application.

VPC *: TestVPC - vpc-af6230c0 (10.0.0.0/16)

ELB Scheme *: Not applicable Security Group *: NATGroup (sg-47c01028)

ELB Subnet *: Not applicable

Instances Subnet *: Public - subnet-a66230c9 (10.0.0.0/24 - us-east-1c)

To run AWS Elastic Beanstalk applications inside a VPC, you will need to configure at least the following:

- Create two subnets: one for your EC2 instances and one for your Elastic Load Balancer.
- Traffic must be able to be routed from your Elastic Load Balancer to your EC2 instances.
- Your EC2 instances must be able to connect to the Internet and AWS endpoints.

Elastic Load Balancer settings are not applicable to 'Single Instance' environment types.

For more information visit [AWS Elastic Beanstalk Developer Guide](#)

Cancel Back Next Finish

- Seleccione el ID de la VPC en la que le gustaría lanzar el entorno.
- En los entornos escalables con balanceo de carga, seleccione private (privado) en ELB Scheme (Esquema de ELB) si no quiere que haya un Elastic Load Balancer disponible en Internet.

En los entornos con una única instancia, esta opción no es aplicable, ya que el entorno no tiene un balanceador de carga. Para obtener más información, consulte [Tipos de entornos](#).

- Si se trata de un entorno con balanceo de carga y escalable, seleccione las subredes del Elastic Load Balancer y las instancias EC2. Si creó subredes públicas y privadas, asegúrese de que el balanceador de tipo Elastic Load Balancer y las instancias EC2 están asociados con la subred correcta. Amazon VPC crea por defecto una subred pública predeterminada con 10.0.0.0/24 y una subred privada con 10.0.1.0/24. Puede ver las subredes existentes en la consola de Amazon VPC en <https://console.aws.amazon.com/vpc/>.

Si se trata de un entorno con una única instancia, la VPC solo necesita una subred pública para la instancia. No podrá seleccionar una subred para el balanceador de carga, ya que el entorno no tiene ninguno. Para obtener más información, consulte [Tipos de entornos](#).

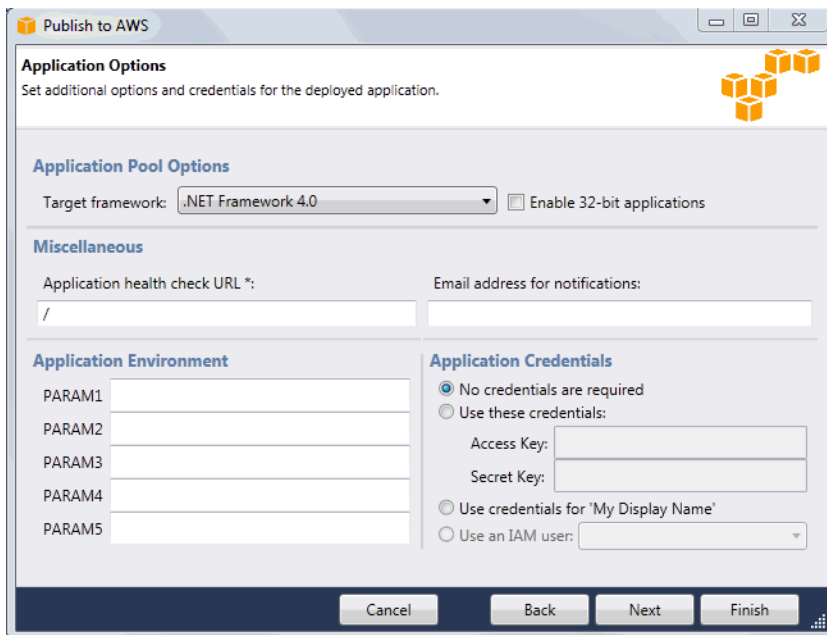
- d. En un entorno escalable con balanceo de carga, seleccione el grupo de seguridad que creó para sus instancias, si procede.

Si el entorno solo tiene una instancia, no necesita un dispositivo NAT. Seleccione el grupo de seguridad predeterminado. Elastic Beanstalk asigna una dirección IP elástica a la instancia que permite a esta obtener acceso a Internet.

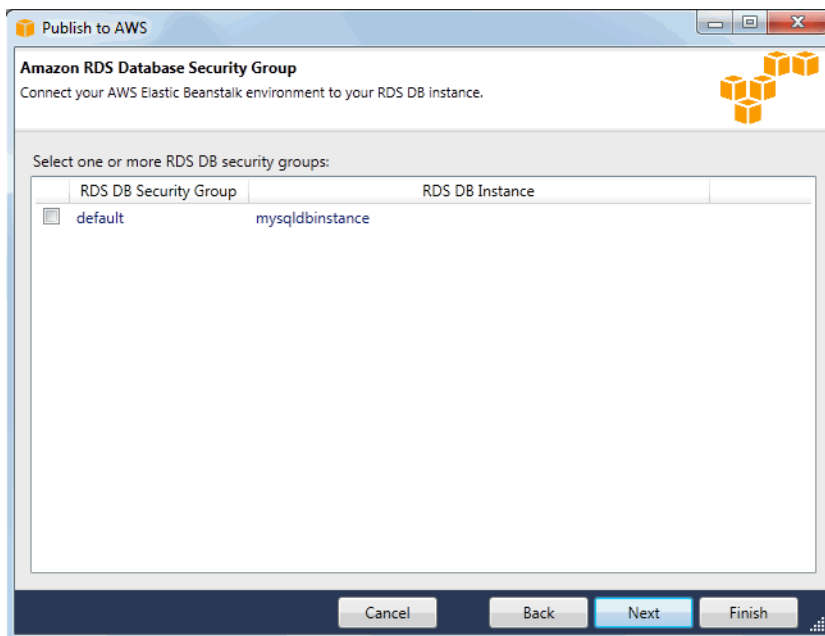
- e. Haga clic en Next (Siguiente).

7. En la página Application Options (Opciones de aplicación), configure las opciones de la aplicación.

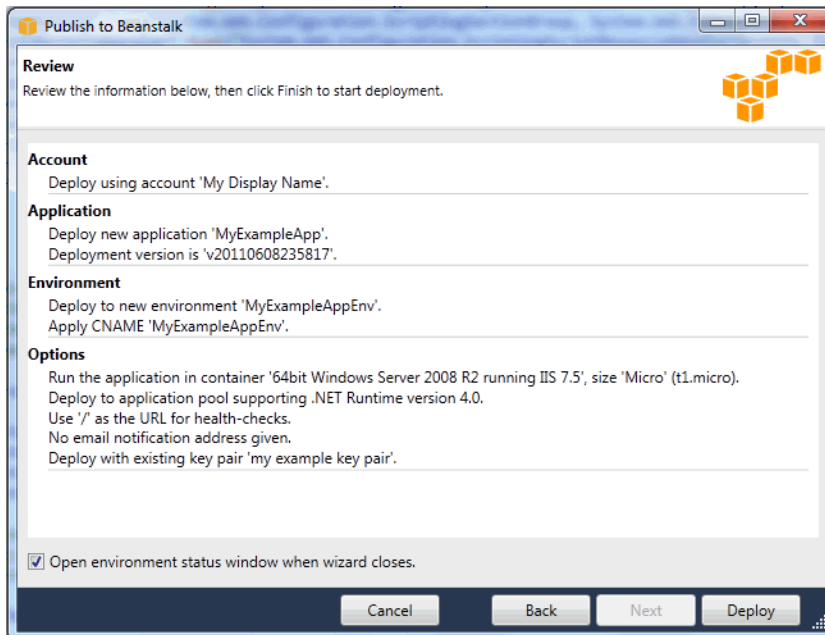
- a. En Target framework, seleccione .NET Framework 4.0.
- b. Elastic Load Balancing utiliza una comprobación de estado para determinar si las instancias de Amazon EC2 que ejecutan la aplicación están en buen estado. La comprobación de estado determina el estado de una instancia sondeando una URL especificada en los intervalos indicados. Puede anular la URL predeterminada para que se ajuste a un recurso existente de la aplicación (por ejemplo, `/myapp/index.aspx`) utilizando el cuadro Application Health Check URL (URL de comprobación de estado de aplicación). Para obtener más información sobre las comprobaciones de estado de la aplicación, consulte [Comprobación de estado](#).
- c. Escriba una dirección de correo electrónico si desea recibir notificaciones de Amazon Simple Notification Service (Amazon SNS) sobre eventos importantes que afectan a la aplicación.
- d. La sección Application Environment (Entorno de aplicación) le permite especificar variables de entorno en las instancias de Amazon EC2 que ejecutan la aplicación. Esta configuración ofrece una mayor portabilidad al eliminar la necesidad de recompilar su código fuente cuando cambia de un entorno a otro.
- e. En Application Credentials, seleccione la opción que desee utilizar para implementar la aplicación.



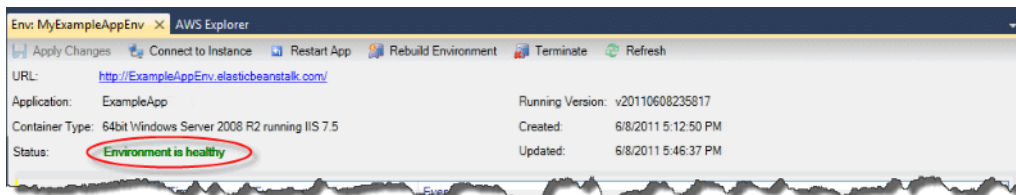
- f. Haga clic en Siguiente.
8. Si previamente configuró una base de datos de Amazon RDS, aparece la página Amazon RDS DB Security Group (Grupo de seguridad de base de datos Amazon RDS). Si desea conectar el entorno de Elastic Beanstalk con la instancia de base de datos de Amazon RDS, seleccione uno o varios grupos de seguridad. De lo contrario, continúe en el siguiente paso. Cuando esté preparado, haga clic en Next (Siguiente).



9. Revise las opciones de implementación. Si todo está tal y como quiere, haga clic en Deploy (Implementar).



El proyecto de ASP.NET se exportará como un archivo de implementación web, se cargará en Amazon S3 y se registrará como una nueva versión de la aplicación con Elastic Beanstalk. La característica de implementación de Elastic Beanstalk monitoreará el entorno hasta que esté disponible con el código que acaba de implementar. En la pestaña env:<nombre de entorno>, aparecerá el estado del entorno.



Terminación de un entorno

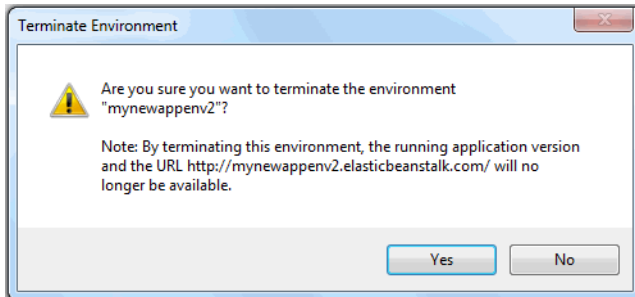
Para evitar que se le apliquen cargos por recursos de AWS que no utiliza, puede terminar un entorno en ejecución con AWS Toolkit for Visual Studio.

Note

Siempre podrá lanzar más adelante un nuevo entorno con la misma versión.

Para terminar un entorno

1. Expanda el nodo Elastic Beanstalk y el nodo de la aplicación en AWS Explorer. Haga clic con el botón derecho en el entorno de la aplicación y seleccione Terminate Environment (Terminar entorno).
2. Cuando le pregunten, haga clic en Yes (Sí) para confirmar que desea terminar el entorno. Elastic Beanstalk tarda unos minutos en terminar los recursos de AWS que se ejecutan en el entorno.



Note

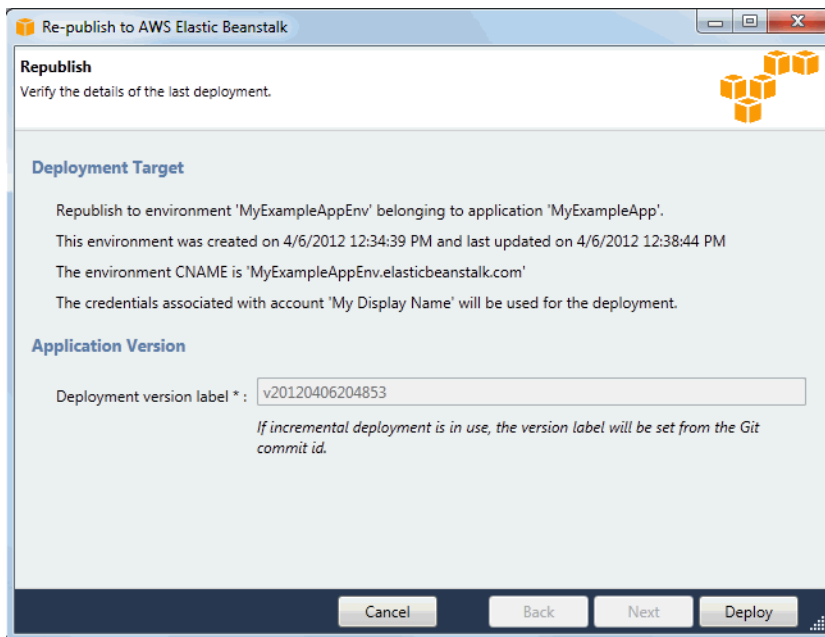
Cuando termine su entorno, el CNAME asociado con el entorno terminado quedará a disposición de cualquier persona que desee utilizarlo.

Implementación en su entorno

Ahora que ha probado su aplicación, es fácil editarla e implementarla y ver los resultados inmediatamente.

Para editar y volver a implementar su aplicación web ASP.NET

1. En Solution Explorer (Explorador de soluciones), haga clic con el botón derecho en la aplicación y después haga clic en Republish to Environment (Volver a publicar en el entorno) **<el nombre de su entorno>**. Se abre el asistente Re-publish to AWS Elastic Beanstalk (Volver a publicar en AEBlong).



2. Revise los detalles de la implementación y haga clic en Deploy (Implementar).

Note

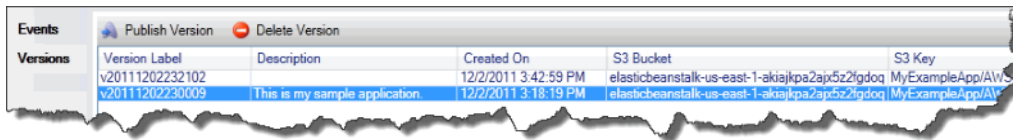
Si desea cambiar parte de la configuración, puede hacer clic en Cancel (Cancelar) y utilizar, en cambio, el asistente Publish to AWS (Publicar en AWS). Para obtener instrucciones, consulte [Cree un entorno de Elastic Beanstalk](#).

El proyecto web ASP.NET actualizado se exportará como un archivo de implementación web con la nueva etiqueta de versión, se cargará en Amazon S3 y se registrará como una nueva versión de la aplicación con Elastic Beanstalk. La característica de implementación de Elastic Beanstalk monitorea el entorno existente hasta que esté disponible con el código que acaba de implementar. En la pestaña env:<**nombre del entorno**>, verá el estado del entorno.

También puede implementar una aplicación en un entorno existente si, por ejemplo, necesita restaurar una versión de la aplicación anterior.

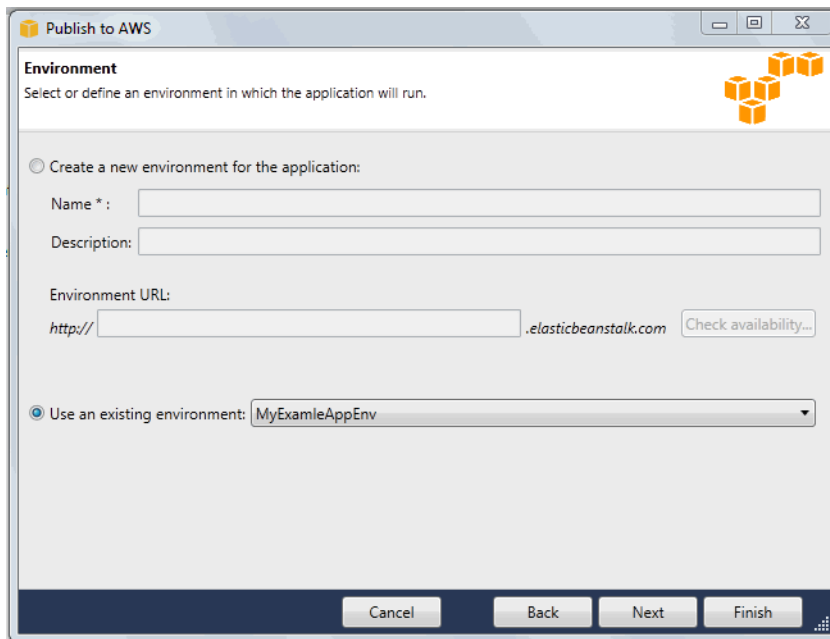
Para implementar una versión de la aplicación en un entorno existente

1. Haga clic con el botón derecho sobre la aplicación Elastic Beanstalk expandiendo el nodo Elastic Beanstalk en AWS Explorer (Explorador de AWS). Seleccione View Status (Ver estado).
2. En la pestaña App: <**nombre de aplicación**>, haga clic en Versions (Versiones).



Version Label	Description	Created On	S3 Bucket	S3 Key
v20111202232102		12/2/2011 3:42:59 PM	elasticbeanstalk-us-east-1-akiajka2ap5z2fgdoq	MyExampleApp/AW
v20111202232009	This is my sample application	12/2/2011 3:18:19 PM	elasticbeanstalk-us-east-1-akiajka2ap5z2fgdoq	MyExampleApp/AW

- Haga clic en la versión de la aplicación que desea implementar y haga clic en Publish Version (Publicar versión).
- En el asistente Publish Application Version (Publicar versión de la aplicación), haga clic en Next (Siguiente).



Publish to AWS

Environment
Select or define an environment in which the application will run.

Create a new environment for the application:

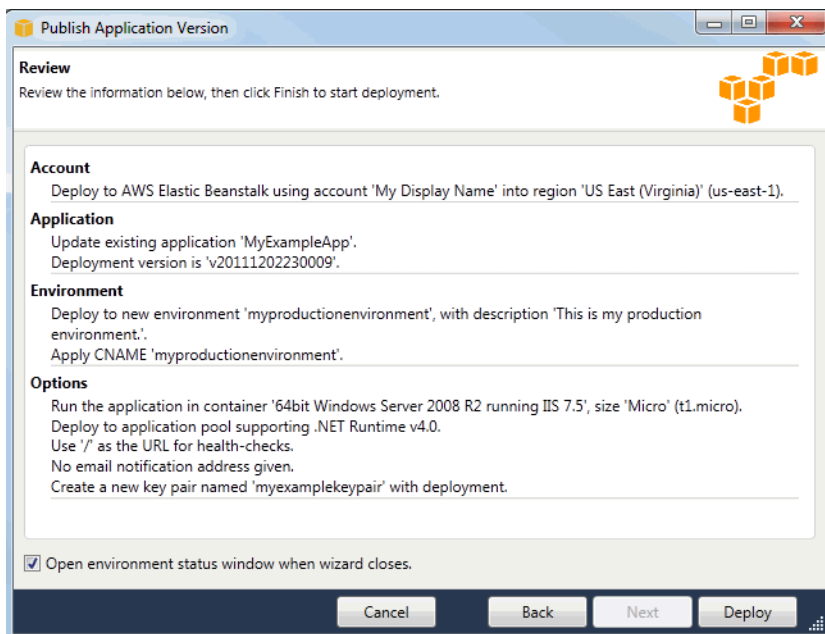
Name * :

Description:

Environment URL:
http:// .elasticbeanstalk.com

Use an existing environment:

- Revise las opciones de implementación y haga clic en Deploy (Implementar).



Publish Application Version

Review
Review the information below, then click Finish to start deployment.

Account
Deploy to AWS Elastic Beanstalk using account 'My Display Name' into region 'US East (Virginia)' (us-east-1).

Application
Update existing application 'MyExampleApp'.
Deployment version is 'v20111202232009'.

Environment
Deploy to new environment 'myproductionenvironment', with description 'This is my production environment'.
Apply CNAME 'myproductionenvironment'.

Options
Run the application in container '64bit Windows Server 2008 R2 running IIS 7.5', size 'Micro' (t1.micro).
Deploy to application pool supporting .NET Runtime v4.0.
Use '/' as the URL for health-checks.
No email notification address given.
Create a new key pair named 'myexamplekeypair' with deployment.

Open environment status window when wizard closes.

El proyecto ASP.NET se exportará como un archivo de implementación web y se cargará en Amazon S3. La característica de implementación de Elastic Beanstalk monitoreará el entorno hasta que esté disponible con el código que acaba de implementar. En la pestaña env:<**nombre de entorno**>, aparecerá el estado del entorno.

Administración de entornos de aplicaciones de Elastic Beanstalk

Con AWS Toolkit for Visual Studio y la Management Console de AWS, puede cambiar el aprovisionamiento y la configuración de los recursos de AWS que se utilizan en los entornos de aplicaciones. Para obtener más información sobre la administración de sus entornos de aplicaciones a través de la Management Console de AWS, consulte [Administración de entornos](#). En esta sección, se abordan las configuraciones específicas del servicio que se pueden editar en el AWS Toolkit for Visual Studio durante la configuración del entorno de aplicaciones.

Modificación de las opciones de configuración del entorno

Cuando implementa su aplicación, Elastic Beanstalk configura una serie de servicios de informática en la nube de AWS. Con el AWS Toolkit for Visual Studio puede controlar la forma en que se configura cada uno de estos servicios.

Para editar la configuración del entorno de una aplicación

- Expanda el nodo Elastic Beanstalk y el nodo de la aplicación. A continuación, haga clic con el botón derecho en el entorno de Elastic Beanstalk en AWS Explorer (Explorador de AWS). Seleccione View Status (Ver estado).

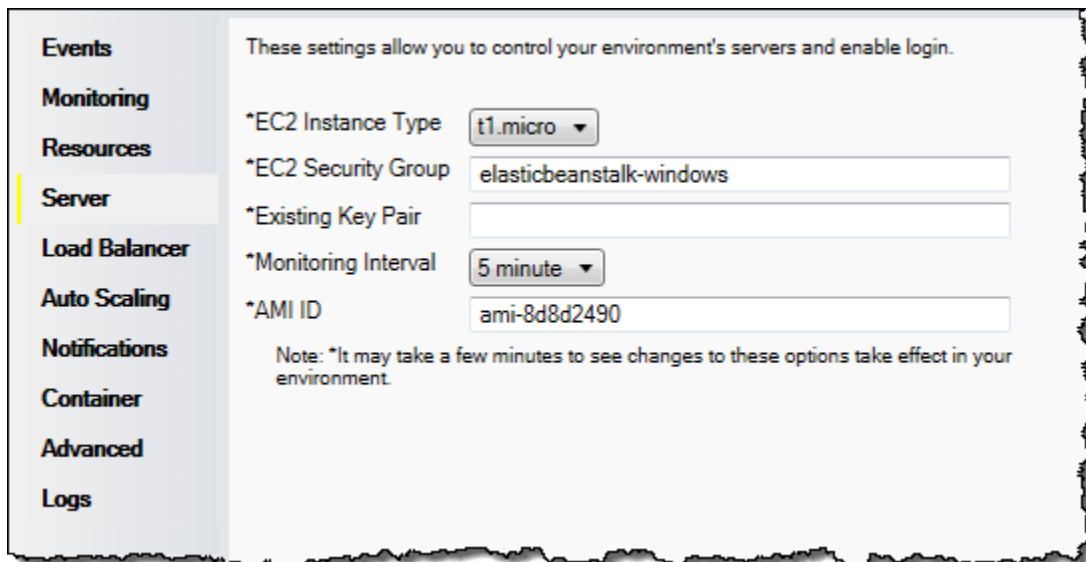
Ahora, podrá configurar las opciones de los elementos siguientes:

- Servidor
- Balanceo de carga
- Autoescalado
- Notificaciones
- Propiedades del entorno

Configuración de instancias de servidor EC2 mediante el AWS Toolkit for Visual Studio

Amazon Elastic Compute Cloud (Amazon EC2) es un servicio web que se utiliza para lanzar y administrar instancias de servidor en los centros de datos de Amazon. Puede utilizar instancias de servidor Amazon EC2 en cualquier momento, durante el tiempo que las necesite y con cualquier finalidad legal. Las instancias están disponibles en diferentes tamaños y configuraciones. Para obtener más información, visite [Amazon EC2](#).

Puede editar la configuración de la instancia de Amazon EC2 del entorno de Elastic Beanstalk con la pestaña Server (Servidor) incluida en la pestaña del entorno de la aplicación en AWS Toolkit for Visual Studio.



Tipos de instancias de Amazon EC2

Instance type (Tipo de instancia) muestra los tipos de instancias disponibles para la aplicación de Elastic Beanstalk. Cambie el tipo de instancia para seleccionar un servidor con las características más adecuadas para su aplicación (entre ellas, el tamaño de la memoria y la potencia de la CPU). Por ejemplo, las aplicaciones de operaciones intensivas de ejecución prolongada pueden necesitar más memoria o CPU.

Para obtener más información acerca de los tipos de instancias de Amazon EC2 disponibles con la aplicación de Elastic Beanstalk, consulte [Tipos de instancias](#) en la Guía del usuario de Amazon Elastic Compute Cloud.

Grupos de seguridad de Amazon EC2

Puede controlar el acceso a la aplicación de Elastic Beanstalk mediante un grupo de seguridad de Amazon EC2. Los grupos de seguridad definen las reglas del firewall para las instancias. Estas reglas especifican qué tráfico de la red de entrada debe entregarse en la instancia. El resto del tráfico de entrada se descartará. Puede modificar las reglas de un grupo en cualquier momento. Las nuevas reglas se aplicarán automáticamente a todas las instancias que estén en ejecución y a las que se lancen en el futuro.

Puede configurar los grupos de seguridad de Amazon EC2 a través de la Management Console de AWS o del AWS Toolkit for Visual Studio. Puede especificar qué grupos de seguridad de Amazon EC2 controlan el acceso a la aplicación de Elastic Beanstalk especificando los nombres de uno o varios grupos de seguridad de Amazon EC2 (delimitados por comas) en el cuadro de texto EC2 Security Groups (Grupos de seguridad de EC2).

Note

Asegúrese de que el puerto 80 (HTTP) está accesible desde 0.0.0.0/0 como el intervalo de CIDR de origen si desea habilitar las comprobaciones de estado para su aplicación. Para obtener más información sobre las comprobaciones de estado, consulte [Comprobaciones de estado](#).

Para crear un grupo de seguridad con el AWS Toolkit for Visual Studio

1. Vaya a Visual Studio, en AWS Explorer (Explorador de AWS) expanda el nodo Amazon EC2 y, a continuación, haga doble clic en Security Groups (Grupos de seguridad).
2. Haga clic en Create Security Group (Crear grupo de seguridad) e introduzca un nombre y una descripción para el grupo de seguridad.
3. Haga clic en OK (Aceptar).

Para obtener más información sobre los grupos de seguridad de Amazon EC2, consulte [Uso de grupos de seguridad](#) en la Guía del usuario de Amazon Elastic Compute Cloud.

Pares de claves de Amazon EC2

Puede iniciar sesión de forma segura en las instancias de Amazon EC2 provisionadas por la aplicación de Elastic Beanstalk con un par de claves de Amazon EC2.

⚠ Important

Debe crear un par de claves de Amazon EC2 y configurar las instancias de Amazon EC2 aprovisionadas por Elastic Beanstalk de forma que utilicen el par de claves de Amazon EC2 para poder obtener acceso a las instancias de Amazon EC2 aprovisionadas por Elastic Beanstalk. Puede crear el par de claves con el asistente Publish to AWS (Publicar en AWS) en el AWS Toolkit for Visual Studio cuando implemente la aplicación en Elastic Beanstalk. Si desea crear pares de claves adicionales mediante el conjunto de herramientas, siga los pasos que se indican a continuación. Si lo desea, también puede configurar el par de claves de Amazon EC2 con la [Management Console de AWS](#). Si desea obtener instrucciones acerca de cómo crear un par de claves para Amazon EC2, consulte la [Guía de introducción de Amazon Elastic Compute Cloud](#).

El cuadro de texto Existing Key Pair (Par de claves existente) le permite especificar el nombre de un par de claves de Amazon EC2 que puede utilizar para iniciar sesión de forma segura en las instancias de Amazon EC2 que ejecutan su aplicación de Elastic Beanstalk.

Para especificar el nombre de un par de claves de Amazon EC2

1. Expanda el nodo Amazon EC2 y haga doble clic en Key Pairs (Pares de claves).
2. Haga clic en Create Key Pair (Crear par de claves) e introduzca el nombre del par de claves.
3. Haga clic en OK (Aceptar).

Para obtener más información sobre los pares de claves de Amazon EC2, consulte [Uso de credenciales de Amazon EC2](#) en la Guía del usuario de Amazon Elastic Compute Cloud. Para obtener más información acerca de cómo conectarse a instancias de Amazon EC2, consulte [Listado y conexión a instancias de servidor](#).

Intervalo de monitorización

De manera predeterminada, solo están habilitadas las métricas básicas de Amazon CloudWatch. Estas devuelven datos en periodos de cinco minutos. Puede habilitar métricas de CloudWatch de un minuto pormenorizada al seleccionar 1 minute (1 minuto) en el Monitoring Interval (Intervalo de monitoreo) de la sección Server (Servidor) en la pestaña Configuration (Configuración) del entorno en el AWS Toolkit for Eclipse.

Note

Se pueden aplicar gastos por las métricas en intervalos de un minuto del servicio Amazon CloudWatch. Consulte [Amazon CloudWatch](#) para obtener más información.

ID de AMI personalizada

Puede anular la AMI predeterminada utilizada en las instancias de Amazon EC2 y sustituirla por la suya personalizada al ingresar el identificador de la AMI personalizada en el cuadro Custom AMI ID (ID de AMI personalizada) de la sección Server (Servidor) en la pestaña Configuration (Configuración) del entorno en el AWS Toolkit for Eclipse.

Important

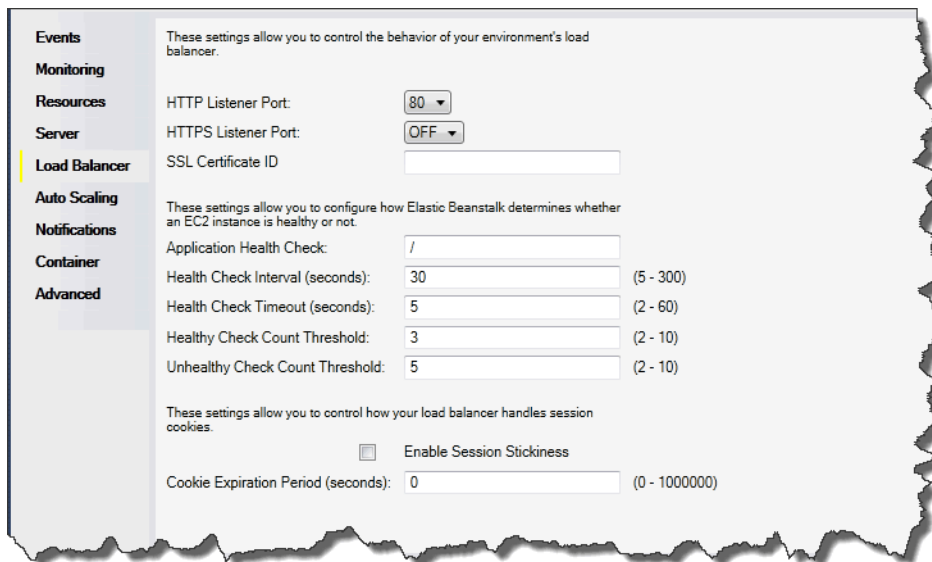
El uso de una AMI propia es una tarea avanzada que debe llevarse a cabo con precaución. Si necesita una AMI personalizada, le recomendamos que comience con la AMI predeterminada de Elastic Beanstalk y después la modifique. Para darlas por buenas, Elastic Beanstalk espera que las instancias de Amazon EC2 satisfagan una serie de requisitos, entre ellos disponer de un administrador host en ejecución. Si no se cumplen, es posible que el entorno no funcione correctamente.

Configuración de Elastic Load Balancing mediante AWS Toolkit for Visual Studio

Elastic Load Balancing es un servicio web de Amazon que le ayuda a mejorar la disponibilidad y escalabilidad de la aplicación. Este servicio le permite distribuir fácilmente las cargas de la aplicación entre dos o más instancias de Amazon EC2. Elastic Load Balancing habilita la disponibilidad a través de la redundancia y admite el aumento del tráfico en la aplicación.

Elastic Load Balancing le permite distribuir y balancear automáticamente el tráfico entrante de la aplicación entre todas las instancias de servidor que están en ejecución. Además, el servicio facilita la incorporación de nuevas instancias cuando es necesario aumentar la capacidad de la aplicación.

Cuando se implementa una aplicación, Elastic Beanstalk aprovisiona automáticamente Elastic Load Balancing. Puede editar la configuración de las instancias de Amazon EC2 del entorno de Elastic Beanstalk con la pestaña Load Balancer (Equilibrador de carga) incluida en la pestaña del entorno de la aplicación en el AWS Toolkit for Visual Studio.



Events

Monitoring

Resources

Server

Load Balancer

Auto Scaling

Notifications

Container

Advanced

These settings allow you to control the behavior of your environment's load balancer.

HTTP Listener Port: 80

HTTPS Listener Port: OFF

SSL Certificate ID

These settings allow you to configure how Elastic Beanstalk determines whether an EC2 instance is healthy or not.

Application Health Check: /

Health Check Interval (seconds): 30 (5 - 300)

Health Check Timeout (seconds): 5 (2 - 60)

Healthy Check Count Threshold: 3 (2 - 10)

Unhealthy Check Count Threshold: 5 (2 - 10)

These settings allow you to control how your load balancer handles session cookies.

Enable Session Stickiness

Cookie Expiration Period (seconds): 0 (0 - 1000000)

En las siguientes secciones se describen los parámetros de Elastic Load Balancing que puede configurar para la aplicación.

Puertos

El balanceador de carga provisionado para atender las solicitudes de la aplicación de Elastic Beanstalk envía solicitudes a las instancias de Amazon EC2 donde se ejecuta la aplicación. El equilibrador de carga provisionado puede atender las solicitudes de los puertos HTTP y HTTPS y direccionarlas a las instancias de Amazon EC2 de la aplicación de AWS Elastic Beanstalk. De forma predeterminada, el balanceador de carga atiende las solicitudes en el puerto HTTP. Al menos uno de los puertos (HTTP o HTTPS) debe estar activo.



These settings allow you to control the behavior of your environment's load balancer.

HTTP Listener Port: 80

HTTPS Listener Port: OFF

SSL Certificate ID

Important

Asegúrese de que el puerto especificado no esté bloqueado; de lo contrario, los usuarios no podrán conectarse a la aplicación de Elastic Beanstalk.

Control del puerto HTTP

Para desactivar el puerto HTTP, seleccione OFF en HTTP Listener Port (HTTP Puerto Listener).
Para activar el puerto HTTP, seleccione un puerto HTTP (por ejemplo, 80) en la lista.

Note

Para tener acceso al entorno usando un puerto diferente del predeterminado 80, por ejemplo, el puerto 8080, añada un agente de escucha al balanceador de carga existente y configúrelo para que escuche en dicho puerto.

Por ejemplo, si usa la [AWS CLI para los equilibradores de carga clásicos](#), escriba el siguiente comando, reemplazando *LOAD_BALANCER_NAME* por el nombre de su equilibrador de carga para Elastic Beanstalk.

```
aws elb create-load-balancer-listeners --load-balancer-name LOAD_BALANCER_NAME
--listeners "Protocol=HTTP, LoadBalancerPort=8080, InstanceProtocol=HTTP,
InstancePort=80"
```

Por ejemplo, si usa la [AWS CLI para los equilibradores de carga de aplicaciones](#), escriba el siguiente comando, reemplazando *LOAD_BALANCER_ARN* por el ARN de su equilibrador de carga para Elastic Beanstalk.

```
aws elbv2 create-listener --load-balancer-arn LOAD_BALANCER_ARN --protocol HTTP
--port 8080
```

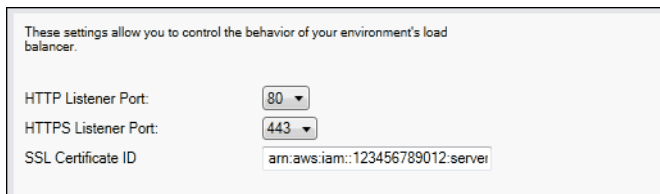
Si quiere que Elastic Beanstalk supervise el entorno, no elimine el agente de escucha del puerto 80.

Control del puerto HTTPS

Elastic Load Balancing admite el protocolo HTTPS/TLS para permitir el cifrado del tráfico en las conexiones cliente con el balanceador de carga. Las conexiones entre el balanceador de carga y las instancias EC2 usan cifrado de texto sin formato. De forma predeterminada, el puerto HTTPS está desactivado.

Para activar el puerto HTTPS

1. Cree un certificado nuevo con AWS Certificate Manager (ACM) o cargue un certificado y una clave en AWS Identity and Access Management (IAM). Para obtener más información sobre cómo solicitar un certificado de ACM, consulte [Solicitar un certificado](#) en la Guía del usuario de AWS Certificate Manager. Para obtener más información sobre la importación de certificados de terceros en ACM, consulte [Importar certificados](#) en la Guía del usuario de AWS Certificate Manager. Si ACM no está [disponible en su región](#), use AWS Identity and Access Management (IAM) para cargar un certificado de terceros. Los servicios de ACM e IAM guardan el certificado y proporcionan un nombre de recurso de Amazon (ARN) para el certificado SSL. Para obtener más información sobre la creación y la carga de certificados en IAM, consulte [Uso de certificados de servidor](#) en la Guía del usuario de IAM.
2. Especifique el puerto HTTPS seleccionando un puerto en HTTPS Listener Port (HTTP Puerto Listener).



These settings allow you to control the behavior of your environment's load balancer.

HTTP Listener Port:	80
HTTPS Listener Port:	443
SSL Certificate ID	arn:aws:iam::123456789012:server

3. En SSL Certificate ID (ID de certificado de SSL), escriba el nombre de recurso de Amazon (ARN) de su certificado SSL. Por ejemplo, **arn:aws:iam::123456789012:server-certificate/abc/certs/build o arn:aws:acm:us-east-2:123456789012:certificate/12345678-12ab-34cd-56ef-12345678**. Utilice el certificado SSL que ha creado o cargado en el paso 1.

Para desactivar el puerto HTTPS, seleccione OFF en HTTPS Listener Port (HTTP Puerto Listener).

Comprobaciones de estado

La definición de comprobación de estado incluye una URL que se utiliza para las consultas del estado de las instancias. De forma predeterminada, Elastic Beanstalk usa contenedores no heredados en TCP:80 y contenedores heredados en HTTP:80. Puede invalidar la URL predeterminada para que coincida con un recurso existente en la aplicación (por ejemplo, /myapp/default.aspx) utilizando el cuadro Application Health Check URL (URL de comprobación de estado de la aplicación). Si anula la URL predeterminada, Elastic Beanstalk utilizará HTTP para

consultar el recurso. Para comprobar si está usando un tipo de contenedor heredado, consulte [the section called “¿Por qué algunas versiones de la plataforma están marcadas como heredadas?”](#)

Puede administrar la configuración de la comprobación de estado a través de la sección EC2 Instance Health Check (Comprobación de estado de instancia EC2) del panel Load Balancing (Balanceo de carga).

These settings allow you to configure how Elastic Beanstalk determines whether an EC2 instance is healthy or not.

Application Health Check:	<input type="text" value="/"/>	
Health Check Interval (seconds):	<input type="text" value="30"/>	(5 - 300)
Health Check Timeout (seconds):	<input type="text" value="5"/>	(2 - 60)
Healthy Check Count Threshold:	<input type="text" value="3"/>	(2 - 10)
Unhealthy Check Count Threshold:	<input type="text" value="5"/>	(2 - 10)

La definición de comprobación de estado incluye una URL que se utiliza para las consultas del estado de las instancias. Invalide la URL predeterminada para que coincida con un recurso existente en la aplicación (por ejemplo, `/myapp/index.jsp`) utilizando el cuadro Application Health Check URL (URL de comprobación de estado de aplicación).

En la siguiente lista se describen los parámetros de comprobación de estado que puede definir para la aplicación.

- En Health Check Interval (seconds) (Intervalo de comprobación de estado [segundos]), especifique el número de segundos que Elastic Load Balancing espera entre las comprobaciones de estado de las instancias de Amazon EC2 de su aplicación.
- En Health Check Timeout (seconds) (Tiempo de espera de comprobación de estado [segundos]), especifique el número de segundos que Elastic Load Balancing espera para recibir una respuesta antes de considerar que la instancia no responde.
- En Healthy Check Count Threshold (Umbral de recuento de estado correcto) y Unhealthy Check Count Threshold (Umbral de recuento de estado incorrecto), especifique el número de sondeos de URL consecutivos que deben realizarse correcta o incorrectamente antes de que Elastic Load Balancing cambie el estado de la instancia. Por ejemplo, si se especifica el valor **5** en Unhealthy Check Count Threshold (Umbral de recuento de estado incorrecto), la dirección URL devuelve un mensaje de error o un mensaje de tiempo de espera agotado cinco veces seguidas antes de que Elastic Load Balancing considere que no se ha superado la comprobación de estado.

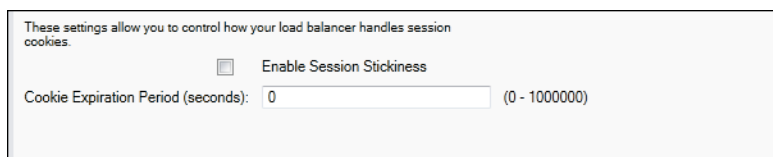
Sesiones

De forma predeterminada, un balanceador de carga direcciona cada solicitud por separado a la instancia de servidor con menor carga. Por su parte, una sesión sticky vincula la sesión de un

usuario con una instancia de servidor específica para que todas las solicitudes que provengan de ese usuario durante la sesión se envíen a la misma instancia de servidor.

Elastic Beanstalk utiliza cookies HTTP generadas por el balanceador de carga cuando las sesiones sticky están habilitadas en una aplicación. El balanceador de carga utiliza una cookie especial generada por el balanceador de carga para hacer un seguimiento de la instancia de aplicación de cada solicitud. Cuando el balanceador de carga recibe una solicitud, primero comprueba si esta cookie está presente en la solicitud. Si lo está, la solicitud se envía a la instancia de aplicación especificada en la cookie. Si no hay ninguna cookie, el balanceador de carga elige una instancia de aplicación en función del algoritmo de balanceo de carga existente. Para vincular las solicitudes posteriores del mismo usuario a esa instancia de aplicación, se inserta una cookie en la respuesta. La configuración de la política define el vencimiento de la cookie, que establece el periodo de validez de cada cookie.

Puede usar la sección Sessions (Sesiones) de la pestaña Load Balancer (Balanceador de carga) para especificar si el balanceador de carga de la aplicación permite o no sesiones sticky.



The screenshot shows a configuration box for session stickiness. At the top, it says "These settings allow you to control how your load balancer handles session cookies". Below this, there is a checkbox labeled "Enable Session Stickiness" which is currently unchecked. Underneath the checkbox, there is a text input field for "Cookie Expiration Period (seconds)" with the value "0" entered. To the right of the input field, the range "(0 - 1000000)" is displayed.

Para obtener más información sobre Elastic Load Balancing, consulte la [Guía para desarrolladores de Elastic Load Balancing](#).

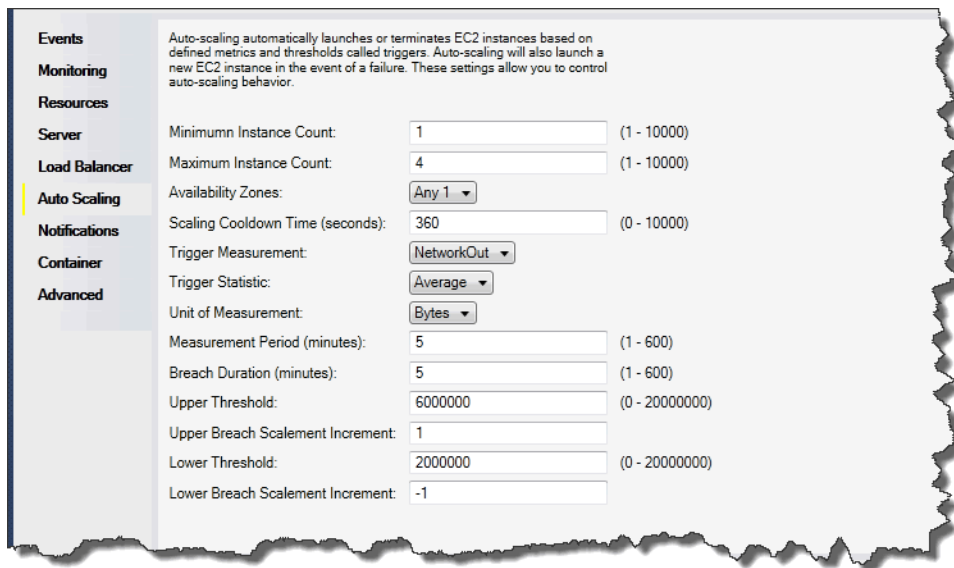
Configuración de Auto Scaling mediante AWS Toolkit for Visual Studio

Amazon EC2 Auto Scaling es un servicio web de Amazon diseñado para lanzar o terminar automáticamente instancias de Amazon EC2 en función de disparadores definidos por el usuario. Los usuarios pueden configurar grupos de Auto Scaling y asociar disparadores a esos grupos para escalar automáticamente los recursos informáticos en función de determinadas métricas, como el uso de ancho de banda o la utilización de CPU. Amazon EC2 Auto Scaling trabaja con Amazon CloudWatch para recuperar las métricas de las instancias de servidor en las que se ejecuta la aplicación.

Amazon EC2 Auto Scaling le permite tomar un grupo de instancias de Amazon EC2 y definir diversos parámetros para hacer que el número de instancias aumente o se reduzca automáticamente. Amazon EC2 Auto Scaling puede añadir o quitar instancias de Amazon EC2 de ese grupo, lo que le ayudará a administrar eficazmente los cambios de tráfico de la aplicación.

Amazon EC2 Auto Scaling también monitoriza el estado de cada instancia de Amazon EC2 que lanza. Si una instancia termina de forma inesperada, Amazon EC2 Auto Scaling lo detecta y lanza una instancia de sustitución. Esta función le permite mantener un número fijo de instancias de Amazon EC2 automáticamente.

Elastic Beanstalk aprovisiona Amazon EC2 Auto Scaling a su aplicación. Puede editar la configuración de las instancias de Amazon EC2 del entorno de Elastic Beanstalk en la pestaña Auto Scaling incluida en la pestaña del entorno de la aplicación en AWS Toolkit for Visual Studio.



Auto-scaling automatically launches or terminates EC2 instances based on defined metrics and thresholds called triggers. Auto-scaling will also launch a new EC2 instance in the event of a failure. These settings allow you to control auto-scaling behavior.

Minimum Instance Count:	<input type="text" value="1"/>	(1 - 10000)
Maximum Instance Count:	<input type="text" value="4"/>	(1 - 10000)
Availability Zones:	<input type="text" value="Any 1"/>	
Scaling Cooldown Time (seconds):	<input type="text" value="360"/>	(0 - 10000)
Trigger Measurement:	<input type="text" value="NetworkOut"/>	
Trigger Statistic:	<input type="text" value="Average"/>	
Unit of Measurement:	<input type="text" value="Bytes"/>	
Measurement Period (minutes):	<input type="text" value="5"/>	(1 - 600)
Breach Duration (minutes):	<input type="text" value="5"/>	(1 - 600)
Upper Threshold:	<input type="text" value="6000000"/>	(0 - 20000000)
Upper Breach Scalement Increment:	<input type="text" value="1"/>	
Lower Threshold:	<input type="text" value="2000000"/>	(0 - 20000000)
Lower Breach Scalement Increment:	<input type="text" value="-1"/>	

En la siguiente sección se explica cómo se configuran los parámetros de Auto Scaling para la aplicación.

Lanzar la configuración

Puede editar la configuración de lanzamiento para controlar la forma en que la aplicación de Elastic Beanstalk aprovisiona los recursos de Amazon EC2 Auto Scaling.

Los cuadros Minimum Instance Count (Número mínimo de instancias) y Maximum Instance Count (Número máximo de instancias) le permiten especificar el tamaño mínimo y máximo del grupo de Auto Scaling que la aplicación de Elastic Beanstalk utiliza.

Auto-scaling automatically launches or terminates EC2 instances based on defined metrics and thresholds called triggers. Auto-scaling will also launch a new EC2 instance in the event of a failure. These settings allow you to control auto-scaling behavior.

Minimum Instance Count:	<input type="text" value="1"/>	(1 - 10000)
Maximum Instance Count:	<input type="text" value="4"/>	(1 - 10000)
Availability Zones:	<input type="text" value="Any"/>	
Scaling Cooldown Time (seconds):	<input type="text" value="360"/>	(0 - 10000)

Note

Para mantener un número fijo de instancias de Amazon EC2, establezca los cuadros Minimum Instance Count (Número mínimo de instancias) y Maximum Instance Count (Número máximo de instancias) en el mismo valor.

El cuadro Availability Zones (Zonas de disponibilidad) le permite especificar el número de zonas de disponibilidad en las que desea que se lancen las instancias de Amazon EC2. Es importante definir este número si desea crear aplicaciones tolerantes a errores. Si una zona de disponibilidad deja de funcionar, las instancias se ejecutarán en las otras zonas de disponibilidad.

Note

En la actualidad, no es posible especificar en qué zona de disponibilidad se encontrará la instancia.

Desencadenadores

Un disparador es un mecanismo de Amazon EC2 Auto Scaling que se define para indicar al sistema cuándo debe aumentar (escalado ascendente) y reducir (escalado descendente) el número de instancias. Puede configurar los disparadores para que se activen de acuerdo con el valor de cualquiera de las métricas publicadas en Amazon CloudWatch, como el uso de la CPU, y determinar si se han cumplido las condiciones especificadas. Cuando se supera el umbral superior o inferior de las condiciones que ha especificado para la métrica durante el periodo de tiempo especificado, el disparador lanza un proceso de larga ejecución llamado actividad de escalado.

Puede definir un desencadenador de escalado para su aplicación de Elastic Beanstalk utilizando AWS Toolkit for Visual Studio.

Trigger Measurement:	<input type="text" value="NetworkOut"/>	
Trigger Statistic:	<input type="text" value="Average"/>	
Unit of Measurement:	<input type="text" value="Bytes"/>	
Measurement Period (minutes):	<input type="text" value="5"/>	(1 - 600)
Breach Duration (minutes):	<input type="text" value="5"/>	(1 - 600)
Upper Threshold:	<input type="text" value="6000000"/>	(0 - 20000000)
Upper Breach Scalement Increment:	<input type="text" value="1"/>	
Lower Threshold:	<input type="text" value="2000000"/>	(0 - 20000000)
Lower Breach Scalement Increment:	<input type="text" value="-1"/>	

Los disparadores de Amazon EC2 Auto Scaling funcionan mediante la observación de una métrica de Amazon CloudWatch específica para una instancia. Se incluyen disparadores para el uso de la CPU, el tráfico de red y la actividad del disco. Use la opción Trigger Measurement (Medición de desencadenador) para seleccionar una métrica para el desencadenador.

En la siguiente lista se describen los parámetros de desencadenador que puede configurar en la Management Console de AWS.

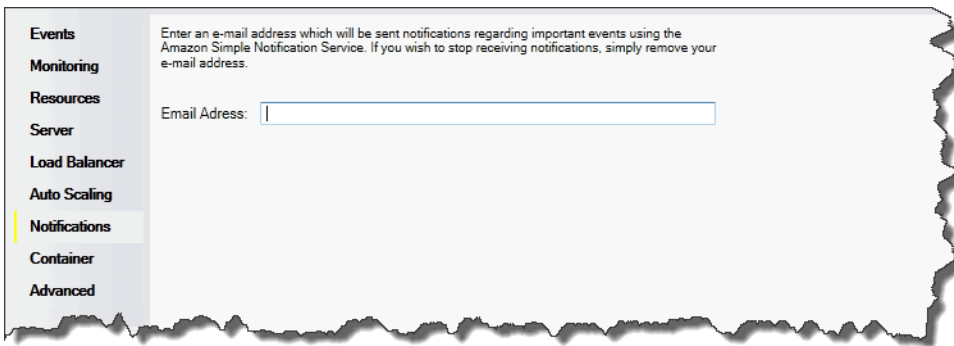
- Puede especificar qué estadística debe usar el disparador. Puede seleccionar Minimum (Mínimo), Maximum (Máximo), Sum (Suma) o Average (Media) para Trigger Statistic (Estadística de desencadenador).
- En Unit of Measurement (Unidad de medición), especifique la unidad de medida del disparador.
- El valor del cuadro Measurement Period (Periodo de medición) especifica la frecuencia con la que Amazon CloudWatch calcula las métricas del disparador. El valor de Breach Duration (Duración de la interrupción) es la cantidad de tiempo que una métrica puede sobrepasar el límite definido (según lo especificado en Upper Threshold (Umbral superior) y Lower Threshold (Umbral inferior)) antes de que se active el disparador.
- En Upper breach scale increment (Incremento de escalado de infracción superior) y Lower breach scale increment (Incremento de escalado de infracción inferior), especifique el número de instancias de Amazon EC2 que se van a agregar o quitar al efectuar una actividad de escalado.

Para obtener más información sobre Amazon EC2 Auto Scaling, consulte la sección Auto Scaling de Amazon EC2 en la [Guía de documentación de Amazon Elastic Compute Cloud](#).

Configuración de notificaciones con AWS Toolkit for Visual Studio

Elastic Beanstalk utiliza Amazon Simple Notification Service (Amazon SNS) para enviarle notificaciones sobre eventos importantes que afectan a la aplicación. Para habilitar las notificaciones

de Amazon SNS, solo tiene que introducir su dirección de correo electrónico en el cuadro Email Address (Dirección de correo electrónico). Para desactivar estas notificaciones, quite la dirección de correo electrónico del cuadro.



Configuración de contenedores .NET con AWS Toolkit for Visual Studio

El panel Container/.NET Options le permite ajustar el comportamiento de las instancias de Amazon EC2 y habilitar o deshabilitar la rotación de registros de Amazon S3. Puede utilizar AWS Toolkit for Visual Studio para configurar la información del contenedor.

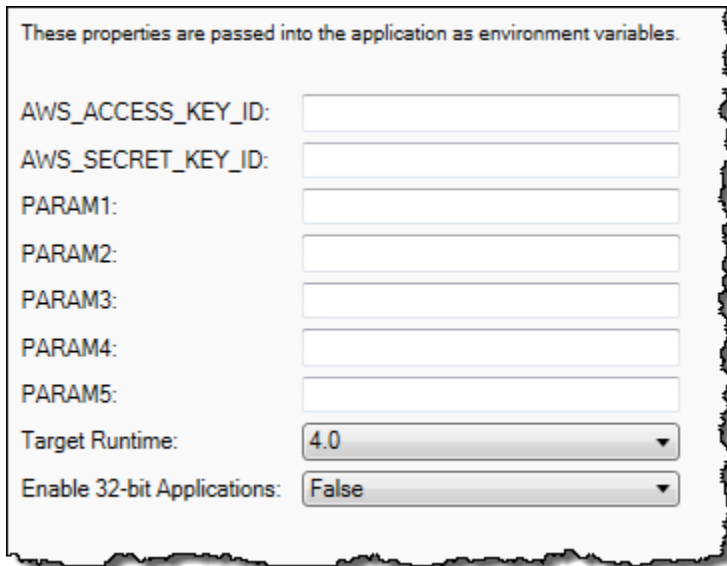
Note

Puede modificar la configuración sin tiempo de inactividad intercambiando el CNAME de los entornos. Para obtener más información, consulte [Implementaciones Blue/Green \(azul/verde\) con Elastic Beanstalk](#).

Si lo desea, puede ampliar el número de parámetros. Para obtener información sobre cómo ampliar los parámetros, consulte [Opciones de configuración](#).

Para obtener acceso al panel Container/.NET Options de su aplicación de Elastic Beanstalk

1. En AWS Toolkit for Visual Studio, expanda el nodo Elastic Beanstalk y el nodo de la aplicación.
2. En AWS Explorer (Explorador de AWS), haga doble clic en su entorno de Elastic Beanstalk.
3. En la parte inferior del panel Overview (Información general), haga clic en la pestaña Configuration (Configuración).
4. En Container (Contenedor), puede configurar las opciones del contenedor.



These properties are passed into the application as environment variables.

AWS_ACCESS_KEY_ID:

AWS_SECRET_KEY_ID:

PARAM1:

PARAM2:

PARAM3:

PARAM4:

PARAM5:

Target Runtime:

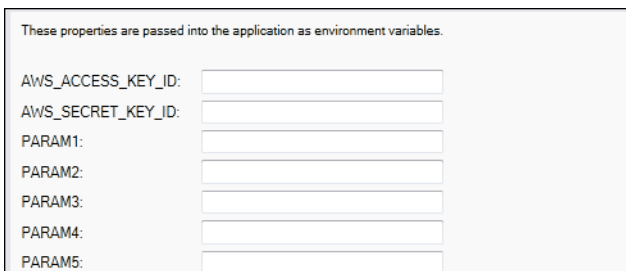
Enable 32-bit Applications:

Opciones del contenedor .NET

Puede elegir la versión de .NET Framework de la aplicación. Elija 2.0 o 4.0 en Target runtime (Tiempo de ejecución objetivo). Seleccione Enable 32-bit Applications (Habilitar aplicaciones de 32 bits) si desea habilitar las aplicaciones de 32 bits.

Application settings

La sección Application Settings (Configuración de aplicación) le permite especificar variables de entorno que se pueden leer desde el código de la aplicación.



These properties are passed into the application as environment variables.

AWS_ACCESS_KEY_ID:

AWS_SECRET_KEY_ID:

PARAM1:

PARAM2:

PARAM3:

PARAM4:

PARAM5:

Administrar cuentas

Si desea configurar diferentes cuentas de AWS para realizar diversas tareas, como pruebas, ensayo y producción, puede agregar, editar y eliminar cuentas con AWS Toolkit for Visual Studio.

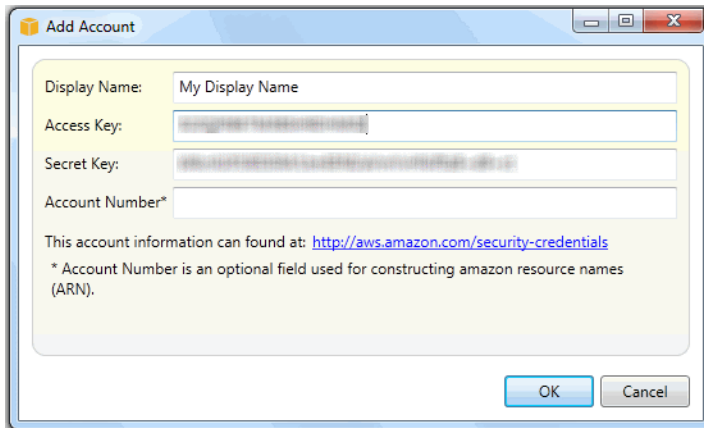
Para administrar varias cuentas

1. Vaya a Visual Studio y en el menú View (Ver), haga clic en AWS Explorer (Explorador de AWS).

2. Junto a la lista Account (Cuenta), haga clic en el botón Add Account (Añadir cuenta).



Se abre el cuadro de diálogo Add Account (Añadir cuenta).



3. Especifique la información solicitada.
4. Ahora, la información de la cuenta aparecerá en la pestaña AWS Explorer (Explorador de AWS). Cuando publique en Elastic Beanstalk, podrá seleccionar la cuenta que le gustaría utilizar.

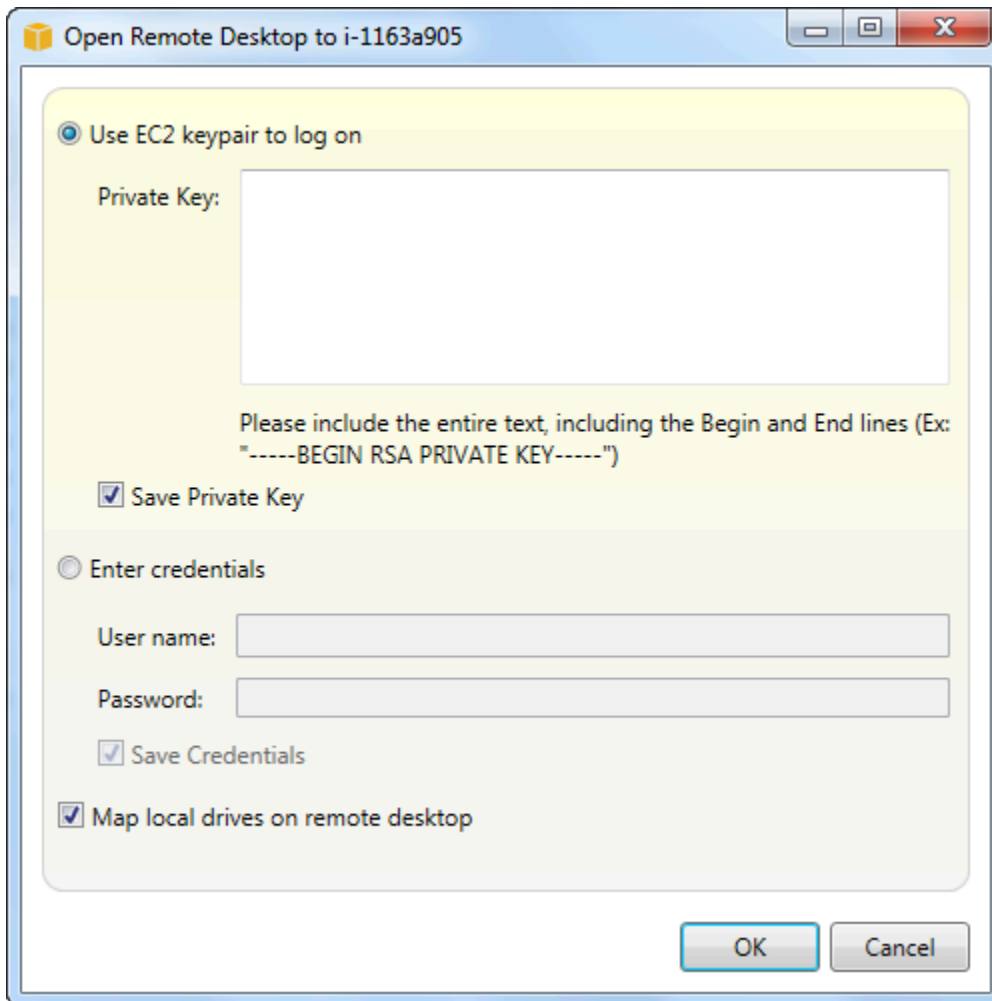
Listado y conexión a instancias de servidor

Puede consultar una lista de las instancias de Amazon EC2 que se ejecutan en el entorno de su aplicación Elastic Beanstalk a través de AWS Toolkit for Visual Studio o de la Management Console de AWS. Puede conectarse a estas instancias a través de Conexión a Escritorio remoto. Para obtener información sobre cómo mostrar las instancias de su servidor y conectarse a ellas a través de la Management Console de AWS, consulte [Listado y conexión a instancias de servidor](#). En la siguiente sección, se detalla un procedimiento para consultar las instancias de servidor y conectarse a ellas a través de AWS Toolkit for Visual Studio.

Para ver las instancias de Amazon EC2 de un entorno y conectarse a ellas

1. Vaya a Visual Studio y en AWS Explorer (Explorador de AWS), expanda el nodo Amazon EC2 y haga doble clic en Instancias (Instancias).

2. En la columna Instance (Instancia), haga clic con el botón derecho en el ID de la instancia de Amazon EC2 que está en ejecución en el balanceador de carga de la aplicación y seleccione Open Remote Desktop (Abrir escritorio remoto) en el menú contextual.



3. Seleccione Use EC2 keypair to log on (Usar par de claves de EC2 para iniciar sesión) y pegue el contenido del archivo de clave privada que utilizó para implementar la aplicación en el cuadro Clave privada. Si lo desea, también puede escribir el nombre de usuario y la contraseña en los cuadros de texto User name (Nombre de usuario) y Password (Contraseña).

Note

Si el par de claves está guardado dentro del conjunto de herramientas, el cuadro de texto no aparecerá.

4. Haga clic en OK (Aceptar).

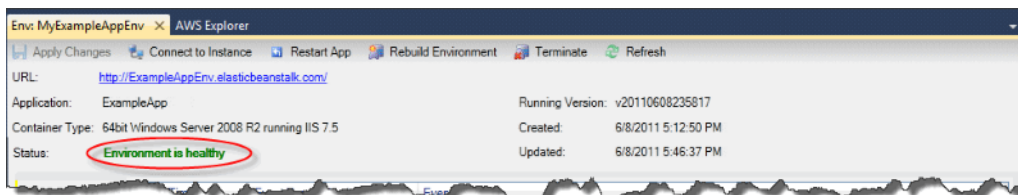
Monitorización del estado de mantenimiento de la aplicación

Cuando ejecuta un sitio web de producción, es importante saber que su aplicación está disponible y responde a las solicitudes. Para ayudarle a monitorizar la capacidad de respuesta de su aplicación, Elastic Beanstalk cuenta con características con las que podrá monitorizar las estadísticas de la aplicación y crear alertas que se activen cuando se superen determinados umbrales.

Para obtener información sobre la monitorización de estado proporcionada por Elastic Beanstalk, consulte [Informes de estado básicos](#).

Para obtener acceso a la información sobre el funcionamiento de su aplicación, puede utilizar el AWS Toolkit for Visual Studio o la Management Console de AWS.

En el conjunto de herramientas, se muestra el estado del entorno y la aplicación en el campo Status (Estado).



Para monitorear el estado de la aplicación

1. En AWS Toolkit for Visual Studio, vaya a AWS Explorer (Explorador de AWS), expanda el nodo Elastic Beanstalk y, a continuación, el nodo de la aplicación.
2. Haga clic con el botón derecho en el entorno de Elastic Beanstalk y, a continuación, haga clic en View Status (Ver estado).
3. En la pestaña del entorno de la aplicación, haga clic en Monitoring (Monitorización).

El panel Monitoring (Monitorización) contiene un conjunto de gráficos donde se muestra el uso que hace de los recursos un entorno de aplicación concreto.



Note

De forma predeterminada, el intervalo de tiempo está establecido en la última hora. Para modificar este valor, en la lista Time Range (Intervalo de tiempo), haga clic en un intervalo de tiempo diferente.

Para ver los eventos asociados a su aplicación, puede utilizar el AWS Toolkit for Visual Studio o la Management Console de AWS.

Para ver los eventos de la aplicación

1. Vaya a AWS Toolkit for Visual Studio y en AWS Explorer (Explorador de AWS), expanda el nodo Elastic Beanstalk y el nodo de la aplicación.
2. Vaya a AWS Explorer (Explorador de AWS), haga clic con el botón derecho en el entorno de Elastic Beanstalk y, a continuación, haga clic en View Status (Ver estado).
3. En la pestaña del entorno de la aplicación, haga clic en Events (Eventos).

Event Time	Event Type	Version Label	Event Details
12/2/2011 3:43:19 PM	INFO	v20111202232102	Environment update completed successfully.
12/2/2011 3:43:19 PM	INFO	v20111202232102	New application version was deployed to running EC2 instances.
12/2/2011 3:43:06 PM	INFO	v20111202232102	Waiting for 2 seconds while EC2 instances download the updated application version.
12/2/2011 3:43:04 PM	INFO	v20111202232102	Deploying version v20111202232102 to 1 instance(s).
12/2/2011 3:42:59 PM	INFO	v20111202230009	Environment update is starting.
12/2/2011 3:30:38 PM	INFO	v20111202230009	Environment health has transitioned from RED to GREEN
12/2/2011 3:29:37 PM	WARN	v20111202230009	Environment health has been set to RED
12/2/2011 3:28:39 PM	INFO	v20111202230009	Launched environment: MyExampleAppEnv. However, there were issues during launch. See event log for details.
12/2/2011 3:28:35 PM	INFO	v20111202230009	Exceeded maximum amount time to wait for the application to become available. Setting environment Ready.
12/2/2011 3:18:13 PM	INFO	v20111202230009	Adding instance 'i-93d6e680' to your environment.
12/2/2011 3:18:50 PM	INFO	v20111202230009	Added EC2 instance 'i-93d6e680' to Auto Scaling Group 'aws-MyExampleAppEnv-5y330GVvOm'.
12/2/2011 3:18:47 PM	INFO	v20111202230009	An EC2 instance has been launched. Waiting for it to be added to Auto Scaling...
12/2/2011 3:18:34 PM	INFO	v20111202230009	Waiting for an EC2 instance to be launched...
12/2/2011 3:18:33 PM	INFO	v20111202230009	Adding Auto Scaling Group 'aws-MyExampleAppEnv-5y330GVvOm' to your environment.
12/2/2011 3:18:33 PM	INFO	v20111202230009	Added URLCheck healthcheck for 'http://MyExampleAppEnv.elasticbeanstalk.com/80/'
12/2/2011 3:18:31 PM	INFO	v20111202230009	Created Auto Scaling trigger named: aws-MyExampleAppEnv-5y330GVvOm.
12/2/2011 3:18:30 PM	INFO	v20111202230009	Created Auto Scaling group named: aws-MyExampleAppEnv-5y330GVvOm.
12/2/2011 3:18:30 PM	INFO	v20111202230009	Created Auto Scaling launch configuration named: aws-MyExampleAppEnv-JDwsd1JA.
12/2/2011 3:18:29 PM	INFO	v20111202230009	Created load balancer named: aws-MyExampleAppEnv.
12/2/2011 3:18:28 PM	INFO	v20111202230009	Created security group named: elasticbeanstalk-windows.
12/2/2011 3:18:28 PM	INFO	v20111202230009	Using elasticbeanstalk-us-east-1-049020475370 as Amazon S3 storage bucket for environment data.

Implementación de aplicaciones de Elastic Beanstalk en .NET con la herramienta de implementación

AWS Toolkit for Visual Studio contiene una herramienta de implementación, una herramienta de línea de comando que proporciona la misma funcionalidad que el asistente de implementación en AWS Toolkit. Puede utilizar la herramienta de implementación en la canalización de compilación o en otros scripts para automatizar las implementaciones en Elastic Beanstalk.

La herramienta de implementación es compatible con implementaciones iniciales y con nuevas implementaciones. Si anteriormente utilizó la herramienta de implementación para implementar la aplicación, puede utilizar el asistente de implementación en Visual Studio para volver a implementarla. Del mismo modo, si la implementación se hizo con el asistente, puede llevarse a cabo de nuevo con la herramienta de implementación.

Note

La herramienta de implementación no aplica [valores recomendados](#) para las opciones de configuración, como la consola o la CLI de EB. Utilice [archivos de configuración](#) para asegurarse de que, al lanzar el entorno, se configuran todas las opciones necesarias.

En este capítulo, se explica paso a paso la implementación de una aplicación .NET de ejemplo en Elastic Beanstalk con la herramienta de implementación. A continuación, se explica una nueva implementación de la aplicación a través de una implementación incremental. Para ver un análisis más exhaustivo sobre la herramienta de implementación, incluidas las opciones de parámetros, consulte [Herramienta de implementación](#).

Requisitos previos

Para utilizar la herramienta de implementación, debe instalar AWS Toolkit for Visual Studio. Si necesita más información sobre los requisitos previos y las instrucciones de instalación, consulte [AWS Toolkit for Microsoft Visual Studio](#).

La herramienta de implementación normalmente se instala en uno de los siguientes directorios de Windows:

32 bits	64 bits
C:\Program Files\AWS Tools\Deployment Tool\awsdeploy.exe	C:\Program Files (x86)\AWS Tools\Deployment Tool\awsdeploy.exe

Implementar en Elastic Beanstalk

Para implementar la aplicación de ejemplo en Elastic Beanstalk con la herramienta de implementación, primero debe modificar el archivo de configuración `ElasticBeanstalkDeploymentSample.txt` que se incluye en el directorio `Samples`. Este archivo de configuración contiene la información necesaria para implementar su aplicación, como el nombre y la versión de la aplicación, el nombre del entorno y las credenciales de acceso de AWS. Después de modificar el archivo de configuración, puede utilizar la línea de comando para implementar la aplicación de ejemplo. El archivo de implementación web se carga en Amazon S3 y se registra como una nueva versión de la aplicación con Elastic Beanstalk. La implementación de la aplicación tarda unos minutos. Una vez que el entorno tiene un estado correcto, la herramienta de implementación genera una URL para la aplicación en ejecución.

Para implementar una aplicación .NET en Elastic Beanstalk

1. En el subdirectorio `Samples` en el que se instaló la herramienta de implementación, abra `ElasticBeanstalkDeploymentSample.txt` y especifique la clave de acceso de AWS y la clave secreta de AWS, tal y como se muestra en el siguiente ejemplo.

```
### AWS Access Key and Secret Key used to create and deploy the application instance
AWSAccessKey = AKIAIOSFODNN7EXAMPLE
AWSSecretKey = wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
```

Note

Para acceder a la API, necesita un ID de clave de acceso y una clave de acceso secreta. Utilice las claves de acceso de usuario de IAM en lugar de las claves de acceso de Usuario raíz de la cuenta de AWS. Para obtener más información sobre la creación de

claves de acceso, consulte [Administración de las claves de acceso de los usuarios de IAM](#) en la Guía del usuario de IAM

2. En la línea de comando, escriba lo siguiente:

```
C:\Program Files (x86)\AWS Tools\Deployment Tool>awsdeploy.exe /w Samples
\ElasticBeanstalkDeploymentSample.txt
```

La aplicación tarda unos minutos en implementarse. Si la implementación se realiza correctamente, aparecerá el mensaje, `Application deployment completed; environment health is Green`.

Note

Si aparece el error siguiente, el CNAME ya existe.

```
[Error]: Deployment to AWS Elastic Beanstalk failed with exception: DNS name
(MyAppEnv.elasticbeanstalk.com) is not available.
```

Como el CNAME debe ser único, debe cambiar `Environment.CNAME` en `ElasticBeanstalkDeploymentSample.txt`.

3. En el navegador web, vaya a la URL de la aplicación en ejecución. La URL tendrá el formato `<CNAME.elasticbeanstalk.com >` (por ejemplo, **MyAppEnv.elasticbeanstalk.com**).

Migración de la aplicación .NET local a Elastic Beanstalk

Si está pensando en migrar su aplicación .NET de un servidor en las instalaciones a Amazon Web Services (AWS), el Asistente de migración de .NET de AWS Elastic Beanstalk podría serle útil. El asistente es una utilidad interactiva de PowerShell que migra una aplicación .NET desde una instancia de Windows Server con IIS que se ejecuta en las instalaciones a AWS Elastic Beanstalk. El asistente puede migrar un sitio web completo a Elastic Beanstalk con cambios mínimos o sin ningún cambio.

Para obtener más información sobre el Asistente de migración de .NET para AWS Elastic Beanstalk y descargarlo, consulte el repositorio <https://github.com/aws-labs/windows-web-app-migration-assistant> de GitHub.

Si la aplicación contiene bases de datos de Microsoft SQL Server, la documentación del asistente de GitHub dispone de varias opciones para migrarlas.

Implementación de aplicaciones Node.js en Elastic Beanstalk

AWS Elastic Beanstalk for Node.js facilita la implementación, la administración y el escalado de las aplicaciones web de Node.js mediante Amazon Web Services. Elastic Beanstalk para Node.js está disponible para todo aquel que desee desarrollar o alojar una aplicación web con Node.js. En este capítulo se proporcionan step-by-step instrucciones para implementar la aplicación web Node.js en Elastic Beanstalk y se proporcionan tutoriales para tareas comunes, como la integración de bases de datos y el trabajo con el marco Express.

Después de implementar la aplicación de Elastic Beanstalk, puede seguir utilizando la CLI de EB para administrar la aplicación y el entorno, o puede usar la consola de Elastic Beanstalk o las API. AWS CLI

Temas

- [QuickStart: Implemente una aplicación Node.js en Elastic Beanstalk](#)
- [Configuración del entorno de desarrollo de Node.js](#)
- [Uso de la plataforma Node.js de Elastic Beanstalk](#)
- [Más ejemplos de aplicaciones y tutoriales para Node.js](#)
- [Implementación de una aplicación Express en Elastic Beanstalk](#)
- [Implementación de una aplicación Express con clústeres en Elastic Beanstalk](#)
- [Implementación de una aplicación Node.js con DynamoDB en Elastic Beanstalk](#)
- [Agregar una instancia de base de datos de Amazon RDS DB a un entorno de aplicaciones Node.js](#)
- [Recursos](#)

QuickStart: Implemente una aplicación Node.js en Elastic Beanstalk

Este QuickStart tutorial explica el proceso de creación de una aplicación Node.js e implementarla en un AWS Elastic Beanstalk entorno.

Note

Este QuickStart tutorial está destinado a fines de demostración. No utilice la aplicación creada en este tutorial para el tráfico de producción.

Secciones

- [Tu AWS cuenta](#)
- [Requisitos previos](#)
- [Paso 1: Cree una aplicación Node.js](#)
- [Paso 2: Ejecute la aplicación localmente](#)
- [Paso 3: Implemente la aplicación Node.js con la CLI de EB](#)
- [Paso 4: Ejecute la aplicación en Elastic Beanstalk](#)
- [Paso 5: Eliminar](#)
- [AWS recursos para su aplicación](#)
- [Siguiendo pasos](#)
- [Implemente con la consola de Elastic Beanstalk](#)

Tu AWS cuenta

Si aún no eres AWS cliente, debes crear una AWS cuenta. El registro le permite acceder a Elastic Beanstalk AWS y a otros servicios que necesite.

Si ya tiene una AWS cuenta, puede pasar a [Requisitos previos](#)

Crea una AWS cuenta

Inscríbese en una Cuenta de AWS

Si no tiene una Cuenta de AWS, complete los siguientes pasos para crearlo.

Para suscribirte a una Cuenta de AWS

1. Abra <https://portal.aws.amazon.com/billing/signup>.
2. Siga las instrucciones que se le indiquen.

Parte del procedimiento de registro consiste en recibir una llamada telefónica e indicar un código de verificación en el teclado del teléfono.

Cuando te registras en un Cuenta de AWS, Usuario raíz de la cuenta de AWS se crea un. El usuario raíz tendrá acceso a todos los Servicios de AWS y recursos de esa cuenta. Como práctica recomendada de seguridad, asigne acceso administrativo a un usuario y utilice únicamente el usuario raíz para realizar [tareas que requieren acceso de usuario raíz](#).

AWS te envía un correo electrónico de confirmación una vez finalizado el proceso de registro. Puede ver la actividad de la cuenta y administrar la cuenta en cualquier momento entrando en <https://aws.amazon.com/> y seleccionando Mi cuenta.

Creación de un usuario con acceso administrativo

Después de crear un usuario administrativo Cuenta de AWS, asegúrelo Usuario raíz de la cuenta de AWS AWS IAM Identity Center, habilite y cree un usuario administrativo para no usar el usuario root en las tareas diarias.

Proteja su Usuario raíz de la cuenta de AWS

1. Inicie sesión [AWS Management Console](#) como propietario de la cuenta seleccionando el usuario root e introduciendo su dirección de Cuenta de AWS correo electrónico. En la siguiente página, escriba su contraseña.

Para obtener ayuda para iniciar sesión con el usuario raíz, consulte [Iniciar sesión como usuario raíz](#) en la Guía del usuario de AWS Sign-In .

2. Active la autenticación multifactor (MFA) para el usuario raíz.

Para obtener instrucciones, consulte [Habilitar un dispositivo MFA virtual para el usuario Cuenta de AWS raíz \(consola\)](#) en la Guía del usuario de IAM.

Creación de un usuario con acceso administrativo

1. Activar IAM Identity Center.

Consulte las instrucciones en [Activar AWS IAM Identity Center](#) en la Guía del usuario de AWS IAM Identity Center .

2. En IAM Identity Center, conceda acceso administrativo a un usuario.

Para ver un tutorial sobre su uso Directorio de IAM Identity Center como fuente de identidad, consulte [Configurar el acceso de los usuarios con la configuración predeterminada Directorio de IAM Identity Center en la](#) Guía del AWS IAM Identity Center usuario.

Iniciar sesión como usuario con acceso de administrador

- Para iniciar sesión con el usuario de IAM Identity Center, utilice la URL de inicio de sesión que se envió a la dirección de correo electrónico cuando creó el usuario de IAM Identity Center.

Para obtener ayuda para iniciar sesión con un usuario del Centro de identidades de IAM, consulte [Iniciar sesión en el portal de AWS acceso](#) en la Guía del AWS Sign-In usuario.

Concesión de acceso a usuarios adicionales

1. En IAM Identity Center, cree un conjunto de permisos que siga la práctica recomendada de aplicar permisos de privilegios mínimos.

Para conocer las instrucciones, consulte [Create a permission set](#) en la Guía del usuario de AWS IAM Identity Center .

2. Asigne usuarios a un grupo y, a continuación, asigne el acceso de inicio de sesión único al grupo.

Para conocer las instrucciones, consulte [Add groups](#) en la Guía del usuario de AWS IAM Identity Center .

Requisitos previos

Para seguir los procedimientos de esta guía, necesitará un shell o un terminal de línea de comando donde pueda ejecutar los comandos. Los comandos aparecen en listas y van precedidos del símbolo del sistema (\$) y del nombre del directorio actual, si es aplicable.

```
~/eb-project$ this is a command  
this is output
```

En Linux y macOS, puede utilizar el administrador de shell y paquetes preferido. En Windows, puede [instalar el subsistema de Windows para Linux](#) y obtener una versión de Ubuntu y Bash integrada en Windows.

CLI DE EB

En este tutorial también se utiliza la interfaz de línea de comandos de Elastic Beanstalk (CLI de EB). Para obtener detalles sobre la instalación y configuración de la CLI de EB, consulte [Instalación de la CLI de EB](#) y [Configuración de la CLI de EB](#).

Node.js

Instale Node.js en su máquina local siguiendo [Cómo instalar Node.js](#) en el sitio web Node.js.

Compruebe la instalación de Node.js ejecutando el siguiente comando.

```
~$ node -v
```

Paso 1: Cree una aplicación Node.js

Cree el directorio del proyecto.

```
~$ mkdir eb-nodejs
~$ cd eb-nodejs
```

A continuación, vamos a crear una aplicación que implementará con Elastic Beanstalk. Crearemos el servicio web RESTful "Hello World".

Example `~/eb-nodejs/server.js`

```
const http = require('node:http');

const hostname = '127.0.0.1';
const port = 8080;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello Elastic Beanstalk!\n');
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```


Esta aplicación abre un oyente en el puerto 8080. Elastic Beanstalk reenvía las solicitudes a la aplicación en el puerto 8080 de forma predeterminada para Node.js.

Paso 2: Ejecute la aplicación localmente

Ejecute el siguiente comando para ejecutar la aplicación localmente.

```
~/eb-nodejs$ node server.js
```

Debería ver el siguiente texto.

```
Server running at http://127.0.0.1:8080/
```

Introduce la dirección URL `http://127.0.0.1:8080/` en tu navegador web. El navegador debería mostrar «¡Hola, Elastic Beanstalk!».

Paso 3: Implemente la aplicación Node.js con la CLI de EB

Ejecute los siguientes comandos para crear un entorno de Elastic Beanstalk para esta aplicación.

Para crear un entorno e implementar la aplicación Node.js

1. Inicialice el repositorio de la CLI de EB con el comando `eb init`.

```
~/eb-nodejs$ eb init -p node.js nodejs-tutorial --region us-east-2
```

Este comando crea una aplicación con el nombre `nodejs-tutorial` y configura el repositorio local para crear entornos con la versión más reciente de la plataforma Node.js.

2. (Opcional) Ejecute de nuevo `eb init` para configurar un par de claves predeterminadas de forma que pueda usar SSH para conectarse a la instancia EC2 donde se ejecuta la aplicación.

```
~/eb-nodejs$ eb init
Do you want to set up SSH for your instances?
(y/n): y
Select a keypair.
1) my-keypair
2) [ Create new KeyPair ]
```

Seleccione un par de claves si ya tiene uno o siga las instrucciones para crear uno. Si no ve el símbolo del sistema o más adelante necesita cambiar la configuración, ejecute `eb init -i`.

3. Cree un entorno e implemente la aplicación en él con `eb create`. Elastic Beanstalk crea automáticamente un archivo zip para la aplicación y lo despliega en una instancia EC2 del entorno. Tras implementar la aplicación, Elastic Beanstalk la inicia en el puerto 8080.

```
~/eb-nodejs$ eb create nodejs-env
```

Elastic Beanstalk tarda unos cinco minutos en crear el entorno.

Paso 4: Ejecute la aplicación en Elastic Beanstalk

Cuando finalice el proceso de creación del entorno, abra su sitio web con `eb open`

```
~/eb-nodejs$ eb open
```

¡Enhorabuena! ¡Ha implementado una aplicación de Node.js con Elastic Beanstalk! Se abre una ventana del navegador con el nombre de dominio creado para su aplicación.

Paso 5: Eliminar

Puede cerrar el entorno cuando termine de trabajar con la aplicación. Elastic Beanstalk AWS cancela todos los recursos asociados a su entorno.

Para finalizar el entorno de Elastic Beanstalk con la CLI de EB, ejecute el siguiente comando.

```
~/eb-nodejs$ eb terminate
```

AWS recursos para su aplicación

Acabas de crear una aplicación de instancia única. Sirve como una aplicación de muestra sencilla con una sola instancia EC2, por lo que no requiere balanceo de carga ni escalado automático. Para las aplicaciones de instancia única, Elastic Beanstalk crea los siguientes recursos: AWS

- EC2 instance (Instancia EC2): máquina virtual de Amazon EC2 configurada para ejecutar aplicaciones web en la plataforma que elija.

Cada plataforma ejecuta un conjunto distinto de software, archivos de configuración y scripts compatibles con una determinada versión de lenguaje, marco y contenedor web (o una combinación de ellos). La mayoría de las plataformas utilizan Apache o nginx como un proxy

inverso que procesa el tráfico web delante de la aplicación web, reenvía las solicitudes a esta, administra los recursos estáticos y genera registros de acceso y errores.

- Instance security group (Grupo de seguridad de la instancia): grupo de seguridad de Amazon EC2 configurado para permitir el tráfico entrante en el puerto 80. Este recurso permite que el tráfico HTTP procedente del balanceador de carga llegue a la instancia EC2 en la que se ejecuta la aplicación web. De forma predeterminada, el tráfico no está permitido en otros puertos.
- Bucket de Amazon S3: ubicación de almacenamiento para el código fuente, los registros y otros artefactos que se crean al utilizar Elastic Beanstalk.
- CloudWatch Alarmas de Amazon: dos CloudWatch alarmas que monitorean la carga de las instancias de su entorno y se activan si la carga es demasiado alta o demasiado baja. Cuando se activa una alarma, en respuesta, el grupo de Auto Scaling aumenta o reduce los recursos.
- AWS CloudFormation pila: Elastic AWS CloudFormation Beanstalk se utiliza para lanzar los recursos de su entorno y propagar los cambios de configuración. Los recursos se definen en una plantilla que puede verse en la [consola de AWS CloudFormation](#).
- Nombre de dominio: nombre de dominio que direcciona el tráfico a la aplicación web con el formato *subdominio.región.elasticbeanstalk.com*.

Elastic Beanstalk administra todos estos recursos. Cuando termina su entorno, Elastic Beanstalk termina todos los recursos que este contiene.

Siguientes pasos

Una vez que disponga de un entorno que ejecute una aplicación, podrá implementar una nueva versión de la aplicación o una aplicación distinta en cualquier momento. La implementación de una nueva versión de la aplicación es una tarea muy rápida, ya que no se requiere aprovisionar ni reiniciar instancias EC2. También puede explorar su nuevo entorno con la consola de Elastic Beanstalk. Para ver los pasos detallados, consulte [Explore su entorno](#) en el capítulo Introducción de esta guía.

Pruebe más tutoriales

Si desea probar otros tutoriales con diferentes aplicaciones de ejemplo, consulte [Más ejemplos de aplicaciones y tutoriales para Node.js](#).

Una vez que haya implementado una o dos aplicaciones de ejemplo y esté listo para empezar a desarrollar y ejecutar aplicaciones de Node.js de forma local, consulte [Configuración del entorno de desarrollo de Node.js](#).

Implemente con la consola de Elastic Beanstalk

También puede usar la consola de Elastic Beanstalk para iniciar la aplicación de muestra. Para ver los pasos detallados, consulte [Crear una aplicación de ejemplo](#) en el capítulo Introducción de esta guía.

Configuración del entorno de desarrollo de Node.js

Configure el entorno de desarrollo de Node.js para probar la aplicación localmente antes de implementarla en AWS Elastic Beanstalk. En este tema, se explican los pasos de configuración del entorno de desarrollo y se incluyen enlaces a páginas de instalación donde encontrará herramientas útiles.

Para ver procedimientos de configuración y herramientas comunes que se utilizan en todos los lenguajes, consulte [Configuración de su máquina de desarrollo](#).

Temas

- [Instale Node.js](#)
- [Confirmar la instalación de npm](#)
- [Instalar el SDK de AWS para Node.js](#)
- [Instale el generador Express](#)
- [Configurar una plataforma y un servidor Express](#)

Instale Node.js

Instale Node.js para ejecutar aplicaciones Node.js localmente. Si no tiene ninguna preferencia, obtenga la versión más reciente admitida por Elastic Beanstalk. Consulte [Node.js](#) en el documento AWS Elastic Beanstalk Plataformas para ver una lista de las versiones que se admiten.

Descargue Node.js en nodejs.org.

Confirmar la instalación de npm

Node.js usa el administrador de paquetes npm como ayuda para instalar herramientas y marcos para usarlos en su aplicación. Dado que npm se distribuye con Node.js, lo instalará automáticamente

cuando descargue e instale Node.js. Para confirmar que tiene npm instalado, puede ejecutar el siguiente comando:

```
$ npm -v
```

Para obtener más información sobre npm, visite el sitio web de [npmjs](https://www.npmjs.com/).

Instalar el SDK de AWS para Node.js

Si necesita administrar recursos de AWS en la aplicación, instale el SDK for JavaScript en Node.js de AWS. Instale el SDK con npm:

```
$ npm install aws-sdk
```

Para obtener más información, consulte la página de inicio del [SDK for JavaScript en Node.js de AWS](https://aws.amazon.com/sdk-for-javascript/).

Instale el generador Express

Express es una plataforma de aplicaciones web que se ejecuta en Node.js. Para usarlo, primero debe instalar la aplicación de línea de comandos del generador Express. Una vez instalado el generador Express, puede ejecutar el comando `express` para generar una estructura de proyecto base para su aplicación web. Una vez el proyecto base, los archivos y las dependencias están instalados, puede iniciar un servidor Express local en su máquina de desarrollo.

Note

- En estos pasos, se describe la instalación del generador Express en un sistema operativo Linux.
- En el caso de Linux, en función de su nivel de permiso para los directorios del sistema, es posible que tenga que escribir delante de algunos de estos comandos con `sudo`.

Para instalar el generador Express en su entorno de desarrollo

1. Cree un directorio de trabajo para su plataforma y servidor Express.

```
~$ mkdir node-express
```

```
~$ cd node-express
```

2. Instale Express globalmente para disponer de acceso al comando `express`.

```
~/node-express$ npm install -g express-generator
```

3. En función del sistema operativo, es posible que tenga que definir la ruta para ejecutar el comando `express`. La salida del paso anterior proporciona información si necesita definir la variable de ruta. A continuación se muestra un ejemplo para Linux.

```
~/node-express$ export PATH=$PATH:/usr/local/share/npm/bin/express
```

Cuando siga los tutoriales de este capítulo, tendrá que ejecutar el comando `express` desde distintos directorios. Cada tutorial configura una estructura de proyecto Express de base en su propio directorio.

Ya tiene instalado el generador de línea de comandos Express. Puede usarlo para crear un directorio marco para su aplicación web, configurar dependencias e iniciar el servidor de aplicaciones web. A continuación, analizaremos los pasos para lograrlo en el directorio de `node-express` que hemos creado.

Configurar una plataforma y un servidor Express

Siga estos pasos para crear los directorios y el contenido del marco de Express de base. Los tutoriales en este capítulo también incluyen estos pasos para configurar el marco de Express de base en cada uno de los directorios de aplicaciones del tutorial.

Para configurar un marco y un servidor Express

1. Ejecute el comando `express`. Esto genera `package.json`, `app.js` y unos directorios.

```
~/node-express$ express
```

Cuando se le pregunte, escriba **y** si desea continuar.

2. Configure las dependencias locales.

```
~/node-express$ npm install
```

3. Compruebe que el servidor de aplicaciones web se inicie.

```
~/node-express$ npm start
```

Debería ver un resultado similar a este:

```
> nodejs@0.0.0 start /home/local/user/node-express
> node ./bin/www
```

El servidor se ejecuta en el puerto 3000 de forma predeterminada. Para probarlo, ejecute `curl http://localhost:3000` en otro terminal o abra un navegador en el equipo local e ingrese el `http://localhost:3000` de la dirección URL.

Presione Ctrl+C para detener el servidor.

Uso de la plataforma Node.js de Elastic Beanstalk

La plataforma Node.js de AWS Elastic Beanstalk es un conjunto de [versiones de la plataforma](#) de aplicaciones web Node.js que se ejecutan en un servidor proxy NGINX.

Elastic Beanstalk proporciona [opciones de configuración](#) que se pueden utilizar para personalizar el software que se ejecuta en las instancias de EC2 del entorno de Elastic Beanstalk. Puede [configurar las variables de entorno](#) que necesita la aplicación, habilitar la rotación de registros en Amazon S3 y asignar carpetas del código fuente de la aplicación que contengan archivos estáticos a rutas proporcionadas por el servidor proxy.

En la consola de Elastic Beanstalk hay opciones de configuración disponibles para [modificar la configuración de un entorno en ejecución](#). Para evitar perder la configuración del entorno cuando lo termina, puede usar las [configuraciones guardadas](#) para guardar la configuración y aplicarla posteriormente a otro entorno.

Para guardar la configuración en el código fuente, puede incluir [archivos de configuración](#). Los valores de configuración de los archivos de configuración se aplican cada vez que crea un entorno o que implementa la aplicación. También puede usar archivos de configuración para instalar paquetes, ejecutar scripts y llevar a cabo otras operaciones de personalización de instancias durante las implementaciones.

Puede [incluir un archivo Package.json](#) en el paquete de código fuente para instalar paquetes durante la implementación, para proporcionar un comando de inicio y para especificar la versión de

Node.js que quiere que la aplicación utilice. Puede incluir un [archivo npm-shrinkwrap.json](#) para bloquear las versiones de dependencia.

La plataforma Node.js incluye un servidor proxy para distribuir recursos estáticos, reenviar el tráfico a la aplicación y comprimir las respuestas. Puede [ampliar o invalidar la configuración del proxy predeterminada](#) para escenarios avanzados.

Hay varias opciones para iniciar su aplicación. Puede agregar un [Procfile](#) a su paquete de código fuente para especificar el comando que inicia la aplicación. Si no proporciona un Procfile, pero proporciona un archivo `package.json`, Elastic Beanstalk ejecuta `npm start`. Si no proporciona ninguno, Elastic Beanstalk busca los archivos `app.js` o `server.js`, en este orden, y ejecuta el script.

La configuración aplicada en la consola de Elastic Beanstalk anula la misma configuración en los archivos de configuración, si existe. Esto le permite tener la configuración predeterminada en los archivos de configuración y anularla con la configuración específica del entorno en la consola. Para obtener más información acerca de la prioridad y otros métodos para cambiar valores de configuración, consulte [Opciones de configuración](#).

Para obtener más información sobre las diversas formas en las que puede ampliar una plataforma Elastic Beanstalk basada en Linux, consulte [the section called “Ampliación de plataformas Linux”](#).

Configuración de su entorno de Node.js

Puede utilizar la configuración de la plataforma Node.js para ajustar el comportamiento de las instancias de Amazon EC2. Puede editar la configuración de la instancia de Amazon EC2 para el entorno de Elastic Beanstalk mediante la consola de Elastic Beanstalk.

Utilice la consola de Elastic Beanstalk para habilitar la rotación de registros en Amazon S3 y configurar variables que la aplicación pueda leer desde el entorno.

Para configurar su entorno Node.js en la consola de Elastic Beanstalk

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Actualizaciones, supervisión y registro, seleccione Editar.

Opciones de contenedor

Puede especificar estas opciones específicas de la plataforma:

- Proxy server (Servidor proxy): el servidor proxy que utilizar en las instancias del entorno. Se utiliza NGNIX de forma predeterminada.

Log Options (Opciones de registro)

La sección Log Options (Opciones de registro) tiene dos valores:

- Perfil de instancia: especifica el perfil de instancia que tiene permiso para obtener acceso al bucket de Amazon S3 asociado a la aplicación.
- Enable log file rotation to Amazon S3 (Habilitar la rotación de archivos de registro para Amazon S3): especifica si los archivos de registro de las instancias de Amazon EC2 de la aplicación se copian en el bucket de Amazon S3 asociado a la aplicación.

Archivos estáticos

Para mejorar el rendimiento, puede utilizar la sección Static files (Archivos estáticos) para configurar el servidor proxy para entregar archivos estáticos de servidor (por ejemplo, HTML o imágenes) desde un conjunto de directorios dentro de su aplicación web. Para cada directorio, se establece la ruta virtual para la asignación de directorios. Cuando el servidor proxy recibe una solicitud de un archivo en la ruta especificada, proporciona directamente el archivo en lugar de direccionar la solicitud a la aplicación.

Para obtener más información sobre la configuración de archivos estáticos mediante archivos de configuración o la consola de Elastic Beanstalk, consulte [the section called “Archivos estáticos”](#).

Propiedades del entorno

Utilice la sección Environment Properties (Propiedades de entorno) para especificar opciones de configuración del entorno en las instancias Amazon EC2 que ejecutan la aplicación. Estos ajustes se pasan como pares clave-valor a la aplicación.

Dentro del entorno Node.js que se ejecuta en AWS Elastic Beanstalk, puede acceder a las variables del entorno mediante `process.env.ENV_VARIABLE`.

```
var endpoint = process.env.API_ENDPOINT
```

La plataforma Node.js establece la variable de entorno PORT en el puerto al que el servidor proxy pasa el tráfico. Para obtener más información, consulte [Configuración del servidor proxy](#).

Para obtener más información, consulte [Propiedades del entorno y otras opciones del software](#).

Configuración de un entorno Node.js de la AMI de Amazon Linux (anterior a Amazon Linux 2)

Las siguientes categorías de configuración del software de la consola solo se admite en un entorno Node.js de Elastic Beanstalk que utiliza una versión de la plataforma de la AMI de Amazon Linux (anterior a Amazon Linux 2).

Notas

- La información de este tema solo se aplica a ramificaciones de plataforma basadas en la AMI de Amazon Linux (AL1). Las ramificaciones de la plataforma AL2023/AL2 son incompatibles con las versiones anteriores de la plataforma de la AMI de Amazon Linux (AL1) y requieren ajustes de configuración diferentes.
- El [18 de julio de 2022](#), Elastic Beanstalk estableció el estado de todas las ramificaciones de la plataforma basadas en la AMI de Amazon Linux (AL1) como retirado. Para obtener más información sobre la migración a una ramificación de la plataforma Amazon Linux 2023 actual y totalmente compatible, consulte [Migración de su aplicación de Linux de Elastic Beanstalk a Amazon Linux 2023 o Amazon Linux 2](#).

Opciones de contenedores: AMI de Amazon Linux (AL1)

En la página de configuración, especifique lo siguiente:

- Proxy Server (Servidor proxy): especifica el servidor web que se utiliza para enviar las conexiones a Node.js. Se utiliza NGINX de forma predeterminada. Si selecciona none (ninguno), las asignaciones de archivos estáticos no se aplican y se desactiva la compresión GZIP.
- Versión de Node.js: especifica la versión de Node.js. Para obtener una lista de las versiones de Node.js compatibles, consulte [Node.js](#) en la guía Plataformas de AWS Elastic Beanstalk.
- Compresión GZIP: especifica si la compresión GZIP está habilitada. La compresión GZIP está habilitada de forma predeterminada.
- Node command (Comando de Node): permite escribir el comando que se utiliza para iniciar la aplicación de Node.js. Una cadena vacía (la predeterminada) significa que Elastic Beanstalk usa `app.js`, `server.js` y, luego, `npm start`.

Espacio de nombres de configuración de Node.js

Puede usar un [archivo de configuración](#) para definir opciones de configuración y realizar otras tareas de configuración en las instancias durante las implementaciones. Las opciones de configuración se pueden definir a través del servicio de Elastic Beanstalk o la plataforma que utilice y están organizadas por espacios de nombres.

Puede elegir el proxy que se utilizará en las instancias para el entorno mediante el uso del espacio de nombres `aws:elasticbeanstalk:environment:proxy`. En el siguiente ejemplo se configura el entorno para utilizar el servidor proxy Apache HTTPD.

Example `.ebextensions/nodejs-settings.config`

```
option_settings:
  aws:elasticbeanstalk:environment:proxy:
    ProxyServer: apache
```

Puede configurar el proxy para que sirva archivos estáticos mediante el espacio de nombres `aws:elasticbeanstalk:environment:proxy:staticfiles`. Para obtener más información y un ejemplo, consulte [the section called “Archivos estáticos”](#).

Elastic Beanstalk cuenta con numerosas opciones de configuración para personalizar el entorno. Además de los archivos de configuración, también puede definir opciones en la consola, configuraciones guardadas, la CLI de EB o la AWS CLI. Para obtener más información, consulte [Opciones de configuración](#).

Plataforma Node.js de la AMI de Amazon Linux (anterior a Amazon Linux 2)

Si el entorno Node.js de Elastic Beanstalk utiliza una versión de la plataforma de la AMI de Amazon Linux (anterior a Amazon Linux 2), tenga en cuenta las configuraciones y las recomendaciones específicas de esta sección.

Notas

- La información de este tema solo se aplica a ramificaciones de plataforma basadas en la AMI de Amazon Linux (AL1). Las ramificaciones de la plataforma AL2023/AL2 son incompatibles con las versiones anteriores de la plataforma de la AMI de Amazon Linux (AL1) y requieren ajustes de configuración diferentes.
- El [18 de julio de 2022](#), Elastic Beanstalk estableció el estado de todas las ramificaciones de la plataforma basadas en la AMI de Amazon Linux (AL1) como retirado. Para obtener más información sobre la migración a una ramificación de la plataforma Amazon Linux 2023 actual y totalmente compatible, consulte [Migración de su aplicación de Linux de Elastic Beanstalk a Amazon Linux 2023 o Amazon Linux 2](#).

Opciones de configuración específicas de la plataforma de Node.js: AMI de Amazon Linux (AL1)

Elastic Beanstalk admite algunas opciones de configuración específicas de la plataforma para las versiones de la plataforma Node.js de la AMI de Amazon Linux. Puede elegir qué servidor proxy se ejecuta delante de su aplicación, la versión específica de Node.js que se debe ejecutar y el comando utilizado para ejecutar la aplicación.

Para el servidor proxy, puede utilizar un servidor proxy NGINX o Apache. Puede establecer el valor none en la opción ProxyServer. Con esta configuración, Elastic Beanstalk ejecuta la aplicación de forma independiente, sin que se relacione a ningún servidor proxy. Si el entorno ejecuta una aplicación independiente, actualice el código para escuchar al puerto al que NGINX reenvía tráfico.

```
var port = process.env.PORT || 8080;

app.listen(port, function() {
  console.log('Server running at http://127.0.0.1:%s', port);
});
```

Versiones de lenguajes de Node.js: AMI de Amazon Linux (AL1)

En términos de la versión de lenguaje compatible, la plataforma Node.js de la AMI de Amazon Linux es diferente de otras plataformas administradas por Elastic Beanstalk. Esto se debe a que cada versión de la plataforma Node.js admite solo unas pocas versiones del lenguaje de Node.js. Para obtener una lista de las versiones de Node.js compatibles, consulte [Node.js](#) en la guía Plataformas de AWS Elastic Beanstalk.

Puede utilizar una opción de configuración específica de la plataforma para establecer la versión del lenguaje. Para obtener instrucciones, consulte [the section called “Configuración de su entorno de Node.js”](#). Como alternativa, utilice la consola de Elastic Beanstalk para actualizar la versión Node.js que utiliza el entorno como parte de la actualización de la versión de la plataforma.

Note

Cuando la compatibilidad de la versión de Node.js que utiliza se elimine de la plataforma, deberá cambiar o eliminar la configuración de la versión antes de realizar una [actualización de la plataforma](#). Esto puede suceder cuando se identifica una vulnerabilidad de seguridad en una o varias versiones de Node.js

Cuando esto sucede, si se intenta actualizar a una nueva versión de la plataforma no compatible con la [NodeVersion](#) configurada, se producirá un error. Para evitar la necesidad de crear un nuevo entorno, cambie la opción de configuración NodeVersion por una versión de Node.js compatible con la antigua y la nueva versión de la plataforma, o [elimine la opción de configuración](#) y, a continuación, lleve a cabo la actualización de la plataforma.

Para configurar la versión Node.js del entorno en la consola de Elastic Beanstalk

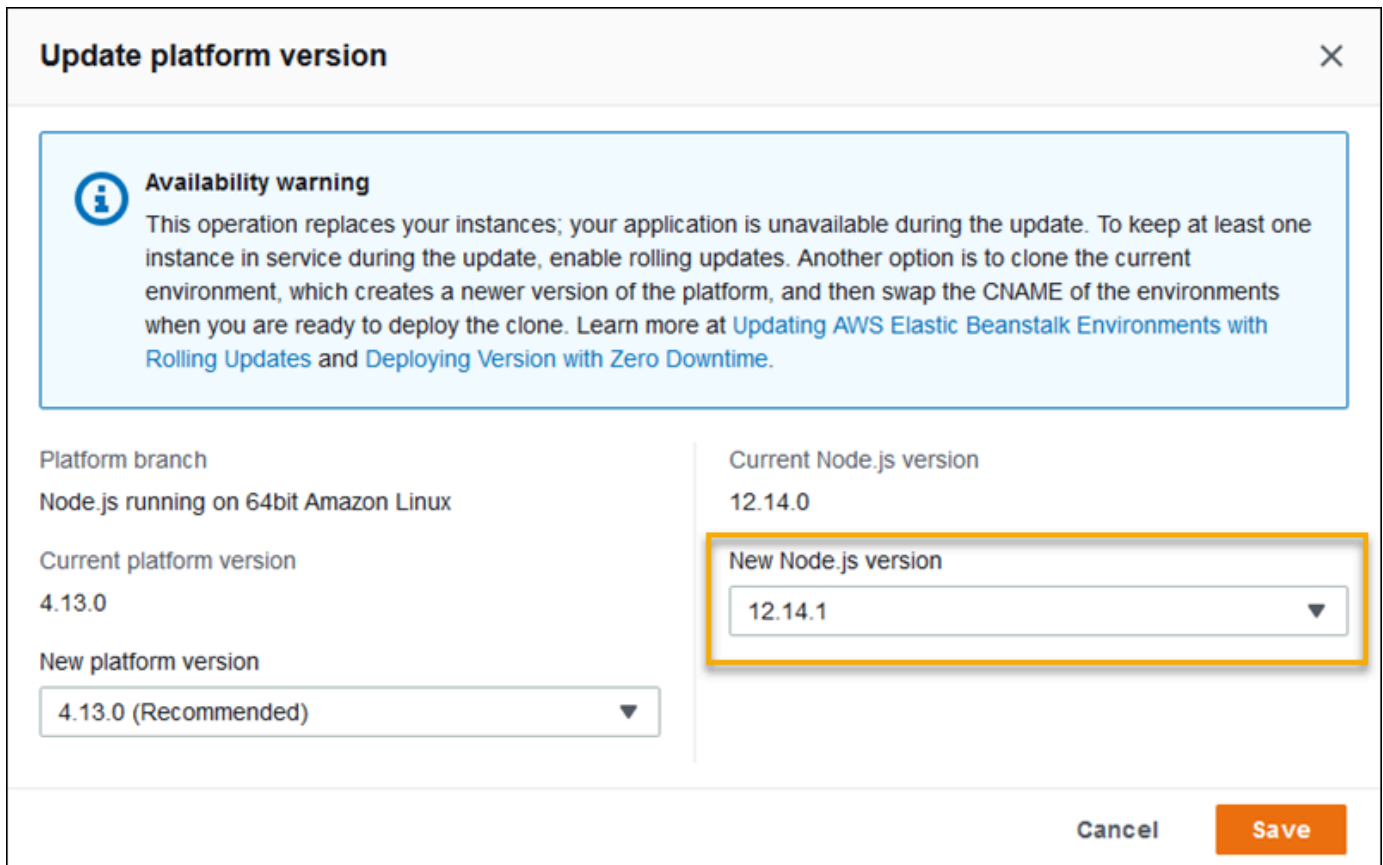
1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En la página de información general del entorno, en Platform (Plataforma), elija Change (Cambiar).

4. En el cuadro de diálogo Update platform version (Actualización de versiones de la plataforma), seleccione una versión de Node.js.



Update platform version [X]

Availability warning
This operation replaces your instances; your application is unavailable during the update. To keep at least one instance in service during the update, enable rolling updates. Another option is to clone the current environment, which creates a newer version of the platform, and then swap the CNAME of the environments when you are ready to deploy the clone. Learn more at [Updating AWS Elastic Beanstalk Environments with Rolling Updates and Deploying Version with Zero Downtime](#).

Platform branch
Node.js running on 64bit Amazon Linux

Current platform version
4.13.0

New platform version
4.13.0 (Recommended) ▼

Current Node.js version
12.14.0

New Node.js version
12.14.1 ▼

Cancel Save

5. Seleccione Save.

Espacios de nombres de configuración de Node.js: AMI de Amazon Linux (AL1)

La plataforma Node.js de la AMI de Amazon Linux define opciones adicionales en los espacios de nombres `aws:elasticbeanstalk:container:nodejs:staticfiles` y `aws:elasticbeanstalk:container:nodejs`.

El siguiente archivo de configuración indica a Elastic Beanstalk que utilicen `pm start` para ejecutar la aplicación. También establece el tipo de proxy en Apache y habilita la compresión. Por último, configura el proxy para que ofrezca archivos estáticos desde dos directorios fuente. Una fuente son los archivos HTML en la ruta `html` de la raíz del sitio web del directorio fuente `statichtml`. El otro origen son los archivos de imagen en la ruta `images` de la raíz del sitio web desde el directorio de origen `staticimages`.

Example .ebextensions/node-settings.config

```
option_settings:
  aws:elasticbeanstalk:container:nodejs:
    NodeCommand: "npm start"
    ProxyServer: apache
    GzipCompression: true
  aws:elasticbeanstalk:container:nodejs:staticfiles:
    /html: statichtml
    /images: staticimages
```

Elastic Beanstalk cuenta con numerosas opciones de configuración para personalizar el entorno. Además de los archivos de configuración, también puede definir opciones en la consola, configuraciones guardadas, la CLI de EB o la AWS CLI. Para obtener más información, consulte [Opciones de configuración](#).

Configuración del proceso de la aplicación con un Procfile

Puede incluir un archivo llamado `Procfile` en la raíz del paquete de código fuente para especificar el comando que inicia la aplicación.

Example Procfile

```
web: node index.js
```

Para obtener más información sobre el uso de `Procfile`, expanda la sección `Buildfile` y `Procfile` en [the section called “Ampliación de plataformas Linux”](#).

Note

Esta característica reemplaza la opción heredada `NodeCommand` en el espacio de nombres `aws:elasticbeanstalk:container:nodejs`.

Configurar las dependencias de su aplicación

La aplicación puede tener dependencias en algunos módulos Node.js, como los que se especifican en las declaraciones `require()`. Los módulos se almacenan en un directorio `node_modules`. Cuando se ejecuta la aplicación, Node.js carga los módulos de este directorio. Para obtener más

información, consulte [Loading from node_modules folders](#) (Carga desde carpetas `node_modules`) en la documentación de Node.js.

Puede especificar las dependencias de módulos mediante un archivo `package.json`. Si Elastic Beanstalk detecta este archivo y no hay un directorio `node_modules`, ejecuta `npm install` como el usuario `webapp`. El comando `npm install` instala las dependencias en el directorio `node_modules`, que Elastic Beanstalk crea previamente. El comando `npm install` accede a los paquetes que figuran en el archivo `package.json` desde el registro npm público o de otras ubicaciones. Para obtener más información, consulte el sitio web [npm Docs](#).

Si Elastic Beanstalk detecta el directorio `node_modules`, no ejecuta `npm install`, incluso si existe un archivo `package.json`. Elastic Beanstalk asume que los paquetes de dependencias están disponibles en el directorio `node_modules` para que Node.js pueda acceder y cargar.

En las siguientes secciones, se ofrece más información sobre cómo establecer las dependencias de los módulos de Node.js de la aplicación.

Note

Si sufre algún problema de implementación cuando Elastic Beanstalk está ejecutando `npm install`, considere un enfoque alternativo. Incluya el directorio `node_modules` con los módulos de dependencias en el paquete de código fuente de la aplicación. De esta manera se pueden evitar problemas relacionados con la instalación de dependencias del registro npm público mientras se investiga el problema. Como los módulos de dependencia provienen de un directorio local, hacer esto también podría ayudar a reducir el tiempo de implementación. Para obtener más información, consultar [Inclusión de dependencias Node.js en un directorio `node_modules`](#)

Especificación de dependencias de Node.js con un archivo `package.json`

Incluya un archivo `package.json` en la raíz del proyecto de origen para especificar paquetes de dependencias y proporcionar un comando de inicio. Cuando hay un archivo `package.json` y no hay un directorio `node_modules` en la raíz de la fuente del proyecto, Elastic Beanstalk ejecuta `npm install` como el usuario `webapp` para instalar las dependencias del registro npm público. Elastic Beanstalk también utiliza el comando `start` para iniciar la aplicación. Para obtener más información sobre el archivo `package.json`, consulte [Especificación de dependencias en un archivo `package.json`](#) en el sitio web npm Docs.

Utilice la palabra clave `scripts` para proporcionar un comando de inicio. En la actualidad, la palabra clave `scripts` se utiliza en lugar de la opción heredada `NodeCommand` en el espacio de nombres `aws:elasticbeanstalk:container:nodejs`.

Example `package.json`: Express

```
{
  "name": "my-app",
  "version": "0.0.1",
  "private": true,
  "dependencies": {
    "ejs": "latest",
    "aws-sdk": "latest",
    "express": "latest",
    "body-parser": "latest"
  },
  "scripts": {
    "start": "node app.js"
  }
}
```

Modo de producción y dependencias de desarrollo

Para especificar sus dependencias en el archivo `package.json`, utilice los atributos `dependencies` y `devDependencies`. El atributo `dependencies` designa los paquetes que necesita la aplicación en producción. El atributo `devDependencies` designa los paquetes que solo se necesitan para el desarrollo y las pruebas locales.

Elastic Beanstalk ejecuta `npm install` como usuario `webapp` con los siguientes comandos. Las opciones de comando varían según la versión de `npm` incluida en la ramificación de la plataforma en la que se ejecute la aplicación.

- `npm v6`: de forma predeterminada, Elastic Beanstalk instala dependencias en modo de producción. Usa el comando `npm install --production`.
- `npm v7` o superior: Elastic Beanstalk omite el atributo `devDependencies`. Usa el comando `npm install --omit=dev`.

Los dos comandos enumerados anteriormente no instalan los paquetes que son atributos `devDependencies`.

Si necesita instalar los paquetes `devDependencies`, defina la propiedad del entorno `NPM_USE_PRODUCTION` en `false`. Con esta configuración, no utilizaremos las opciones anteriores cuando ejecutemos la instalación de `npm`. Esto hará que se instalen los paquetes de atributos `devDependencies`.

SSH y HTTPS

A partir del lanzamiento de la plataforma Amazon Linux 2 del 7 de marzo de 2023, también podrá usar los protocolos SSH y HTTPS para recuperar paquetes de un repositorio de Git. La ramificación de la plataforma 16 de Node.js admite los protocolos SSH y HTTPS. 14 de Node.js solo admite el protocolo HTTPS.

Example `package.json`: 16 de Node.js admite HTTPS y SSH

```
...
"dependencies": {
  "aws-sdk": "https://github.com/aws/aws-sdk-js.git",
  "aws-chime": "git+ssh://git@github.com:aws/amazon-chime-sdk-js.git"
}
```

Versiones y rangos de las versiones

Important

La característica para especificar los rangos de las versiones no está disponible para las ramificaciones de la plataforma de Node.js que se ejecutan en AL2023. Solo admitimos una versión de Node.js en una ramificación específica de Node.js en AL2023. Si el archivo `package.json` especifica un rango de versiones, lo ignoraremos y utilizaremos de forma predeterminada la versión de la ramificación de la plataforma de Node.js.

Utilice la palabra clave `engines` del archivo `package.json` para especificar la versión de Node.js que desea que utilice la aplicación. También puede especificar un rango de versiones mediante la notación `npm`. Para obtener más información sobre la sintaxis de los rangos de versiones, consulte [Semantic Versioning using npm](#) (Control de versiones semánticas mediante `npm`) en el sitio web de Node.js. La palabra clave `engines` del archivo `package.json` de Node.js reemplaza la opción heredada `NodeVersion` en el espacio de nombres `aws:elasticbeanstalk:container:nodejs`.

Example `package.json`: versión de Node.js única

```
{
  ...
  "engines": { "node" : "14.16.0" }
}
```

Example `package.json`: rango de versiones de Node.js

```
{
  ...
  "engines": { "node" : ">=10 <11" }
}
```

Cuando se indica un rango de versiones, Elastic Beanstalk instala la última versión de Node.js que la plataforma tiene disponible dentro del rango. En este ejemplo, el rango indica que la versión debe ser mayor o igual que la versión 10, pero inferior que la versión 11. Como resultado, Elastic Beanstalk instala la última versión 10.x.y de Node.js, que está disponible en la [plataforma compatible](#).

Tenga en cuenta que solo puede especificar una versión de Node.js que corresponda con la ramificación de la plataforma. Por ejemplo, si utiliza la ramificación de la plataforma 16 de Node.js, puede especificar solo una versión 16.x.y de Node.js. Puede utilizar las opciones del rango de versiones admitidas por npm para aumentar la flexibilidad. Para obtener versiones válidas de Node.js para cada una de las ramificaciones de la plataforma, consulte [Node.js](#) en la guía Plataformas de AWS Elastic Beanstalk.

Note

Cuando la compatibilidad de la versión de Node.js que utiliza se elimine de la plataforma, deberá cambiar o eliminar la configuración de la versión de Node.js antes de realizar una [actualización de la plataforma](#). Esto puede suceder cuando se identifica una vulnerabilidad de seguridad en una o varias versiones de Node.js

Cuando esto sucede, si se intenta actualizar a una nueva versión de la plataforma no compatible con la versión de Node.js configurada, se producirá un error. Para evitar la necesidad de crear un nuevo entorno, cambie la configuración de la versión `Node.jspackage.json` a una versión Node.js compatible con las versiones anterior y nueva de la plataforma. Tiene la opción de especificar un rango de versiones de Node.js que incluya

una versión compatible, como se describió anteriormente en este tema. También tiene la opción de eliminar la configuración e implementar el nuevo paquete de origen.

Inclusión de dependencias Node.js en un directorio `node_modules`

Para implementar paquetes de dependencias en instancias del entorno junto con el código de la aplicación, inclúyalos en un directorio llamado `node_modules` en la raíz del origen del proyecto. Para obtener más información, consulte [Downloading and installing packages locally](#) (Descarga e instalación de paquetes a nivel local) en el sitio web npm Docs.

Cuando implementa un directorio `node_modules` en una versión de la plataforma de Node.js de Amazon Linux 2, Elastic Beanstalk asume que usted proporciona sus propios paquetes de dependencias y, por lo tanto, no instala dependencias especificadas en un archivo [package.json](#). Node.js busca dependencias en el directorio `node_modules`. Para obtener más información, consulte [Loading from node_modules Folders](#) (Carga desde carpetas `node_modules`) en la documentación de Node.js.

Note

Si sufre algún problema de implementación cuando Elastic Beanstalk ejecuta `npm install`, considere la posibilidad de utilizar el enfoque descrito en este tema como solución alternativa mientras investiga el problema.

Bloqueo de dependencias con `npm shrinkwrap`

La plataforma Node.js ejecuta `npm install` como el usuario `webapp` cada vez que se realiza una implementación. Las versiones de dependencias nuevas disponibles se instalan cuando usted implementa la aplicación, lo que puede ocasionar demoras durante el proceso de implementación.

Puede evitar que se actualicen las dependencias creando un archivo `npm-shrinkwrap.json` que bloquee las dependencias de la aplicación en la versión actual.

```
$ npm install
$ npm shrinkwrap
wrote npm-shrinkwrap.json
```

Incluya este archivo en su paquete de código fuente para asegurarse de que las dependencias solo se instalan una vez.

Configuración del servidor proxy

Elastic Beanstalk puede utilizar NGINX o Apache HTTPD como proxy inverso para asignar la aplicación al equilibrador de carga de Elastic Load Balancing en el puerto 80. El valor predeterminado es NGINX. Elastic Beanstalk proporciona una configuración de proxy predeterminada que puede ampliar o invalidar completamente su propia configuración.

De forma predeterminada, Elastic Beanstalk configura el proxy para reenviar las solicitudes a la aplicación en el puerto 5000. Puede invalidar el puerto predeterminado al establecer la `PORT` [propiedad del entorno](#) en el puerto en el que la aplicación principal esté a la escucha.

Note

El puerto en el que su aplicación escucha no afecta al puerto en el que el servidor NGINX escucha para recibir solicitudes del equilibrador de carga.

Configuración del servidor proxy en la versión de su plataforma

Todas las plataformas AL2023/AL2 son compatibles con una característica de configuración de proxy uniforme. Para obtener más información sobre la configuración del servidor proxy en las versiones de su plataforma que ejecutan AL2023/AL2, expanda la sección Configuración del proxy inverso de [the section called “Ampliación de plataformas Linux”](#).

Configuración del proxy en la AMI de Amazon Linux (anterior a Amazon Linux 2)

Si el entorno Node.js de Elastic Beanstalk utiliza una versión de la plataforma de la AMI de Amazon Linux (anterior a Amazon Linux 2), lea la información de esta sección.

Notas

- La información de este tema solo se aplica a ramificaciones de plataforma basadas en la AMI de Amazon Linux (AL1). Las ramificaciones de la plataforma AL2023/AL2 son incompatibles con las versiones anteriores de la plataforma de la AMI de Amazon Linux (AL1) y requieren ajustes de configuración diferentes.
- El [18 de julio de 2022](#), Elastic Beanstalk estableció el estado de todas las ramificaciones de la plataforma basadas en la AMI de Amazon Linux (AL1) como retirado. Para obtener

más información sobre la migración a una ramificación de la plataforma Amazon Linux 2023 actual y totalmente compatible, consulte [Migración de su aplicación de Linux de Elastic Beanstalk a Amazon Linux 2023](#) o [Amazon Linux 2](#).

Ampliación y anulación de la configuración predeterminada del proxy: AMI de Amazon Linux (AL1)

La plataforma Node.js utiliza un proxy inverso para transmitir las solicitudes del puerto 80 de la instancia a la aplicación que está a la escucha en el puerto 8081. Elastic Beanstalk proporciona una configuración de proxy predeterminada que puede ampliar o invalidar completamente su propia configuración.

Para ampliar la configuración predeterminada, añade archivos `.conf` a `/etc/nginx/conf.d` con un archivo de configuración. Para ver un ejemplo específico, consulte [Terminación de HTTPS en instancias EC2 que ejecutan Node.js](#).

La plataforma Node.js establece la variable de entorno `PORT` en el puerto al que el servidor proxy pasa el tráfico. Lea esta variable en el código para configurar el puerto de la aplicación.

```
var port = process.env.PORT || 3000;

var server = app.listen(port, function () {
  console.log('Server running at http://127.0.0.1:' + port + '/');
});
```

La configuración predeterminada de NGINX reenvía el tráfico a un servidor ascendente denominado `nodejs` en `127.0.0.1:8081`. Puede eliminar la configuración predeterminada y proporcionar la propia en un [archivo de configuración](#).

Example `.ebextensions/proxy.config`

El siguiente ejemplo elimina la configuración predeterminada y agrega una configuración personalizada que reenvía el tráfico al puerto 5000, en lugar del puerto 8081.

```
files:
  /etc/nginx/conf.d/proxy.conf:
    mode: "000644"
    owner: root
    group: root
    content: |
      upstream nodejs {
```

```
server 127.0.0.1:5000;
keepalive 256;
}

server {
    listen 8080;

    if ($time_iso8601 ~ "^(\\d{4})-(\\d{2})-(\\d{2})T(\\d{2})") {
        set $year $1;
        set $month $2;
        set $day $3;
        set $hour $4;
    }
    access_log /var/log/nginx/healthd/application.log.$year-$month-$day-$hour
healthd;
    access_log /var/log/nginx/access.log main;

    location / {
        proxy_pass http://nodejs;
        proxy_set_header    Connection "";
        proxy_http_version 1.1;
        proxy_set_header     Host          $host;
        proxy_set_header     X-Real-IP     $remote_addr;
        proxy_set_header     X-Forwarded-For $proxy_add_x_forwarded_for;
    }

    gzip on;
    gzip_comp_level 4;
    gzip_types text/html text/plain text/css application/json application/x-
javascript text/xml application/xml application/xml+rss text/javascript;

    location /static {
        alias /var/app/current/static;
    }

}

/opt/elasticbeanstalk/hooks/configdeploy/post/99_kill_default_nginx.sh:
mode: "000755"
owner: root
group: root
content: |
    #!/bin/bash -xe
    rm -f /etc/nginx/conf.d/00_elastic_beanstalk_proxy.conf
```

```
service nginx stop
service nginx start

container_commands:
  removeconfig:
    command: "rm -f /tmp/deployment/config/
#etc#nginx#conf.d#00_elastic_beanstalk_proxy.conf /etc/nginx/
conf.d/00_elastic_beanstalk_proxy.conf"
```

La configuración de ejemplo (`/etc/nginx/conf.d/proxy.conf`) utiliza la configuración predeterminada de `/etc/nginx/conf.d/00_elastic_beanstalk_proxy.conf` como punto de partida para incluir el bloque de servidores predeterminado con configuración de compresión y registros y un mapeo de archivos estáticos.

El comando `removeconfig` elimina la configuración predeterminada del contenedor para asegurarse de que el servidor proxy utiliza la configuración personalizada. Elastic Beanstalk vuelve a crear la configuración predeterminada cuando se implementa cada configuración. Para tener en cuenta esto, en el siguiente ejemplo, se agrega un enlace de implementación posterior a la configuración (`/opt/elasticbeanstalk/hooks/configdeploy/post/99_kill_default_nginx.sh`). Esto elimina la configuración predeterminada y reinicia el servidor proxy.

Note

La configuración predeterminada puede cambiar en las próximas versiones de la plataforma Node.js. Utilice la última versión de la configuración como punto de partida para las personalizaciones con el fin de garantizar la compatibilidad.

Si anula la configuración predeterminada, debe definir las asignaciones de archivos estáticos y la compresión GZIP. Esto se debe a que la plataforma no puede aplicar la [configuración estándar](#).

Más ejemplos de aplicaciones y tutoriales para Node.js

Para empezar a utilizar las aplicaciones de Node.js AWS Elastic Beanstalk, lo único que necesita es un [paquete de código fuente](#) de la aplicación para cargarlo como primera versión de la aplicación e implementarlo en un entorno. En [QuickStart para Node.js](#) este tema se explica cómo iniciar la aplicación Node.js de ejemplo con la CLI de EB. En esta sección se proporcionan aplicaciones y tutoriales adicionales.

Lanzar un entorno con una aplicación de Node.js de ejemplo

Elastic Beanstalk proporciona aplicaciones de muestra de una sola página para cada plataforma, así como ejemplos más complejos que muestran el uso de recursos AWS adicionales, como Amazon RDS, y características y API específicas de cada idioma o plataforma.

Note

Siga los pasos del archivo README .md del paquete de origen para implementarlo.

Muestras

Tipo de entorno	Paquete de código fuente	Descripción
Servidor web	nodejs.zip	<p>Aplicación de una sola página.</p> <p>Para iniciar la aplicación de ejemplo con la CLI de EB, consulte QuickStart para Node.js.</p> <p>También puede usar la consola de Elastic Beanstalk para iniciar la aplicación de muestra. Para ver los pasos detallados, consulte Crear una aplicación de ejemplo en el capítulo Introducción de esta guía.</p>
Servidor web con Amazon RDS	nodejs-ex-ample-express-rds.zip	<p>Aplicación de registros de hiking que utiliza el marco Express y un Amazon Relational Database Service (RDS).</p> <p>Tutorial</p>
Servidor web con Amazon ElastiCache	nodejs-ex-ample-express-elas	<p>Aplicación web Express que utiliza Amazon ElastiCache para la agrupación en clústeres. La agrupación en clústeres mejora la disponibilidad, el desempeño y la seguridad de su aplicación web.</p>

Tipo de entorno	Paquete de código fuente	Descripción
	ticache.zip	Tutorial
Servidor web con DynamoDB, Amazon SNS y Amazon SQS	nodejs-ex-ample-dyn-amo.zip	Sitio web Express que recopila información de contacto de los usuarios para la campaña de marketing de una empresa. Utiliza el AWS SDK de Node.js para escribir entradas JavaScript en una tabla de DynamoDB y los archivos de configuración de Elastic Beanstalk para crear recursos en DynamoDB, Amazon SNS y Amazon SQS. Tutorial

Siguientes pasos

Una vez que disponga de un entorno que ejecute una aplicación, podrá implementar una nueva versión de la aplicación o una aplicación totalmente diferente en cualquier momento. La implementación de una nueva versión de la aplicación es una tarea muy rápida, ya que no se requiere aprovisionar ni reiniciar instancias EC2. Para obtener más información sobre la implementación de aplicaciones, consulte [Implementar una nueva versión de la aplicación](#).

Una vez que haya implementado una o dos aplicaciones de muestra y esté listo para empezar a desarrollar y ejecutar las aplicaciones de Node.js de forma local, asegúrese de [Configuración del entorno de desarrollo de Node.js](#) configurar un entorno de desarrollo Node.js con todas las herramientas que necesitará.

Implementación de una aplicación Express en Elastic Beanstalk

En esta sección, se describe cómo se implementa una aplicación de ejemplo en Elastic Beanstalk con la interfaz de línea de comando de Elastic Beanstalk (CLI de EB), y cómo se la actualiza después para utilizar el marco de [Express](#).

Requisitos previos

Este tutorial tiene siguientes los requisitos previos:

- Los tiempos de ejecución de Node.js
- El software de gestión de paquetes de Node.js predeterminado, npm
- El generador de línea de comandos Express
- La interfaz de línea de comando de Elastic Beanstalk (CLI de EB)

Para obtener más detalles acerca de la instalación de los primeros tres componentes y la configuración de su entorno de desarrollo local, consulte [Configuración del entorno de desarrollo de Node.js](#). Para este tutorial, no necesitas instalar el AWS SDK para Node.js, que también se menciona en el tema al que se hace referencia.

Para obtener detalles sobre la instalación y configuración de la CLI de EB, consulte [Instalación de la CLI de EB](#) y [Configuración de la CLI de EB](#).

Cree un entorno de Elastic Beanstalk

Su directorio de aplicaciones

Este tutorial usa un directorio llamado `nodejs-example-express-rds` para el paquete de origen de la aplicación. Cree el directorio `nodejs-example-express-rds` para este tutorial.

```
~$ mkdir nodejs-example-express-rds
```

Note

Cada tutorial de este capítulo usa su propio directorio para el paquete de origen de la aplicación. El nombre del directorio coincide con el nombre de la aplicación de ejemplo utilizada en el tutorial.

Cambie su directorio de trabajo actual a `nodejs-example-express-rds`.

```
~$ cd nodejs-example-express-rds
```

Ahora configuremos un entorno de Elastic Beanstalk que ejecute la plataforma Node.js y la aplicación de ejemplo. Se usará la interfaz de línea de comandos de Elastic Beanstalk (CLI de EB)

Para configurar un repositorio en la CLI de EB para la aplicación y crear un entorno de Elastic Beanstalk que ejecute la plataforma Node.js

1. Cree un repositorio con el comando [eb init](#).

```
~/nodejs-example-express-rds$ eb init --platform node.js --region <region>
```

Este comando crea un archivo de configuración en una carpeta llamada `.elasticbeanstalk` que especifica los ajustes para crear los entornos de la aplicación y crea una aplicación de Elastic Beanstalk con el nombre de la carpeta actual.

2. Cree un entorno que ejecute una aplicación de ejemplo con el comando [eb create](#).

```
~/nodejs-example-express-rds$ eb create --sample nodejs-example-express-rds
```

Este comando crea un entorno con balanceador de carga utilizando la configuración predeterminada de la plataforma de Node.js y los siguientes recursos:

- Instancia de EC2: una máquina virtual de Amazon Elastic Compute Cloud (Amazon EC2) configurada para ejecutar aplicaciones web en la plataforma que elija.

Cada plataforma ejecuta un conjunto específico de software, archivos de configuración y scripts compatibles con una determinada versión de lenguaje, marco de trabajo y contenedor web (o una combinación de estos). La mayoría de las plataformas utilizan Apache o nginx como un proxy inverso que se sitúa delante de la aplicación web, reenvía las solicitudes a esta, administra los recursos estáticos y genera registros de acceso y errores.

- Instance security group (Grupo de seguridad de la instancia): grupo de seguridad de Amazon EC2 configurado para permitir el tráfico entrante en el puerto 80. Este recurso permite que el tráfico HTTP procedente del balanceador de carga llegue a la instancia EC2 en la que se ejecuta la aplicación web. De forma predeterminada, el tráfico no está permitido en otros puertos.
- Balanceador de carga: balanceador de carga de Elastic Load Balancing configurado para distribuir solicitudes a las instancias que se ejecutan en la aplicación. Los balanceadores de carga también permiten que las instancias no estén expuestas directamente a Internet.
- Grupo de seguridad del balanceador de carga: grupo de seguridad de Amazon EC2 configurado para permitir el tráfico entrante en el puerto 80. Este recurso permite que el tráfico HTTP procedente de Internet llegue al balanceador de carga. De forma predeterminada, el tráfico no está permitido en otros puertos.

- Grupo de Auto Scaling: grupo de Auto Scaling configurado para reemplazar una instancia si termina o deja de estar disponible.
- Bucket de Amazon S3: ubicación de almacenamiento para el código fuente, los registros y otros artefactos que se crean al utilizar Elastic Beanstalk.
- CloudWatch Alarmas de Amazon: dos CloudWatch alarmas que monitorean la carga de las instancias de su entorno y que se activan si la carga es demasiado alta o demasiado baja. Cuando se activa una alarma, en respuesta, el grupo de Auto Scaling aumenta o reduce los recursos.
- AWS CloudFormation pila: Elastic AWS CloudFormation Beanstalk se utiliza para lanzar los recursos de su entorno y propagar los cambios de configuración. Los recursos se definen en una plantilla que puede verse en la [consola de AWS CloudFormation](#).
- Nombre de dominio: nombre de dominio que direcciona el tráfico a la aplicación web con el formato *subdominio.región.elasticbeanstalk.com*.

Note

Para aumentar la seguridad de las aplicaciones de Elastic Beanstalk, el dominio `elasticbeanstalk.com` está registrado en la [lista de sufijos públicos \(PSL\)](#). Para mayor seguridad, se recomienda que utilice cookies con un prefijo `__Host-` en caso de que necesite configurar cookies confidenciales en el nombre de dominio predeterminado de sus aplicaciones de Elastic Beanstalk. Esta práctica le ayudará a defender su dominio contra intentos de falsificación de solicitudes entre sitios (CSRF). Para obtener más información, consulte la página de [configuración de cookies](#) en la red de desarrolladores de Mozilla.

3. Cuando se complete la creación del entorno, utilice el comando [eb open](#) para abrir la URL del entorno en el navegador predeterminado.

```
~/nodejs-example-express-rds$ eb open
```

Ahora ha creado un entorno de Elastic Beanstalk para Node.js con una aplicación de ejemplo. Puede actualizarlo con su propia aplicación. Luego, actualizamos la aplicación de ejemplo para que use el marco de Express.

Actualizar la aplicación para que use Express

Una vez que haya creado el entorno con una aplicación de ejemplo, puede actualizarlo con su propia aplicación. En este procedimiento, ejecutamos primero los comandos `express` y `npm install` para configurar el marco de Express en el directorio de la aplicación. Luego use la CLI de EB para actualizar su entorno Elastic Beanstalk con la aplicación actualizada.

Si desea actualizar la aplicación para que use Express

1. Ejecute el comando `express`. Esto genera `package.json`, `app.js` y unos directorios.

```
~/nodejs-example-express-rds$ express
```

Cuando se le pregunte, escriba **y** si desea continuar.

Note

Si el comando `express` no funciona, es posible que no haya instalado el generador de línea de comandos Express tal y como se describe en la sección Requisitos previos anterior. O bien, puede que sea necesario configurar la ruta del directorio de su máquina local para ejecutar el comando `express`. Consulte la sección Requisitos previos para conocer los pasos detallados sobre la configuración del entorno de desarrollo, de modo que pueda continuar con este tutorial.

2. Configure las dependencias locales.

```
~/nodejs-example-express-rds$ npm install
```

3. (Opcional) Compruebe que el servidor de aplicaciones web se inicie.

```
~/nodejs-example-express-rds$ npm start
```

Debería ver una salida similar a esta:

```
> nodejs@0.0.0 start /home/local/user/node-express  
> node ./bin/www
```

El servidor se ejecuta en el puerto 3000 de forma predeterminada. Para probarlo, ejecute `curl http://localhost:3000` en otro terminal o abra un navegador en el equipo local e ingrese el `http://localhost:3000` de la dirección URL.

Presione `Ctrl+C` para detener el servidor.

4. Implemente los cambios en su entorno Elastic Beanstalk con el comando [eb deploy](#).

```
~/nodejs-example-express-rds$ eb deploy
```

5. Una vez que el entorno esté listo, actualice la dirección URL para verificar que ha funcionado. Debería aparecer una página web que diga Welcome to Express.

A continuación, vamos a actualizar la aplicación de Express para atender archivos estáticos y agregar una nueva página.

Para configurar archivos estáticos y agregar una nueva página a la aplicación de Express

1. Agregue un segundo archivo de configuración en la carpeta [.ebextensions](#) con el contenido siguiente:

nodejs-example-express-rds/.ebextensions/staticfiles.config

```
option_settings:
  aws:elasticbeanstalk:environment:proxy:staticfiles:
    /stylesheets: public/stylesheets
```

Esta opción configura el servidor proxy que proporcione los archivos en la carpeta `public` de la ruta `/public` de la aplicación. Si los archivos se sirven de forma estática desde el servidor proxy, se reduce la carga en la aplicación. Para obtener más información, consulte [Archivos estáticos](#) mencionada previamente en este capítulo.

2. (Opcional) Para confirmar que las asignaciones estáticas están configuradas correctamente, comente la configuración de asignación estática en `nodejs-example-express-rds/app.js`. Esto elimina la asignación de la aplicación de nodos.

```
// app.use(express.static(path.join(__dirname, 'public')));
```

Las asignaciones de archivos estáticas del archivo `staticfiles.config` del paso anterior deberían seguir cargando la hoja de estilos correctamente, incluso después de comentar esta línea. Para comprobar que las asignaciones de archivos estáticas se cargan mediante la configuración de archivos estáticos del proxy, en lugar de mediante la aplicación expés, elimine los valores siguientes `option_settings`:. Una vez que se haya eliminado de la configuración del archivo estático y de la aplicación de nodo, la hoja de estilos no se cargará.

Recuerde restablecer el contenido de `nodejs-example-express-rds/app.js` y de `staticfiles.config` cuando haya terminado las pruebas.

3. Add `nodejs-example-express-rds/routes/hike.js`. Escriba lo siguiente:

```
exports.index = function(req, res) {
  res.render('hike', {title: 'My Hiking Log'});
};

exports.add_hike = function(req, res) {
};
```

4. Actualice `nodejs-example-express-rds/app.js` para incluir tres nuevas líneas.

En primer lugar, inserte la siguiente línea para agregar un objeto `require` para esta ruta:

```
var hike = require('./routes/hike');
```

El archivo debe ser similar al siguiente fragmento:

```
var express = require('express');
var path = require('path');
var hike = require('./routes/hike');
```

A continuación, agregue las dos líneas siguientes a `nodejs-example-express-rds/app.js` detrás de `var app = express();`.

```
app.get('/hikes', hike.index);
app.post('/add_hike', hike.add_hike);
```

El archivo debe ser similar al siguiente fragmento:


```
var app = express();
app.get('/hikes', hike.index);
app.post('/add_hike', hike.add_hike);
```

5. Copie `nodejs-example-express-rds/views/index.jade` en `nodejs-example-express-rds/views/hike.jade`.

```
~/nodejs-example-express-rds$ cp views/index.jade views/hike.jade
```

6. Implemente los cambios con el comando [eb deploy](#).

```
~/nodejs-example-express-rds$ eb deploy
```

7. Su entorno se actualizará al cabo de unos minutos. Cuando el entorno esté listo, asegúrese de que funciona actualizando el navegador y agregando **hikes** al final de la URL (por ejemplo, `http://node-express-env-syypntcz2q.elasticbeanstalk.com/hikes`).

Debería ver una página web con el título My Hiking Log.

Ya ha creado una aplicación web que utiliza el marco Express. En la siguiente sección, modificaremos la aplicación para utilizar un Amazon Relational Database Service (RDS) para almacenar un registro de hiking.

Actualice la aplicación para que use Amazon RDS.

En el siguiente paso, actualizamos la aplicación para usar Amazon RDS para MySQL.

Para actualizar la aplicación para que use RDS para MySQL

1. Para crear una base de datos RDS para MySQL acoplada a su entorno de Elastic Beanstalk, siga las instrucciones del tema [Agregar una base de datos](#) que se incluye más adelante en este capítulo. Añadir una instancia de base de datos demora aproximadamente 10 minutos.
2. Actualice la sección de dependencias en `package.json` con el siguiente contenido:

```
"dependencies": {
  "async": "^3.2.4",
  "express": "4.18.2",
  "jade": "1.11.0",
  "mysql": "2.18.1",
```

```
"node-uuid": "^1.4.8",
"body-parser": "^1.20.1",
"method-override": "^3.0.0",
"morgan": "^1.10.0",
"errorhandler": "^1.5.1"
}
```

3. Ejecute npm install.

```
~/nodejs-example-express-rds$ npm install
```

4. Actualice app.js para conectarse a la base de datos, crear una tabla e insertar un único registro de hiking predeterminado. Cada vez que se implemente esta aplicación, eliminará la tabla de hiking anterior y la volverá a crear.

```
/**
 * Module dependencies.
 */

const express = require('express')
, routes = require('./routes')
, hike = require('./routes/hike')
, http = require('http')
, path = require('path')
, mysql = require('mysql')
, async = require('async')
, bodyParser = require('body-parser')
, methodOverride = require('method-override')
, morgan = require('morgan')
, errorHandler = require('errorhandler');

const { connect } = require('http2');

const app = express()

app.set('views', __dirname + '/views')
app.set('view engine', 'jade')
app.use(methodOverride())
app.use(bodyParser.json())
app.use(bodyParser.urlencoded({ extended: true }))
app.use(express.static(path.join(__dirname, 'public')))
```

```
app.set('connection', mysql.createConnection({
  host: process.env.RDS_HOSTNAME,
  user: process.env.RDS_USERNAME,
  password: process.env.RDS_PASSWORD,
  port: process.env.RDS_PORT}));

function init() {
  app.get('/', routes.index);
  app.get('/hikes', hike.index);
  app.post('/add_hike', hike.add_hike);
}

const client = app.get('connection');
async.series([
  function connect(callback) {
    client.connect(callback);
    console.log('Connected!');
  },
  function clear(callback) {
    client.query('DROP DATABASE IF EXISTS mynode_db', callback);
  },
  function create_db(callback) {
    client.query('CREATE DATABASE mynode_db', callback);
  },
  function use_db(callback) {
    client.query('USE mynode_db', callback);
  },
  function create_table(callback) {
    client.query('CREATE TABLE HIKES (' +
      'ID VARCHAR(40), ' +
      'HIKE_DATE DATE, ' +
      'NAME VARCHAR(40), ' +
      'DISTANCE VARCHAR(40), ' +
      'LOCATION VARCHAR(40), ' +
      'WEATHER VARCHAR(40), ' +
      'PRIMARY KEY(ID))', callback);
  },
  function insert_default(callback) {
    const hike = {HIKE_DATE: new Date(), NAME: 'Hazard Stevens',
      LOCATION: 'Mt Rainier', DISTANCE: '4,027m vertical', WEATHER: 'Bad', ID:
      '12345'};
    client.query('INSERT INTO HIKES set ?', hike, callback);
  }
], function (err, results) {
```

```
    if (err) {
      console.log('Exception initializing database.');
```

```
      throw err;
    } else {
      console.log('Database initialization complete.');
```

```
      init();
    }
  });

module.exports = app
```

5. Agregue el siguiente contenido a `routes/hike.js`. Esto permitirá a las rutas insertar nuevos registros de hiking en la base de datos de HIKES.

```
const uuid = require('node-uuid');
exports.index = function(req, res) {
  res.app.get('connection').query( 'SELECT * FROM HIKES', function(err,
rows) {
    if (err) {
      res.send(err);
    } else {
      console.log(JSON.stringify(rows));
      res.render('hike', {title: 'My Hiking Log', hikes: rows});
    }
  });
};

exports.add_hike = function(req, res){
  const input = req.body.hike;
  const hike = { HIKE_DATE: new Date(), ID: uuid.v4(), NAME: input.NAME,
LOCATION: input.LOCATION, DISTANCE: input.DISTANCE, WEATHER: input.WEATHER};
  console.log('Request to log hike:' + JSON.stringify(hike));
  req.app.get('connection').query('INSERT INTO HIKES set ?', hike, function(err) {
    if (err) {
      res.send(err);
    } else {
      res.redirect('/hikes');
    }
  });
};
```

6. Reemplace todo el contenido de `routes/index.js` por lo siguiente:

```
/*
 * GET home page.
```

```
*/  
  
exports.index = function(req, res){  
  res.render('index', { title: 'Express' });  
};
```

7. Añada la siguiente plantilla de jade a `views/hike.jade` para proporcionar la interfaz de usuario para añadir registros de hiking.

```
extends layout  
  
block content  
  h1= title  
  p Welcome to #{title}  
  
  form(action="/add_hike", method="post")  
    table(border="1")  
      tr  
        td Your Name  
        td  
          input(name="hike[NAME]", type="text")  
      tr  
        td Location  
        td  
          input(name="hike[LOCATION]", type="text")  
      tr  
        td Distance  
        td  
          input(name="hike[DISTANCE]", type="text")  
      tr  
        td Weather  
        td  
          input(name="hike[WEATHER]", type="radio", value="Good")  
          | Good  
          input(name="hike[WEATHER]", type="radio", value="Bad")  
          | Bad  
          input(name="hike[WEATHER]", type="radio", value="Seattle", checked)  
          | Seattle  
      tr  
        td colspan="2"  
          input(type="submit", value="Record Hike")  
  
div
```

```
h3 Hikes
table(border="1")
  tr
    td Date
    td Name
    td Location
    td Distance
    td Weather
  each hike in hikes
    tr
      td #{hike.HIKE_DATE.toDateString()}
      td #{hike.NAME}
      td #{hike.LOCATION}
      td #{hike.DISTANCE}
      td #{hike.WEATHER}
```

8. Implemente los cambios con el comando [eb deploy](#).

```
~/nodejs-example-express-rds$ eb deploy
```

Limpieza

Si ha terminado de trabajar con Elastic Beanstalk, puede terminar su entorno.

Utilice el comando `eb terminate` para terminar el entorno y todos los recursos que contiene.

```
~/nodejs-example-express-rds$ eb terminate
The environment "nodejs-example-express-rds-env" and all associated instances will be
terminated.
To confirm, type the environment name: nodejs-example-express-rds-env
INFO: terminateEnvironment is starting.
...
```

Implementación de una aplicación Express con clústeres en Elastic Beanstalk

[En este tutorial, se explica cómo implementar una aplicación de ejemplo en Elastic Beanstalk mediante la interfaz de línea de comandos de Elastic Beanstalk \(EB CLI\) y, a continuación, se actualiza la aplicación para que utilice el marco Express, Amazon y los clústeres. ElastiCache](#) La agrupación en clústeres mejora la disponibilidad, el desempeño y la seguridad de su aplicación web.

Para obtener más información sobre Amazon ElastiCache, consulta [¿Qué es Amazon ElastiCache para Memcached?](#) en la Guía del usuario ElastiCache de Amazon for Memcached.

Note

En este ejemplo se crean AWS recursos, por los que es posible que se le cobre. Para obtener más información sobre AWS los precios, consulte <https://aws.amazon.com/pricing/>. Algunos servicios forman parte de la capa de uso AWS gratuito. Si es un cliente nuevo, puede probar estos servicios de forma gratuita. Para obtener más información, consulte <https://aws.amazon.com/free/>.

Requisitos previos

Este tutorial tiene siguientes los requisitos previos:

- Los tiempos de ejecución de Node.js
- El software de gestión de paquetes de Node.js predeterminado, npm
- El generador de línea de comandos Express
- La interfaz de línea de comando de Elastic Beanstalk (CLI de EB)

Para obtener más detalles acerca de la instalación de los primeros tres componentes y la configuración de su entorno de desarrollo local, consulte [Configuración del entorno de desarrollo de Node.js](#). Para este tutorial, no es necesario instalar el AWS SDK para Node.js, que también se menciona en el tema al que se hace referencia.

Para obtener detalles sobre la instalación y configuración de la CLI de EB, consulte [Instalación de la CLI de EB](#) y [Configuración de la CLI de EB](#).

Cree un entorno de Elastic Beanstalk

Su directorio de aplicaciones

Este tutorial usa un directorio llamado `nodejs-example-express-elasticache` para el paquete de origen de la aplicación. Cree el directorio `nodejs-example-express-elasticache` para este tutorial.

```
~$ mkdir nodejs-example-express-elasticache
```

Note

Cada tutorial de este capítulo usa su propio directorio para el paquete de origen de la aplicación. El nombre del directorio coincide con el nombre de la aplicación de ejemplo utilizada en el tutorial.

Cambie su directorio de trabajo actual a `nodejs-example-express-elasticache`.

```
~$ cd nodejs-example-express-elasticache
```

Ahora configuremos un entorno de Elastic Beanstalk que ejecute la plataforma Node.js y la aplicación de ejemplo. Se usará la interfaz de línea de comandos de Elastic Beanstalk (CLI de EB)

Para configurar un repositorio en la CLI de EB para la aplicación y crear un entorno de Elastic Beanstalk que ejecute la plataforma Node.js

1. Cree un repositorio con el comando [eb init](#).

```
~/nodejs-example-express-elasticache$ eb init --platform node.js --region <region>
```

Este comando crea un archivo de configuración en una carpeta llamada `.elasticbeanstalk` que especifica los ajustes para crear los entornos de la aplicación y crea una aplicación de Elastic Beanstalk con el nombre de la carpeta actual.

2. Cree un entorno que ejecute una aplicación de ejemplo con el comando [eb create](#).

```
~/nodejs-example-express-elasticache$ eb create --sample nodejs-example-express-elasticache
```

Este comando crea un entorno con balanceador de carga utilizando la configuración predeterminada de la plataforma de Node.js y los siguientes recursos:

- Instancia de EC2: una máquina virtual de Amazon Elastic Compute Cloud (Amazon EC2) configurada para ejecutar aplicaciones web en la plataforma que elija.

Cada plataforma ejecuta un conjunto específico de software, archivos de configuración y scripts compatibles con una determinada versión de lenguaje, marco de trabajo y contenedor web (o una combinación de estos). La mayoría de las plataformas utilizan Apache o nginx

como un proxy inverso que se sitúa delante de la aplicación web, reenvía las solicitudes a esta, administra los recursos estáticos y genera registros de acceso y errores.

- Instance security group (Grupo de seguridad de la instancia): grupo de seguridad de Amazon EC2 configurado para permitir el tráfico entrante en el puerto 80. Este recurso permite que el tráfico HTTP procedente del balanceador de carga llegue a la instancia EC2 en la que se ejecuta la aplicación web. De forma predeterminada, el tráfico no está permitido en otros puertos.
- Balanceador de carga: balanceador de carga de Elastic Load Balancing configurado para distribuir solicitudes a las instancias que se ejecutan en la aplicación. Los balanceadores de carga también permiten que las instancias no estén expuestas directamente a Internet.
- Grupo de seguridad del balanceador de carga: grupo de seguridad de Amazon EC2 configurado para permitir el tráfico entrante en el puerto 80. Este recurso permite que el tráfico HTTP procedente de Internet llegue al balanceador de carga. De forma predeterminada, el tráfico no está permitido en otros puertos.
- Grupo de Auto Scaling: grupo de Auto Scaling configurado para reemplazar una instancia si termina o deja de estar disponible.
- Bucket de Amazon S3: ubicación de almacenamiento para el código fuente, los registros y otros artefactos que se crean al utilizar Elastic Beanstalk.
- CloudWatch Alarmas de Amazon: dos CloudWatch alarmas que monitorean la carga de las instancias de su entorno y que se activan si la carga es demasiado alta o demasiado baja. Cuando se activa una alarma, en respuesta, el grupo de Auto Scaling aumenta o reduce los recursos.
- AWS CloudFormation pila: Elastic AWS CloudFormation Beanstalk se utiliza para lanzar los recursos de su entorno y propagar los cambios de configuración. Los recursos se definen en una plantilla que puede verse en la [consola de AWS CloudFormation](#).
- Nombre de dominio: nombre de dominio que direcciona el tráfico a la aplicación web con el formato *subdominio.región*.elasticbeanstalk.com.

Note

Para aumentar la seguridad de las aplicaciones de Elastic Beanstalk, el dominio elasticbeanstalk.com está registrado en la [lista de sufijos públicos \(PSL\)](#). Para mayor seguridad, se recomienda que utilice cookies con un prefijo `__Host-` en caso de que necesite configurar cookies confidenciales en el nombre de dominio predeterminado de sus aplicaciones de Elastic Beanstalk. Esta práctica le ayudará a defender su

dominio contra intentos de falsificación de solicitudes entre sitios (CSRF). Para obtener más información, consulte la página de [configuración de cookies](#) en la red de desarrolladores de Mozilla.

3. Cuando se complete la creación del entorno, utilice el comando `eb open` para abrir la URL del entorno en el navegador predeterminado.

```
~/nodejs-example-express-elasticache$ eb open
```

Ahora ha creado un entorno de Elastic Beanstalk para Node.js con una aplicación de ejemplo. Puede actualizarlo con su propia aplicación. Luego, actualizamos la aplicación de ejemplo para que use el marco de Express.

Actualizar la aplicación para que use Express

Actualice la aplicación de ejemplo en el entorno de Elastic Beanstalk para que utilice el marco de Express.

Puede descargar el [nodejs-example-express-elasticachecódigo](#) fuente final desde .zip.

Si desea actualizar la aplicación para que use Express

Una vez que haya creado el entorno con una aplicación de ejemplo, puede actualizarlo con su propia aplicación. En este procedimiento, ejecutamos primero los comandos `express` y `npm install` para configurar el marco de Express en el directorio de la aplicación.

1. Ejecute el comando `express`. Esto genera `package.json`, `app.js` y unos directorios.

```
~/nodejs-example-express-elasticache$ express
```

Cuando se le pregunte, escriba `y` si desea continuar.

Note

Si el comando `express` no funciona, es posible que no haya instalado el generador de línea de comandos Express tal y como se describe en la sección Requisitos previos anterior. O bien, puede que sea necesario configurar la ruta del directorio de su máquina local para ejecutar el comando `express`. Consulte la sección Requisitos previos para

conocer los pasos detallados sobre la configuración del entorno de desarrollo, de modo que pueda continuar con este tutorial.

2. Configure las dependencias locales.

```
~/nodejs-example-express-elasticache$ npm install
```

3. (Opcional) Compruebe que el servidor de aplicaciones web se inicie.

```
~/nodejs-example-express-elasticache$ npm start
```

Debería ver una salida similar a esta:

```
> nodejs@0.0.0 start /home/local/user/node-express
> node ./bin/www
```

El servidor se ejecuta en el puerto 3000 de forma predeterminada. Para probarlo, ejecute `curl http://localhost:3000` en otro terminal o abra un navegador en el equipo local e ingrese el `http://localhost:3000` de la dirección URL.

Presione `Ctrl+C` para detener el servidor.

4. Cambie el nombre de `nodejs-example-express-elasticache/app.js` a `nodejs-example-express-elasticache/express-app.js`.

```
~/nodejs-example-express-elasticache$ mv app.js express-app.js
```

5. Actualice la línea `var app = express();` en `nodejs-example-express-elasticache/express-app.js` a la siguiente:

```
var app = module.exports = express();
```

6. En el equipo local, cree un archivo llamado `nodejs-example-express-elasticache/app.js` con el siguiente código.

```
/**
 * Module dependencies.
 */

const express = require('express'),
```

```
session = require('express-session'),
bodyParser = require('body-parser'),
methodOverride = require('method-override'),
cookieParser = require('cookie-parser'),
fs = require('fs'),
filename = '/var/nodelist',
app = express();

let MemcachedStore = require('connect-memcached')(session);

function setup(cacheNodes) {
  app.use(bodyParser.raw());
  app.use(methodOverride());
  if (cacheNodes.length > 0) {
    app.use(cookieParser());

    console.log('Using memcached store nodes:');
    console.log(cacheNodes);

    app.use(session({
      secret: 'your secret here',
      resave: false,
      saveUninitialized: false,
      store: new MemcachedStore({ 'hosts': cacheNodes })
    }));
  } else {
    console.log('Not using memcached store.');
```

```
    app.use(session({
      resave: false,
      saveUninitialized: false, secret: 'your secret here'
    }));
  }

  app.get('/', function (req, resp) {
    if (req.session.views) {
      req.session.views++
      resp.setHeader('Content-Type', 'text/html')
      resp.send(`You are session: ${req.session.id}. Views: ${req.session.views}`)
    } else {
      req.session.views = 1
      resp.send(`You are session: ${req.session.id}. No views yet, refresh the page!`
    )
    }
  });
}
```

```
if (!module.parent) {
  console.log('Running express without cluster. Listening on port %d',
    process.env.PORT || 5000)
  app.listen(process.env.PORT || 5000)
}
}

console.log("Reading elastic cache configuration")
// Load elasticache configuration.
fs.readFile(filename, 'UTF8', function (err, data) {
  if (err) throw err;

  let cacheNodes = []
  if (data) {
    let lines = data.split('\n');
    for (let i = 0; i < lines.length; i++) {
      if (lines[i].length > 0) {
        cacheNodes.push(lines[i])
      }
    }
  }

  setup(cacheNodes)
});

module.exports = app;
```

7. Reemplace el contenido del archivo `nodejs-example-express-elasticache/bin/www` por lo siguiente:

```
#!/usr/bin/env node

/**
 * Module dependencies.
 */

const app = require('../app');
const cluster = require('cluster');
const debug = require('debug')('nodejs-example-express-elasticache:server');
const http = require('http');
const workers = {},
  count = require('os').cpus().length;
```

```
function spawn() {
  const worker = cluster.fork();
  workers[worker.pid] = worker;
  return worker;
}

/**
 * Get port from environment and store in Express.
 */

const port = normalizePort(process.env.PORT || '3000');
app.set('port', port);

if (cluster.isMaster) {
  for (let i = 0; i < count; i++) {
    spawn();
  }

  // If a worker dies, log it to the console and start another worker.
  cluster.on('exit', function (worker, code, signal) {
    console.log('Worker ' + worker.process.pid + ' died.');
```

```
    cluster.fork();
  });

  // Log when a worker starts listening
  cluster.on('listening', function (worker, address) {
    console.log('Worker started with PID ' + worker.process.pid + '.');
```

```
  });
} else {
  /**
   * Create HTTP server.
   */

  let server = http.createServer(app);

  /**
   * Event listener for HTTP server "error" event.
   */

  function onError(error) {
    if (error.syscall !== 'listen') {
```

```
    throw error;
  }

  const bind = typeof port === 'string'
    ? 'Pipe ' + port
    : 'Port ' + port;

  // handle specific listen errors with friendly messages
  switch (error.code) {
    case 'EACCES':
      console.error(bind + ' requires elevated privileges');
      process.exit(1);
      break;
    case 'EADDRINUSE':
      console.error(bind + ' is already in use');
      process.exit(1);
      break;
    default:
      throw error;
  }
}

/**
 * Event listener for HTTP server "listening" event.
 */

function onListening() {
  const addr = server.address();
  const bind = typeof addr === 'string'
    ? 'pipe ' + addr
    : 'port ' + addr.port;
  debug('Listening on ' + bind);
}

/**
 * Listen on provided port, on all network interfaces.
 */

server.listen(port);
server.on('error', onError);
server.on('listening', onListening);
}

/**
```

```
* Normalize a port into a number, string, or false.
*/

function normalizePort(val) {
  const port = parseInt(val, 10);

  if (isNaN(port)) {
    // named pipe
    return val;
  }

  if (port >= 0) {
    // port number
    return port;
  }

  return false;
}
```

8. Implemente los cambios en su entorno Elastic Beanstalk con el comando [eb deploy](#).

```
~/nodejs-example-express-elasticache$ eb deploy
```

9. Su entorno se actualizará al cabo de unos minutos. Una vez que el entorno esté listo, actualice la dirección URL para verificar que ha funcionado. Debería aparecer una página web que diga “Welcome to Express”.

Puede obtener acceso a los logs de las instancias EC2 que ejecutan la aplicación. Para obtener instrucciones acerca del acceso a los logs, consulte [Visualización de registros de instancias de Amazon EC2 en su entorno de Elastic Beanstalk](#).

A continuación, vamos a actualizar la aplicación Express para usar Amazon ElastiCache.

Para actualizar tu aplicación Express para usar Amazon ElastiCache

1. En su equipo local, cree un directorio `.ebextensions` en el directorio de nivel superior del paquete de código fuente. En este ejemplo, usaremos `nodejs-example-express-elasticache/.ebextensions`.
2. Cree un archivo de configuración `nodejs-example-express-elasticache/.ebextensions/elasticache-iam-with-script.config` con el siguiente fragmento. Para obtener más información sobre el archivo de configuración, consulte

[Espacio de nombres de configuración de Node.js](#). Se crea un usuario de IAM con los permisos necesarios para descubrir los nodos de elasticache y se escribe en un archivo siempre que cambia la caché. También puedes copiar el archivo desde un [nodejs-example-express-elasticachearchivo.zip](#). Para obtener más información sobre las ElastiCache propiedades, consulte [Ejemplo: ElastiCache](#).

 Note

YAML usa la sangría uniforme. Utilice el mismo nivel de sangría cuando sustituya el contenido en el archivo de configuración de ejemplo y asegúrese de que el editor de texto utiliza espacios para la sangría, no tabuladores.

```
Resources:
  MyCacheSecurityGroup:
    Type: 'AWS::EC2::SecurityGroup'
    Properties:
      GroupDescription: "Lock cache down to webserver access only"
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort:
            Fn::GetOptionSetting:
              OptionName: CachePort
              DefaultValue: 11211
          ToPort:
            Fn::GetOptionSetting:
              OptionName: CachePort
              DefaultValue: 11211
          SourceSecurityGroupName:
            Ref: AWSEBSecurityGroup
  MyElastiCache:
    Type: 'AWS::ElastiCache::CacheCluster'
    Properties:
      CacheNodeType:
        Fn::GetOptionSetting:
          OptionName: CacheNodeType
          DefaultValue: cache.t2.micro
      NumCacheNodes:
        Fn::GetOptionSetting:
          OptionName: NumCacheNodes
          DefaultValue: 1
```

```
Engine:
  Fn::GetOptionSetting:
    OptionName: Engine
    DefaultValue: redis
VpcSecurityGroupIds:
  -
  Fn::GetAtt:
    - MyCacheSecurityGroup
    - GroupId
AWSEBAutoScalingGroup :
  Metadata :
    ElastiCacheConfig :
      CacheName :
        Ref : MyElastiCache
      CacheSize :
        Fn::GetOptionSetting:
          OptionName : NumCacheNodes
          DefaultValue: 1
WebServerUser :
  Type : AWS::IAM::User
  Properties :
    Path : "/"
    Policies:
      -
        PolicyName: root
        PolicyDocument :
          Statement :
            -
              Effect : Allow
              Action :
                - cloudformation:DescribeStackResource
                - cloudformation:ListStackResources
                - elasticache:DescribeCacheClusters
              Resource : "*"
WebServerKeys :
  Type : AWS::IAM::AccessKey
  Properties :
    UserName :
      Ref: WebServerUser

Outputs:
  WebsiteURL:
    Description: sample output only here to show inline string function parsing
    Value: |
```

```

    http://`{ "Fn::GetAtt" : [ "AWSEBLoadBalancer", "DNSName" ] }`
MyElastiCacheName:
  Description: Name of the elasticache
  Value:
    Ref : MyElastiCache
NumCacheNodes:
  Description: Number of cache nodes in MyElastiCache
  Value:
    Fn::GetOptionSetting:
      OptionName : NumCacheNodes
      DefaultValue: 1

files:
  "/etc/cfn/cfn-credentials" :
    content : |
      AWSAccessKeyId=`{ "Ref" : "WebServerKeys" }`
      AWSSecretKey=`{ "Fn::GetAtt" : ["WebServerKeys", "SecretAccessKey"] }`
    mode : "000400"
    owner : root
    group : root

  "/etc/cfn/get-cache-nodes" :
    content : |
      # Define environment variables for command line tools
      export AWS_ELASTICACHE_HOME="/home/ec2-user/elasticache/$(ls /home/ec2-user/
elasticache/)"
      export AWS_CLOUDFORMATION_HOME=/opt/aws/apitools/cfn
      export PATH=$AWS_CLOUDFORMATION_HOME/bin:$AWS_ELASTICACHE_HOME/bin:$PATH
      export AWS_CREDENTIAL_FILE=/etc/cfn/cfn-credentials
      export JAVA_HOME=/usr/lib/jvm/jre

      # Grab the Cache node names and configure the PHP page
      aws cloudformation list-stack-resources --stack `{ "Ref" :
"AWS::StackName" }` --region `{ "Ref" : "AWS::Region" }` --output text | grep
MyElastiCache | awk '{print $4}' | xargs -I {} aws elasticache describe-cache-
clusters --cache-cluster-id {} --region `{ "Ref" : "AWS::Region" }` --show-
cache-node-info --output text | grep '^ENDPOINT' | awk '{print $2 ":" $3}' >
`{ "Fn::GetOptionSetting" : { "OptionName" : "NodeListPath", "DefaultValue" : "/"
var/www/html/nodelist" } }`
    mode : "000500"
    owner : root
    group : root

  "/etc/cfn/hooks.d/cfn-cache-change.conf" :

```

```

"content": |
  [cfn-cache-size-change]
  triggers=post.update
  path=Resources.AWSEBAutoScalingGroup.Metadata.ElastiCacheConfig
  action=/etc/cfn/get-cache-nodes
  runas=root

sources :
  "/home/ec2-user/elasticache" : "https://elasticache-downloads.s3.amazonaws.com/
  AmazonElastiCacheCli-latest.zip"

commands:
  make-elasticache-executable:
    command: chmod -R ugo+x /home/ec2-user/elasticache/*/bin/*

packages :
  "yum" :
    "aws-apitools-cfn" : []

container_commands:
  initial_cache_nodes:
    command: /etc/cfn/get-cache-nodes

```

3. En su equipo local, cree un archivo de configuración `nodejs-example-express-elasticache/.ebextensions/elasticache_settings.config` con el siguiente fragmento de código para configurarlo. ElastiCache

```

option_settings:
  "aws:elasticbeanstalk:customoption":
    CacheNodeType: cache.t2.micro
    NumCacheNodes: 1
    Engine: memcached
    NodeListPath: /var/nodelist

```

4. En el equipo local, sustituya `nodejs-example-express-elasticache/express-app.js` con el siguiente fragmento. Este archivo lee la lista de nodos del disco (`/var/nodelist`) y configura Express para que utilice `memcached` como un almacén de sesiones si los nodos están presentes. Su archivo debería tener el siguiente aspecto.

```

/**
 * Module dependencies.
 */

```

```
var express = require('express'),
    session = require('express-session'),
    bodyParser = require('body-parser'),
    methodOverride = require('method-override'),
    cookieParser = require('cookie-parser'),
    fs = require('fs'),
    filename = '/var/nodelist',
    app = module.exports = express();

var MemcachedStore = require('connect-memcached')(session);

function setup(cacheNodes) {
  app.use(bodyParser.raw());
  app.use(methodOverride());
  if (cacheNodes) {
    app.use(cookieParser());

    console.log('Using memcached store nodes:');
    console.log(cacheNodes);

    app.use(session({
      secret: 'your secret here',
      resave: false,
      saveUninitialized: false,
      store: new MemcachedStore({'hosts': cacheNodes})
    }));
  } else {
    console.log('Not using memcached store. ');
    app.use(cookieParser('your secret here'));
    app.use(session());
  }

  app.get('/', function(req, resp){
  if (req.session.views) {
    req.session.views++
    resp.setHeader('Content-Type', 'text/html')
    resp.write('Views: ' + req.session.views)
    resp.end()
  } else {
    req.session.views = 1
    resp.end('Refresh the page!')
  }
});
```

```
if (!module.parent) {
  console.log('Running express without cluster.');
```

```
  app.listen(process.env.PORT || 5000);
}
}
```

```
// Load elasticache configuration.
fs.readFile(filename, 'UTF8', function(err, data) {
  if (err) throw err;

  var cacheNodes = [];
  if (data) {
    var lines = data.split('\n');
    for (var i = 0 ; i < lines.length ; i++) {
      if (lines[i].length > 0) {
        cacheNodes.push(lines[i]);
      }
    }
  }
  setup(cacheNodes);
});
```

5. En el equipo local, actualice `package.json` con el siguiente contenido:

```
"dependencies": {
  "cookie-parser": "~1.4.4",
  "debug": "~2.6.9",
  "express": "~4.16.1",
  "http-errors": "~1.6.3",
  "jade": "~1.11.0",
  "morgan": "~1.9.1",
  "connect-memcached": "*",
  "express-session": "*",
  "body-parser": "*",
  "method-override": "*"
}
```

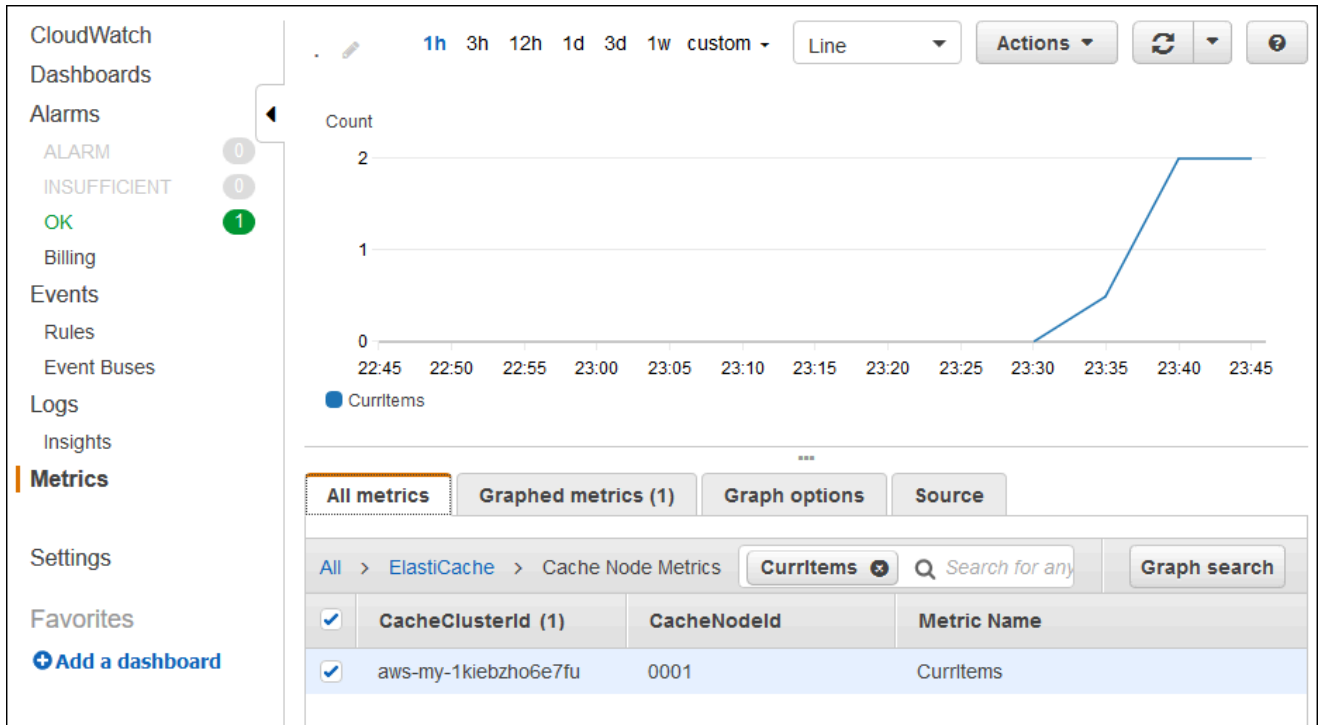
6. Ejecute `npm install`.

```
~/nodejs-example-express-elasticache$ npm install
```

7. Implemente la aplicación actualizada.

```
~/nodejs-example-express-elasticache$ eb deploy
```

8. Su entorno se actualizará al cabo de unos minutos. Una vez que su entorno esté listo, verifique que el código ha funcionado.
 - a. Consulta la [CloudWatch consola de Amazon](#) para ver tus ElastiCache estadísticas. Para ver tus ElastiCache métricas, selecciona Métricas en el panel izquierdo y, a continuación, busca CurrItems. Seleccione ElastiCache > Métricas del nodo de caché y, a continuación, seleccione su nodo de caché para ver el número de elementos de la caché.



Note

Asegúrese de buscar en la misma región en la que ha implementado su aplicación.

Si copias y pegas la URL de tu aplicación en otro navegador web y actualizas la página, el CurrItem recuento debería aumentar al cabo de 5 minutos.

- b. Tome una instantánea de los registros.. Para obtener más información acerca de la recuperación de registros, consulte [Visualización de registros de instancias de Amazon EC2 en su entorno de Elastic Beanstalk](#).

- c. Compruebe el archivo `/var/log/nodejs/nodejs.log` en el paquete del registro. Debería ver algo similar a lo siguiente:

```
Using memcached store nodes:  
[ 'aws-my-1oys9co8zt1uo.1iwtrn.0001.use1.cache.amazonaws.com:11211' ]
```

Limpieza

Si ya no desea ejecutar la aplicación, puede limpiar los recursos terminando el entorno y eliminando la aplicación.

Utilice el comando `eb terminate` para finalizar el entorno y el comando `eb delete` para eliminar la aplicación.

Para terminar su entorno

En el directorio en el que creó el repositorio local, ejecute `eb terminate`.

```
$ eb terminate
```

Este proceso puede tardar unos minutos. Elastic Beanstalk muestra un mensaje cuando el entorno termina correctamente.

Implementación de una aplicación Node.js con DynamoDB en Elastic Beanstalk

Este tutorial y su aplicación de ejemplo, [nodejs-example-dynamo.zip](#), explican el proceso de implementación de una aplicación de Node.js que utiliza el AWS SDK de Node.js para JavaScript interactuar con el servicio Amazon DynamoDB. Creará una tabla de DynamoDB que esté en una base de datos desacoplada o externa del entorno. AWS Elastic Beanstalk También configurará la aplicación para que utilice una base de datos desacoplada. En un entorno de producción, se recomienda utilizar una base de datos que esté desacoplada del entorno de Elastic Beanstalk para que sea independiente del ciclo de vida del entorno. Esta práctica también permite realizar [implementaciones azul/verde](#).

La aplicación de ejemplo ilustra lo siguiente:

- Una tabla de DynamoDB que almacena los datos de texto proporcionados por el usuario.
- Los [archivos de configuración](#) para crear la tabla.

- Un tema de Amazon Simple Notification Service.
- Uso de un [archivo package.json](#) para instalar paquetes durante la implementación.

Secciones

- [Requisitos previos](#)
- [Cree un entorno de Elastic Beanstalk](#)
- [Añadir permisos a las instancias del entorno](#)
- [Implementar la aplicación de ejemplo](#)
- [Creación de una tabla de DynamoDB](#)
- [Actualización de los archivos de configuración de la aplicación](#)
- [Configuración de su entorno para alta disponibilidad](#)
- [Limpieza](#)
- [Siguiendo pasos](#)

Requisitos previos

Este tutorial tiene siguientes los requisitos previos:

- Los tiempos de ejecución de Node.js
- El software de gestión de paquetes de Node.js predeterminado, npm
- El generador de línea de comandos Express
- La interfaz de línea de comando de Elastic Beanstalk (CLI de EB)

Para obtener más detalles acerca de la instalación de los primeros tres componentes y la configuración de su entorno de desarrollo local, consulte [Configuración del entorno de desarrollo de Node.js](#). Para este tutorial, no necesita instalar el AWS SDK para Node.js, que también se menciona en el tema al que se hace referencia.

Para obtener detalles sobre la instalación y configuración de la CLI de EB, consulte [Instalación de la CLI de EB](#) y [Configuración de la CLI de EB](#).

Cree un entorno de Elastic Beanstalk

Su directorio de aplicaciones

Este tutorial usa un directorio llamado `nodejs-example-dynamo` para el paquete de origen de la aplicación. Cree el directorio `nodejs-example-dynamo` para este tutorial.

```
~$ mkdir nodejs-example-dynamo
```

Note

Cada tutorial de este capítulo usa su propio directorio para el paquete de origen de la aplicación. El nombre del directorio coincide con el nombre de la aplicación de ejemplo utilizada en el tutorial.

Cambie su directorio de trabajo actual a `nodejs-example-dynamo`.

```
~$ cd nodejs-example-dynamo
```

Ahora configuremos un entorno de Elastic Beanstalk que ejecute la plataforma Node.js y la aplicación de ejemplo. Se usará la interfaz de línea de comandos de Elastic Beanstalk (CLI de EB)

Para configurar un repositorio en la CLI de EB para la aplicación y crear un entorno de Elastic Beanstalk que ejecute la plataforma Node.js

1. Cree un repositorio con el comando [eb init](#).

```
~/nodejs-example-dynamo$ eb init --platform node.js --region <region>
```

Este comando crea un archivo de configuración en una carpeta llamada `.elasticbeanstalk` que especifica los ajustes para crear los entornos de la aplicación y crea una aplicación de Elastic Beanstalk con el nombre de la carpeta actual.

2. Cree un entorno que ejecute una aplicación de ejemplo con el comando [eb create](#).

```
~/nodejs-example-dynamo$ eb create --sample nodejs-example-dynamo
```

Este comando crea un entorno con balanceador de carga utilizando la configuración predeterminada de la plataforma de Node.js y los siguientes recursos:

- Instancia de EC2: una máquina virtual de Amazon Elastic Compute Cloud (Amazon EC2) configurada para ejecutar aplicaciones web en la plataforma que elija.

Cada plataforma ejecuta un conjunto específico de software, archivos de configuración y scripts compatibles con una determinada versión de lenguaje, marco de trabajo y contenedor web (o una combinación de estos). La mayoría de las plataformas utilizan Apache o nginx como un proxy inverso que se sitúa delante de la aplicación web, reenvía las solicitudes a esta, administra los recursos estáticos y genera registros de acceso y errores.

- Instance security group (Grupo de seguridad de la instancia): grupo de seguridad de Amazon EC2 configurado para permitir el tráfico entrante en el puerto 80. Este recurso permite que el tráfico HTTP procedente del balanceador de carga llegue a la instancia EC2 en la que se ejecuta la aplicación web. De forma predeterminada, el tráfico no está permitido en otros puertos.
- Balanceador de carga: balanceador de carga de Elastic Load Balancing configurado para distribuir solicitudes a las instancias que se ejecutan en la aplicación. Los balanceadores de carga también permiten que las instancias no estén expuestas directamente a Internet.
- Grupo de seguridad del balanceador de carga: grupo de seguridad de Amazon EC2 configurado para permitir el tráfico entrante en el puerto 80. Este recurso permite que el tráfico HTTP procedente de Internet llegue al balanceador de carga. De forma predeterminada, el tráfico no está permitido en otros puertos.
- Grupo de Auto Scaling: grupo de Auto Scaling configurado para reemplazar una instancia si termina o deja de estar disponible.
- Bucket de Amazon S3: ubicación de almacenamiento para el código fuente, los registros y otros artefactos que se crean al utilizar Elastic Beanstalk.
- CloudWatch Alarmas de Amazon: dos CloudWatch alarmas que monitorean la carga de las instancias de su entorno y que se activan si la carga es demasiado alta o demasiado baja. Cuando se activa una alarma, en respuesta, el grupo de Auto Scaling aumenta o reduce los recursos.
- AWS CloudFormation pila: Elastic AWS CloudFormation Beanstalk se utiliza para lanzar los recursos de su entorno y propagar los cambios de configuración. Los recursos se definen en una plantilla que puede verse en la [consola de AWS CloudFormation](#).
- Nombre de dominio: nombre de dominio que direcciona el tráfico a la aplicación web con el formato *subdominio.región.elasticbeanstalk.com*.

Note

Para aumentar la seguridad de las aplicaciones de Elastic Beanstalk, el dominio elasticbeanstalk.com está registrado en la [lista de sufijos públicos \(PSL\)](#). Para mayor

seguridad, se recomienda que utilice cookies con un prefijo `__Host-` en caso de que necesite configurar cookies confidenciales en el nombre de dominio predeterminado de sus aplicaciones de Elastic Beanstalk. Esta práctica le ayudará a defender su dominio contra intentos de falsificación de solicitudes entre sitios (CSRF). Para obtener más información, consulte la página de [configuración de cookies](#) en la red de desarrolladores de Mozilla.

3. Cuando se complete la creación del entorno, utilice el comando `eb open` para abrir la URL del entorno en el navegador predeterminado.

```
~/nodejs-example-dynamo$ eb open
```

Ahora ha creado un entorno de Elastic Beanstalk para Node.js con una aplicación de ejemplo. Puede actualizarlo con su propia aplicación. Luego, actualizamos la aplicación de ejemplo para que use el marco de Express.

Añadir permisos a las instancias del entorno

La aplicación se ejecuta en una o varias instancias EC2 detrás de un balanceador de carga y se encarga de servir solicitudes HTTP desde Internet. Cuando recibe una solicitud que requiere el uso de servicios, la aplicación utiliza los permisos de la instancia en la que se ejecuta para acceder a esos AWS servicios.

La aplicación de ejemplo usa permisos de instancia para escribir datos en una tabla de DynamoDB y enviar notificaciones a un tema de Amazon SNS con el SDK de Node.js. JavaScript Añada las siguientes políticas administradas al [perfil de instancia](#) predeterminado para conceder a las instancias EC2 del entorno permiso para tener acceso a DynamoDB y Amazon SNS:

- AmazonDynamoBASE DE DATOS FullAccess
- Amazon SNS FullAccess

Para agregar políticas al perfil de instancia predeterminado

1. Abra la [página Roles](#) en la consola de IAM.
2. Elija 2 funciones `saws-elasticbeanstalk-ec`.
3. En la pestaña Permissions (Permisos), elija Attach policies (Adjuntar políticas).

4. Seleccione la política administrada que se aplicará a los demás servicios que utilice la aplicación. Para este tutorial, seleccione `AmazonSNSFullAccess` y `AmazonDynamoDBFullAccess`.
5. Elija Asociar política.

Para obtener más información acerca de cómo administrar perfiles de instancia, consulte [Administración de perfiles de instancia de Elastic Beanstalk](#).

Implementar la aplicación de ejemplo

Ahora su entorno está listo para implementar y ejecutar la aplicación de ejemplo de este tutorial: [nodejs-example-dynamo.zip](#).

Para implementar y ejecutar la aplicación de ejemplo del tutorial

1. Cambie el directorio de trabajo actual a `nodejs-example-dynamo` de directorio de la aplicación.

```
~$ cd nodejs-example-dynamo
```

2. Descargue y extraiga el contenido del paquete de código fuente de la aplicación de ejemplo ([nodejs-example-dynamo.zip](#)) en el directorio de la aplicación. `nodejs-example-dynamo`
3. Implemente la aplicación de ejemplo en su entorno Elastic Beanstalk con el comando [eb deploy](#).

```
~/nodejs-example-dynamo$ eb deploy
```

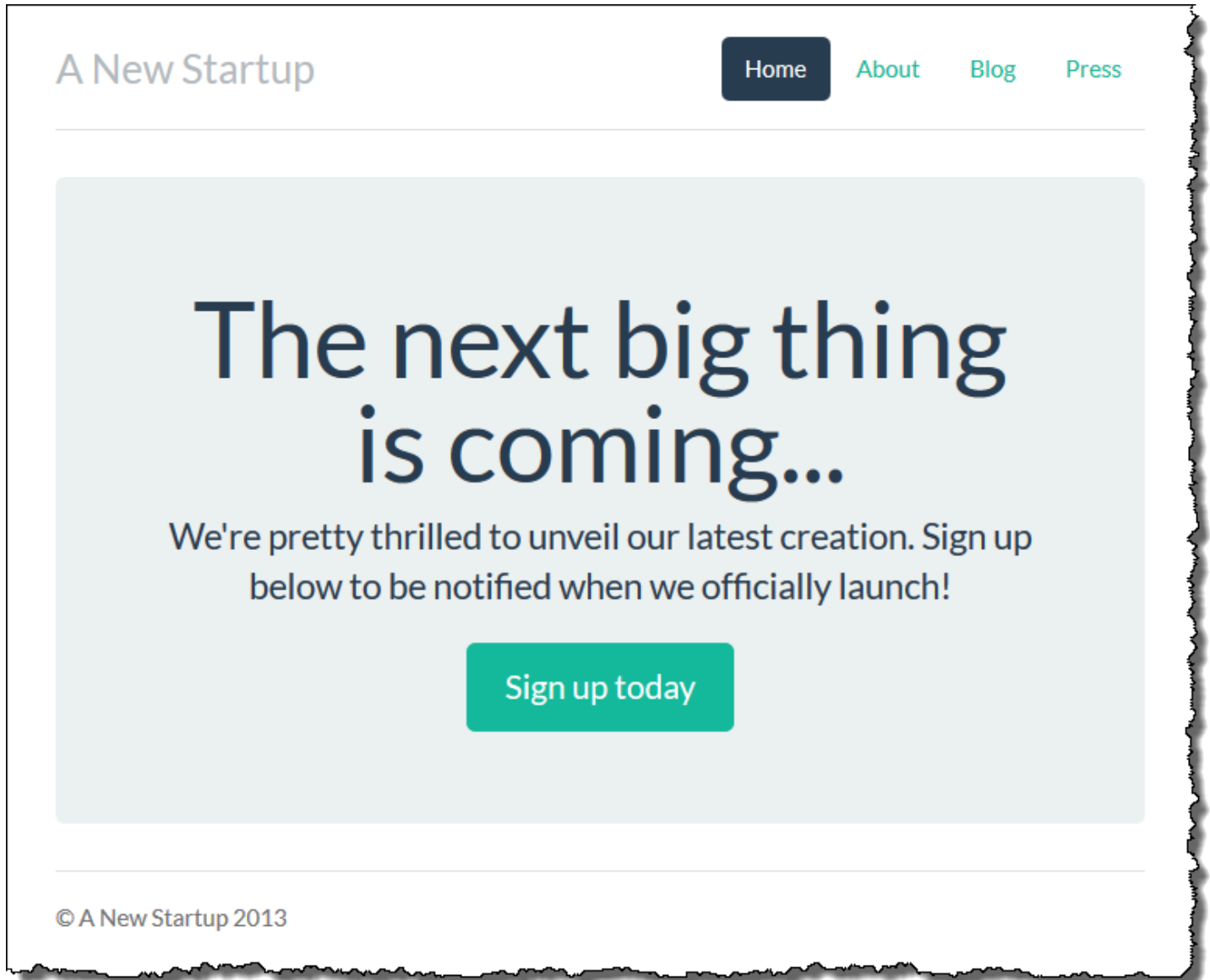
Note

De forma predeterminada, el `eb deploy` comando crea un archivo ZIP de la carpeta del proyecto. Puede configurar la CLI de EB para implementar un artefacto de su proceso de compilación en lugar de crear un archivo ZIP de la carpeta del proyecto. Para obtener más información, consulte [Implementar un artefacto en lugar de la carpeta del proyecto](#).

4. Cuando se complete la creación del entorno, utilice el comando [eb open](#) para abrir la URL del entorno en el navegador predeterminado.

```
~/nodejs-example-dynamo$ eb open
```

El sitio recopila la información de contacto del usuario y utiliza una tabla de DynamoDB para almacenar los datos. Para agregar una entrada, elija Sign up today (Regístrese hoy mismo), introduzca un nombre y dirección de correo electrónico y, a continuación, elija Sign Up! (Regístrese) La aplicación web escribe el contenido del formulario en la tabla y activa una notificación por correo electrónico de Amazon SNS.



De momento, el tema de Amazon SNS está configurado con un marcador de posición para el correo electrónico de las notificaciones. Actualizará la configuración en breve, pero entretanto puede verificar la tabla de DynamoDB y el tema de Amazon SNS en la AWS Management Console.

Para ver la tabla

1. Abra la [página Tables](#) en la consola de DynamoDB.

2. Busque la tabla que creó la aplicación. El nombre comienza por `awseb` y contiene `StartupSignupsTable`.
3. Seleccione la tabla, elija `Items (Elementos)` y, a continuación, elija `Start search (Iniciar búsqueda)` para ver todos los elementos de la tabla.

La tabla contiene una entrada para cada dirección de correo electrónico enviada en el sitio de inscripción. Además de escribir en la tabla, la aplicación envía un mensaje a un tema de Amazon SNS que tiene dos suscripciones: una para las notificaciones por correo electrónico que se le envían a usted y otra para una cola de Amazon Simple Queue Service que una aplicación de trabajo puede leer para procesar las solicitudes y enviar mensajes de correo electrónico a los clientes interesados.

Para ver el tema

1. Abra la página [Topics \(Temas\)](#) en la consola de Amazon SNS.
2. Busque el tema que creó la aplicación. El nombre comienza por `awseb` y contiene `NewSignupTopic`.
3. Seleccione el tema para ver sus suscripciones.

La aplicación ([app.js](#)) define dos rutas. La ruta raíz (`/`) devuelve una página web renderizada a partir de una plantilla integrada JavaScript (EJS) con un formulario que el usuario rellena para registrar su nombre y dirección de correo electrónico. Al enviar el formulario se envía una solicitud POST con los datos del formulario a la ruta `/signup`, que escribe una entrada en la tabla de DynamoDB y publica un mensaje en el tema de Amazon SNS para enviar una notificación al propietario de la inscripción.

La aplicación de ejemplo incluye [archivos de configuración](#) que crean la tabla de DynamoDB, el tema de Amazon SNS y la cola de Amazon SQS usados por la aplicación. Esto le permite crear un nuevo entorno y probar la funcionalidad de forma inmediata, pero tiene el inconveniente de que la tabla de DynamoDB está asociada al entorno. Para un entorno de producción, debería crear la tabla de DynamoDB fuera de su entorno para evitar perderla cuando termine el entorno o actualice su configuración.

Creación de una tabla de DynamoDB

Para utilizar una tabla de DynamoDB externa con una aplicación que se ejecuta en Elastic Beanstalk, primero cree una tabla en DynamoDB. Cuando crea una tabla fuera de Elastic Beanstalk, es

completamente independiente de Elastic Beanstalk y de sus entornos de Elastic Beanstalk, y Elastic Beanstalk no la finalizará.

Cree una tabla con la siguiente configuración:

- Table name (Nombre de la tabla) – **nodejs-tutorial**
- Primary key (Clave principal) – **email**
- Tipo de clave principal: String (Cadena)

Para crear una tabla de DynamoDB

1. Abra la [página Tables](#) en la consola de administración de DynamoDB.
2. Seleccione Create table (Crear tabla).
3. Escriba un Table name (Nombre de tabla) y una Primary key (Clave principal).
4. Elija el tipo de clave principal.
5. Seleccione Crear.

Actualización de los archivos de configuración de la aplicación

Actualice los [archivos de configuración](#) del origen de la aplicación para que utilicen la tabla nodejs-tutorial en lugar de crear una nueva.

Para actualizar la aplicación de ejemplo para su uso en producción

1. Cambie el directorio de trabajo actual a `nodejs-example-dynamo` de directorio de la aplicación.

```
~$ cd nodejs-example-dynamo
```

2. Abra `.ebextensions/options.config` y cambie los valores de las siguientes opciones:
 - `NewSignupEmail`— Su dirección de correo electrónico.
 - `STARTUP_SIGNUP_TABLE`: `nodejs-tutorial`

Example .ebextensions/options.config

```
option_settings:
  aws:elasticbeanstalk:customoption:
    NewSignupEmail: you@example.com
  aws:elasticbeanstalk:application:environment:
    THEME: "flatly"
    AWS_REGION: '`{"Ref" : "AWS::Region"}`'
    STARTUP_SIGNUP_TABLE: nodejs-tutorial
    NEW_SIGNUP_TOPIC: '`{"Ref" : "NewSignupTopic"}`'
  aws:elasticbeanstalk:container:nodejs:
    ProxyServer: nginx
  aws:elasticbeanstalk:container:nodejs:staticfiles:
    /static: /static
  aws:autoscaling:asg:
    Cooldown: "120"
  aws:autoscaling:trigger:
    Unit: "Percent"
    Period: "1"
    BreachDuration: "2"
    UpperThreshold: "75"
    LowerThreshold: "30"
    MeasureName: "CPUUtilization"
```

Esto se aplica a las siguientes configuraciones para la aplicación:

- La dirección de correo electrónico que el tema de Amazon SNS utiliza para las notificaciones está configurado en su dirección o en la que introduzca en el archivo `options.config`.
- Se utilizará la tabla `nodejs-tutorial` en lugar de la creada por `.ebextensions/create-dynamodb-table.config`.

3. Remove `.ebextensions/create-dynamodb-table.config`.

```
~/nodejs-tutorial$ rm .ebextensions/create-dynamodb-table.config
```

La próxima vez que implemente la aplicación, se eliminará la tabla creada por este archivo de configuración.

4. Implemente la aplicación actualizada en su entorno Elastic Beanstalk con el comando [eb deploy](#).

```
~/nodejs-example-dynamo$ eb deploy
```

5. Cuando se complete la creación del entorno, utilice el comando [eb open](#) para abrir la URL del entorno en el navegador predeterminado.

```
~/nodejs-example-dynamo$ eb open
```

Cuando lo implemente, Elastic Beanstalk actualizará la configuración del tema de Amazon SNS y eliminará la tabla de DynamoDB que se creó cuando implementó la primera versión de la aplicación.

Ahora, cuando termine el entorno, no se eliminará la tabla `nodejs-tutorial`. Esto le permite realizar implementaciones blue/green, modificar los archivos de configuración o retirar su sitio web sin arriesgarse a perder datos.

Abra su sitio en un navegador y verifique que el formulario funciona según lo previsto. Cree un par de entradas y, a continuación, consulte la consola de DynamoDB para verificar la tabla.

Para ver la tabla

1. Abra la [página Tables](#) en la consola de DynamoDB.
2. Busque la tabla `nodejs-tutorial`.
3. Seleccione la tabla, elija `Items (Elementos)` y, a continuación, elija `Start search (Iniciar búsqueda)` para ver todos los elementos de la tabla.

También puede ver que Elastic Beanstalk eliminó la tabla que creó anteriormente.

Configuración de su entorno para alta disponibilidad

Por último, configure el grupo de Auto Scaling de su entorno con un número mínimo de instancias mayor. Ejecute al menos dos instancias en todo momento para evitar que los servidores web de su entorno se conviertan en un único punto de error y para poder implementar cambios sin que el sitio se quede fuera de servicio.

Si desea configurar el grupo de Auto Scaling del entorno para disponer de una alta disponibilidad

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija `Environments (Entornos)` y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Capacity (Capacidad), elija Edit (Editar).
5. En la sección Auto Scaling group (grupo de Auto Scaling) defina Min instances (Instancias mínimas) en **2**.
6. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

Limpieza

Cuando termine de trabajar con Elastic Beanstalk, puede terminar su entorno. [Elastic Beanstalk AWS cancela todos los recursos asociados al entorno, como las instancias de Amazon EC2, las instancias de bases de datos, los balanceadores de carga, los grupos de seguridad y las alarmas.](#)

Para cerrar el entorno de Elastic Beanstalk desde la consola

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. Elija Actions (Acciones) y, a continuación, Terminate Environment (Terminar el entorno).
4. Utilice el cuadro de diálogo en pantalla para confirmar la terminación del entorno.

Con Elastic Beanstalk, puede crear fácilmente un nuevo entorno para su aplicación en cualquier momento.

También puede eliminar las tablas de DynamoDB externas que haya creado.

Para eliminar una tabla de DynamoDB

1. Abra la [página Tables](#) en la consola de DynamoDB.

2. Seleccione una tabla.
3. Seleccione Actions (Acciones) y Delete table (Eliminar tabla).
4. Elija Eliminar.

Siguientes pasos

La aplicación de ejemplo utiliza archivos de configuración para configurar los ajustes del software y crear AWS recursos como parte de su entorno. Para obtener más información sobre los archivos de configuración y cómo se utilizan, consulte [Personalización avanzada de entornos con archivos de configuración \(.ebextensions\)](#).

La aplicación de ejemplo de este tutorial utiliza la plataforma web Express para Node.js. Para obtener más información sobre Express, consulte la documentación oficial en expressjs.com.

Por último, si pretende utilizar la aplicación en un entorno de producción, [configure un nombre de dominio personalizado](#) para el entorno y [habilite HTTPS](#) para las conexiones seguras.

Agregar una instancia de base de datos de Amazon RDS DB a un entorno de aplicaciones Node.js

Puede utilizar una instancia de base de datos de Amazon Relational Database Service (Amazon RDS) para almacenar datos recopilados y modificados por su aplicación. La base de datos se puede adjuntar al entorno y administrarse con Elastic Beanstalk, o bien, crearse como desacoplada y administrarse de forma externa a través de otro servicio. En este tema se proporcionan instrucciones para crear un Amazon RDS mediante la consola de Elastic Beanstalk. La base de datos se acoplará al entorno y se administrará con Elastic Beanstalk. Para obtener más información sobre el uso de Elastic Beanstalk con Amazon RDS, consulte [Añadir una base de datos al entorno de Elastic Beanstalk](#).

Secciones

- [Añadir una instancia de base de datos al entorno](#)
- [Descargar un controlador](#)
- [Conexión a una base de datos](#)

Añadir una instancia de base de datos al entorno

Para agregar una instancia de base de datos al entorno

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Database (Base de datos), seleccione Edit (Editar).
5. Seleccione un motor de base de datos y escriba el nombre de usuario y la contraseña.
6. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

Se tarda alrededor de 10 minutos en añadir una instancia de base de datos. Cuando se haya completado la actualización del entorno, el nombre de host de la instancia de base de datos y otra información de conexión estarán disponibles para su aplicación mediante las siguientes propiedades de entorno

Nombre de la propiedad	Descripción	Valor de la propiedad
RDS_HOSTNAME	El nombre de host de la instancia de base de datos.	En la pestaña Connectivity & security (Conectividad y seguridad) de la consola de Amazon RDS: Endpoint (Punto de enlace).
RDS_PORT	El puerto en donde la instancia de base de datos acepta las conexiones. El valor predeterminado varía	En la pestaña Connectivity & security (Conectividad y seguridad) de la consola de Amazon RDS: Port (Puerto).

Nombre de la propiedad	Descripción	Valor de la propiedad
	dependiendo del motor de base de datos.	
RDS_DB_NAME	El nombre de la base de dato, ebdb .	En la pestaña Configuration (Configuración) de la consola de Amazon RDS: DB Name (Nombre de la base de datos).
RDS_USERNAME	El nombre de usuario que ha configurado para la base de datos.	En la pestaña Configuration (Configuración) de la consola de Amazon RDS: Master username (Nombre de usuario maestro).
RDS_PASSWORD	La contraseña que ha configurado para la base de datos.	No está disponible como referencia en la consola de Amazon RDS.

Para obtener más información sobre la configuración de una instancia de bases de datos acopladas a un entorno de Elastic Beanstalk, consulte [Añadir una base de datos al entorno de Elastic Beanstalk](#).

Descargar un controlador

Añada el controlador de base de datos al archivo [package.json del proyecto](#) bajo `dependencies`.

Example **package.json**: Express con MySQL

```
{
  "name": "my-app",
  "version": "0.0.1",
  "private": true,
  "dependencies": {
    "ejs": "latest",
    "aws-sdk": "latest",
    "express": "latest",
    "body-parser": "latest",
    "mysql": "latest"
  }
}
```

```
},
"scripts": {
  "start": "node app.js"
}
}
```

Paquetes de controladores comunes para Node.js

- MySQL — [mysql](#)
- PostgreSQL — [node-postgres](#)
- SQL Server — [node-mssql](#)
- Oracle — [node-oracledb](#)

Conexión a una base de datos

Elastic Beanstalk proporciona información de conexión para instancias de base de datos asociadas en propiedades de entorno. Use `process.env.VARIABLE` para leer las propiedades y configurar una conexión con la base de datos.

Example app.js: conexión con una base de datos MySQL

```
var mysql = require('mysql');

var connection = mysql.createConnection({
  host      : process.env.RDS_HOSTNAME,
  user      : process.env.RDS_USERNAME,
  password  : process.env.RDS_PASSWORD,
  port      : process.env.RDS_PORT
});

connection.connect(function(err) {
  if (err) {
    console.error('Database connection failed: ' + err.stack);
    return;
  }

  console.log('Connected to database.');
```

```
connection.end();
```

Para obtener más información sobre cómo crear cadena de conexión a través de node-mysql, consulte npmjs.org/package/mysql.

Recursos

Existen varios lugares a los que puede acudir para obtener ayuda adicional con el desarrollo de sus aplicaciones Node.js:

Recurso	Descripción
GitHub	Instalar el SDK de AWS para Node.js mediante GitHub.
Foro de desarrollo de Node.js	Publique sus preguntas y obtenga comentarios.
AWS SDK de para Node.js (vista previa para desarrolladores)	Un único lugar para obtener código de ejemplo, documentación, herramientas y recursos adicionales.

Creación e implementación de aplicaciones PHP en Elastic Beanstalk

AWS Elastic Beanstalk for PHP facilita la implementación, la administración y el escalado de sus aplicaciones web PHP mediante Amazon Web Services. En este capítulo se proporcionan instrucciones para implementar la aplicación web PHP en Elastic Beanstalk. Puede implementar la aplicación en solo unos minutos mediante la interfaz de línea de comandos de Elastic Beanstalk (EB CLI) o la consola de Elastic Beanstalk.

En este capítulo se proporcionan los siguientes tutoriales:

- [QuickStart para PHP](#)— Implemente una aplicación PHP Hello World mediante la CLI de EB.
- [Ejemplos de aplicaciones y tutoriales](#)— Tutoriales detallados para marcos comunes como CakePHP y Symfony, además de añadir una instancia de Amazon RDS a su entorno de aplicaciones PHP.

Si necesita ayuda para desarrollar aplicaciones en PHP, hay varios lugares a los que puede acudir:

- [GitHub](#)— Instala el AWS SDK for PHP usando GitHub.

- [Centro para desarrolladores de PHP](#): ventanilla única para códigos de muestra, documentación, herramientas y recursos adicionales.
- [AWS Preguntas frecuentes sobre SDK for PHP](#): obtén respuestas a las preguntas más frecuentes.

Temas

- [QuickStart: Implementar una aplicación PHP en Elastic Beanstalk](#)
- [Configuración del entorno de desarrollo de PHP](#)
- [Uso de la plataforma PHP de Elastic Beanstalk](#)
- [Más ejemplos de aplicaciones y tutoriales para PHP](#)

QuickStart: Implementar una aplicación PHP en Elastic Beanstalk

Este QuickStart tutorial explica el proceso de creación de una aplicación PHP y su despliegue en un AWS Elastic Beanstalk entorno.

Note

Este QuickStart tutorial está destinado a fines de demostración. No utilice la aplicación creada en este tutorial para el tráfico de producción.

Secciones

- [Tu AWS cuenta](#)
- [Requisitos previos](#)
- [Paso 1: Crea una aplicación PHP](#)
- [Paso 2: Ejecute la aplicación localmente](#)
- [Paso 3: Implemente la aplicación PHP con la CLI de EB](#)
- [Paso 4: Ejecute la aplicación en Elastic Beanstalk](#)
- [Paso 5: Eliminar](#)
- [AWS recursos para su aplicación](#)
- [Siguiendo pasos](#)
- [Implemente con la consola de Elastic Beanstalk](#)

Tu AWS cuenta

Si aún no eres AWS cliente, debes crear una AWS cuenta. El registro le permite acceder a Elastic Beanstalk AWS y a otros servicios que necesite.

Si ya tiene una AWS cuenta, puede pasar a [Requisitos previos](#)

Crear una cuenta de AWS

Inscríbese en una Cuenta de AWS

Si no tiene una Cuenta de AWS, complete los siguientes pasos para crearlo.

Para suscribirse a una Cuenta de AWS

1. Abra <https://portal.aws.amazon.com/billing/signup>.
2. Siga las instrucciones que se le indiquen.

Parte del procedimiento de registro consiste en recibir una llamada telefónica e indicar un código de verificación en el teclado del teléfono.

Cuando te registras en una Cuenta de AWS, Usuario raíz de la cuenta de AWS se crea una. El usuario raíz tendrá acceso a todos los Servicios de AWS y recursos de esa cuenta. Como práctica recomendada de seguridad, asigne acceso administrativo a un usuario y utilice únicamente el usuario raíz para realizar [tareas que requieren acceso de usuario raíz](#).

AWS te envía un correo electrónico de confirmación una vez finalizado el proceso de registro. Puede ver la actividad de la cuenta y administrar la cuenta en cualquier momento entrando en <https://aws.amazon.com/> y seleccionando Mi cuenta.

Creación de un usuario con acceso administrativo

Después de crear un usuario administrativo Cuenta de AWS, asegúrelo Usuario raíz de la cuenta de AWS AWS IAM Identity Center, habilite y cree un usuario administrativo para no usar el usuario root en las tareas diarias.

Proteja su Usuario raíz de la cuenta de AWS

1. Inicie sesión [AWS Management Console](#) como propietario de la cuenta seleccionando el usuario root e introduciendo su dirección de Cuenta de AWS correo electrónico. En la siguiente página, escriba su contraseña.

Para obtener ayuda para iniciar sesión con el usuario raíz, consulte [Iniciar sesión como usuario raíz](#) en la Guía del usuario de AWS Sign-In .

2. Active la autenticación multifactor (MFA) para el usuario raíz.

Para obtener instrucciones, consulte [Habilitar un dispositivo MFA virtual para el usuario Cuenta de AWS raíz \(consola\)](#) en la Guía del usuario de IAM.

Creación de un usuario con acceso administrativo

1. Activar IAM Identity Center.

Consulte las instrucciones en [Activar AWS IAM Identity Center](#) en la Guía del usuario de AWS IAM Identity Center .

2. En IAM Identity Center, conceda acceso administrativo a un usuario.

Para ver un tutorial sobre su uso Directorio de IAM Identity Center como fuente de identidad, consulte [Configurar el acceso de los usuarios con la configuración predeterminada Directorio de IAM Identity Center en la](#) Guía del AWS IAM Identity Center usuario.

Iniciar sesión como usuario con acceso de administrador

- Para iniciar sesión con el usuario de IAM Identity Center, utilice la URL de inicio de sesión que se envió a la dirección de correo electrónico cuando creó el usuario de IAM Identity Center.

Para obtener ayuda para iniciar sesión con un usuario del Centro de identidades de IAM, consulte [Iniciar sesión en el portal de AWS acceso](#) en la Guía del AWS Sign-In usuario.

Concesión de acceso a usuarios adicionales

1. En IAM Identity Center, cree un conjunto de permisos que siga la práctica recomendada de aplicar permisos de privilegios mínimos.

Para conocer las instrucciones, consulte [Create a permission set](#) en la Guía del usuario de AWS IAM Identity Center .

2. Asigne usuarios a un grupo y, a continuación, asigne el acceso de inicio de sesión único al grupo.

Para conocer las instrucciones, consulte [Add groups](#) en la Guía del usuario de AWS IAM Identity Center .

Requisitos previos

Para seguir los procedimientos de esta guía, necesitará un shell o un terminal de línea de comando donde pueda ejecutar los comandos. Los comandos aparecen en listas y van precedidos del símbolo del sistema (\$) y del nombre del directorio actual, si es aplicable.

```
~/eb-project$ this is a command  
this is output
```

En Linux y macOS, puede utilizar el administrador de shell y paquetes preferido. En Windows, puede [instalar el subsistema de Windows para Linux](#) y obtener una versión de Ubuntu y Bash integrada en Windows.

CLI DE EB

En este tutorial también se utiliza la interfaz de línea de comandos de Elastic Beanstalk (CLI de EB). Para obtener detalles sobre la instalación y configuración de la CLI de EB, consulte [Instalación de la CLI de EB](#) y [Configuración de la CLI de EB](#).

PHP

Instale PHP en su máquina local siguiendo los pasos de [Instalación y configuración](#) del sitio web de PHP.

Paso 1: Crea una aplicación PHP

En este ejemplo, creamos una aplicación PHP Hello World. Las aplicaciones PHP se pueden crear con una sobrecarga mínima.

Cree el directorio del proyecto.

```
~$ mkdir eb-php  
~$ cd eb-php
```

A continuación, cree un `index.php` archivo en el directorio del proyecto. Este archivo se publica de forma predeterminada cuando se ejecuta PHP.

```
~/eb-php/  
|-- index.php
```

Añada el siguiente contenido a su `index.php` archivo.

Example `~/eb-php/index.php`

```
echo "Hello Elastic Beanstalk! This is a PHP application.";
```

Paso 2: Ejecute la aplicación localmente

Ejecute el siguiente comando para ejecutar la aplicación localmente.

```
~/eb-php$ php -S localhost:5000
```

Introduzca la dirección URL `http://localhost:5000` en su navegador web. El navegador debería mostrar «¡Hola, Elastic Beanstalk! Se trata de una aplicación PHP».

Paso 3: Implemente la aplicación PHP con la CLI de EB

Ejecute los siguientes comandos para crear un entorno de Elastic Beanstalk para esta aplicación.

Para crear un entorno e implementar su aplicación PHP

1. Inicialice el repositorio de la CLI de EB con el comando `eb init`.

```
~/eb-php$ eb init -p php php-tutorial --region us-east-2
```

Este comando crea una aplicación con un nombre `php-tutorial` y configura su repositorio local para crear entornos con la última versión de la plataforma PHP.

2. (Opcional) Ejecute de nuevo `eb init` para configurar un par de claves predeterminadas de forma que pueda usar SSH para conectarse a la instancia EC2 donde se ejecuta la aplicación.

```
~/eb-php$ eb init  
Do you want to set up SSH for your instances?  
(y/n): y  
Select a keypair.  
1) my-keypair
```

```
2) [ Create new KeyPair ]
```

Seleccione un par de claves si ya tiene uno o siga las instrucciones para crear uno. Si no ve el símbolo del sistema o más adelante necesita cambiar la configuración, ejecute `eb init -i`.

3. Cree un entorno e implemente la aplicación en él con `eb create`. Elastic Beanstalk crea automáticamente un archivo zip para la aplicación y lo despliega en una instancia EC2 del entorno. Tras implementar la aplicación, Elastic Beanstalk la inicia en el puerto 5000.

```
~/eb-php$ eb create php-env
```

Elastic Beanstalk tarda unos cinco minutos en crear el entorno.

Paso 4: Ejecute la aplicación en Elastic Beanstalk

Cuando finalice el proceso de creación del entorno, abra su sitio web con `eb open`

```
~/eb-php$ eb open
```

¡Enhorabuena! ¡Ha implementado una aplicación PHP con Elastic Beanstalk! Se abre una ventana del navegador con el nombre de dominio creado para su aplicación.

Paso 5: Eliminar

Puede cerrar el entorno cuando termine de trabajar con la aplicación. Elastic Beanstalk AWS cancela todos los recursos asociados a su entorno.

Para finalizar el entorno de Elastic Beanstalk con la CLI de EB, ejecute el siguiente comando.

```
~/eb-php$ eb terminate
```

AWS recursos para su aplicación

Acabas de crear una aplicación de instancia única. Sirve como una aplicación de muestra sencilla con una sola instancia EC2, por lo que no requiere balanceo de carga ni escalado automático. Para las aplicaciones de instancia única, Elastic Beanstalk crea los siguientes recursos: AWS

- EC2 instance (Instancia EC2): máquina virtual de Amazon EC2 configurada para ejecutar aplicaciones web en la plataforma que elija.

Cada plataforma ejecuta un conjunto distinto de software, archivos de configuración y scripts compatibles con una determinada versión de lenguaje, marco y contenedor web (o una combinación de ellos). La mayoría de las plataformas utilizan Apache o nginx como un proxy inverso que procesa el tráfico web delante de la aplicación web, reenvía las solicitudes a esta, administra los recursos estáticos y genera registros de acceso y errores.

- Instance security group (Grupo de seguridad de la instancia): grupo de seguridad de Amazon EC2 configurado para permitir el tráfico entrante en el puerto 80. Este recurso permite que el tráfico HTTP procedente del balanceador de carga llegue a la instancia EC2 en la que se ejecuta la aplicación web. De forma predeterminada, el tráfico no está permitido en otros puertos.
- Bucket de Amazon S3: ubicación de almacenamiento para el código fuente, los registros y otros artefactos que se crean al utilizar Elastic Beanstalk.
- CloudWatch Alarmas de Amazon: dos CloudWatch alarmas que monitorean la carga de las instancias de su entorno y se activan si la carga es demasiado alta o demasiado baja. Cuando se activa una alarma, en respuesta, el grupo de Auto Scaling aumenta o reduce los recursos.
- AWS CloudFormation pila: Elastic AWS CloudFormation Beanstalk se utiliza para lanzar los recursos de su entorno y propagar los cambios de configuración. Los recursos se definen en una plantilla que puede verse en la [consola de AWS CloudFormation](#).
- Nombre de dominio: nombre de dominio que direcciona el tráfico a la aplicación web con el formato *subdominio.región.elasticbeanstalk.com*.

Elastic Beanstalk administra todos estos recursos. Cuando termina su entorno, Elastic Beanstalk termina todos los recursos que este contiene.

Siguientes pasos

Una vez que disponga de un entorno que ejecute una aplicación, podrá implementar una nueva versión de la aplicación o una aplicación distinta en cualquier momento. La implementación de una nueva versión de la aplicación es una tarea muy rápida, ya que no se requiere aprovisionar ni reiniciar instancias EC2. También puede explorar su nuevo entorno con la consola de Elastic Beanstalk. Para ver los pasos detallados, consulte [Explore su entorno](#) en el capítulo Introducción de esta guía.

Pruebe más tutoriales

Si desea probar otros tutoriales con diferentes aplicaciones de ejemplo, consulte [Más ejemplos de aplicaciones y tutoriales para PHP](#).

Una vez que haya implementado una o dos aplicaciones de ejemplo y esté listo para empezar a desarrollar y ejecutar aplicaciones PHP de forma local, consulte [Configuración del entorno de desarrollo de PHP](#).

Implemente con la consola de Elastic Beanstalk

También puede usar la consola de Elastic Beanstalk para iniciar la aplicación de muestra. Para ver los pasos detallados, consulte [Crear una aplicación de ejemplo](#) en el capítulo Introducción de esta guía.

Configuración del entorno de desarrollo de PHP

Configure el entorno de desarrollo de PHP para probar la aplicación localmente antes de implementarla en AWS Elastic Beanstalk. En este tema, se explican los pasos de configuración del entorno de desarrollo y se incluyen enlaces a páginas de instalación donde encontrará herramientas útiles.

Para ver procedimientos de configuración y herramientas comunes que se utilizan en todos los lenguajes, consulte [Configuración de su máquina de desarrollo](#).

Secciones

- [Instalación de PHP](#)
- [Instalación de Composer](#)
- [Instalación del SDK para PHP de AWS](#)
- [Instalación de un IDE o un editor de texto](#)

Instalación de PHP

Instale PHP y algunas extensiones comunes. Si no tiene ninguna preferencia, obtenga la versión más reciente. Los pasos pueden variar en función de la plataforma y el administrador de paquetes disponible.

En Amazon Linux, use yum:

```
$ sudo yum install php
$ sudo yum install php-mbstring
$ sudo yum install php-intl
```

Note

Para obtener versiones específicas del paquete PHP que coincidan con su [versión de la plataforma PHP](#) de Elastic Beanstalk, utilice el comando `yum search php` para buscar las versiones disponibles del paquete, por ejemplo, `php72`, `php72-mbstring` y `php72-intl`. A continuación, utilice `sudo yum install package` para instalarlas.

En Ubuntu, use apt:

```
$ sudo apt install php-all-dev
$ sudo apt install php-intl
$ sudo apt install php-mbstring
```

En OS-X, use brew:

```
$ brew install php
$ brew install php-intl
```

Note

Para obtener versiones específicas del paquete PHP que coincidan con su [versión de la plataforma PHP](#) de Elastic Beanstalk, consulte en [Fórmulas de Homebrew](#) las versiones disponibles de PHP, por ejemplo, `php@7.2`. A continuación, utilice `brew install package` para instalarlas.

En función de la versión, `php-intl` podría incluirse en el paquete PHP principal y no existir como paquete independiente.

En Windows 10, [instale Windows Subsystem for Linux](#) para obtener Ubuntu e instalar PHP con apt. En versiones anteriores, visite la página de descargas windows.php.net para obtener PHP y consulte [esta página](#) para obtener más información sobre las extensiones.

Después de instalar PHP, vuelva a abrir el terminal y ejecute `php --version` para garantizar que la nueva versión se ha instalado y es la opción predeterminada.

Instalación de Composer

Composer es un administrador de dependencias para PHP. Puede utilizarlo para instalar bibliotecas, realizar un seguimiento de las dependencias de la aplicación y generar proyectos para los marcos PHP populares.

Instale Composer con el script de PHP que encontrará en getcomposer.org.

```
$ curl -s https://getcomposer.org/installer | php
```

El instalador genera un archivo PHAR en el directorio actual. Mueva este archivo a una ubicación de PATH de su entorno para que pueda utilizarlo como archivo ejecutable.

```
$ mv composer.phar ~/.local/bin/composer
```

Instale las bibliotecas con el comando `require`.

```
$ composer require twig/twig
```

Composer agrega las bibliotecas que se instalan localmente al [composer.json archivo](#) del proyecto. Al implementar el código del proyecto, Elastic Beanstalk utiliza Composer para instalar las bibliotecas que se indican en este archivo en las instancias de aplicación del entorno.

Si tiene problemas al instalar Composer, consulte la [documentación de Composer](#).

Instalación del SDK para PHP de AWS

Si necesita administrar recursos de AWS en la aplicación, instale AWS SDK for PHP. Por ejemplo, con el SDK para PHP, puede utilizar Amazon DynamoDB (DynamoDB) para almacenar información de usuario y sesión sin crear una base de datos relacional.

Instale el SDK para PHP con Composer.

```
$ composer require aws/aws-sdk-php
```

Visite la [página de inicio de AWS SDK for PHP](#) si desea obtener más información e instrucciones de instalación.

Instalación de un IDE o un editor de texto

Los entornos de desarrollo integrado (IDE) cuentan con una amplia gama de características que facilitan el desarrollo de aplicaciones. Si no ha utilizado un IDE para el desarrollo de PHP, pruebe Eclipse Y PHPStorm y determine cuál es el que mejor se adapta a sus necesidades.

- [Instalación de Eclipse](#)
- [Instalación de PhpStorm](#)

Note

Un IDE podría agregar archivos a la carpeta de proyectos que tal vez no quiera confirmar en el control de código fuente. Para evitar confirmar estos archivos en el control de código fuente, utilice `.gitignore` o la característica equivalente de la herramienta de control de código fuente.

Si simplemente quiere comenzar a escribir el código y no necesita todas las características de un IDE, considere la posibilidad de [instalar Sublime Text](#).

Uso de la plataforma PHP de Elastic Beanstalk

AWS Elastic Beanstalk admite varias plataformas para diferentes versiones del lenguaje de programación PHP. Estas plataformas admiten aplicaciones web PHP que se pueden ejecutar solas o con Composer. Obtenga más información en [PHP](#) en el documento Plataformas de AWS Elastic Beanstalk .

Elastic Beanstalk proporciona [opciones de configuración](#) que se pueden usar para personalizar el software que se ejecuta en las instancias EC2 del entorno de Elastic Beanstalk. Puede [configurar las variables de entorno](#) que necesita la aplicación, habilitar la rotación de registros en Amazon S3, asignar carpetas en el origen de la aplicación que contengan archivos estáticos a rutas entregadas por el servidor proxy y establecer parámetros comunes de inicialización de PHP.

En la consola de Elastic Beanstalk hay opciones de configuración disponibles para [modificar la configuración de un entorno en ejecución](#). Para evitar perder la configuración del entorno cuando lo termina, puede usar las [configuraciones guardadas](#) para guardar la configuración y aplicarla posteriormente a otro entorno.

Para guardar la configuración en el código fuente, puede incluir [archivos de configuración](#). Los valores de configuración de los archivos de configuración se aplican cada vez que crea un entorno o que implementa la aplicación. También puede usar archivos de configuración para instalar paquetes, ejecutar scripts y llevar a cabo otras operaciones de personalización de instancias durante las implementaciones.

Si utiliza Composer, puede [incluir un archivo `composer.json`](#) en el paquete de código fuente para instalar paquetes durante la implementación.

Si desea definir valores de configuración avanzados en PHP y configurar ajustes de PHP que no están disponibles como opciones de configuración, puede [utilizar archivos de configuración y proporcionar un archivo INI](#) para ampliar o anular la configuración predeterminada que aplica Elastic Beanstalk o para instalar extensiones adicionales.

La configuración aplicada en la consola de Elastic Beanstalk anula la misma configuración en los archivos de configuración, si existe. Esto le permite tener la configuración predeterminada en los archivos de configuración y anularla con la configuración específica del entorno en la consola. Para obtener más información acerca de la prioridad y otros métodos para cambiar valores de configuración, consulte [Opciones de configuración](#).

Para obtener más información sobre las diversas formas en las que puede ampliar una plataforma Elastic Beanstalk basada en Linux, consulte [the section called “Ampliación de plataformas Linux”](#).

Consideraciones sobre PHP 8.1 en Amazon Linux 2

Si utiliza la ramificación de la plataforma PHP 8.1 en Amazon Linux 2, lea esta sección.

Consideraciones sobre PHP 8.1 en Amazon Linux 2

Note

La información de este tema solo se aplica a la ramificación de la plataforma PHP 8.1 en Amazon Linux 2. No se aplica a las ramificaciones de la plataforma PHP basadas en AL2023. Tampoco se aplica a la ramificación de la plataforma PHP 8.0 en Amazon Linux 2.

Elastic Beanstalk almacena los paquetes de RPM relacionados con PHP 8.1 para la ramificación de la plataforma PHP 8.1 en Amazon Linux 2 en las instancias de EC2 en un directorio local, en lugar del repositorio de Amazon Linux. Se puede utilizar `rpm -i` para instalar los paquetes. A partir de la

[versión 3.5.0 de la plataforma PHP 8.1](#), Elastic Beanstalk almacena los paquetes RPM relacionados con PHP 8.1 en el siguiente directorio local de EC2.

```
/opt/elasticbeanstalk/RPMS
```

En el siguiente ejemplo se instala el paquete php-debuginfo.

```
$rpm -i /opt/elasticbeanstalk/RPMS/php-debuginfo-8.1.8-1.amzn2.x86_64.rpm
```

La versión del nombre del paquete variará según la versión real que aparezca en el directorio local de EC2 `/opt/elasticbeanstalk/RPMS`. Utilice la misma sintaxis para instalar otros paquetes PHP 8.1 RPM.

Amplíe la siguiente sección para mostrar una lista de los paquetes RPM que ofrecemos.

Paquetes RPM

La siguiente lista proporciona los paquetes de RMP que ofrece la plataforma PHP 8.1 de Elastic Beanstalk en Amazon Linux 2. Estos archivos están ubicados en el directorio `/opt/elasticbeanstalk/RPMS`.

Los números de versión 8.1.8-1 y 3.7.0-1 de los nombres de paquetes enumerados son solo un ejemplo.

- php-8.1.8-1.amzn2.x86_64.rpm
- php-bcmath-8.1.8-1.amzn2.x86_64.rpm
- php-cli-8.1.8-1.amzn2.x86_64.rpm
- php-common-8.1.8-1.amzn2.x86_64.rpm
- php-dba-8.1.8-1.amzn2.x86_64.rpm
- php-dbg-8.1.8-1.amzn2.x86_64.rpm
- php-debuginfo-8.1.8-1.amzn2.x86_64.rpm
- php-devel-8.1.8-1.amzn2.x86_64.rpm
- php-embedded-8.1.8-1.amzn2.x86_64.rpm
- php-enchant-8.1.8-1.amzn2.x86_64.rpm
- php-fpm-8.1.8-1.amzn2.x86_64.rpm
- php-gd-8.1.8-1.amzn2.x86_64.rpm

- php-gmp-8.1.8-1.amzn2.x86_64.rpm
- php-intl-8.1.8-1.amzn2.x86_64.rpm
- php-ldap-8.1.8-1.amzn2.x86_64.rpm
- php-mbstring-8.1.8-1.amzn2.x86_64.rpm
- php-mysqlnd-8.1.8-1.amzn2.x86_64.rpm
- php-odbc-8.1.8-1.amzn2.x86_64.rpm
- php-opcache-8.1.8-1.amzn2.x86_64.rpm
- php-pdo-8.1.8-1.amzn2.x86_64.rpm
- php-pear-1.10.13-1.amzn2.noarch.rpm
- php-pgsql-8.1.8-1.amzn2.x86_64.rpm
- php-process-8.1.8-1.amzn2.x86_64.rpm
- php-pspell-8.1.8-1.amzn2.x86_64.rpm
- php-snmp-8.1.8-1.amzn2.x86_64.rpm
- php-soap-8.1.8-1.amzn2.x86_64.rpm
- php-sodium-8.1.8-1.amzn2.x86_64.rpm
- php-xml-8.1.8-1.amzn2.x86_64.rpm
- php-pecl-imagick-3.7.0-1.amzn2.x86_64.rpm
- php-pecl-imagick-debuginfo-3.7.0-1.amzn2.x86_64.rpm
- php-pecl-imagick-devel-3.7.0-1.amzn2.noarch.rpm

Puede utilizar los paquetes PEAR y PECL para instalar extensiones comunes. Para obtener más información sobre PEAR, consulte el sitio web del [repositorio de aplicaciones y extensiones PHP de PEAR](#). Para obtener más información sobre PECL, consulte el sitio web de la [extensión PECL](#).

Los siguientes comandos de ejemplo instalan las extensiones de Memcached.

```
$pecl install memcache
```

O también puede utilizar lo siguiente:

```
$pear install pecl/memcache
```

Los siguientes comandos de ejemplo instalan las extensiones de Redis.

```
$pecl install redis
```

O también puede utilizar lo siguiente:

```
$pear install pecl/redis
```

Configuración del entorno PHP

Puede utilizar la consola de Elastic Beanstalk para habilitar la rotación de registros en Amazon S3, configurar variables que la aplicación pueda leer desde el entorno y cambiar la configuración de PHP.

Para configurar su entorno PHP en la consola de Elastic Beanstalk

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Actualizaciones, supervisión y registro, seleccione Editar.

Configuración de PHP

- Proxy server (Servidor proxy): el servidor proxy que utilizar en las instancias del entorno. De forma predeterminada, se utiliza nginx.
- Document root (Raíz del documento): carpeta que contiene la página predeterminada del sitio. Si la página de bienvenida no está en la raíz del paquete de código fuente, especifique la carpeta donde se encuentre con respecto a la ruta raíz. Por ejemplo, /public si la página de bienvenida está en una carpeta llamada public.
- Memory limit (Límite de memoria): cantidad máxima de memoria que un script puede asignar. Por ejemplo, 512M.
- Zlib output compression (Compresión de salida Zlib): establézcalo en On para comprimir las respuestas.

- **Allow URL fopen (Permitir fopen de URL):** establézcalo en `Off` para impedir que los scripts descarguen archivos de ubicaciones remotas.
- **Display errors (Mostrar errores):** establézcalo en `On` para mostrar mensajes de error internos sobre depuración.
- **Max execution time (Tiempo máximo de ejecución):** tiempo máximo en segundos que un script puede ejecutarse antes de que el entorno lo termine.

Log Options (Opciones de registro)

La sección Log Options tiene dos valores:

- **Instance profile (Perfil de instancia):** especifica el perfil de instancia que tiene permiso para obtener acceso al bucket de Amazon S3 asociado con la aplicación.
- **Enable log file rotation to Amazon S3 (Habilitar la rotación de archivos de registro para Amazon S3):** especifica si los archivos de registro de las instancias de Amazon EC2 de la aplicación se copian en el bucket de Amazon S3 asociado a la aplicación.

Archivos estáticos

Para mejorar el rendimiento, puede utilizar la sección Static files (Archivos estáticos) para configurar el servidor proxy para entregar archivos estáticos de servidor (por ejemplo, HTML o imágenes) desde un conjunto de directorios dentro de su aplicación web. Para cada directorio, se establece la ruta virtual para la asignación de directorios. Cuando el servidor proxy recibe una solicitud de un archivo en la ruta especificada, proporciona directamente el archivo en lugar de direccionar la solicitud a la aplicación.

Para obtener más información sobre la configuración de archivos estáticos mediante archivos de configuración o la consola de Elastic Beanstalk, consulte [the section called “Archivos estáticos”](#).

Propiedades del entorno

La sección Environment Properties (Propiedades de entorno) le permite especificar opciones de configuración del entorno en las instancias de Amazon EC2 que ejecutan la aplicación. Estos ajustes se pasan como pares clave-valor a la aplicación.

El código de la aplicación puede acceder a las propiedades del entorno mediante `$_SERVER` o la función `get_cfg_var`.


```
$endpoint = $_SERVER['API_ENDPOINT'];
```

Para obtener más información, consulte [Propiedades del entorno y otras opciones del software](#).

El espacio de nombres `aws:elasticbeanstalk:container:php:phpini`

Puede usar un [archivo de configuración](#) para definir opciones de configuración y realizar otras tareas de configuración en las instancias durante las implementaciones. Las opciones de configuración se pueden definir a través del servicio de Elastic Beanstalk o la plataforma que utilice y están organizadas por espacios de nombres.

Puede utilizar el espacio de nombres `aws:elasticbeanstalk:environment:proxy` para elegir el servidor proxy del entorno.

Puede usar el espacio de nombres

`aws:elasticbeanstalk:environment:proxy:staticfiles` para configurar el proxy del entorno para que sirva archivos estáticos. Defina mapeos de rutas virtuales a directorios de aplicaciones.

La plataforma PHP define opciones en el espacio de

`aws:elasticbeanstalk:container:php:phpini` nombres, incluida una que no está disponible en la consola de Elastic Beanstalk. `composer_options` establece las opciones personalizadas que se utilizarán al instalar dependencias mediante Composer a través de `composer.phar install`. Para obtener más información, incluidas las opciones disponibles, visite <http://getcomposer.org/doc/03-cli.md#install>.

El siguiente [archivo de configuración](#) de ejemplo especifica una opción de archivos estáticos que asigna un directorio llamado `staticimages` a la ruta `/images` y muestra la configuración de cada una de las opciones disponibles en el espacio de nombres

`aws:elasticbeanstalk:container:php:phpini`:

Example `.ebextensions/php-settings.config`

```
option_settings:
  aws:elasticbeanstalk:environment:proxy:
    ProxyServer: apache
  aws:elasticbeanstalk:environment:proxy:staticfiles:
    /images: staticimages
  aws:elasticbeanstalk:container:php:phpini:
    document_root: /public
```

```
memory_limit: 128M
zlib.output_compression: "Off"
allow_url_fopen: "On"
display_errors: "Off"
max_execution_time: 60
composer_options: vendor/package
```

Note

El espacio de nombres `aws:elasticbeanstalk:environment:proxy:staticfiles` no está definido en las ramas de la plataforma PHP de la AMI de Amazon Linux (antes de Amazon Linux 2).

Elastic Beanstalk cuenta con numerosas opciones de configuración para personalizar el entorno. Además de los archivos de configuración, también puede definir opciones en la consola, configuraciones guardadas, la CLI de EB o la AWS CLI. Para obtener más información, consulte [Opciones de configuración](#).

Instalación de las dependencias de su aplicación

La aplicación podría tener dependencias en otros paquetes PHP. Puede configurar su aplicación para instalar estas dependencias en las instancias de Amazon Elastic Compute Cloud (Amazon EC2) del entorno. Como alternativa, puede incluir las dependencias de la aplicación en el paquete de código fuente e implementarlas con la aplicación. En la siguiente sección se explican ambas formas.

Utilizar un archivo de Composer para instalar dependencias en instancias

Utilice un archivo `composer.json` en la raíz del código fuente del proyecto si desea utilizar Composer para instalar los paquetes que necesita las instancias de Amazon EC2 del entorno.

Example `composer.json`

```
{
  "require": {
    "monolog/monolog": "1.0.*"
  }
}
```

Cuando hay un archivo `composer.json`, Elastic Beanstalk ejecuta `composer.phar install` para instalar las dependencias. Puede agregar otras opciones al

comando estableciendo la opción [composer_options](#) del espacio de nombres `aws:elasticbeanstalk:container:php:phpini`.

Incluir dependencias en el paquete de código fuente

Si la aplicación tiene un gran número de dependencias, instalarlas puede tardar mucho. Esto puede aumentar la implementación y las operaciones de escalado, ya que las dependencias se instalan en cada nueva instancia.

Para evitar el impacto negativo en el momento de la implementación, utilice Composer en su entorno de desarrollo para resolver dependencias e instalarlas en la carpeta `vendor`.

Para incluir dependencias en el paquete de código fuente de la aplicación

1. Ejecute el siguiente comando:

```
% composer install
```

2. Incluya la carpeta `vendor` generada en la raíz del paquete de código fuente de la aplicación.

Cuando Elastic Beanstalk encuentra una carpeta `vendor` en la instancia, no tiene en cuenta el archivo `composer.json` (aunque exista). A continuación, su aplicación utiliza dependencias de la carpeta `vendor`.

Actualización de Composer

Puede que tenga que actualizar Composer si ve un error al intentar instalar paquetes con un archivo de Composer o si no puede utilizar la versión más reciente de la plataforma. Entre las actualizaciones de la plataforma, puede actualizar Composer en las instancias de su entorno mediante el uso de los archivos de configuración de su [.ebextensions](#) carpeta.

Puede actualizar Composer automáticamente con la siguiente configuración.

```
commands:
  01updateComposer:
    command: /usr/bin/composer.phar self-update 2.7.0
```

La siguiente [configuración de opción](#) establece la variable de `COMPOSER_HOME` entorno, que configura la ubicación de la memoria caché de Composer.

```
option_settings:
```

```
- namespace: aws:elasticbeanstalk:application:environment
  option_name: COMPOSER_HOME
  value: /home/webapp/composer-home
```

Puede combinar ambas opciones en el mismo archivo de configuración de su `.ebextensions` carpeta.

Example `.ebextensions/composer.config`

```
commands:
  01updateComposer:
    command: /usr/bin/composer.phar self-update 2.7.0

option_settings:
  - namespace: aws:elasticbeanstalk:application:environment
    option_name: COMPOSER_HOME
    value: /home/webapp/composer-home
```

Note

Debido a las actualizaciones de la instalación de Composer en la versión AL2023 del [22 de febrero de 2024](#) y en la versión de la plataforma AL2 del [28 de febrero de 2024](#), la actualización automática de Composer puede fallar si `COMPOSER_HOME` se configura cuando se ejecuta la actualización automática.

No se ejecutarán los siguientes comandos combinados: `export COMPOSER_HOME=/home/webapp/composer-home && /usr/bin/composer.phar self-update 2.7.0`. Sin embargo, el ejemplo anterior funcionará. En el ejemplo anterior, la opción configurada para no se `COMPOSER_HOME` transferirá a la `01updateComposer` ejecución y tampoco se establecerá cuando se ejecute el comando `self-update`.

Important

Si omite el número de versión en el comando `composer.phar self-update`, Composer actualizará a la última versión disponible cada vez que implemente el código fuente y cuando Auto Scaling aprovisiona nuevas instancias. Esto podría provocar errores en las operaciones de escalado y las implementaciones si se publica una versión de Composer que no es compatible con la aplicación.

Para obtener más información sobre las plataformas PHP Elastic Beanstalk, incluida la versión de Composer, consulte las [versiones de la plataforma PHP](#) en el documento Plataformas de AWS Elastic Beanstalk .

Ampliación de php.ini

Utilice un archivo de configuración con un bloque `files` para agregar un archivo `.ini` a `/etc/php.d/` en las instancias del entorno. El principal archivo de configuración, `php.ini`, extrae la configuración de los archivos de esta carpeta en orden alfabético. Muchas extensiones están habilitadas de forma predeterminada por los archivos de esta carpeta.

Example `.ebextensions/mongo.config`

```
files:
  "/etc/php.d/99mongo.ini":
    mode: "000755"
    owner: root
    group: root
    content: |
      extension=mongo.so
```

Más ejemplos de aplicaciones y tutoriales para PHP

Para empezar a utilizar las aplicaciones PHP AWS Elastic Beanstalk, lo único que necesita es un [paquete de código fuente](#) de la aplicación para cargarlo como primera versión de la aplicación e implementarlo en un entorno. En [QuickStart para PHP](#) este tema se explica cómo iniciar la aplicación PHP de ejemplo con la CLI de EB. En esta sección se proporcionan tutoriales más detallados.

Tutoriales de PHP

- [Implementación de una aplicación de Laravel en Elastic Beanstalk](#)
- [Implementación de una aplicación de CakePHP en Elastic Beanstalk](#)
- [Implementación de una aplicación Symfony en Elastic Beanstalk](#)
- [Implementación de una aplicación PHP de alta disponibilidad con una base de datos externa de Amazon RDS en Elastic Beanstalk](#)
- [Implementación de un WordPress sitio web de alta disponibilidad con una base de datos externa de Amazon RDS en Elastic Beanstalk](#)
- [Implementación de un sitio web de Drupal de alta disponibilidad con una base de datos externa de Amazon RDS en Elastic Beanstalk](#)

- [Agregar una instancia de base de datos de Amazon RDS a un entorno de aplicaciones PHP](#)

Implementación de una aplicación de Laravel en Elastic Beanstalk

Laravel es un framework de código abierto model-view-controller (MVC) para PHP. En este tutorial, se explica el proceso de generación de una aplicación Laravel, su despliegue en un AWS Elastic Beanstalk entorno y su configuración para que se conecte a una instancia de base de datos de Amazon Relational Database Service (Amazon RDS).

Secciones

- [Requisitos previos](#)
- [Lance un entorno de Elastic Beanstalk](#)
- [Instalación de Laravel y creación de un sitio web](#)
- [Implementar la aplicación](#)
- [Configuración de los ajustes de Composer](#)
- [Agregar una base de datos al entorno](#)
- [Limpieza](#)
- [Siguiendo los pasos](#)

Requisitos previos

Este tutorial asume que tiene conocimiento de las operaciones básicas de Elastic Beanstalk y de la consola de Elastic Beanstalk. Si aún no lo ha hecho, siga las instrucciones que se indican en [Introducción a Elastic Beanstalk](#) para lanzar su primer entorno de Elastic Beanstalk.

Para seguir los procedimientos de esta guía, necesitará un shell o un terminal de línea de comando donde pueda ejecutar los comandos. Los comandos aparecen en listas y van precedidos del símbolo del sistema (\$) y del nombre del directorio actual, si es aplicable.

```
~/eb-project$ this is a command  
this is output
```

En Linux y macOS, puede utilizar el administrador de shell y paquetes preferido. En Windows, puede [instalar el subsistema de Windows para Linux para](#) obtener una versión de Ubuntu y Bash integrada en Windows.

Laravel 6 requiere PHP 7.2 o posterior. También requiere las extensiones de PHP enumeradas en el tema [requisitos del servidor](#) en la documentación oficial de Laravel. Siga las instrucciones del tema [Configuración del entorno de desarrollo de PHP](#) para instalar PHP y Composer.

Para obtener información sobre soporte y mantenimiento de Laravel, consulte el tema de la [política de soporte](#) en la documentación oficial de Laravel.

Lance un entorno de Elastic Beanstalk

Utilice la consola de Elastic Beanstalk para crear un entorno de Elastic Beanstalk. Elija la plataforma PHP y acepte la configuración predeterminada y el código de muestra.

Para lanzar un entorno (consola)

1. [Abra la consola de Elastic Beanstalk mediante este enlace preconfigurado: console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=Tutorials&EnvironmentType=LoadBalanced](https://console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=Tutorials&EnvironmentType=LoadBalanced)
2. En Platform (Plataforma), seleccione la plataforma y la ramificación de plataforma que coincidan con el idioma utilizado por la aplicación.
3. En Application code (Código de aplicación), seleccione Sample application (Aplicación de muestra).
4. Elija Review and launch (Revisar y lanzar).
5. Revise las opciones disponibles. Elija la opción disponible que quiera usar y, cuando esté listo, elija Create app (Crear aplicación).

El entorno tarda unos cinco minutos en crearse e incluye los recursos siguientes:

- Instancia de EC2: una máquina virtual de Amazon Elastic Compute Cloud (Amazon EC2) configurada para ejecutar aplicaciones web en la plataforma que elija.

Cada plataforma ejecuta un conjunto específico de software, archivos de configuración y scripts compatibles con una determinada versión de lenguaje, marco de trabajo y contenedor web (o una combinación de estos). La mayoría de las plataformas utilizan Apache o nginx como un proxy inverso que se sitúa delante de la aplicación web, reenvía las solicitudes a esta, administra los recursos estáticos y genera registros de acceso y errores.

- Instance security group (Grupo de seguridad de la instancia): grupo de seguridad de Amazon EC2 configurado para permitir el tráfico entrante en el puerto 80. Este recurso permite que el tráfico

HTTP procedente del balanceador de carga llegue a la instancia EC2 en la que se ejecuta la aplicación web. De forma predeterminada, el tráfico no está permitido en otros puertos.

- Balanceador de carga: balanceador de carga de Elastic Load Balancing configurado para distribuir solicitudes a las instancias que se ejecutan en la aplicación. Los balanceadores de carga también permiten que las instancias no estén expuestas directamente a Internet.
- Grupo de seguridad del balanceador de carga: grupo de seguridad de Amazon EC2 configurado para permitir el tráfico entrante en el puerto 80. Este recurso permite que el tráfico HTTP procedente de Internet llegue al balanceador de carga. De forma predeterminada, el tráfico no está permitido en otros puertos.
- Grupo de Auto Scaling: grupo de Auto Scaling configurado para reemplazar una instancia si termina o deja de estar disponible.
- Bucket de Amazon S3: ubicación de almacenamiento para el código fuente, los registros y otros artefactos que se crean al utilizar Elastic Beanstalk.
- CloudWatch Alarmas de Amazon: dos CloudWatch alarmas que monitorean la carga de las instancias de su entorno y que se activan si la carga es demasiado alta o demasiado baja. Cuando se activa una alarma, en respuesta, el grupo de Auto Scaling aumenta o reduce los recursos.
- AWS CloudFormation pila: Elastic AWS CloudFormation Beanstalk se utiliza para lanzar los recursos de su entorno y propagar los cambios de configuración. Los recursos se definen en una plantilla que puede verse en la [consola de AWS CloudFormation](#).
- Nombre de dominio: nombre de dominio que direcciona el tráfico a la aplicación web con el formato *subdominio.región.elasticbeanstalk.com*.

Note

Para aumentar la seguridad de las aplicaciones de Elastic Beanstalk, el dominio `elasticbeanstalk.com` está registrado en la [lista de sufijos públicos \(PSL\)](#). Para mayor seguridad, se recomienda que utilice cookies con un prefijo `__Host-` en caso de que necesite configurar cookies confidenciales en el nombre de dominio predeterminado de sus aplicaciones de Elastic Beanstalk. Esta práctica le ayudará a defender su dominio contra intentos de falsificación de solicitudes entre sitios (CSRF). Para obtener más información, consulte la página de [configuración de cookies](#) en la red de desarrolladores de Mozilla.

Todos estos recursos los administra Elastic Beanstalk. Cuando termina su entorno, Elastic Beanstalk termina todos los recursos que este contiene.

Note

El bucket de Amazon S3 que Elastic Beanstalk crea se comparte entre los entornos y no se elimina durante la terminación del entorno. Para obtener más información, consulte [Usar Elastic Beanstalk con Amazon S3](#).

Instalación de Laravel y creación de un sitio web

Composer puede instalar Laravel y crear un proyecto de trabajo con un único comando:

```
~$ composer create-project --prefer-dist laravel/laravel eb-laravel
```

Composer instala Laravel y sus dependencias, y genera un proyecto predeterminado.

Si tiene algún problema al instalar Laravel, vaya al tema de instalación en la documentación oficial: <https://laravel.com/docs/6.x>.

Implementar la aplicación

Cree un [paquete de código fuente](#) que contenga los archivos creados por Composer. El comando siguiente crea un paquete de código fuente denominado `laravel-default.zip`. Excluye los archivos de la carpeta `vendor`, que ocupan mucho espacio y no se requieren para implementar la aplicación en Elastic Beanstalk.

```
~/eb-laravel$ zip ../laravel-default.zip -r * .[^.]* -x "vendor/*"
```

Cargue el paquete de origen en Elastic Beanstalk para implementar Laravel en su entorno.

Para implementar un paquete de código fuente

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

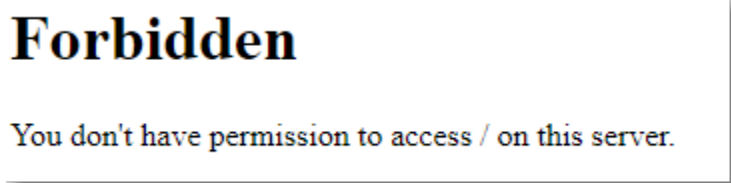
3. En la página de información general del entorno, elija Upload and deploy (Cargar e implementar).
4. Utilice el cuadro de diálogo en pantalla para cargar el paquete de código fuente.
5. Elija Deploy (Implementar).
6. Cuando se complete la implementación, seleccione la URL del para abrir el sitio web en una nueva pestaña.

Note

Para optimizar aun más el paquete de código fuente, inicialice un repositorio Git y use el [comando git archive](#) para crear el paquete de código fuente. El proyecto Laravel predeterminado incluye un archivo `.gitignore` que indica a Git que debe excluir la carpeta `vendor` y los demás archivos que no se requieren para la implementación.

Configuración de los ajustes de Composer

Cuando la implementación se complete, haga clic en la URL para abrir la aplicación de Laravel en el navegador:



Forbidden

You don't have permission to access / on this server.

¿Qué es esto? De forma predeterminada, Elastic Beanstalk proporciona la raíz del proyecto en la ruta raíz del sitio web. En este caso, sin embargo, la página predeterminada (`index.php`) está un nivel por debajo de la carpeta `public`. Puede comprobarlo agregando `/public` a la URL. Por ejemplo, <http://laravel.us-east-2.elasticbeanstalk.com/public>.

Para que la aplicación Laravel esté disponible en la ruta raíz, utilice la consola Elastic Beanstalk con el fin de configurar la raíz del documento del el sitio web.

Para configurar la raíz de documentos del sitio web

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Actualizaciones, supervisión y registro, seleccione Editar.
5. En Document Root (Raíz del documento), introduzca **/public**.
6. Para guardar los cambios, elija Aplicar en la parte inferior de la página.
7. Cuando la actualización se haya completado, haga clic en la URL para volver a abrir el sitio en el navegador.



Hasta el momento, todo fue bien. A continuación, debe agregar una base de datos al entorno y configurar Laravel para que se conecte a ella.

Agregar una base de datos al entorno

Inicie una instancia de base de datos de RDS en su entorno de Elastic Beanstalk. Puede utilizar bases de datos de MySQL, SQLServer o PostgreSQL con Laravel en Elastic Beanstalk. En este ejemplo, utilizaremos MySQL.

Para agregar una instancia de base de datos de RDS a su entorno Elastic Beanstalk

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Database (Base de datos), seleccione Edit (Editar).
5. En Engine (Motor), seleccione mysql.
6. Escriba un nombre de usuario y contraseña maestros. Elastic Beanstalk proporcionará estos valores a la aplicación a través de propiedades de entorno.
7. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

Se tarda alrededor de 10 minutos en crear una instancia de base de datos. Para obtener más información sobre las bases de datos acopladas a un entorno de Elastic Beanstalk, consulte [Añadir una base de datos al entorno de Elastic Beanstalk](#).

Mientras tanto, puede actualizar el código fuente para que pueda leer la información de conexión del entorno. Elastic Beanstalk proporciona la información de la conexión a través de variables de entorno, como RDS_HOSTNAME, a las que puede obtener acceso desde la aplicación.

La configuración de base de datos de Laravel se guarda en un archivo llamado `database.php` en la carpeta `config` del proyecto de código. Busque la entrada `mysql` y modifique las variables `host`, `database`, `username` y `password` de forma que lean los valores correspondientes en Elastic Beanstalk:

Example `~/Eb-laravel/config/database.php`

```
...
'connections' => [

    'sqlite' => [
        'driver' => 'sqlite',
        'database' => env('DB_DATABASE', database_path('database.sqlite')),
        'prefix' => '',
```

```

    ],

    'mysql' => [
        'driver' => 'mysql',
        'host' => env('RDS_HOSTNAME', '127.0.0.1'),
        'port' => env('RDS_PORT', '3306'),
        'database' => env('RDS_DB_NAME', 'forge'),
        'username' => env('RDS_USERNAME', 'forge'),
        'password' => env('RDS_PASSWORD', ''),
        'unix_socket' => env('DB_SOCKET', ''),
        'charset' => 'utf8mb4',
        'collation' => 'utf8mb4_unicode_ci',
        'prefix' => '',
        'strict' => true,
        'engine' => null,
    ],

    ...

```

Para comprobar que la conexión de la base de datos está configurada correctamente, agregue código a `index.php` para conectarse a la base de datos y también para la respuesta predeterminada:

Example `~/Eb-laravel/public/index.php`

```

...
if(DB::connection()->getDatabaseName())
{
    echo "Connected to database ".DB::connection()->getDatabaseName();
}
$response->send();
...

```

Cuando finalice el lanzamiento de la instancia de base de datos, empaquete e implemente la aplicación actualizada en el entorno.

Para actualizar su entorno de Elastic Beanstalk

1. Cree un nuevo paquete de código fuente:

```
~/eb-laravel$ zip ../laravel-v2-rds.zip -r * .[^.]* -x "vendor/*"
```

2. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS

3. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

4. Elija Upload and Deploy (Cargar e implementar).
5. Seleccione Browse (Examinar) y cargue `laravel-v2-rds.zip`.
6. Elija Implementar.

La implementación de una nueva versión de la aplicación tarda menos de un minuto. Cuando se haya completado, actualice la página web de nuevo para comprobar que la conexión con la base de datos es correcta:

Connected to database ebdb

Laravel

DOCUMENTATION

LARACASTS

NEWS

FORGE

GITHUB

Limpieza

Cuando termine de trabajar con Elastic Beanstalk, puede terminar su entorno. [Elastic Beanstalk AWS cancela todos los recursos asociados al entorno, como las instancias de Amazon EC2, las instancias de bases de datos, los balanceadores de carga, los grupos de seguridad y las alarmas.](#)

Para cerrar el entorno de Elastic Beanstalk desde la consola

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. Elija Actions (Acciones) y, a continuación, Terminate Environment (Terminar el entorno).
4. Utilice el cuadro de diálogo en pantalla para confirmar la terminación del entorno.

Con Elastic Beanstalk, puede crear fácilmente un nuevo entorno para su aplicación en cualquier momento.

Además, puede terminar los recursos de base de datos creados fuera del entorno de Elastic Beanstalk. Cuando termine una instancia de base de datos de Amazon RDS, puede hacer una instantánea para restaurar los datos posteriormente en otra instancia.

Para terminar la instancia de base de datos de RDS

1. Abra la [consola de Amazon RDS](#).
2. Seleccione Databases (Bases de datos).
3. Seleccione la instancia de base de datos.
4. Elija Acciones y, a continuación, elija Eliminar.
5. Decida si desea crear una instantánea y seleccione Delete (Eliminar).

Siguientes pasos

Para obtener más información sobre Laravel, visite el sitio web oficial de Laravel en laravel.com.

A medida que siga desarrollando la aplicación, es probable que desee contar con un mecanismo que le permita administrar los entornos e implementar la aplicación sin tener que crear manualmente un archivo .zip y cargarlo en la consola de Elastic Beanstalk. La interfaz de [línea de comandos de Elastic Beanstalk](#) (EB CLI easy-to-use) proporciona comandos para crear, configurar e implementar aplicaciones en entornos de Elastic Beanstalk desde la línea de comandos.

En este tutorial, ha utilizado la consola de Elastic Beanstalk para configurar las opciones de Composer. Para integrar esta configuración en el código fuente de la aplicación, puede utilizar un archivo de configuración similar al siguiente.

Example .ebextensions/composer.config

```
option_settings:
  aws:elasticbeanstalk:container:php:phpini:
    document_root: /public
```

Para obtener más información, consulte [Personalización avanzada de entornos con archivos de configuración \(.ebextensions\)](#).

Ejecutar una instancia de base de datos de Amazon RDS; en el entorno de Elastic Beanstalk es ideal para desarrollo y pruebas, pero supedita el ciclo de vida de la base de datos al entorno. Consulte [Agregar una instancia de base de datos de Amazon RDS a un entorno de aplicaciones PHP](#) para obtener instrucciones acerca de cómo conectarse a una base de datos que se ejecute fuera del entorno.

Por último, si pretende utilizar la aplicación en un entorno de producción, querrá [configurar un nombre de dominio personalizado](#) para el entorno y [habilitar HTTPS](#) para las conexiones seguras.

Implementación de una aplicación de CakePHP en Elastic Beanstalk

CakePHP es un marco MVC de código abierto para PHP. En este tutorial, se explica paso a paso cómo se genera un proyecto de CakePHP, cómo se implementa en un entorno de Elastic Beanstalk y cómo se configura para conectarlo a una instancia de base de datos de Amazon RDS.

Secciones

- [Requisitos previos](#)
- [Lance un entorno de Elastic Beanstalk](#)
- [Instalación de CakePHP y creación de un sitio web](#)
- [Implementar la aplicación](#)
- [Agregar una base de datos al entorno](#)
- [Limpieza](#)
- [Siguiendo pasos](#)

Requisitos previos

Este tutorial asume que tiene conocimiento de las operaciones básicas de Elastic Beanstalk y de la consola de Elastic Beanstalk. Si aún no lo ha hecho, siga las instrucciones que se indican en [Introducción a Elastic Beanstalk](#) para lanzar su primer entorno de Elastic Beanstalk.

Para seguir los procedimientos de esta guía, necesitará un shell o un terminal de línea de comando donde pueda ejecutar los comandos. Los comandos aparecen en listas y van precedidos del símbolo del sistema (\$) y del nombre del directorio actual, si es aplicable.

```
~/eb-project$ this is a command  
this is output
```

En Linux y macOS, puede utilizar el administrador de shell y paquetes preferido. En Windows, puede [instalar el subsistema de Windows para Linux para](#) obtener una versión de Ubuntu y Bash integrada en Windows.

CakePHP 4 requiere PHP 7.2 o posterior. También requiere las extensiones de PHP enumeradas en la documentación oficial de [instalación de CakePHP](#). Siga las instrucciones del tema [Configuración del entorno de desarrollo de PHP](#) para instalar PHP y Composer.

Lance un entorno de Elastic Beanstalk

Utilice la consola de Elastic Beanstalk para crear un entorno de Elastic Beanstalk. Elija la plataforma PHP y acepte la configuración predeterminada y el código de muestra.

Para lanzar un entorno (consola)

1. [Abra la consola de Elastic Beanstalk mediante este enlace preconfigurado:](https://console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=Tutorials&EnvironmentType=LoadBalanced)
console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=Tutorials&EnvironmentType=LoadBalanced
2. En Platform (Plataforma), seleccione la plataforma y la ramificación de plataforma que coincidan con el idioma utilizado por la aplicación.
3. En Application code (Código de aplicación), seleccione Sample application (Aplicación de muestra).
4. Elija Review and launch (Revisar y lanzar).
5. Revise las opciones disponibles. Elija la opción disponible que quiera usar y, cuando esté listo, elija Create app (Crear aplicación).

El entorno tarda unos cinco minutos en crearse e incluye los recursos siguientes:

- Instancia de EC2: una máquina virtual de Amazon Elastic Compute Cloud (Amazon EC2) configurada para ejecutar aplicaciones web en la plataforma que elija.

Cada plataforma ejecuta un conjunto específico de software, archivos de configuración y scripts compatibles con una determinada versión de lenguaje, marco de trabajo y contenedor web (o una combinación de estos). La mayoría de las plataformas utilizan Apache o nginx como un proxy inverso que se sitúa delante de la aplicación web, reenvía las solicitudes a esta, administra los recursos estáticos y genera registros de acceso y errores.

- Instance security group (Grupo de seguridad de la instancia): grupo de seguridad de Amazon EC2 configurado para permitir el tráfico entrante en el puerto 80. Este recurso permite que el tráfico HTTP procedente del balanceador de carga llegue a la instancia EC2 en la que se ejecuta la aplicación web. De forma predeterminada, el tráfico no está permitido en otros puertos.
- Balanceador de carga: balanceador de carga de Elastic Load Balancing configurado para distribuir solicitudes a las instancias que se ejecutan en la aplicación. Los balanceadores de carga también permiten que las instancias no estén expuestas directamente a Internet.
- Grupo de seguridad del balanceador de carga: grupo de seguridad de Amazon EC2 configurado para permitir el tráfico entrante en el puerto 80. Este recurso permite que el tráfico HTTP procedente de Internet llegue al balanceador de carga. De forma predeterminada, el tráfico no está permitido en otros puertos.
- Grupo de Auto Scaling: grupo de Auto Scaling configurado para reemplazar una instancia si termina o deja de estar disponible.
- Bucket de Amazon S3: ubicación de almacenamiento para el código fuente, los registros y otros artefactos que se crean al utilizar Elastic Beanstalk.
- CloudWatch Alarmas de Amazon: dos CloudWatch alarmas que monitorean la carga de las instancias de su entorno y que se activan si la carga es demasiado alta o demasiado baja. Cuando se activa una alarma, en respuesta, el grupo de Auto Scaling aumenta o reduce los recursos.
- AWS CloudFormation pila: Elastic AWS CloudFormation Beanstalk se utiliza para lanzar los recursos de su entorno y propagar los cambios de configuración. Los recursos se definen en una plantilla que puede verse en la [consola de AWS CloudFormation](#).
- Nombre de dominio: nombre de dominio que direcciona el tráfico a la aplicación web con el formato *subdominio.región.elasticbeanstalk.com*.

Note

Para aumentar la seguridad de las aplicaciones de Elastic Beanstalk, el dominio elasticbeanstalk.com está registrado en la [lista de sufijos públicos \(PSL\)](#). Para mayor seguridad, se recomienda que utilice cookies con un prefijo `__Host-` en caso de que necesite configurar cookies confidenciales en el nombre de dominio predeterminado de

sus aplicaciones de Elastic Beanstalk. Esta práctica le ayudará a defender su dominio contra intentos de falsificación de solicitudes entre sitios (CSRF). Para obtener más información, consulte la página de [configuración de cookies](#) en la red de desarrolladores de Mozilla.

Todos estos recursos los administra Elastic Beanstalk. Cuando termina su entorno, Elastic Beanstalk termina todos los recursos que este contiene.

Note

El bucket de Amazon S3 que Elastic Beanstalk crea se comparte entre los entornos y no se elimina durante la terminación del entorno. Para obtener más información, consulte [Usar Elastic Beanstalk con Amazon S3](#).

Instalación de CakePHP y creación de un sitio web

Composer puede instalar CakePHP y crear un proyecto de trabajo con un único comando:

```
~$ composer create-project --prefer-dist cakephp/app eb-cake
```

Composer instala CakePHP y unas 20 dependencias, y genera un proyecto predeterminado.

Si tiene algún problema al instalar CakePHP, visite el tema de instalación en la documentación oficial: <http://book.cakephp.org/4.0/en/installation.html>

Implementar la aplicación

Cree un [paquete de código fuente](#) que contenga los archivos creados por Composer. El comando siguiente crea un paquete de código fuente denominado `cake-default.zip`. Excluye los archivos de la carpeta `vendor`, que ocupan mucho espacio y no se requieren para implementar la aplicación en Elastic Beanstalk.

```
eb-cake zip ../cake-default.zip -r * .[^.]* -x "vendor/*"
```

Cargue el paquete de origen en Elastic Beanstalk para implementar CakePHP en su entorno.

Para implementar un paquete de código fuente

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En la página de información general del entorno, elija Upload and deploy (Cargar e implementar).
4. Utilice el cuadro de diálogo en pantalla para cargar el paquete de código fuente.
5. Elija Deploy (Implementar).
6. Cuando se complete la implementación, seleccione la URL del para abrir el sitio web en una nueva pestaña.

Note

Para optimizar aun más el paquete de código fuente, inicialice un repositorio Git y use el [comando git archive](#) para crear el paquete de código fuente. El proyecto Symfony predeterminado incluye un archivo `.gitignore` que indica a Git que debe excluir la carpeta `vendor` y los demás archivos que no se requieren para la implementación.

Cuando el proceso se complete, haga clic en la URL para abrir su aplicación CakePHP en el navegador.


Hasta el momento, todo fue bien. A continuación, va a agregar una base de datos al entorno y a configurar CakePHP para que se conecte a ella.

Agregar una base de datos al entorno

Inicie una instancia de base de datos de Amazon RDS en su entorno de Elastic Beanstalk. Puede utilizar bases de datos de MySQL, SQLServer o PostgreSQL con CakePHP en Elastic Beanstalk. En este ejemplo, vamos a utilizar PostgreSQL.

Para agregar una instancia de base de datos de Amazon RDS a su entorno de Elastic Beanstalk

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

 Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En Database (Base de datos), elija Edit (Editar).
5. En DB engine (Motor de base de datos), seleccione postgres.
6. Escriba un nombre de usuario y contraseña maestros. Elastic Beanstalk proporcionará estos valores a la aplicación a través de propiedades de entorno.
7. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

Se tarda alrededor de 10 minutos en crear una instancia de base de datos. Mientras tanto, puede actualizar el código fuente para que pueda leer la información de conexión del entorno. Elastic Beanstalk proporciona la información de la conexión a través de variables de entorno, como RDS_HOSTNAME, a las que puede obtener acceso desde la aplicación.

La configuración de base de datos de CakePHP se encuentra en un archivo llamado `app.php` de la carpeta `config` del proyecto de código. Abra este archivo y agregue código que permita leer las variables de entorno de `$_SERVER` y las asigne a variables locales. Inserte las líneas resaltadas en el siguiente ejemplo, detrás de la primera línea (`<?php`):

Example `~/Eb-cake/config/app.php`

```
<?php
if (!defined('RDS_HOSTNAME')) {
    define('RDS_HOSTNAME', $_SERVER['RDS_HOSTNAME']);
    define('RDS_USERNAME', $_SERVER['RDS_USERNAME']);
    define('RDS_PASSWORD', $_SERVER['RDS_PASSWORD']);
    define('RDS_DB_NAME', $_SERVER['RDS_DB_NAME']);
}
return [
    ...
```

La conexión de la base de datos está configurada más adelante en el archivo `app.php`. Busque la sección siguiente y modifique la configuración predeterminada de las fuentes de datos con el nombre del controlador que corresponda al motor de base de datos (MySQL, Sqlserver o Postgres) y configure las variables `host`, `username`, `password` y `database` para que lean los valores correspondientes en Elastic Beanstalk:

Example `~/Eb-cake/config/app.php`

```
...
/**
 * Connection information used by the ORM to connect
 * to your application's datastores.
 * Drivers include MySQL Postgres Sqlite Sqlserver
 * See vendor\cakephp\cakephp\src\Database\Driver for complete list
 */
'Datasources' => [
    'default' => [
        'className' => 'Cake\Database\Connection',
        'driver' => 'Cake\Database\Driver\Postgres',
        'persistent' => false,
        'host' => RDS_HOSTNAME,
        /*
         * CakePHP will use the default DB port based on the driver selected
         * MySQL on MAMP uses port 8889, MAMP users will want to uncomment
         * the following line and set the port accordingly
         */
        //'port' => 'non_standard_port_number',
        'username' => RDS_USERNAME,
        'password' => RDS_PASSWORD,
        'database' => RDS_DB_NAME,
        /*
         * You do not need to set this flag to use full utf-8 encoding (internal
         default since CakePHP 3.6).
         */
        //'encoding' => 'utf8mb4',
        'timezone' => 'UTC',
        'flags' => [],
        'cacheMetadata' => true,
        'log' => false,
    ]
]
...
```


Cuando finalice el lanzamiento de la instancia de base de datos, empaquete e implemente la aplicación actualizada en el entorno:

Para actualizar su entorno de Elastic Beanstalk

1. Cree un nuevo paquete de código fuente:

```
~/eb-cake$ zip ../cake-v2-rds.zip -r * .[^.]* -x "vendor/*"
```

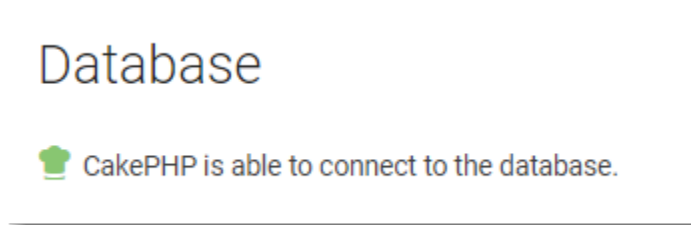
2. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
3. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

 Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

4. Elija Upload and Deploy (Cargar e implementar).
5. Seleccione Browse (Examinar) y cargue `cake-v2-rds.zip`.
6. Elija Implementar.

La implementación de una nueva versión de la aplicación tarda menos de un minuto. Cuando se haya completado, actualice la página web de nuevo para comprobar que la conexión con la base de datos es correcta:




Limpieza

Cuando termine de trabajar con Elastic Beanstalk, puede terminar su entorno. [Elastic Beanstalk AWS cancela todos los recursos asociados al entorno, como las instancias de Amazon EC2, las instancias de bases de datos, los balanceadores de carga, los grupos de seguridad y las alarmas.](#)

Para cerrar el entorno de Elastic Beanstalk desde la consola

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

 Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. Elija Actions (Acciones) y, a continuación, Terminate Environment (Terminar el entorno).
4. Utilice el cuadro de diálogo en pantalla para confirmar la terminación del entorno.

Con Elastic Beanstalk, puede crear fácilmente un nuevo entorno para su aplicación en cualquier momento.

Además, puede terminar los recursos de base de datos creados fuera del entorno de Elastic Beanstalk. Cuando termine una instancia de base de datos de Amazon RDS, puede hacer una instantánea para restaurar los datos posteriormente en otra instancia.

Para terminar la instancia de base de datos de RDS

1. Abra la [consola de Amazon RDS](#).
2. Seleccione Databases (Bases de datos).
3. Seleccione la instancia de base de datos.
4. Elija Acciones y, a continuación, elija Eliminar.
5. Decida si desea crear una instantánea y seleccione Delete (Eliminar).

Siguientes pasos

Para obtener más información sobre CakePHP, consulte el libro en book.cakephp.org.

A medida que siga desarrollando la aplicación, es probable que desee contar con un mecanismo que le permita administrar los entornos e implementar la aplicación sin tener que crear manualmente un archivo .zip y cargarlo en la consola de Elastic Beanstalk. La interfaz de [línea de comandos de Elastic Beanstalk](#) (EB CLI easy-to-use) proporciona comandos para crear, configurar e implementar aplicaciones en entornos de Elastic Beanstalk desde la línea de comandos.

Ejecutar una instancia de base de datos de Amazon RDS; en el entorno de Elastic Beanstalk es ideal para desarrollo y pruebas, pero supedita el ciclo de vida de la base de datos al entorno. Consulte [Agregar una instancia de base de datos de Amazon RDS a un entorno de aplicaciones PHP](#) para obtener instrucciones acerca de cómo conectarse a una base de datos que se ejecute fuera del entorno.

Por último, si pretende utilizar la aplicación en un entorno de producción, querrá [configurar un nombre de dominio personalizado](#) para el entorno y [habilitar HTTPS](#) para las conexiones seguras.

Implementación de una aplicación Symfony en Elastic Beanstalk

[Symfony](#) es un marco de código abierto para el desarrollo de aplicaciones web PHP dinámicas. Este tutorial te guía por el proceso de generar una aplicación Symfony y desplegarla en un AWS Elastic Beanstalk entorno.

Secciones

- [Requisitos previos](#)
- [Lance un entorno de Elastic Beanstalk](#)
- [Instalación de Symfony y generación de un sitio web](#)
- [Implementar la aplicación](#)
- [Configuración de los ajustes de Composer](#)
- [Limpieza](#)
- [Sigüientes pasos](#)

Requisitos previos

Este tutorial asume que tiene conocimiento de las operaciones básicas de Elastic Beanstalk y de la consola de Elastic Beanstalk. Si aún no lo ha hecho, siga las instrucciones que se indican en [Introducción a Elastic Beanstalk](#) para lanzar su primer entorno de Elastic Beanstalk.

Para seguir los procedimientos de esta guía, necesitará un shell o un terminal de línea de comando donde pueda ejecutar los comandos. Los comandos aparecen en listas y van precedidos del símbolo del sistema (\$) y del nombre del directorio actual, si es aplicable.

```
~/eb-project$ this is a command  
this is output
```

En Linux y macOS, puede utilizar el administrador de shell y paquetes preferido. En Windows, puede [instalar el subsistema de Windows para Linux para](#) obtener una versión de Ubuntu y Bash integrada en Windows.

Symfony 4.4.9 requiere PHP 7.1.3 o posterior. También requiere las extensiones de PHP enumeradas en el tema de [requisitos técnicos](#) en la documentación oficial de instalación de Symfony. En este tutorial, utilizamos PHP 7.2 y la [versión correspondiente de la plataforma](#) Elastic Beanstalk. Siga las instrucciones del tema [Configuración del entorno de desarrollo de PHP](#) para instalar PHP y Composer.

Para obtener información de soporte y mantenimiento de Symfony, consulte el tema de [versiones de Symfony](#) en el sitio web de Symfony. Con el fin de obtener más información acerca de las actualizaciones relacionadas con el soporte de versiones de PHP para Symfony 4.4.9, consulte el tema de [notas de la versión de Symfony 4.4.9](#) en el sitio web de Symfony.

Lance un entorno de Elastic Beanstalk

Utilice la consola de Elastic Beanstalk para crear un entorno de Elastic Beanstalk. Elija la plataforma PHP y acepte la configuración predeterminada y el código de muestra.

Para lanzar un entorno (consola)

1. [Abra la consola de Elastic Beanstalk mediante este enlace preconfigurado: console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=Tutorials&EnvironmentType=LoadBalanced](#)
2. En Platform (Plataforma), seleccione la plataforma y la ramificación de plataforma que coincidan con el idioma utilizado por la aplicación.
3. En Application code (Código de aplicación), seleccione Sample application (Aplicación de muestra).
4. Elija Review and launch (Revisar y lanzar).
5. Revise las opciones disponibles. Elija la opción disponible que quiera usar y, cuando esté listo, elija Create app (Crear aplicación).

El entorno tarda unos cinco minutos en crearse e incluye los recursos siguientes:

- Instancia de EC2: una máquina virtual de Amazon Elastic Compute Cloud (Amazon EC2) configurada para ejecutar aplicaciones web en la plataforma que elija.

Cada plataforma ejecuta un conjunto específico de software, archivos de configuración y scripts compatibles con una determinada versión de lenguaje, marco de trabajo y contenedor web (o una combinación de estos). La mayoría de las plataformas utilizan Apache o nginx como un proxy inverso que se sitúa delante de la aplicación web, reenvía las solicitudes a esta, administra los recursos estáticos y genera registros de acceso y errores.

- Instance security group (Grupo de seguridad de la instancia): grupo de seguridad de Amazon EC2 configurado para permitir el tráfico entrante en el puerto 80. Este recurso permite que el tráfico HTTP procedente del balanceador de carga llegue a la instancia EC2 en la que se ejecuta la aplicación web. De forma predeterminada, el tráfico no está permitido en otros puertos.
- Balanceador de carga: balanceador de carga de Elastic Load Balancing configurado para distribuir solicitudes a las instancias que se ejecutan en la aplicación. Los balanceadores de carga también permiten que las instancias no estén expuestas directamente a Internet.
- Grupo de seguridad del balanceador de carga: grupo de seguridad de Amazon EC2 configurado para permitir el tráfico entrante en el puerto 80. Este recurso permite que el tráfico HTTP procedente de Internet llegue al balanceador de carga. De forma predeterminada, el tráfico no está permitido en otros puertos.
- Grupo de Auto Scaling: grupo de Auto Scaling configurado para reemplazar una instancia si termina o deja de estar disponible.
- Bucket de Amazon S3: ubicación de almacenamiento para el código fuente, los registros y otros artefactos que se crean al utilizar Elastic Beanstalk.
- CloudWatch Alarmas de Amazon: dos CloudWatch alarmas que monitorean la carga de las instancias de su entorno y que se activan si la carga es demasiado alta o demasiado baja. Cuando se activa una alarma, en respuesta, el grupo de Auto Scaling aumenta o reduce los recursos.
- AWS CloudFormation pila: Elastic AWS CloudFormation Beanstalk se utiliza para lanzar los recursos de su entorno y propagar los cambios de configuración. Los recursos se definen en una plantilla que puede verse en la [consola de AWS CloudFormation](#).
- Nombre de dominio: nombre de dominio que direcciona el tráfico a la aplicación web con el formato *subdominio.región.elasticbeanstalk.com*.

Note

Para aumentar la seguridad de las aplicaciones de Elastic Beanstalk, el dominio elasticbeanstalk.com está registrado en la [lista de sufijos públicos \(PSL\)](#). Para mayor seguridad, se recomienda que utilice cookies con un prefijo `__Host-` en caso de que necesite configurar cookies confidenciales en el nombre de dominio predeterminado de

sus aplicaciones de Elastic Beanstalk. Esta práctica le ayudará a defender su dominio contra intentos de falsificación de solicitudes entre sitios (CSRF). Para obtener más información, consulte la página de [configuración de cookies](#) en la red de desarrolladores de Mozilla.

Todos estos recursos los administra Elastic Beanstalk. Cuando termina su entorno, Elastic Beanstalk termina todos los recursos que este contiene.

Note

El bucket de Amazon S3 que Elastic Beanstalk crea se comparte entre los entornos y no se elimina durante la terminación del entorno. Para obtener más información, consulte [Usar Elastic Beanstalk con Amazon S3](#).

Instalación de Symfony y generación de un sitio web

Composer puede instalar Symfony y crear un proyecto de trabajo con un único comando:

```
~$ composer create-project symfony/website-skeleton eb-symfony
```

Composer instala Symfony y sus dependencias y genera un proyecto predeterminado.

Si tiene algún problema al instalar Symfony, vaya al tema de [instalación](#) en la documentación oficial de Symfony.

Implementar la aplicación

Vaya al directorio del proyecto.

```
~$ cd eb-symfony
```


Cree un [paquete de código fuente](#) que contenga los archivos creados por Composer. El comando siguiente crea un paquete de código fuente denominado `symfony-default.zip`. Excluye los archivos de la carpeta `vendor`, que ocupan mucho espacio y no se requieren para implementar la aplicación en Elastic Beanstalk.

```
eb-symfony$ zip ../symfony-default.zip -r * .[^.]* -x "vendor/*"
```

Cargue el paquete de origen en Elastic Beanstalk para implementar Symfony en su entorno.


Para implementar un paquete de código fuente

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

 Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En la página de información general del entorno, elija Upload and deploy (Cargar e implementar).
4. Utilice el cuadro de diálogo en pantalla para cargar el paquete de código fuente.
5. Elija Deploy (Implementar).
6. Cuando se complete la implementación, seleccione la URL del para abrir el sitio web en una nueva pestaña.

 Note

Para optimizar aun más el paquete de código fuente, inicialice un repositorio Git y use el [comando git archive](#) para crear el paquete de código fuente. El proyecto Symfony predeterminado incluye un archivo `.gitignore` que indica a Git que debe excluir la carpeta `vendor` y los demás archivos que no se requieren para la implementación.

Configuración de los ajustes de Composer


Cuando la implementación se complete, haga clic en la URL para abrir la aplicación de Symfony en el navegador.

¿Qué es esto? De forma predeterminada, Elastic Beanstalk proporciona la raíz del proyecto en la ruta raíz del sitio web. En este caso, sin embargo, la página predeterminada (`app.php`) está un nivel por debajo de la carpeta web. Puede comprobarlo agregando `/public` a la URL. Por ejemplo, <http://symfony.us-east-2.elasticbeanstalk.com/public>.

Para que la aplicación Symfony esté disponible en la ruta raíz, utilice la consola de Elastic Beanstalk con el fin de configurar la raíz del documento del sitio web.

Para configurar la raíz de documentos del sitio web

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

 Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.


3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Actualizaciones, supervisión y registro, seleccione Editar.
5. En Document root (Raíz del documento), introduzca **/public**.
6. Para guardar los cambios, elija Aplicar en la parte inferior de la página.
7. Cuando la actualización se haya completado, haga clic en la URL para volver a abrir el sitio en el navegador.

Limpieza

Cuando termine de trabajar con Elastic Beanstalk, puede terminar su entorno. [Elastic Beanstalk AWS cancela todos los recursos asociados al entorno, como las instancias de Amazon EC2, las instancias de bases de datos, los balanceadores de carga, los grupos de seguridad y las alarmas.](#)

Para cerrar el entorno de Elastic Beanstalk desde la consola

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

 Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. Elija Actions (Acciones) y, a continuación, Terminate Environment (Terminar el entorno).

4. Utilice el cuadro de diálogo en pantalla para confirmar la terminación del entorno.

Con Elastic Beanstalk, puede crear fácilmente un nuevo entorno para su aplicación en cualquier momento.

Siguientes pasos

Para obtener más información acerca de Symfony, consulte [What is Symfony?](#) en symfony.com.

A medida que siga desarrollando la aplicación, es probable que desee contar con un mecanismo que le permita administrar los entornos e implementar la aplicación sin tener que crear manualmente un archivo .zip y cargarlo en la consola de Elastic Beanstalk. La interfaz de [línea de comandos de Elastic Beanstalk](#) (EB CLI easy-to-use) proporciona comandos para crear, configurar e implementar aplicaciones en entornos de Elastic Beanstalk desde la línea de comandos.

En este tutorial, ha utilizado la consola de Elastic Beanstalk para configurar las opciones de Composer. Para integrar esta configuración en el código fuente de la aplicación, puede utilizar un archivo de configuración similar al siguiente.

Example .ebextensions/composer.config

```
option_settings:
  aws:elasticbeanstalk:container:php:phpini:
    document_root: /public
```

Para obtener más información, consulte [Personalización avanzada de entornos con archivos de configuración \(.ebextensions\)](#).

Symfony utiliza sus propios archivos de configuración para configurar las conexiones a las bases de datos. Para obtener instrucciones acerca de cómo conectarse a una base de datos con Symfony, consulte [Conexión a una base de datos con Symfony](#).

Por último, si pretende utilizar la aplicación en un entorno de producción, querrá [configurar un nombre de dominio personalizado](#) para el entorno y [habilitar HTTPS](#) para las conexiones seguras.

Implementación de una aplicación PHP de alta disponibilidad con una base de datos externa de Amazon RDS en Elastic Beanstalk

Este tutorial explica el proceso de [lanzar una instancia de base de datos de RDS](#) externa y configurar un entorno de alta disponibilidad que ejecute una aplicación PHP para conectarse a él. AWS Elastic Beanstalk Si se ejecuta una instancia de base de datos externa en Elastic Beanstalk, la

base de datos se desvincula del ciclo de vida del entorno. Esto le permite conectarse a la misma base de datos desde diferentes entornos, intercambiar una base de datos por otra o realizar una implementación "blue/green" sin que ello afecte a la base de datos.

En este tutorial, se utiliza una [aplicación PHP de ejemplo](#) que emplea una base de datos MySQL para guardar los datos de texto proporcionados por el usuario. La aplicación de ejemplo utiliza [archivos de configuración](#) para configurar los [ajustes de PHP](#) y crear una tabla en la base de datos para que la aplicación la utilice. También se muestra cómo utilizar un [archivo Composer](#) para instalar paquetes durante la implementación.

Secciones

- [Requisitos previos](#)
- [Inicie una instancia de base de datos en Amazon RDS](#)
- [Cree un entorno de Elastic Beanstalk](#)
- [Configuración de grupos de seguridad, propiedades de entorno y escalado](#)
- [Implemente la aplicación de muestra.](#)
- [Limpieza](#)
- [Siguiendo los pasos](#)

Requisitos previos

[Antes de empezar, descargue el paquete fuente de la aplicación de muestra desde GitHub: - app-1.3.zip eb-demo-php-simple](#)

Los procedimientos de este tutorial para las tareas de Amazon Relational Database Service (Amazon RDS) suponen que está iniciando recursos en una [Amazon Virtual Private Cloud](#) (Amazon VPC) predeterminada. Todas las nuevas cuentas contienen una VPC predeterminada en cada región. Si no dispone de una VPC predeterminada, los procedimientos serán diferentes. Consulte [Uso de Elastic Beanstalk con Amazon RDS](#) para obtener instrucciones sobre la plataforma de EC2 y de una VPC personalizada.


Inicie una instancia de base de datos en Amazon RDS

Para utilizar una base de datos externa con una aplicación que se ejecuta en Elastic Beanstalk, primero lance una instancia de base de datos con Amazon RDS. Cuando inicia una instancia con Amazon RDS, es completamente independiente de Elastic Beanstalk y sus entornos Elastic Beanstalk, y Elastic Beanstalk no la finalizará ni supervisará.

Use la consola de Amazon RDS para lanzar una instancia de base de datos MySQL Multi-AZ. La elección de una implementación Multi-AZ garantiza que la base de datos realice una conmutación por error y siga estando disponible en caso de que la instancia de base de datos de origen quede fuera de servicio.

Para lanzar una instancia de base de datos de RDS en una VPC predeterminada

1. Abra la [consola de RDS](#).
2. En el panel de navegación, seleccione Databases (Bases de datos).
3. Elija Create database (Crear base de datos).
4. Elija Standard Create (Creación estándar).

 Important

No elija la opción Easy Create (Creación sencilla). Si lo elige, no puede configurar los ajustes necesarios para lanzar esta base de datos de RDS.

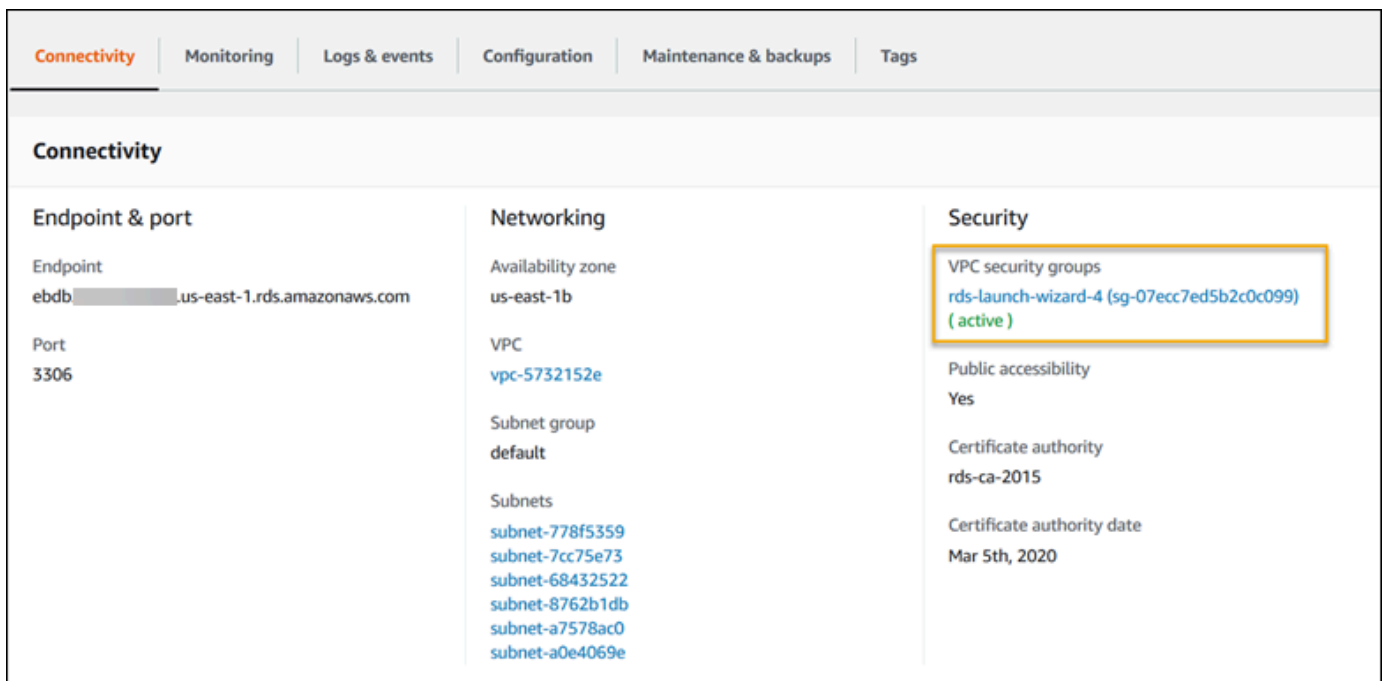
5. En Additional configuration (Configuración adicional), escriba **ebdb** en Initial database name (Nombre de base de datos inicial).
6. Revise la configuración predeterminada y ajústela de acuerdo con sus requisitos específicos. Preste atención a las siguientes opciones:
 - DB instance class (Clase de instancia de base de datos): elija un tamaño de instancia que cuente con la memoria y la potencia de CPU apropiadas para su carga de trabajo.
 - Multi-AZ deployment (Implementación Multi-AZ): para obtener alta disponibilidad, establezca esta opción en Create an Aurora Replica/Reader node in a different AZ (Crear una réplica de Aurora/nodo de lector en otra zona de disponibilidad).
 - Master username (Nombre de usuario maestro) y Master password (Contraseña maestra): nombre de usuario y contraseña de la base de datos. Anote esta configuración, ya que la necesitará posteriormente.
7. Compruebe los valores predeterminados de las demás opciones y, a continuación, elija Create database (Crear base de datos).

A continuación, modifique el grupo de seguridad asociado a la instancia de base de datos para permitir el tráfico entrante en el puerto correspondiente. Este es el mismo grupo de seguridad que

asociará a su entorno de Elastic Beanstalk más adelante, por lo que la regla que añada concederá permiso de entrada a otros recursos del mismo grupo de seguridad.

Para modificar las reglas de entrada del grupo de seguridad adjuntas a la instancia de RDS

1. Abra la [consola de Amazon RDS](#).
2. Seleccione Databases (Bases de datos).
3. Elija el nombre de la instancia de base de datos para ver sus detalles.
4. En la sección Connectivity (Conectividad), tome nota de los valores de Subnets (Subredes), Security groups (Grupos de seguridad) y Endpoint (Punto de conexión) que aparecen en esta página. Esto es para que pueda utilizar esta información más adelante.
5. En Security (Seguridad) se muestra el grupo de seguridad que está asociado a la instancia de base de datos. Abra el enlace para ver el grupo de seguridad en la consola de Amazon EC2.



6. En los detalles del grupo de seguridad, elija la pestaña Inbound (Entrada).
7. Elija Edit (Editar).
8. Seleccione Add Rule (Agregar regla).
9. En Type (Tipo), elija el motor de base de datos que utiliza la aplicación.
10. En Source (Origen), escriba **sg-** para ver una lista de grupos de seguridad disponibles. Elija el grupo de seguridad que está asociado al grupo de Auto Scaling que se utiliza con su entorno de Elastic Beanstalk. Esto es para que las instancias de Amazon EC2 en el entorno puedan acceder a la base de datos.

Edit inbound rules

Type	Protocol	Port Range	Source	Description
MySQL/Aurora	TCP	3306	Custom 72.21.198.67/32	e.g. SSH for Admin Desktop
MySQL/Aurora	TCP	3306	Custom sg-	e.g. SSH for Admin Desktop

NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

Buttons: Add Rule, Cancel, Save

11. Seleccione Save (Guardar).

Se tarda alrededor de 10 minutos en crear una instancia de base de datos. Mientras tanto, cree su entorno de Elastic Beanstalk.

Cree un entorno de Elastic Beanstalk

Utilice la consola de Elastic Beanstalk para crear un entorno de Elastic Beanstalk. Elija la plataforma PHP y acepte la configuración predeterminada y el código de muestra. Tras lanzar el entorno, puede configurarlo para que se conecte a la base de datos y, a continuación, implementar la aplicación de muestra desde la que la descargó. GitHub

Para lanzar un entorno (consola)

1. [Abra la consola de Elastic Beanstalk mediante este enlace preconfigurado:](https://console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=Tutorials&EnvironmentType=LoadBalanced)
console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=Tutorials&EnvironmentType=LoadBalanced
2. En Platform (Plataforma), seleccione la plataforma y la ramificación de plataforma que coincidan con el idioma utilizado por la aplicación.
3. En Application code (Código de aplicación), seleccione Sample application (Aplicación de muestra).
4. Elija Review and launch (Revisar y lanzar).
5. Revise las opciones disponibles. Elija la opción disponible que quiera usar y, cuando esté listo, elija Create app (Crear aplicación).

El entorno tarda unos cinco minutos en crearse e incluye los recursos siguientes:

- Instancia de EC2: una máquina virtual de Amazon Elastic Compute Cloud (Amazon EC2) configurada para ejecutar aplicaciones web en la plataforma que elija.

Cada plataforma ejecuta un conjunto específico de software, archivos de configuración y scripts compatibles con una determinada versión de lenguaje, marco de trabajo y contenedor web (o una combinación de estos). La mayoría de las plataformas utilizan Apache o nginx como un proxy inverso que se sitúa delante de la aplicación web, reenvía las solicitudes a esta, administra los recursos estáticos y genera registros de acceso y errores.

- Instance security group (Grupo de seguridad de la instancia): grupo de seguridad de Amazon EC2 configurado para permitir el tráfico entrante en el puerto 80. Este recurso permite que el tráfico HTTP procedente del balanceador de carga llegue a la instancia EC2 en la que se ejecuta la aplicación web. De forma predeterminada, el tráfico no está permitido en otros puertos.
- Balanceador de carga: balanceador de carga de Elastic Load Balancing configurado para distribuir solicitudes a las instancias que se ejecutan en la aplicación. Los balanceadores de carga también permiten que las instancias no estén expuestas directamente a Internet.
- Grupo de seguridad del balanceador de carga: grupo de seguridad de Amazon EC2 configurado para permitir el tráfico entrante en el puerto 80. Este recurso permite que el tráfico HTTP procedente de Internet llegue al balanceador de carga. De forma predeterminada, el tráfico no está permitido en otros puertos.
- Grupo de Auto Scaling: grupo de Auto Scaling configurado para reemplazar una instancia si termina o deja de estar disponible.
- Bucket de Amazon S3: ubicación de almacenamiento para el código fuente, los registros y otros artefactos que se crean al utilizar Elastic Beanstalk.
- CloudWatch Alarmas de Amazon: dos CloudWatch alarmas que monitorean la carga de las instancias de su entorno y que se activan si la carga es demasiado alta o demasiado baja. Cuando se activa una alarma, en respuesta, el grupo de Auto Scaling aumenta o reduce los recursos.
- AWS CloudFormation pila: Elastic AWS CloudFormation Beanstalk se utiliza para lanzar los recursos de su entorno y propagar los cambios de configuración. Los recursos se definen en una plantilla que puede verse en la [consola de AWS CloudFormation](#).
- Nombre de dominio: nombre de dominio que direcciona el tráfico a la aplicación web con el formato *subdominio.región.elasticbeanstalk.com*.

Note

Para aumentar la seguridad de las aplicaciones de Elastic Beanstalk, el dominio elasticbeanstalk.com está registrado en la [lista de sufijos públicos \(PSL\)](#). Para mayor

seguridad, se recomienda que utilice cookies con un prefijo `__Host-` en caso de que necesite configurar cookies confidenciales en el nombre de dominio predeterminado de sus aplicaciones de Elastic Beanstalk. Esta práctica le ayudará a defender su dominio contra intentos de falsificación de solicitudes entre sitios (CSRF). Para obtener más información, consulte la página de [configuración de cookies](#) en la red de desarrolladores de Mozilla.

Todos estos recursos los administra Elastic Beanstalk. Cuando termina su entorno, Elastic Beanstalk termina todos los recursos que este contiene. La instancia de base de datos de RDS que lanzó se encuentra fuera de su entorno, por lo que usted es responsable de administrar su ciclo de vida.

Note


El bucket de Amazon S3 que Elastic Beanstalk crea se comparte entre los entornos y no se elimina durante la terminación del entorno. Para obtener más información, consulte [Usar Elastic Beanstalk con Amazon S3](#).

Configuración de grupos de seguridad, propiedades de entorno y escalado

Agregue el grupo de seguridad de la instancia de base de datos al entorno en ejecución. Este procedimiento hace que Elastic Beanstalk vuelva a aprovisionar todas las instancias de su entorno con el grupo de seguridad adicional asociado.

Para agregar un grupo de seguridad al entorno

- Realice una de las siguientes acciones siguientes:
 - Para añadir un grupo de seguridad mediante la consola de Elastic Beanstalk
 - a. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
 - b. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

 Note


Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

- c. En el panel de navegación, elija Configuration (Configuración).
 - d. En la categoría de configuración Instances (Instancias), elija Edit (Editar).
 - e. En EC2 security groups (Grupos de seguridad de EC2), elija el grupo de seguridad que desea asignar a las instancias, además del grupo de seguridad de instancia que crea Elastic Beanstalk.
 - f. Para guardar los cambios, elija Aplicar en la parte inferior de la página.
 - g. Lea la advertencia y, a continuación, elija Confirm (Confirmar).
- Para agregar un grupo de seguridad con un [archivo de configuración](#), utilice el archivo de ejemplo [securitygroup-addexisting.config](#).

Después, utilice las propiedades de entorno para pasar la información de la conexión al entorno. La aplicación de ejemplo utiliza un conjunto predeterminado de propiedades que coinciden con las que Elastic Beanstalk configura al aprovisionar una base de datos en su entorno.

Para configurar las propiedades de entorno de una instancia de base de datos de Amazon RDS

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

 Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Actualizaciones, supervisión y registro, seleccione Editar.
5. En la sección Environment properties (Propiedades del entorno), defina las variables que lee la aplicación para crear una cadena de conexión. Para conseguir compatibilidad con los entornos que tienen una instancia de base de datos de RDS, utilice los siguientes nombres y valores. Encontrará todos los valores, excepto la contraseña, en la [consola de RDS](#).

Nombre de la propiedad	Descripción	Valor de la propiedad
RDS_HOSTNAME	El nombre de host de la instancia de base de datos.	En la pestaña Connectivity & security (Conectividad y seguridad) de la consola de Amazon RDS: Endpoint (Punto de enlace).
RDS_PORT	El puerto en donde la instancia de base de datos acepta las conexiones. El valor predeterminado varía dependiendo del motor de base de datos.	En la pestaña Connectivity & security (Conectividad y seguridad) de la consola de Amazon RDS: Port (Puerto).
RDS_DB_NAME	El nombre de la base de dato, ebdb .	En la pestaña Configuration (Configuración) de la consola de Amazon RDS: DB Name (Nombre de la base de datos).
RDS_USERNAME	El nombre de usuario que ha configurado para la base de datos.	En la pestaña Configuration (Configuración) de la consola de Amazon RDS: Master username (Nombre de usuario maestro).
RDS_PASSWORD	La contraseña que ha configurado para la base de datos.	No está disponible como referencia en la consola de Amazon RDS.

Environment Properties

The following properties are passed into the application as environment variables. [Learn more.](#)

Property Name	Property Value
RDS_DB_NAME	<input type="text" value="ebdb"/> ✕
RDS_HOSTNAME	<input type="text" value="webapp-db.jxzc b5mpaniu.us-wes"/> ✕
RDS_PORT	<input type="text" value="5432"/> ✕
RDS_USERNAME	<input type="text" value="webapp-admin"/> ✕
<input type="text" value="RDS_PASSWORD"/>	<input type="text" value="kUj5uKxmWDMYc403"/> +

[Cancel](#) [Apply](#)

6. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

Por último, configure el grupo de Auto Scaling de su entorno con un número mínimo de instancias mayor. Ejecute al menos dos instancias en todo momento para evitar que los servidores web de su entorno se conviertan en un único punto de error y para poder implementar cambios sin que el sitio se quede fuera de servicio.

Si desea configurar el grupo de Auto Scaling del entorno para disponer de una alta disponibilidad

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note


Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).

4. En la categoría de configuración Capacity (Capacidad), elija Edit (Editar).
5. En la sección Auto Scaling group (grupo de Auto Scaling) defina Min instances (Instancias mínimas) en **2**.
6. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

Implemente la aplicación de muestra.


Ahora, el entorno está listo para ejecutar la aplicación de ejemplo y conectarse a Amazon RDS. Implemente la aplicación de ejemplo en el entorno.

 Note

[Descargue el paquete fuente desde GitHub, si aún no lo ha hecho: -app-1.3.zip eb-demo-php-simple](#)

Para implementar un paquete de código fuente

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

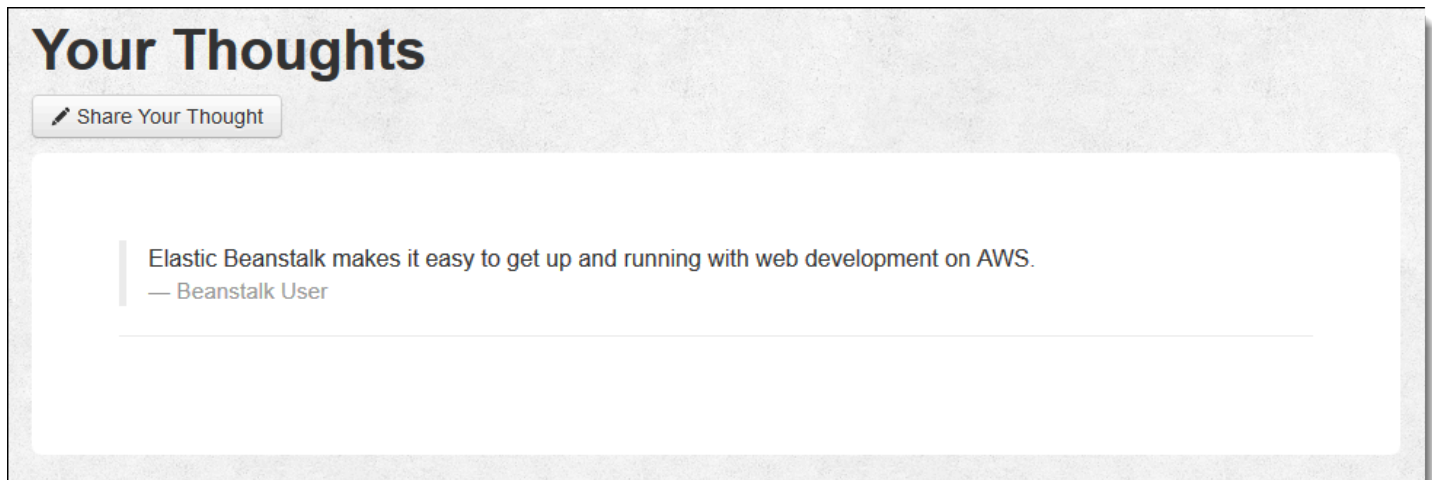
 Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En la página de información general del entorno, elija Upload and deploy (Cargar e implementar).
4. Utilice el cuadro de diálogo en pantalla para cargar el paquete de código fuente.
5. Elija Deploy (Implementar).
6. Cuando se complete la implementación, seleccione la URL del para abrir el sitio web en una nueva pestaña.

El sitio recopila los comentarios de los usuarios y utiliza una base de datos MySQL para almacenar los datos. Para agregar un comentario, elija Share Your Thought (Compartir su opinión), escriba sus impresiones y seleccione Submit Your Thought (Enviar su opinión). La aplicación web escribe el

comentario en la base de datos para que cualquier instancia del entorno pueda leerlo y que no se pierda en caso de que las instancias se queden fuera de servicio.



Limpieza

Cuando termine de trabajar con Elastic Beanstalk, puede terminar su entorno. [Elastic Beanstalk AWS cancela todos los recursos asociados al entorno, como las instancias de Amazon EC2, las instancias de bases de datos, los balanceadores de carga, los grupos de seguridad y las alarmas.](#)

Para cerrar el entorno de Elastic Beanstalk desde la consola

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. Elija Actions (Acciones) y, a continuación, Terminate Environment (Terminar el entorno).
4. Utilice el cuadro de diálogo en pantalla para confirmar la terminación del entorno.

Con Elastic Beanstalk, puede crear fácilmente un nuevo entorno para su aplicación en cualquier momento.

Además, puede terminar los recursos de base de datos creados fuera del entorno de Elastic Beanstalk. Cuando termine una instancia de base de datos de Amazon RDS, puede hacer una instantánea para restaurar los datos posteriormente en otra instancia.

Para terminar la instancia de base de datos de RDS

1. Abra la [consola de Amazon RDS](#).
2. Seleccione Databases (Bases de datos).
3. Seleccione la instancia de base de datos.
4. Elija Acciones y, a continuación, elija Eliminar.
5. Decida si desea crear una instantánea y seleccione Delete (Eliminar).

Siguientes pasos

A medida que siga desarrollando la aplicación, es probable que desee contar con un mecanismo que le permita administrar los entornos e implementar la aplicación sin tener que crear manualmente un archivo .zip y cargarlo en la consola de Elastic Beanstalk. La interfaz de [línea de comandos de Elastic Beanstalk](#) (EB CLI easy-to-use) proporciona comandos para crear, configurar e implementar aplicaciones en entornos de Elastic Beanstalk desde la línea de comandos.

La aplicación de ejemplo utiliza los archivos de configuración para configurar los ajustes de PHP y crear una tabla en la base de datos si no existe todavía. También puede utilizar un archivo de configuración para configurar el grupo de seguridad de las instancias durante la creación del entorno y evitar así la laboriosa tarea de actualizar la configuración. Para obtener más información, consulte [Personalización avanzada de entornos con archivos de configuración \(.ebextensions\)](#).

En las tareas de desarrollo y pruebas, es posible que desee utilizar la funcionalidad de Elastic Beanstalk para agregar una instancia de base de datos administrada directamente al entorno. Para obtener instrucciones acerca de cómo configurar una base de datos dentro del entorno, consulte [Añadir una base de datos al entorno de Elastic Beanstalk](#).

Si necesita una base de datos de alto desempeño, considere la posibilidad de usar [Amazon Aurora](#). Amazon Aurora es un motor de base de datos compatible con MySQL que ofrece funciones de bases de datos comerciales a bajo costo. Para conectar la aplicación a una base de datos diferente, repita los pasos de la [configuración del grupo de seguridad](#) y [actualice las propiedades de entorno relacionadas con RDS](#).

Por último, si pretende utilizar la aplicación en un entorno de producción, querrá [configurar un nombre de dominio personalizado](#) para el entorno y [habilitar HTTPS](#) para las conexiones seguras.

Implementación de un WordPress sitio web de alta disponibilidad con una base de datos externa de Amazon RDS en Elastic Beanstalk

En este tutorial se describe cómo [lanzar una instancia de base de datos de Amazon RDS](#) externa y AWS Elastic Beanstalk, a continuación, cómo configurar un entorno de alta disponibilidad en el que se ejecute un WordPress sitio web para conectarse a él. El sitio web utiliza Amazon Elastic File System (Amazon EFS) como almacenamiento compartido para los archivos cargados.

Si se ejecuta una instancia de base de datos externa en Elastic Beanstalk, la base de datos se desvincula del ciclo de vida del entorno. Esto le permite conectarse a la misma base de datos desde diferentes entornos, intercambiar una base de datos por otra o realizar una [implementación blue/green](#) sin que ello afecte a la base de datos.

Note

Para obtener información actualizada sobre la compatibilidad de las versiones de PHP con WordPress las versiones, consulte [Compatibilidad y WordPress versiones de PHP](#) en el WordPress sitio web. Debe consultar esta información antes de actualizar sus WordPress implementaciones a una nueva versión de PHP.

Temas

- [Requisitos previos](#)
- [Inicie una instancia de base de datos en Amazon RDS](#)
- [Descarga WordPress](#)
- [Lance un entorno de Elastic Beanstalk](#)
- [Configuración de grupos de seguridad y propiedades de entorno](#)
- [Configuración e implementación de la aplicación](#)
- [Instale WordPress](#)
- [Actualización de claves y sales](#)
- [Eliminación de las restricciones de acceso](#)
- [Configure el grupo de Auto Scaling](#)
- [Actualice WordPress](#)

- [Eliminar recursos](#)
- [Siguiendo pasos](#)

Requisitos previos

Este tutorial asume que tiene conocimiento de las operaciones básicas de Elastic Beanstalk y de la consola de Elastic Beanstalk. Si aún no lo ha hecho, siga las instrucciones que se indican en [Introducción a Elastic Beanstalk](#) para lanzar su primer entorno de Elastic Beanstalk.

Para seguir los procedimientos de esta guía, necesitará un shell o un terminal de línea de comando donde pueda ejecutar los comandos. Los comandos aparecen en listas y van precedidos del símbolo del sistema (\$) y del nombre del directorio actual, si es aplicable.

```
~/eb-project$ this is a command  
this is output
```

En Linux y macOS, puede utilizar el administrador de shell y paquetes preferido. En Windows, puede [instalar el subsistema de Windows para Linux para](#) obtener una versión de Ubuntu y Bash integrada en Windows.

VPC predeterminada

Los procedimientos de este tutorial de Amazon Relational Database Service (Amazon RDS) suponen que está iniciando recursos en una [Amazon Virtual Private Cloud](#) (Amazon VPC) predeterminada. Todas las cuentas nuevas incluyen una VPC predeterminada en cada AWS región. Si no dispone de una VPC predeterminada, los procedimientos serán diferentes. Consulte [Uso de Elastic Beanstalk con Amazon RDS](#) para obtener instrucciones sobre la plataforma de EC2 y de una VPC personalizada.

AWS Regiones

La aplicación de ejemplo utiliza Amazon EFS, que solo funciona en AWS las regiones que admiten Amazon EFS. Para obtener más información sobre AWS las regiones compatibles, consulte los [puntos de enlace y las cuotas de Amazon Elastic File System](#) en Referencia general de AWS

Inicie una instancia de base de datos en Amazon RDS


Cuando inicia una instancia con Amazon RDS, es completamente independiente de Elastic Beanstalk y sus entornos Elastic Beanstalk, y Elastic Beanstalk no la finalizará ni supervisará.

En los siguientes pasos, utilizará la consola de Amazon RDS para:

- Lanzar una base de datos con el motor de MySQL.
- Habilitar una Multi-AZ deployment (Implementación Multi-AZ). Con esto se crea una instancia en espera en una zona de disponibilidad (AZ) diferente para proporcionar redundancia de datos, eliminar los bloqueos de E/S y minimizar los picos de latencia durante las copias de seguridad del sistema.

Para lanzar una instancia de base de datos de RDS en una VPC predeterminada

1. Abra la [consola de RDS](#).
2. En el panel de navegación, seleccione Databases (Bases de datos).
3. Elija Create database (Crear base de datos).
4. Elija Standard Create (Creación estándar).

 Important

No elija la opción Easy Create (Creación sencilla). Si lo elige, no puede configurar los ajustes necesarios para lanzar esta base de datos de RDS.

5. En Additional configuration (Configuración adicional), escriba **ebdb** en Initial database name (Nombre de base de datos inicial).
6. Revise la configuración predeterminada y ajústela de acuerdo con sus requisitos específicos. Preste atención a las siguientes opciones:
 - DB instance class (Clase de instancia de base de datos): elija un tamaño de instancia que cuente con la memoria y la potencia de CPU apropiadas para su carga de trabajo.
 - Multi-AZ deployment (Implementación Multi-AZ): para obtener alta disponibilidad, establezca esta opción en Create an Aurora Replica/Reader node in a different AZ (Crear una réplica de Aurora/nodo de lector en otra zona de disponibilidad).
 - Master username (Nombre de usuario maestro) y Master password (Contraseña maestra): nombre de usuario y contraseña de la base de datos. Anote esta configuración, ya que la necesitará posteriormente.
7. Compruebe los valores predeterminados de las demás opciones y, a continuación, elija Create database (Crear base de datos).

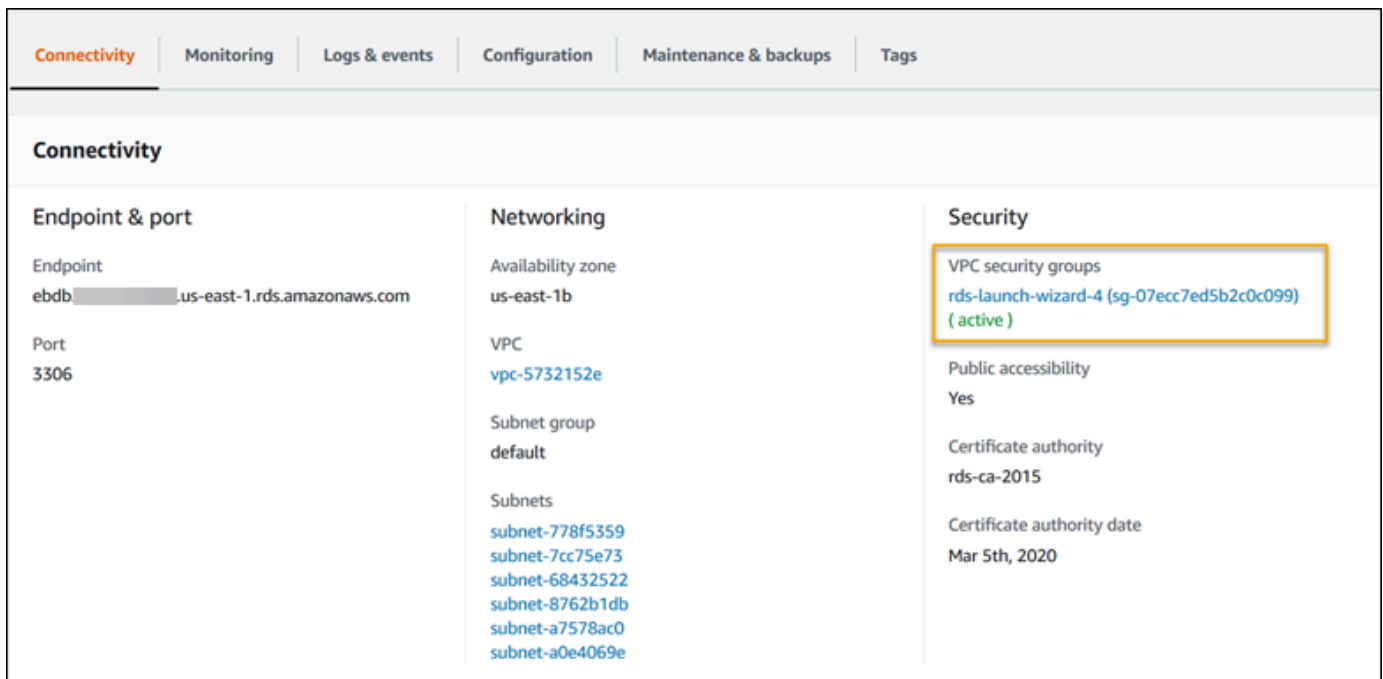
Una vez creada la instancia de base de datos, modifique el grupo de seguridad asociado para permitir el tráfico entrante en el puerto apropiado.

Note

Este es el mismo grupo de seguridad que asociará a su entorno de Elastic Beanstalk más adelante, por lo que la regla que añada ahora concederá permiso de entrada a otros recursos del mismo grupo de seguridad.

Para modificar las reglas de entrada del grupo de seguridad adjuntas a la instancia de RDS

1. Abra la [consola de Amazon RDS](#).
2. Seleccione Databases (Bases de datos).
3. Elija el nombre de la instancia de base de datos para ver sus detalles.
4. En la sección Connectivity (Conectividad), tome nota de los valores de Subnets (Subredes), Security groups (Grupos de seguridad) y Endpoint (Punto de conexión) que aparecen en esta página. Esto es para que pueda utilizar esta información más adelante.
5. En Security (Seguridad) se muestra el grupo de seguridad que está asociado a la instancia de base de datos. Abra el enlace para ver el grupo de seguridad en la consola de Amazon EC2.



The screenshot shows the Amazon RDS console interface. At the top, there are tabs for Connectivity, Monitoring, Logs & events, Configuration, Maintenance & backups, and Tags. The Connectivity tab is selected. Below the tabs, the 'Connectivity' section is displayed, divided into three columns: Endpoint & port, Networking, and Security. The Security column is highlighted with a yellow box, showing the VPC security group 'rds-launch-wizard-4 (sg-07ecc7ed5b2c0c099) (active)'. Other details in the Security column include Public accessibility (Yes), Certificate authority (rds-ca-2015), and Certificate authority date (Mar 5th, 2020).

6. En los detalles del grupo de seguridad, elija la pestaña Inbound (Entrada).
7. Elija Edit (Editar).

8. Seleccione Add Rule (Agregar regla).
9. En Type (Tipo), elija el motor de base de datos que utiliza la aplicación.
10. En Source (Origen), escriba **sg-** para ver una lista de grupos de seguridad disponibles. Elija el grupo de seguridad que está asociado al grupo de Auto Scaling que se utiliza con su entorno de Elastic Beanstalk. Esto es para que las instancias de Amazon EC2 en el entorno puedan acceder a la base de datos.



Edit inbound rules

Type	Protocol	Port Range	Source	Description
MYSQL/Auror.	TCP	3306	Custom 72.21.198.67/32	e.g. SSH for Admin Desktop
MYSQL/Auror.	TCP	3306	Custom sg-	e.g. SSH for Admin Desktop

NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

Cancel Save

11. Seleccione Save (Guardar).

Se tarda alrededor de 10 minutos en crear una instancia de base de datos. Mientras tanto, descargue WordPress y cree su entorno de Elastic Beanstalk.

Descarga WordPress

Para prepararse para la implementación WordPress mediante AWS Elastic Beanstalk, debe copiar los WordPress archivos en su ordenador y proporcionar la información de configuración correcta.

Para crear un WordPress proyecto

1. Descárgalo WordPress desde wordpress.org.

```
~$curl https://wordpress.org/wordpress-6.2.tar.gz -o wordpress.tar.gz
```

2. Descargue los archivos de configuración del repositorio de muestra.

```
~$ wget https://github.com/aws-samples/eb-php-wordpress/releases/download/v1.1/eb-php-wordpress-v1.zip
```

3. Extrae WordPress y cambia el nombre de la carpeta.


```
~$ tar -xvf wordpress.tar.gz
~$ mv wordpress wordpress-beanstalk
~$ cd wordpress-beanstalk
```

4. Extraiga los archivos de configuración de la WordPress instalación.

```
~/wordpress-beanstalk$ unzip ../eb-php-wordpress-v1.zip
creating: .ebextensions/
inflating: .ebextensions/dev.config
inflating: .ebextensions/efs-create.config
inflating: .ebextensions/efs-mount.config
inflating: .ebextensions/loadbalancer-sg.config
inflating: .ebextensions/wordpress.config
inflating: LICENSE
inflating: README.md
inflating: wp-config.php
```

Lance un entorno de Elastic Beanstalk

Utilice la consola de Elastic Beanstalk para crear un entorno de Elastic Beanstalk. Tras lanzar el entorno, puede configurarlo para que se conecte a la base de datos y, a continuación, implementar el WordPress código en el entorno.

En los siguientes pasos, utilizará la consola de Elastic Beanstalk para:

- Crear una aplicación de Elastic Beanstalk mediante la plataforma PHP administrada.
- Aceptar la configuración predeterminada y el código de muestra.

Para lanzar un entorno (consola)

1. [Abra la consola de Elastic Beanstalk mediante este enlace preconfigurado:](https://console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=Tutorials&EnvironmentType=LoadBalanced)
console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=Tutorials&EnvironmentType=LoadBalanced
2. En Platform (Plataforma), seleccione la plataforma y la ramificación de plataforma que coincidan con el idioma utilizado por la aplicación.
3. En Application code (Código de aplicación), seleccione Sample application (Aplicación de muestra).
4. Elija Review and launch (Revisar y lanzar).

5. Revise las opciones disponibles. Elija la opción disponible que quiera usar y, cuando esté listo, elija Create app (Crear aplicación).

El entorno tarda unos cinco minutos en crearse e incluye los siguientes recursos.


Elastic Beanstalk creó recursos

- Instancia de EC2: una máquina virtual de Amazon Elastic Compute Cloud (Amazon EC2) configurada para ejecutar aplicaciones web en la plataforma que elija.

Cada plataforma ejecuta un conjunto específico de software, archivos de configuración y scripts compatibles con una determinada versión de lenguaje, marco de trabajo y contenedor web (o una combinación de estos). La mayoría de las plataformas utilizan Apache o nginx como un proxy inverso que se sitúa delante de la aplicación web, reenvía las solicitudes a esta, administra los recursos estáticos y genera registros de acceso y errores.

- Instance security group (Grupo de seguridad de la instancia): grupo de seguridad de Amazon EC2 configurado para permitir el tráfico entrante en el puerto 80. Este recurso permite que el tráfico HTTP procedente del balanceador de carga llegue a la instancia EC2 en la que se ejecuta la aplicación web. De forma predeterminada, el tráfico no está permitido en otros puertos.
- Balanceador de carga: balanceador de carga de Elastic Load Balancing configurado para distribuir solicitudes a las instancias que se ejecutan en la aplicación. Los balanceadores de carga también permiten que las instancias no estén expuestas directamente a Internet.
- Grupo de seguridad del balanceador de carga: grupo de seguridad de Amazon EC2 configurado para permitir el tráfico entrante en el puerto 80. Este recurso permite que el tráfico HTTP procedente de Internet llegue al balanceador de carga. De forma predeterminada, el tráfico no está permitido en otros puertos.
- Grupo de Auto Scaling: grupo de Auto Scaling configurado para reemplazar una instancia si termina o deja de estar disponible.
- Bucket de Amazon S3: ubicación de almacenamiento para el código fuente, los registros y otros artefactos que se crean al utilizar Elastic Beanstalk.
- CloudWatch Alarmas de Amazon: dos CloudWatch alarmas que monitorean la carga de las instancias de su entorno y que se activan si la carga es demasiado alta o demasiado baja. Cuando se activa una alarma, en respuesta, el grupo de Auto Scaling aumenta o reduce los recursos.
- AWS CloudFormation pila: Elastic AWS CloudFormation Beanstalk se utiliza para lanzar los recursos de su entorno y propagar los cambios de configuración. Los recursos se definen en una plantilla que puede verse en la [consola de AWS CloudFormation](#).


- Nombre de dominio: nombre de dominio que direcciona el tráfico a la aplicación web con el formato *subdominio.región.elasticbeanstalk.com*.

 Note

Para aumentar la seguridad de las aplicaciones de Elastic Beanstalk, el dominio `elasticbeanstalk.com` está registrado en la [lista de sufijos públicos \(PSL\)](#). Para mayor seguridad, se recomienda que utilice cookies con un prefijo `__Host-` en caso de que necesite configurar cookies confidenciales en el nombre de dominio predeterminado de sus aplicaciones de Elastic Beanstalk. Esta práctica le ayudará a defender su dominio contra intentos de falsificación de solicitudes entre sitios (CSRF). Para obtener más información, consulte la página de [configuración de cookies](#) en la red de desarrolladores de Mozilla.

Todos estos recursos los administra Elastic Beanstalk. Cuando termina su entorno, Elastic Beanstalk termina todos los recursos que este contiene.

Dado que la instancia de base de datos de Amazon RDS que lanzó se encuentra fuera de su entorno, usted es responsable de administrar su ciclo de vida.

 Note

El bucket de Amazon S3 que Elastic Beanstalk crea se comparte entre los entornos y no se elimina durante la terminación del entorno. Para obtener más información, consulte [Usar Elastic Beanstalk con Amazon S3](#).


Configuración de grupos de seguridad y propiedades de entorno

Agregue el grupo de seguridad de la instancia de base de datos al entorno en ejecución. Este procedimiento hace que Elastic Beanstalk vuelva a aprovisionar todas las instancias de su entorno con el grupo de seguridad adicional asociado.

Para agregar un grupo de seguridad al entorno

- Realice una de las siguientes acciones siguientes:
 - Para añadir un grupo de seguridad mediante la consola de Elastic Beanstalk

- a. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
- b. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

 Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.


- c. En el panel de navegación, elija Configuration (Configuración).
 - d. En la categoría de configuración Instances (Instancias), elija Edit (Editar).
 - e. En EC2 security groups (Grupos de seguridad de EC2), elija el grupo de seguridad que desea asignar a las instancias, además del grupo de seguridad de instancia que crea Elastic Beanstalk.
 - f. Para guardar los cambios, elija Aplicar en la parte inferior de la página.
 - g. Lea la advertencia y, a continuación, elija Confirm (Confirmar).
- Para agregar un grupo de seguridad con un [archivo de configuración](#), utilice el archivo de ejemplo [securitygroup-addexisting.config](#).

Después, utilice las propiedades de entorno para pasar la información de la conexión al entorno.

La WordPress aplicación utiliza un conjunto predeterminado de propiedades que coinciden con las que Elastic Beanstalk configura al aprovisionar una base de datos en su entorno.

Para configurar las propiedades de entorno de una instancia de base de datos de Amazon RDS

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

 Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).

4. En la categoría de configuración Actualizaciones, supervisión y registro, seleccione Editar.
5. En la sección Environment properties (Propiedades del entorno), defina las variables que lee la aplicación para crear una cadena de conexión. Para conseguir compatibilidad con los entornos que tienen una instancia de base de datos de RDS, utilice los siguientes nombres y valores. Encontrará todos los valores, excepto la contraseña, en la [consola de RDS](#).

Nombre de la propiedad	Descripción	Valor de la propiedad
RDS_HOSTNAME	El nombre de host de la instancia de base de datos.	En la pestaña Connectivity & security (Conectividad y seguridad) de la consola de Amazon RDS: Endpoint (Punto de enlace).
RDS_PORT	El puerto en donde la instancia de base de datos acepta las conexiones. El valor predeterminado varía dependiendo del motor de base de datos.	En la pestaña Connectivity & security (Conectividad y seguridad) de la consola de Amazon RDS: Port (Puerto).
RDS_DB_NAME	El nombre de la base de dato, ebdb .	En la pestaña Configuration (Configuración) de la consola de Amazon RDS: DB Name (Nombre de la base de datos).
RDS_USERNAME	El nombre de usuario que ha configurado para la base de datos.	En la pestaña Configuration (Configuración) de la consola de Amazon RDS: Master username (Nombre de usuario maestro).
RDS_PASSWORD	La contraseña que ha configurado para la base de datos.	No está disponible como referencia en la consola de Amazon RDS.

Environment Properties

The following properties are passed into the application as environment variables. [Learn more.](#)

Property Name	Property Value
RDS_DB_NAME	<input type="text" value="ebdb"/> ✕
RDS_HOSTNAME	<input type="text" value="webapp-db.jxzc b5mpaniu.us-wes"/> ✕
RDS_PORT	<input type="text" value="5432"/> ✕
RDS_USERNAME	<input type="text" value="webapp-admin"/> ✕
<input type="text" value="RDS_PASSWORD"/>	<input type="text" value="kUj5uKxmWDMYc403"/> +

[Cancel](#) [Apply](#)

6. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

Configuración e implementación de la aplicación

Compruebe que la estructura de la carpeta `wordpress-beanstalk` es correcta, según se indica.

```
wordpress-beanstalk$ tree -aL 1
.
### .ebextensions
### index.php
### LICENSE
### license.txt
### readme.html
### README.md
### wp-activate.php
### wp-admin
### wp-blog-header.php
### wp-comments-post.php
### wp-config.php
### wp-config-sample.php
```


```
### wp-content
### wp-cron.php
### wp-includes
### wp-links-opml.php
### wp-load.php
### wp-login.php
### wp-mail.php
### wp-settings.php
### wp-signup.php
### wp-trackback.php
### xmlrpc.php
```

El archivo `wp-config.php` personalizado del repositorio del proyecto utiliza las variables de entorno definidas en el paso anterior para configurar la conexión a la base de datos. La carpeta `.ebextensions` contiene archivos de configuración que crean recursos adicionales en el entorno de Elastic Beanstalk.

Los archivos de configuración requieren modificaciones para que funcionen con su cuenta. Reemplace los valores de los marcadores de posición de los archivos por los ID apropiados y cree un paquete de código fuente.

Para actualizar los archivos de configuración y crear un paquete de código fuente.

1. Modifique los archivos de configuración como se indica a continuación.
 - `.ebextensions/dev.config`— Restringe el acceso a su entorno para protegerlo durante el proceso de instalación. WordPress Sustituya la dirección IP del marcador de posición situada en la parte superior del archivo por la dirección IP pública del ordenador que utilizará para acceder al sitio web de su entorno y completar la instalación. WordPress

 Note

En función de su red, es posible que tenga que utilizar un bloque de direcciones IP.

- `.ebextensions/efs-create.config`: crea un sistema de archivos EFS y puntos de montaje en cada zona de disponibilidad/subred de la VPC. Identifique los ID de la VPC y subred predeterminadas en la [consola de Amazon VPC](#).
2. Cree un [paquete de código fuente](#) que contenga los archivos de la carpeta del proyecto. El comando siguiente crea un paquete de código fuente denominado `wordpress-beanstalk.zip`.

```
~/eb-wordpress$ zip ../wordpress-beanstalk.zip -r * .[^.]*
```

Cargue el paquete de código fuente en Elastic Beanstalk WordPress para implementarlo en su entorno.

Para implementar un paquete de código fuente

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En la página de información general del entorno, elija Upload and deploy (Cargar e implementar).
4. Utilice el cuadro de diálogo en pantalla para cargar el paquete de código fuente.
5. Elija Deploy (Implementar).
6. Cuando se complete la implementación, seleccione la URL del para abrir el sitio web en una nueva pestaña.

Instale WordPress

Para completar la WordPress instalación

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. Elija la URL del entorno para abrir su sitio en un navegador. Se le redirigirá a un asistente WordPress de instalación porque aún no ha configurado el sitio.

4. Realice una instalación estándar. El archivo `wp-config.php` ya está presente en el código fuente y configurado para leer la información de conexión a la base de datos del entorno. No se le pedirá que configure la conexión.

La instalación tarda alrededor de un minuto en completarse.

Actualización de claves y sales

El archivo WordPress de configuración `wp-config.php` también lee los valores de las claves y sales de las propiedades del entorno. En la actualidad, estas propiedades están establecidas en `test` a través del archivo `wordpress.config` de la carpeta `.ebextensions`.

La sal hash puede ser cualquier valor que cumpla los [requisitos de las propiedades de entorno](#), pero no debe almacenarla en el control de código fuente. Utilice la consola de Elastic Beanstalk para establecer estas propiedades directamente en el entorno.

Para actualizar propiedades de entorno

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, seleccione Configuration (Configuración).
4. En Software, seleccione Edit (Editar).
5. En `Environment properties`, modifique las propiedades siguientes:
 - `AUTH_KEY`: el valor elegido para `AUTH_KEY`.
 - `SECURE_AUTH_KEY`: el valor elegido para `SECURE_AUTH_KEY`.
 - `LOGGED_IN_KEY`: el valor elegido para `LOGGED_IN_KEY`.
 - `NONCE_KEY`: el valor elegido para `NONCE_KEY`.
 - `AUTH_SALT`: el valor elegido para `AUTH_SALT`.
 - `SECURE_AUTH_SALT`: el valor elegido para `SECURE_AUTH_SALT`.
 - `LOGGED_IN_SALT`: el valor elegido para `LOGGED_IN_SALT`.

- `NONCE_SALT`: el valor elegido para `NONCE_SALT`.
6. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

Note

El establecimiento de las propiedades directamente en el entorno invalida los valores de `wordpress.config`.

Eliminación de las restricciones de acceso

El proyecto de muestra incluye el archivo de configuración `loadbalancer-sg.config`. Crea un grupo de seguridad y lo asigna al balanceador de carga del entorno, utilizando la dirección IP que configuró en `dev.config` y restringe el acceso HTTP en el puerto 80 a las conexiones procedentes de su red. De lo contrario, una persona externa podría conectarse a su sitio antes de que usted haya instalado WordPress y configurado su cuenta de administrador.

Ahora que lo has instalado WordPress, elimina el archivo de configuración para abrir el sitio al resto del mundo.

Para eliminar la restricción y actualizar el entorno

1. Elimine el archivo `.ebextensions/loadbalancer-sg.config` del directorio del proyecto.

```
~/wordpress-beanstalk$ rm .ebextensions/loadbalancer-sg.config
```

2. Cree un paquete de código fuente.

```
~/eb-wordpress$ zip ../wordpress-beanstalk-v2.zip -r * .[^.]*
```

Cargue el paquete de código fuente en Elastic Beanstalk WordPress para implementarlo en su entorno.

Para implementar un paquete de código fuente

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

 Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.


3. En la página de información general del entorno, elija Upload and deploy (Cargar e implementar).
4. Utilice el cuadro de diálogo en pantalla para cargar el paquete de código fuente.
5. Elija Deploy (Implementar).
6. Cuando se complete la implementación, seleccione la URL del para abrir el sitio web en una nueva pestaña.

Configure el grupo de Auto Scaling

Por último, configure el grupo de Auto Scaling de su entorno con un número mínimo de instancias mayor. Ejecute al menos dos instancias en todo momento para evitar que los servidores web del entorno se conviertan en un único punto de error. Esto también le permite implementar cambios sin que el sitio quede fuera de servicio.

Si desea configurar el grupo de Auto Scaling del entorno para disponer de una alta disponibilidad

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

 Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Capacity (Capacidad), elija Edit (Editar).
5. En la sección Auto Scaling group (grupo de Auto Scaling) defina Min instances (Instancias mínimas) en **2**.
6. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

Con el fin de admitir la carga de contenido en varias instancias, el proyecto de muestra utiliza Amazon EFS para crear un sistema de archivos compartidos. Cree una publicación en el sitio y cargue contenido para almacenarlo en el sistema de archivos compartidos. Consulte la publicación y actualice la página varias veces para utilizar ambas instancias y comprobar que el sistema de archivos compartidos funciona.

Actualice WordPress

Para actualizar a una nueva versión de WordPress, haga una copia de seguridad de su sitio e impleméntelo en un nuevo entorno.

Important

No utilices la funcionalidad de actualización incluida en los archivos fuente WordPress ni actualices una versión nueva para utilizarlos. Estas dos acciones puede hacer que las direcciones URL de publicación devuelvan un error 404 aunque sigan presentes en la base de datos y en el sistema de archivos.

Para actualizar WordPress


1. En la consola de WordPress administración, usa la herramienta de exportación para exportar tus publicaciones a un archivo XML.
2. Implemente e instale la nueva versión de WordPress en Elastic Beanstalk siguiendo los mismos pasos que utilizó para instalar la versión anterior. Para evitar el tiempo de inactividad, puede crear un entorno con la nueva versión.
3. En la nueva versión, instale la herramienta WordPress Importer en la consola de administración y utilícela para importar el archivo XML que contiene las publicaciones. Si el usuario administrador creó las publicaciones en la versión antigua, asígneselas al usuario administrador en el nuevo sitio, en lugar de intentar importar el usuario administrador.
4. Si ha implementado la nueva versión en un entorno distinto, lleve a cabo un [intercambio de CNAME](#) para redirigir a los usuarios del sitio anterior al nuevo.

Eliminar recursos

Cuando termine de trabajar con Elastic Beanstalk, puede terminar su entorno. [Elastic Beanstalk AWS cancela todos los recursos asociados al entorno, como las instancias de Amazon EC2, las instancias de bases de datos, los balanceadores de carga, los grupos de seguridad y las alarmas.](#)

Para cerrar el entorno de Elastic Beanstalk desde la consola

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

 Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. Elija Actions (Acciones) y, a continuación, Terminate Environment (Terminar el entorno).
4. Utilice el cuadro de diálogo en pantalla para confirmar la terminación del entorno.

Con Elastic Beanstalk, puede crear fácilmente un nuevo entorno para su aplicación en cualquier momento.

Además, puede terminar los recursos de base de datos creados fuera del entorno de Elastic Beanstalk. Cuando termine una instancia de base de datos de Amazon RDS, puede hacer una instantánea para restaurar los datos posteriormente en otra instancia.

Para terminar la instancia de base de datos de RDS

1. Abra la [consola de Amazon RDS](#).
2. Seleccione Databases (Bases de datos).
3. Seleccione la instancia de base de datos.
4. Elija Acciones y, a continuación, elija Eliminar.
5. Decida si desea crear una instantánea y seleccione Delete (Eliminar).

Siguientes pasos

A medida que siga desarrollando la aplicación, es probable que desee contar con un mecanismo que le permita administrar los entornos e implementar la aplicación sin tener que crear manualmente un archivo .zip y cargarlo en la consola de Elastic Beanstalk. La interfaz de [línea de comandos de Elastic Beanstalk](#) (EB CLI easy-to-use) proporciona comandos para crear, configurar e implementar aplicaciones en entornos de Elastic Beanstalk desde la línea de comandos.

La aplicación de ejemplo utiliza archivos de configuración para configurar las opciones de PHP y crear una tabla en la base de datos si no existe todavía. También puede utilizar un archivo de configuración para configurar el grupo de seguridad de las instancias durante la creación del entorno y evitar así la laboriosa tarea de actualizar la configuración. Para obtener más información, consulte [Personalización avanzada de entornos con archivos de configuración \(.ebextensions\)](#).

En las tareas de desarrollo y pruebas, es posible que desee utilizar la funcionalidad de Elastic Beanstalk para agregar una instancia de base de datos administrada directamente al entorno. Para obtener instrucciones acerca de cómo configurar una base de datos dentro del entorno, consulte [Añadir una base de datos al entorno de Elastic Beanstalk](#).

Si necesita una base de datos de alto desempeño, considere la posibilidad de usar [Amazon Aurora](#). Amazon Aurora es un motor de base de datos compatible con MySQL que ofrece funciones de bases de datos comerciales a bajo costo. Para conectar la aplicación a una base de datos diferente, repita los pasos de la [configuración del grupo de seguridad](#) y [actualice las propiedades de entorno relacionadas con RDS](#).

Por último, si pretende utilizar la aplicación en un entorno de producción, querrá [configurar un nombre de dominio personalizado](#) para el entorno y [habilitar HTTPS](#) para las conexiones seguras.

Implementación de un sitio web de Drupal de alta disponibilidad con una base de datos externa de Amazon RDS en Elastic Beanstalk

En este tutorial se explica el proceso de [lanzamiento de una instancia de base de datos de RDS](#) externa a AWS Elastic Beanstalk. A continuación, se describe cómo configurar un entorno de alta disponibilidad con un sitio web de Drupal para que se conecte a esa instancia. El sitio web utiliza Amazon Elastic File System (Amazon EFS) como almacenamiento compartido para los archivos cargados. La ejecución de una instancia de base de datos externa a Elastic Beanstalk desacopla la base de datos del ciclo de vida de su entorno y le permite conectarse a la misma base de datos desde varios entornos, reemplazar una base de datos por otra o realizar una implementación blue/green sin que ello afecte a la base de datos.

Secciones

- [Requisitos previos](#)
- [Inicie una instancia de base de datos en Amazon RDS](#)
- [Lance un entorno de Elastic Beanstalk](#)
- [Configuración de ajustes de seguridad y propiedades de entorno](#)
- [Configuración e implementación de la aplicación](#)

- [Instalación de Drupal](#)
- [Actualización de la configuración de Drupal y supresión de las restricciones de acceso](#)
- [Configure el grupo de Auto Scaling](#)
- [Limpieza](#)
- [Siguiendo pasos](#)

Requisitos previos

Este tutorial asume que tiene conocimiento de las operaciones básicas de Elastic Beanstalk y de la consola de Elastic Beanstalk. Si aún no lo ha hecho, siga las instrucciones que se indican en [Introducción a Elastic Beanstalk](#) para lanzar su primer entorno de Elastic Beanstalk.

Para seguir los procedimientos de esta guía, necesitará un shell o un terminal de línea de comando donde pueda ejecutar los comandos. Los comandos aparecen en listas y van precedidos del símbolo del sistema (\$) y del nombre del directorio actual, si es aplicable.

```
~/eb-project$ this is a command  
this is output
```

En Linux y macOS, puede utilizar el administrador de shell y paquetes preferido. En Windows, puede [instalar el subsistema de Windows para Linux](#) y obtener una versión de Ubuntu y Bash integrada en Windows.

Los procedimientos de este tutorial para las tareas de Amazon Relational Database Service (Amazon RDS) suponen que está iniciando recursos en una [Amazon Virtual Private Cloud](#) (Amazon VPC) predeterminada. Todas las nuevas cuentas contienen una VPC predeterminada en cada región. Si no dispone de una VPC predeterminada, los procedimientos serán diferentes. Consulte [Uso de Elastic Beanstalk con Amazon RDS](#) para obtener instrucciones sobre la plataforma de EC2 y de una VPC personalizada.

La aplicación de ejemplo utiliza Amazon EFS. Solo funciona en AWS las regiones que admiten Amazon EFS. Para obtener más información sobre AWS las regiones compatibles, consulte los [puntos de enlace y las cuotas de Amazon Elastic File System](#) en la Referencia general de AWS.

Si la plataforma del entorno Elastic Beanstalk utiliza PHP 7.4 o anterior, recomendamos que utilice la versión 8.9.13 de Drupal para este tutorial. Para plataformas instaladas con PHP 8.0 o posterior, recomendamos usar Drupal 9.1.5.

Para obtener más información acerca de las versiones de Drupal y las versiones de PHP que soportan, consulte [los requisitos de PHP](#) en el sitio web de Drupal. Las versiones principales que Drupal recomienda se enumeran en el sitio web <https://www.drupal.org/project/drupal>.

Inicie una instancia de base de datos en Amazon RDS

Para utilizar una base de datos externa con una aplicación que se ejecuta en Elastic Beanstalk, primero lance una instancia de base de datos con Amazon RDS. Cuando inicia una instancia con Amazon RDS, es completamente independiente de Elastic Beanstalk y sus entornos Elastic Beanstalk, y Elastic Beanstalk no la finalizará ni supervisará.

Use la consola de Amazon RDS para lanzar una instancia de base de datos MySQL Multi-AZ. La elección de una implementación Multi-AZ garantiza que la base de datos realice una conmutación por error y siga estando disponible en caso de que la instancia de base de datos de origen quede fuera de servicio.

Para lanzar una instancia de base de datos de RDS en una VPC predeterminada

1. Abra la [consola de RDS](#).
2. En el panel de navegación, seleccione Databases (Bases de datos).
3. Elija Create database (Crear base de datos).
4. Elija Standard Create (Creación estándar).

 Important

No elija la opción Easy Create (Creación sencilla). Si lo elige, no puede configurar los ajustes necesarios para lanzar esta base de datos de RDS.

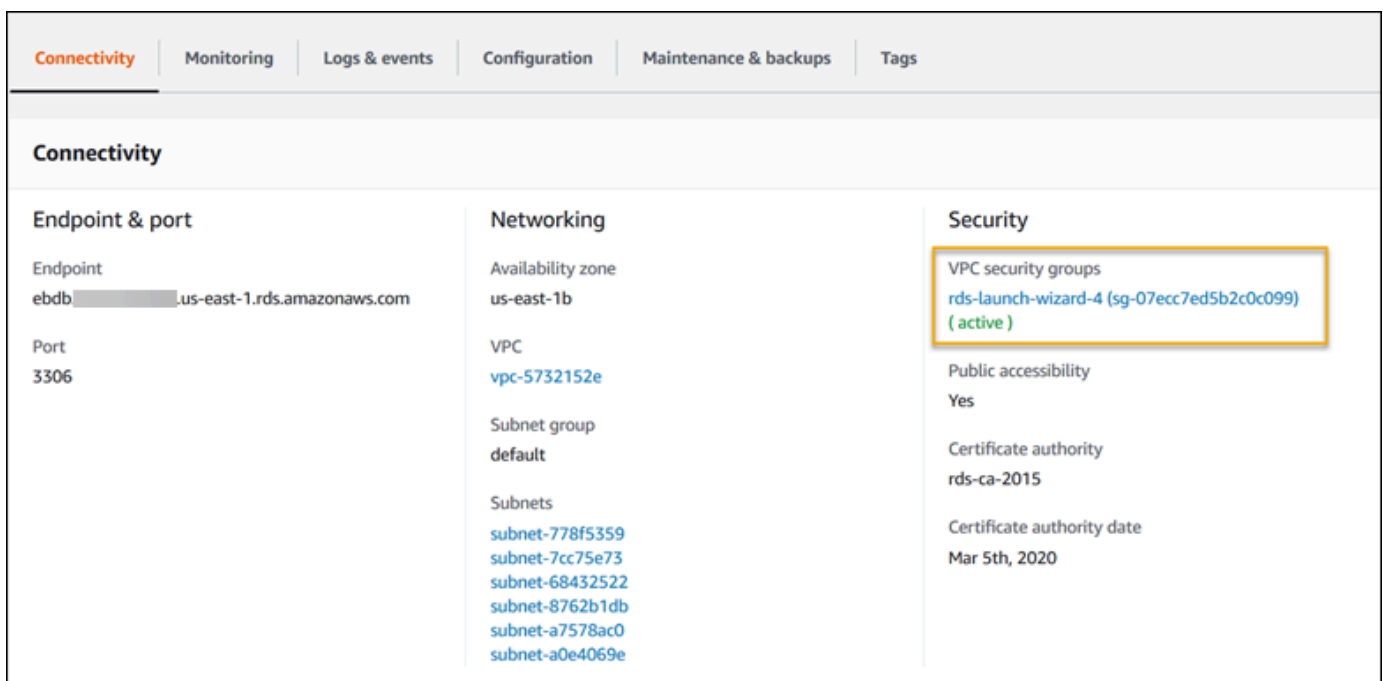
5. En Additional configuration (Configuración adicional), escriba **ebdb** en Initial database name (Nombre de base de datos inicial).
6. Revise la configuración predeterminada y ajústela de acuerdo con sus requisitos específicos. Preste atención a las siguientes opciones:
 - DB instance class (Clase de instancia de base de datos): elija un tamaño de instancia que cuente con la memoria y la potencia de CPU apropiadas para su carga de trabajo.
 - Multi-AZ deployment (Implementación Multi-AZ): para obtener alta disponibilidad, establezca esta opción en Create an Aurora Replica/Reader node in a different AZ (Crear una réplica de Aurora/nodo de lector en otra zona de disponibilidad).

- Master username (Nombre de usuario maestro) y Master password (Contraseña maestra): nombre de usuario y contraseña de la base de datos. Anote esta configuración, ya que la necesitará posteriormente.
7. Compruebe los valores predeterminados de las demás opciones y, a continuación, elija Create database (Crear base de datos).

A continuación, modifique el grupo de seguridad asociado a la instancia de base de datos para permitir el tráfico entrante en el puerto correspondiente. Este es el mismo grupo de seguridad que asociará a su entorno de Elastic Beanstalk más adelante, por lo que la regla que añada concederá permiso de entrada a otros recursos del mismo grupo de seguridad.

Para modificar las reglas de entrada del grupo de seguridad adjuntas a la instancia de RDS

1. Abra la [consola de Amazon RDS](#).
2. Seleccione Databases (Bases de datos).
3. Elija el nombre de la instancia de base de datos para ver sus detalles.
4. En la sección Connectivity (Conectividad), tome nota de los valores de Subnets (Subredes), Security groups (Grupos de seguridad) y Endpoint (Punto de conexión) que aparecen en esta página. Esto es para que pueda utilizar esta información más adelante.
5. En Security (Seguridad) se muestra el grupo de seguridad que está asociado a la instancia de base de datos. Abra el enlace para ver el grupo de seguridad en la consola de Amazon EC2.



The screenshot displays the Amazon RDS console interface for a database instance. The top navigation bar includes tabs for Connectivity, Monitoring, Logs & events, Configuration, Maintenance & backups, and Tags. The Connectivity tab is selected. The main content area is divided into three sections: Endpoint & port, Networking, and Security. The Security section is highlighted with a yellow box, showing the VPC security group 'rds-launch-wizard-4 (sg-07ecc7ed5b2c0c099)' which is active. Other details visible include the endpoint 'ebdb-...us-east-1.rds.amazonaws.com', port '3306', availability zone 'us-east-1b', VPC 'vpc-5732152e', and subnets 'subnet-778f5359', 'subnet-7cc75e73', 'subnet-68432522', 'subnet-8762b1db', 'subnet-a7578ac0', and 'subnet-a0e4069e'.

6. En los detalles del grupo de seguridad, elija la pestaña Inbound (Entrada).
7. Elija Edit (Editar).
8. Seleccione Add Rule (Agregar regla).
9. En Type (Tipo), elija el motor de base de datos que utiliza la aplicación.
10. En Source (Origen), escriba **sg-** para ver una lista de grupos de seguridad disponibles. Elija el grupo de seguridad que está asociado al grupo de Auto Scaling que se utiliza con su entorno de Elastic Beanstalk. Esto es para que las instancias de Amazon EC2 en el entorno puedan acceder a la base de datos.

The screenshot shows the 'Edit inbound rules' dialog in the AWS console. It features a table with columns: Type, Protocol, Port Range, Source, and Description. Two rules are listed, both for MySQL/Aurora on TCP port 3306. The 'Source' field of the second rule is active, showing a dropdown menu with 'sg-' selected. A search result 'sg-07ecc7ed5b2c0c099 - rds-launch-wizard-4' is highlighted in a dark grey box. Below the table is an 'Add Rule' button and a note: 'NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.' At the bottom right are 'Cancel' and 'Save' buttons.

11. Seleccione Save (Guardar).

Se tarda alrededor de 10 minutos en crear una instancia de base de datos. Mientras tanto, inicie el entorno de Elastic Beanstalk.

Lance un entorno de Elastic Beanstalk

Utilice la consola de Elastic Beanstalk para crear un entorno de Elastic Beanstalk. Elija la plataforma PHP y acepte la configuración predeterminada y el código de muestra. Después de iniciar el entorno, puede configurar el entorno para que se conecte a la base de datos e implementar el código Drupal en el entorno.

Para lanzar un entorno (consola)

1. [Abra la consola de Elastic Beanstalk mediante este enlace preconfigurado:](https://console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=Tutorials&EnvironmentType=LoadBalanced)
[console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=Tutorials
&EnvironmentType=LoadBalanced](https://console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=Tutorials&EnvironmentType=LoadBalanced)
2. En Platform (Plataforma), seleccione la plataforma y la ramificación de plataforma que coincidan con el idioma utilizado por la aplicación.

3. En Application code (Código de aplicación), seleccione Sample application (Aplicación de muestra).
4. Elija Review and launch (Revisar y lanzar).
5. Revise las opciones disponibles. Elija la opción disponible que quiera usar y, cuando esté listo, elija Create app (Crear aplicación).


El entorno tarda unos cinco minutos en crearse e incluye los recursos siguientes:

- Instancia de EC2: una máquina virtual de Amazon Elastic Compute Cloud (Amazon EC2) configurada para ejecutar aplicaciones web en la plataforma que elija.

Cada plataforma ejecuta un conjunto específico de software, archivos de configuración y scripts compatibles con una determinada versión de lenguaje, marco de trabajo y contenedor web (o una combinación de estos). La mayoría de las plataformas utilizan Apache o nginx como un proxy inverso que se sitúa delante de la aplicación web, reenvía las solicitudes a esta, administra los recursos estáticos y genera registros de acceso y errores.


- Instance security group (Grupo de seguridad de la instancia): grupo de seguridad de Amazon EC2 configurado para permitir el tráfico entrante en el puerto 80. Este recurso permite que el tráfico HTTP procedente del balanceador de carga llegue a la instancia EC2 en la que se ejecuta la aplicación web. De forma predeterminada, el tráfico no está permitido en otros puertos.
- Balanceador de carga: balanceador de carga de Elastic Load Balancing configurado para distribuir solicitudes a las instancias que se ejecutan en la aplicación. Los balanceadores de carga también permiten que las instancias no estén expuestas directamente a Internet.
- Grupo de seguridad del balanceador de carga: grupo de seguridad de Amazon EC2 configurado para permitir el tráfico entrante en el puerto 80. Este recurso permite que el tráfico HTTP procedente de Internet llegue al balanceador de carga. De forma predeterminada, el tráfico no está permitido en otros puertos.
- Grupo de Auto Scaling: grupo de Auto Scaling configurado para reemplazar una instancia si termina o deja de estar disponible.
- Bucket de Amazon S3: ubicación de almacenamiento para el código fuente, los registros y otros artefactos que se crean al utilizar Elastic Beanstalk.
- CloudWatch Alarmas de Amazon: dos CloudWatch alarmas que monitorean la carga de las instancias de su entorno y que se activan si la carga es demasiado alta o demasiado baja. Cuando se activa una alarma, en respuesta, el grupo de Auto Scaling aumenta o reduce los recursos.

- AWS CloudFormation pila: Elastic AWS CloudFormation Beanstalk se utiliza para lanzar los recursos de su entorno y propagar los cambios de configuración. Los recursos se definen en una plantilla que puede verse en la [consola de AWS CloudFormation](#).
- Nombre de dominio: nombre de dominio que direcciona el tráfico a la aplicación web con el formato *subdominio.región.elasticbeanstalk.com*.

 Note

Para aumentar la seguridad de las aplicaciones de Elastic Beanstalk, el dominio elasticbeanstalk.com está registrado en la [lista de sufijos públicos \(PSL\)](#). Para mayor seguridad, se recomienda que utilice cookies con un prefijo __Host- en caso de que necesite configurar cookies confidenciales en el nombre de dominio predeterminado de sus aplicaciones de Elastic Beanstalk. Esta práctica le ayudará a defender su dominio contra intentos de falsificación de solicitudes entre sitios (CSRF). Para obtener más información, consulte la página de [configuración de cookies](#) en la red de desarrolladores de Mozilla.

Todos estos recursos los administra Elastic Beanstalk. Cuando termina su entorno, Elastic Beanstalk termina todos los recursos que este contiene. La instancia de base de datos de RDS que lanzó se encuentra fuera de su entorno, por lo que usted es responsable de administrar su ciclo de vida.

 Note

El bucket de Amazon S3 que Elastic Beanstalk crea se comparte entre los entornos y no se elimina durante la terminación del entorno. Para obtener más información, consulte [Usar Elastic Beanstalk con Amazon S3](#).


Configuración de ajustes de seguridad y propiedades de entorno

Agregue el grupo de seguridad de la instancia de base de datos al entorno en ejecución. Este procedimiento hace que Elastic Beanstalk vuelva a aprovisionar todas las instancias de su entorno con el grupo de seguridad adicional asociado.

Para agregar un grupo de seguridad al entorno

- Realice una de las siguientes acciones siguientes:

- Para añadir un grupo de seguridad mediante la consola de Elastic Beanstalk
 - a. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
 - b. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

 Note


Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

- c. En el panel de navegación, elija Configuration (Configuración).
 - d. En la categoría de configuración Instances (Instancias), elija Edit (Editar).
 - e. En EC2 security groups (Grupos de seguridad de EC2), elija el grupo de seguridad que desea asignar a las instancias, además del grupo de seguridad de instancia que crea Elastic Beanstalk.
 - f. Para guardar los cambios, elija Aplicar en la parte inferior de la página.
 - g. Lea la advertencia y, a continuación, elija Confirm (Confirmar).
- Para agregar un grupo de seguridad con un [archivo de configuración](#), utilice el archivo de ejemplo [securitygroup-addexisting.config](#).

Después, utilice las propiedades de entorno para pasar la información de la conexión al entorno. La aplicación de ejemplo utiliza un conjunto predeterminado de propiedades que coinciden con las que Elastic Beanstalk configura al aprovisionar una base de datos en su entorno.

Para configurar las propiedades de entorno de una instancia de base de datos de Amazon RDS

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

 Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Actualizaciones, supervisión y registro, seleccione Editar.
5. En la sección Environment properties (Propiedades del entorno), defina las variables que lee la aplicación para crear una cadena de conexión. Para conseguir compatibilidad con los entornos que tienen una instancia de base de datos de RDS, utilice los siguientes nombres y valores. Encontrará todos los valores, excepto la contraseña, en la [consola de RDS](#).

Nombre de la propiedad	Descripción	Valor de la propiedad
RDS_HOSTNAME	El nombre de host de la instancia de base de datos.	En la pestaña Connectivity & security (Conectividad y seguridad) de la consola de Amazon RDS: Endpoint (Punto de enlace).
RDS_PORT	El puerto en donde la instancia de base de datos acepta las conexiones. El valor predeterminado varía dependiendo del motor de base de datos.	En la pestaña Connectivity & security (Conectividad y seguridad) de la consola de Amazon RDS: Port (Puerto).
RDS_DB_NAME	El nombre de la base de dato, ebdb .	En la pestaña Configuration (Configuración) de la consola de Amazon RDS: DB Name (Nombre de la base de datos).
RDS_USERNAME	El nombre de usuario que ha configurado para la base de datos.	En la pestaña Configuration (Configuración) de la consola de Amazon RDS: Master username (Nombre de usuario maestro).
RDS_PASSWORD	La contraseña que ha configurado para la base de datos.	No está disponible como referencia en la consola de Amazon RDS.

Environment Properties

The following properties are passed into the application as environment variables. [Learn more.](#)

Property Name	Property Value
RDS_DB_NAME	<input type="text" value="ebdb"/> ✕
RDS_HOSTNAME	<input type="text" value="webapp-db.jxzc b5mpaniu.us-wes"/> ✕
RDS_PORT	<input type="text" value="5432"/> ✕
RDS_USERNAME	<input type="text" value="webapp-admin"/> ✕
<input type="text" value="RDS_PASSWORD"/>	<input type="text" value="kUj5uKxmWDMYc403"/> +

[Cancel](#) [Apply](#)

6. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

Después de instalar Drupal, tiene que conectarse a la instancia con SSH para recuperar algunos detalles de configuración. Asigne una clave SSH a las instancias del entorno.

Para configurar SSH

1. Si no ha creado previamente un par de claves, abra la [página de pares de claves](#) de la consola de Amazon EC2; y siga las instrucciones para crear uno.
2. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
3. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

4. En el panel de navegación, elija Configuration (Configuración).

5. En Security (Seguridad), seleccione Edit (Editar).
6. En EC2 key pair (Par de claves de EC2), seleccione su par de claves.
7. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

Configuración e implementación de la aplicación

[Para crear un proyecto de Drupal para Elastic Beanstalk, descargue el código fuente de Drupal y combínelo con los archivos del repositorio aws-samples/ de eb-php-drupal](#) GitHub

Para crear un proyecto de Drupal

1. Ejecute el siguiente comando para descargar Drupal desde www.drupal.org/download. Para obtener más información sobre las descargas, consulte [el sitio web de Drupal](#).

Si la plataforma del entorno Elastic Beanstalk utiliza PHP 7.4 o anterior, recomendamos que descargue la versión 8.9.13 de Drupal para este tutorial. Puede ejecutar el siguiente comando para descargarlo.

```
~$ curl https://ftp.drupal.org/files/projects/drupal-8.9.13.tar.gz -o drupal.tar.gz
```

Si la plataforma utiliza PHP 8.0 o posterior, recomendamos que descargue Drupal 9.1.5. Puede usar este comando para descargarlo.

```
~$ curl https://ftp.drupal.org/files/projects/drupal-9.1.5.tar.gz -o drupal.tar.gz
```

Para obtener más información acerca de las versiones de Drupal y las versiones de PHP que soportan, consulte [los requisitos de PHP](#) en la documentación oficial de Drupal. Las versiones principales que Drupal recomienda se enumeran en [el sitio web de Drupal](#).

2. Utilice el siguiente comando para descargar los archivos de configuración del repositorio de ejemplo:

```
~$ wget https://github.com/aws-samples/eb-php-drupal/releases/download/v1.1/eb-php-drupal-v1.zip
```

3. Extraiga Drupal y cambie el nombre de la carpeta.

Si descargó Drupal 8.9.13:


```
~$ tar -xvf drupal.tar.gz
~$ mv drupal-8.9.13 drupal-beanstalk
~$ cd drupal-beanstalk
```

Si descargó Drupal 9.1.5:

```
~$ tar -xvf drupal.tar.gz
~$ mv drupal-9.1.5 drupal-beanstalk
~$ cd drupal-beanstalk
```

4. Extraiga los archivos de configuración a través de la instalación de Drupal.

```
~/drupal-beanstalk$ unzip ../eb-php-drupal-v1.zip
creating: .ebextensions/
inflating: .ebextensions/dev.config
inflating: .ebextensions/drupal.config
inflating: .ebextensions/efs-create.config
inflating: .ebextensions/efs-filesystem.template
inflating: .ebextensions/efs-mount.config
inflating: .ebextensions/loadbalancer-sg.config
inflating: LICENSE
inflating: README.md
inflating: beanstalk-settings.php
```

Compruebe que la estructura de la carpeta `drupal-beanstalk` es correcta, según se indica.

```
drupal-beanstalk$ tree -aL 1
.
### autoload.php
### beanstalk-settings.php
### composer.json
### composer.lock
### core
### .csslintrc
### .ebextensions
### .ebextensions
### .editorconfig
### .eslintignore
### .eslintrc.json
### example.gitignore
```

```
### .gitattributes
### .htaccess
### .ht.router.php
### index.php
### LICENSE
### LICENSE.txt
### modules
### profiles
### README.md
### README.txt
### robots.txt
### sites
### themes
### update.php
### vendor
### web.config
```

El archivo `beanstalk-settings.php` del repositorio del proyecto utiliza las variables de entorno definidas en el paso anterior para configurar la conexión a la base de datos. La carpeta `.ebextensions` contiene archivos de configuración que crean recursos adicionales en el entorno de Elastic Beanstalk.

Los archivos de configuración requieren modificaciones para que funcionen con su cuenta. Reemplace los valores de los marcadores de posición de los archivos por los ID apropiados y cree un paquete de código fuente.

Para actualizar los archivos de configuración y crear un paquete de código fuente.


1. Modifique los archivos de configuración como se indica a continuación.
 - `.ebextensions/dev.config`: limita a su dirección IP el acceso a su entorno con el fin de protegerlo durante el proceso de instalación de Drupal. Cambie la dirección IP del marcador de posición situada cerca de la parte superior del archivo por la dirección IP pública.
 - `.ebextensions/efs-create.config`: crea un sistema de archivos EFS y puntos de montaje en cada zona de disponibilidad/subred de la VPC. Identifique los ID de la VPC y subred predeterminadas en la [consola de Amazon VPC](#).
2. Cree un [paquete de código fuente](#) que contenga los archivos de la carpeta del proyecto. El comando siguiente crea un paquete de código fuente denominado `drupal-beanstalk.zip`. Excluye los archivos de la carpeta `vendor`, que ocupan mucho espacio y no se requieren para implementar la aplicación en Elastic Beanstalk.

```
~/eb-drupal$ zip ../drupal-beanstalk.zip -r * .[^.]* -x "vendor/*"
```

Cargue el paquete de origen en Elastic Beanstalk para implementar Drupal en su entorno.

Para implementar un paquete de código fuente

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

 Note


Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En la página de información general del entorno, elija Upload and deploy (Cargar e implementar).
4. Utilice el cuadro de diálogo en pantalla para cargar el paquete de código fuente.
5. Elija Deploy (Implementar).
6. Cuando se complete la implementación, seleccione la URL del para abrir el sitio web en una nueva pestaña.

Instalación de Drupal

Para completar la instalación de Drupal

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

 Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. Elija la URL del entorno para abrir su sitio en un navegador. Se abrirá un asistente de instalación de Drupal, ya que aún no se ha configurado el sitio.
4. Realice una instalación estándar de la base de datos con la siguiente configuración:

- Database name (Nombre de la base de datos): el nombre de base de datos que se muestra en la consola de Amazon RDS.
- Database username and password: valores de Master Username (Nombre de usuario maestro) y Master Password (Contraseña maestra) que especificó al crear la base de datos.
- Advanced Options > Host (Opciones avanzadas > Host): el punto de enlace de la instancia de base de datos que se muestra en la consola de Amazon RDS.

La instalación tarda alrededor de un minuto en completarse.

Actualización de la configuración de Drupal y supresión de las restricciones de acceso

Durante el proceso de instalación de Drupal, se crea un archivo denominado `settings.php` en la carpeta `sites/default` de la instancia. Este archivo se necesita en el código fuente para evitar restablecer el sitio en las implementaciones subsiguientes; sin embargo, el archivo contiene secretos que no es conveniente incorporar al código fuente. Conéctese a la instancia de aplicación para recuperar la información del archivo de configuración.

Para conectarse a la instancia de aplicación con SSH

1. Abra la [página de instancias](#) de la consola de Amazon EC2.
2. Elija la instancia de aplicación. Es la que tiene el nombre de su entorno de Elastic Beanstalk.
3. Elija Conectar.
4. Siga las instrucciones para conectar la instancia con SSH. El comando tiene un aspecto similar al siguiente.

```
$ ssh -i ~/.ssh/mykey ec2-user@ec2-00-55-33-222.us-west-2.compute.amazonaws.com
```

Obtenga el ID del directorio de sincronización, que aparece en la última línea del archivo de configuración.

```
[ec2-user ~]$ tail -n 1 /var/app/current/sites/default/settings.php
$config_directories['sync'] = 'sites/default/files/
config_4ccfX2sPQm79p1mk5IbUq9S_FokcEN04mxyC-L18-4g_xKj_7j9ydn31kD0Y0gnzMu071Tvc4Q/
sync';
```

El archivo también contiene la clave de hash actual de los sitios, pero puede hacer caso omiso del valor actual y utilizar el suyo propio.

Asigne la ruta del directorio de sincronización y la clave de hash a las propiedades de entorno. El archivo de configuración personalizado del repositorio del proyecto leerá estas propiedades para configurar el sitio durante la implementación, además de las propiedades de conexión de la base de datos establecidas anteriormente.

Propiedades de configuración de Drupal

- SYNC_DIR: la ruta al directorio de sincronización.
- HASH_SALT: cualquier valor de tipo cadena que cumpla los [requisitos de las propiedades del entorno](#).

Para configurar las propiedades del entorno en la consola de Elastic Beanstalk

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Actualizaciones, supervisión y registro, seleccione Editar.
5. Desplácese hacia abajo hasta Propiedades del entorno.
6. Seleccione Agregar propiedad de entorno.
7. Introduzca los pares Nombre y Valor de la propiedad.
8. Si necesita agregar más variables, repita Paso 6 y Paso 7.
9. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

Por último, el proyecto de muestra incluye un archivo de configuración (`loadbalancer-sg.config`) que crea un grupo de seguridad y lo asigna al balanceador de carga del entorno usando la dirección IP que configuró en `dev.config` para restringir el acceso HTTP en el puerto 80

a las conexiones procedentes de su red. De no ser así, una entidad externa podría conectarse al sitio antes de que se instalase Drupal y se hubiese configurado la cuenta de administrador.

Para actualizar la configuración de Drupal y suprimir las restricciones de acceso

1. Elimine el archivo `.ebextensions/loadbalancer-sg.config` del directorio del proyecto.

```
~/drupal-beanstalk$ rm .ebextensions/loadbalancer-sg.config
```

2. Copie el archivo `settings.php` personalizado en la carpeta de sitios.

```
~/drupal-beanstalk$ cp beanstalk-settings.php sites/default/settings.php
```

3. Cree un paquete de código fuente.

```
~/eb-drupal$ zip ../drupal-beanstalk-v2.zip -r * .[^.]* -x "vendor/*"
```

Cargue el paquete de origen en Elastic Beanstalk para implementar Drupal en su entorno.

Para implementar un paquete de código fuente

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En la página de información general del entorno, elija Upload and deploy (Cargar e implementar).
4. Utilice el cuadro de diálogo en pantalla para cargar el paquete de código fuente.
5. Elija Deploy (Implementar).
6. Cuando se complete la implementación, seleccione la URL del para abrir el sitio web en una nueva pestaña.

Configure el grupo de Auto Scaling

Por último, configure el grupo de Auto Scaling de su entorno con un número mínimo de instancias mayor. Ejecute al menos dos instancias en todo momento para evitar que los servidores web de su entorno se conviertan en un único punto de error y para poder implementar cambios sin que el sitio se quede fuera de servicio.

Si desea configurar el grupo de Auto Scaling del entorno para disponer de una alta disponibilidad

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Capacity (Capacidad), elija Edit (Editar).
5. En la sección Auto Scaling group (grupo de Auto Scaling) defina Min instances (Instancias mínimas) en **2**.
6. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

Con el fin de admitir la carga de contenido en varias instancias, el proyecto de muestra utiliza Amazon Elastic File System para crear un sistema de archivos compartidos. Cree una publicación en el sitio y cargue contenido para almacenarlo en el sistema de archivos compartidos. Consulte la publicación y actualice la página varias veces para utilizar ambas instancias y comprobar que el sistema de archivos compartidos funciona.

Limpieza

Cuando termine de trabajar con Elastic Beanstalk, puede terminar su entorno. [Elastic Beanstalk AWS cancela todos los recursos asociados al entorno, como las instancias de Amazon EC2, las instancias de bases de datos, los balanceadores de carga, los grupos de seguridad y las alarmas.](#)

Para cerrar el entorno de Elastic Beanstalk desde la consola

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS

2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

 Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. Elija Actions (Acciones) y, a continuación, Terminate Environment (Terminar el entorno).
4. Utilice el cuadro de diálogo en pantalla para confirmar la terminación del entorno.

Con Elastic Beanstalk, puede crear fácilmente un nuevo entorno para su aplicación en cualquier momento.

Además, puede terminar los recursos de base de datos creados fuera del entorno de Elastic Beanstalk. Cuando termine una instancia de base de datos de Amazon RDS, puede hacer una instantánea para restaurar los datos posteriormente en otra instancia.

Para terminar la instancia de base de datos de RDS

1. Abra la [consola de Amazon RDS](#).
2. Seleccione Databases (Bases de datos).
3. Seleccione la instancia de base de datos.
4. Elija Acciones y, a continuación, elija Eliminar.
5. Decida si desea crear una instantánea y seleccione Delete (Eliminar).

Siguientes pasos

A medida que siga desarrollando la aplicación, es probable que desee contar con un mecanismo que le permita administrar los entornos e implementar la aplicación sin tener que crear manualmente un archivo .zip y cargarlo en la consola de Elastic Beanstalk. La interfaz de [línea de comandos de Elastic Beanstalk](#) (EB CLI easy-to-use) proporciona comandos para crear, configurar e implementar aplicaciones en entornos de Elastic Beanstalk desde la línea de comandos.

La aplicación de ejemplo utiliza los archivos de configuración para configurar los ajustes de PHP y crear una tabla en la base de datos si no existe todavía. También puede utilizar un archivo de configuración para configurar el grupo de seguridad de las instancias durante la creación del entorno

y evitar así las laboriosas tareas de actualización de la configuración. Para obtener más información, consulte [Personalización avanzada de entornos con archivos de configuración \(.ebextensions\)](#).

En las tareas de desarrollo y pruebas, es posible que desee utilizar la funcionalidad de Elastic Beanstalk para agregar una instancia de base de datos administrada directamente al entorno. Para obtener instrucciones acerca de cómo configurar una base de datos dentro del entorno, consulte [Añadir una base de datos al entorno de Elastic Beanstalk](#).

Si necesita una base de datos de alto desempeño, considere la posibilidad de usar [Amazon Aurora](#). Amazon Aurora es un motor de base de datos compatible con MySQL que ofrece funciones de bases de datos comerciales a bajo costo. Para conectar la aplicación a una base de datos diferente, repita los pasos de la [configuración del grupo de seguridad](#) y [actualice las propiedades de entorno relacionadas con RDS](#).

Por último, si pretende utilizar la aplicación en un entorno de producción, querrá [configurar un nombre de dominio personalizado](#) para el entorno y [habilitar HTTPS](#) para las conexiones seguras.

Agregar una instancia de base de datos de Amazon RDS a un entorno de aplicaciones PHP

Puede utilizar una instancia de base de datos de Amazon Relational Database Service (Amazon RDS) para almacenar datos recopilados y modificados por su aplicación. La base de datos se puede adjuntar al entorno y administrarse con Elastic Beanstalk, o bien, crearse como desacoplada y administrarse de forma externa a través de otro servicio. En este tema se proporcionan instrucciones para crear un Amazon RDS mediante la consola de Elastic Beanstalk. La base de datos se acoplará al entorno y se administrará con Elastic Beanstalk. Para obtener más información sobre el uso de Elastic Beanstalk con Amazon RDS, consulte [Añadir una base de datos al entorno de Elastic Beanstalk](#).

Secciones

- [Añadir una instancia de base de datos al entorno](#)
- [Descargar un controlador](#)
- [Conexión a una base de datos con PDO o MySQLi](#)
- [Conexión a una base de datos con Symfony](#)

Añadir una instancia de base de datos al entorno

Para agregar una instancia de base de datos al entorno

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Database (Base de datos), seleccione Edit (Editar).
5. Seleccione un motor de base de datos y escriba el nombre de usuario y la contraseña.
6. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

Se tarda alrededor de 10 minutos en añadir una instancia de base de datos. Cuando se haya completado la actualización del entorno, el nombre de host de la instancia de base de datos y otra información de conexión estarán disponibles para su aplicación mediante las siguientes propiedades de entorno

Nombre de la propiedad	Descripción	Valor de la propiedad
RDS_HOSTNAME	El nombre de host de la instancia de base de datos.	En la pestaña Connectivity & security (Conectividad y seguridad) de la consola de Amazon RDS: Endpoint (Punto de enlace).
RDS_PORT	El puerto en donde la instancia de base de datos acepta las conexiones. El valor predeterminado varía dependiendo del motor de base de datos.	En la pestaña Connectivity & security (Conectividad y seguridad) de la consola de Amazon RDS: Port (Puerto).

Nombre de la propiedad	Descripción	Valor de la propiedad
RDS_DB_NAME	El nombre de la base de dato, ebdb .	En la pestaña Configuration (Configuración) de la consola de Amazon RDS: DB Name (Nombre de la base de datos).
RDS_USERNAME	El nombre de usuario que ha configurado para la base de datos.	En la pestaña Configuration (Configuración) de la consola de Amazon RDS: Master username (Nombre de usuario maestro).
RDS_PASSWORD	La contraseña que ha configurado para la base de datos.	No está disponible como referencia en la consola de Amazon RDS.

Para obtener más información sobre la configuración de una instancia de bases de datos acopladas a un entorno de Elastic Beanstalk, consulte [Añadir una base de datos al entorno de Elastic Beanstalk](#).

Descargar un controlador

Si desea utilizar PHP Data Objects (PDO) para conectarse a la base de datos, instale el driver que corresponda al motor de base de datos elegido.

- MySQL – [PDO_MYSQL](#)
- PostgreSQL – [PDO_PGSQL](#)
- Oracle: [PDO_OCI](#)
- SQL Server – [PDO_SQLSRV](#)

Para obtener más información, consulte <http://php.net/manual/en/pdo.installation.php>.

Conexión a una base de datos con PDO o MySQLi

Puede utilizar `$_SERVER[`VARIABLE`]` para leer la información de conexión del entorno.

Para obtener un PDO, cree un nombre de origen de datos (DSN) a partir del host, el puerto y el nombre. Pase el DSN al [constructor del PDO](#) con el nombre de usuario y la contraseña de la base de datos.

Example Conexión con una base de datos de RDS a través de PDO - MySQL

```
<?php
$dbhost = $_SERVER['RDS_HOSTNAME'];
$dbport = $_SERVER['RDS_PORT'];
$dbname = $_SERVER['RDS_DB_NAME'];
$charset = 'utf8' ;

$dsn = "mysql:host={$dbhost};port={$dbport};dbname={$dbname};charset={$charset}";
$username = $_SERVER['RDS_USERNAME'];
$password = $_SERVER['RDS_PASSWORD'];

$pdo = new PDO($dsn, $username, $password);
?>
```

Si utiliza otros controladores, sustituya `mysql` por el nombre del controlador: `pgsql`, `oci` o `sqlsrv`.

En el caso de MySQLi, pase el nombre del host, el nombre de usuario, la contraseña, el nombre de la base de datos y el puerto al constructor `mysqli`.

Example Conexión con una base de datos de RDS a través de `mysqli_connect()`

```
$link = new mysqli($_SERVER['RDS_HOSTNAME'], $_SERVER['RDS_USERNAME'],
    $_SERVER['RDS_PASSWORD'], $_SERVER['RDS_DB_NAME'], $_SERVER['RDS_PORT']);
```

Conexión a una base de datos con Symfony

Para Symfony versión 3.2 y posteriores, puede utilizar `%env(PROPERTY_NAME)%` para establecer los parámetros de base de datos en un archivo de configuración en función de las propiedades de entorno establecidas por Elastic Beanstalk.

Example `app/config/parameters.yml`

```
parameters:
    database_driver:    pdo_mysql
    database_host:     '%env(RDS_HOSTNAME)%'
    database_port:     '%env(RDS_PORT)%'
```

```
database_name:      '%env(RDS_DB_NAME)%'  
database_user:     '%env(RDS_USERNAME)%'  
database_password: '%env(RDS_PASSWORD)%'
```

Consulte [External Parameters \(Symfony 3.4\)](#) para obtener más información.

Para versiones anteriores de Symfony, las variables de entorno solo están accesibles si comienzan por `SYMFONY__`. Esto significa que las propiedades de entorno definidas por Elastic Beanstalk no están accesibles, por lo que deberá definir expresamente sus propiedades de entorno para pasar a Symfony la información de conexión.

Para conectarse a una base de datos con Symfony 2, [cree una propiedad de entorno](#) para cada parámetro. A continuación, utilice `%property.name%` para obtener acceso a la variable transformada por Symfony en un archivo de configuración. Por ejemplo, a una propiedad del entorno denominada `SYMFONY__DATABASE__USER` se puede tener acceso como `database.user`.

```
database_user:     "%database.user%"
```

Consulte [External Parameters \(Symfony 2.8\)](#) para obtener más información.

Uso de Python

Esta sección contiene tutoriales e información sobre la implementación de aplicaciones de Python con AWS Elastic Beanstalk.

En los temas de este capítulo se supone que tiene conocimientos de los entornos de Elastic Beanstalk. Si no ha usado Elastic Beanstalk antes, pruebe el [tutorial de introducción](#) para conocer los conceptos básicos.

Temas

- [Configuración del entorno de desarrollo de Python](#)
- [Uso de la plataforma Elastic Beanstalk Python](#)
- [Implementación de una aplicación Flask en Elastic Beanstalk](#)
- [Implementación de una aplicación Django en Elastic Beanstalk](#)
- [Agregar una instancia de base de datos de Amazon RDS a su entorno de aplicaciones Python](#)
- [Herramientas y recursos de Python](#)

Configuración del entorno de desarrollo de Python

Configure el entorno de desarrollo de Python para probar la aplicación localmente antes de implementarla en AWS Elastic Beanstalk. En este tema, se explican los pasos de configuración del entorno de desarrollo y se incluyen enlaces a páginas de instalación donde encontrará herramientas útiles.

Para seguir los procedimientos de esta guía, necesitará un shell o un terminal de línea de comando donde pueda ejecutar los comandos. Los comandos aparecen en listas y van precedidos del símbolo del sistema (\$) y del nombre del directorio actual, si es aplicable.

```
~/eb-project$ this is a command  
this is output
```

En Linux y macOS, puede utilizar el administrador de shell y paquetes preferido. En Windows, puede [instalar el subsistema de Windows para Linux para](#) obtener una versión de Ubuntu y Bash integrada en Windows.

Para ver procedimientos de configuración y herramientas comunes que se utilizan en todos los lenguajes, consulte [Configuración de su máquina de desarrollo](#).

Secciones

- [Requisitos previos](#)
- [Mediante un entorno virtual](#)
- [Configuración de un proyecto de Python para Elastic Beanstalk](#)

Requisitos previos

Todas las aplicaciones Python que implemente con Elastic Beanstalk tienen estos requisitos previos comunes:

1. Una versión de Python que coincida con la versión de la plataforma Python de Elastic Beanstalk que utilizará su aplicación.
2. La utilidad `pip`, correspondiente a su versión de Python. Se utiliza para instalar y mostrar las dependencias de su proyecto, de forma que Elastic Beanstalk sepa cómo configurar el entorno de su aplicación.
3. La interfaz de línea de AWS Elastic Beanstalk comandos (CLI de EB). Se utiliza para iniciar la aplicación con los archivos necesarios para su implementación con Elastic Beanstalk.

4. Una instalación de `ssh` de trabajo. Se utiliza para conectar con las instancias en ejecución cuando es necesario examinar o depurar una implementación.
5. El paquete `virtualenv`. Se utiliza para crear un entorno usado para desarrollar y probar la aplicación, de modo que Elastic Beanstalk pueda replicar el entorno sin necesidad de instalar paquetes adicionales que no necesite la aplicación. Instale este paquete con el siguiente comando:

```
$ pip install virtualenv
```

Para obtener instrucciones sobre cómo instalar Python, pip y la CLI de EB, consulte [Instalación de la CLI de EB](#).

Mediante un entorno virtual

Una vez instalados los requisitos previos, configure un entorno virtual con `virtualenv` para instalar las dependencias de la aplicación. Al utilizar un entorno virtual, puede determinar exactamente qué paquetes son necesarios para la aplicación a fin de que los paquetes necesarios se instalen en las instancias EC2 que ejecutan su aplicación.

Para configurar un entorno virtual

1. Abra una ventana de línea de comandos y escriba:

```
$ virtualenv /tmp/eb_python_app
```

Sustituya *eb_python_app* por un nombre que tenga sentido para su aplicación (se recomienda usar el nombre de la aplicación). El comando `virtualenv` crea un entorno virtual automáticamente en el directorio especificado e imprime los resultados de sus acciones:

```
Running virtualenv with interpreter /usr/bin/python
New python executable in /tmp/eb_python_app/bin/python3.7
Also creating executable in /tmp/eb_python_app/bin/python
Installing setuptools, pip...done.
```

2. Una vez que el entorno virtual esté listo, empiece ejecutando el script `activate` ubicado en el directorio `bin` del entorno. Por ejemplo, para iniciar el entorno `eb_python_app` creado en el paso anterior, escribiría:

```
$ source /tmp/eb_python_app/bin/activate
```

El entorno virtual imprime su nombre (por ejemplo: (eb_python_app)) al principio de cada símbolo del sistema para recordarle que se encuentra en un entorno de Python virtual.

3. Para dejar de usar su entorno virtual y volver al intérprete de Python predeterminado del sistema con todas las bibliotecas instaladas, ejecute el comando `deactivate`.

```
(eb_python_app) $ deactivate
```

Note

Una vez creado, puede reiniciar el entorno virtual en cualquier momento ejecutando de nuevo el script `activate`.

Configuración de un proyecto de Python para Elastic Beanstalk

Puede usar la CLI de Elastic Beanstalk para preparar las aplicaciones de Python para su implementación con Elastic Beanstalk.

Para configurar una aplicación Python para su implementación con Elastic Beanstalk

1. Desde dentro de su [entorno virtual](#), vuelva a la parte superior del árbol de directorios de su proyecto (python_eb_app) y escriba:

```
pip freeze >requirements.txt
```

Este comando copia los nombres y las versiones de los paquetes que están instalados en su entorno virtual en `requirements.txt`. Por ejemplo, si el paquete PyYAML, versión 3.11 está instalado en su entorno virtual, el archivo contendrá la línea:

```
PyYAML==3.11
```

Esto permite que Elastic Beanstalk replique el entorno Python de su aplicación con los mismos paquetes y las mismas versiones que utilizó para desarrollar y probar su aplicación.

2. Configure el repositorio de la CLI de EB con el comando `eb init`. Siga las instrucciones para elegir una región, una plataforma y otras opciones. Para obtener instrucciones detalladas, consulte [Administración de entornos de Elastic Beanstalk con la CLI de EB](#).

De forma predeterminada, Elastic Beanstalk busca un archivo llamado `application.py` para iniciar la aplicación. Si este archivo no existe en el proyecto de Python que ha creado, tendrá que realizar algunos ajustes en el entorno de su aplicación. También necesitará configurar las variables de entorno para que puedan cargarse los módulos de la aplicación. Para obtener más información, consulte [Uso de la plataforma Elastic Beanstalk Python](#).

Uso de la plataforma Elastic Beanstalk Python

La plataforma Python de AWS Elastic Beanstalk es un conjunto de [versiones de la plataforma](#) para aplicaciones web Python que se pueden ejecutar detrás de un servidor proxy con WSGI. Cada ramificación de plataforma corresponde a una versión de Python, como Python 3.8.

A partir de las ramas de la plataforma Amazon Linux 2, Elastic Beanstalk proporciona [Gunicorn](#) como el servidor WSGI predeterminado.

Puede añadir un `Procfile` a su paquete de código fuente para especificar y configurar el servidor WSGI de su aplicación. Para obtener más información, consulte [the section called “Procfile”](#).

Puede usar los archivos `Pipfile` y `Pipfile.lock` creados por Pipenv para especificar las dependencias del paquete de Python y otros requisitos. Para obtener más información sobre cómo especificar dependencias, consulte [the section called “Especificación de dependencias”](#).

Elastic Beanstalk proporciona [opciones de configuración](#) que se pueden usar para personalizar el software que se ejecuta en las instancias EC2 del entorno de Elastic Beanstalk. Puede configurar las variables de entorno que necesita la aplicación, habilitar la rotación de logs en Amazon S3 y asignar carpetas del código fuente de la aplicación que contengan archivos estáticos a rutas proporcionadas por el servidor proxy.

En la consola de Elastic Beanstalk hay opciones de configuración disponibles para [modificar la configuración de un entorno en ejecución](#). Para evitar perder la configuración del entorno cuando lo termina, puede usar las [configuraciones guardadas](#) para guardar la configuración y aplicarla posteriormente a otro entorno.

Para guardar la configuración en el código fuente, puede incluir [archivos de configuración](#). Los valores de configuración de los archivos de configuración se aplican cada vez que crea un entorno o que implementa la aplicación. También puede usar archivos de configuración para instalar paquetes,

ejecutar scripts y llevar a cabo otras operaciones de personalización de instancias durante las implementaciones.

La configuración aplicada en la consola de Elastic Beanstalk anula la misma configuración en los archivos de configuración, si existe. Esto le permite tener la configuración predeterminada en los archivos de configuración y anularla con la configuración específica del entorno en la consola. Para obtener más información acerca de la prioridad y otros métodos para cambiar valores de configuración, consulte [Opciones de configuración](#).

En los paquetes de Python disponibles de pip, también puede incluir un archivo de requisitos en la raíz del código fuente de la aplicación. Elastic Beanstalk instala los paquetes de dependencias especificados en un archivo de requisitos durante la implementación. Para obtener más información, consulte [the section called “Especificación de dependencias”](#).

Para obtener más información sobre las diversas formas en las que puede ampliar una plataforma Elastic Beanstalk basada en Linux, consulte [the section called “Ampliación de plataformas Linux”](#).

Configuración del entorno de Python

La configuración de la plataforma Python le permite ajustar el comportamiento de sus instancias de Amazon EC2. Puede editar la configuración de instancia de Amazon EC2 del entorno de Elastic Beanstalk mediante la consola de Elastic Beanstalk.

Utilice la consola de Elastic Beanstalk para configurar los parámetros del proceso Python, habilitar AWS X-Ray, habilitar la rotación de registros en Amazon S3 y configurar variables que su aplicación pueda leer en el entorno.

Para configurar su entorno Python en la consola de Elastic Beanstalk

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Actualizaciones, supervisión y registro, seleccione Editar.

Configuración de Python

- Proxy server (Servidor proxy): el servidor proxy que utilizar en las instancias del entorno. De forma predeterminada, se utiliza nginx.
- WSGI Path (Ruta de WSGI): el nombre de una ruta a su archivo de la aplicación principal. Por ejemplo, `application.py` o `django/wsgi.py`.
- NumProcesses: el número de procesos que se van a ejecutar en cada instancia de la aplicación.
- NumThreads: el número de subprocesos para ejecutar en cada proceso.

Configuración de AWS X-Ray

- X-Ray daemon (Daemon de X-Ray): ejecute el daemon de AWS X-Ray para procesar datos de rastreo desde [AWS X-Ray SDK para Python](#).

Log Options (Opciones de registro)

La sección Log Options tiene dos valores:

- Instance profile (Perfil de instancia): especifica el perfil de instancia que tiene permiso para obtener acceso al bucket de Amazon S3 asociado con la aplicación.
- Enable log file rotation to Amazon S3 (Habilitar la rotación de archivos de registro para Amazon S3): especifica si los archivos de registro de las instancias de Amazon EC2 de la aplicación se copian en el bucket de Amazon S3 asociado a la aplicación.

Archivos estáticos

Para mejorar el rendimiento, puede utilizar la sección Static files (Archivos estáticos) para configurar el servidor proxy para entregar archivos estáticos de servidor (por ejemplo, HTML o imágenes) desde un conjunto de directorios dentro de su aplicación web. Para cada directorio, se establece la ruta virtual para la asignación de directorios. Cuando el servidor proxy recibe una solicitud de un archivo en la ruta especificada, proporciona directamente el archivo en lugar de direccionar la solicitud la aplicación.

Para obtener más información sobre la configuración de archivos estáticos mediante archivos de configuración o la consola de Elastic Beanstalk, consulte [the section called “Archivos estáticos”](#).

De forma predeterminada, el servidor proxy en un entorno de Python proporciona los archivos en una carpeta llamada `static` situada en la ruta `/static`. Por ejemplo, si el código fuente

de la aplicación contiene un archivo llamado `logo.png` en una carpeta denominada `static`, el servidor proxy lo proporciona a los usuarios en `subdomain.elasticbeanstalk.com/static/logo.png`. Puede configurar mapeos adicionales como se explica en esta sección.

Propiedades del entorno

Puede utilizar las propiedades del entorno para proporcionar información a la aplicación y configurar las variables de entorno. Por ejemplo, puede crear una propiedad de entorno llamada `CONNECTION_STRING` que especifique una cadena de conexión que la aplicación pueda usar para conectarse a la base de datos.

Dentro del entorno de Python que se ejecuta en Elastic Beanstalk, estos valores son accesibles a través del diccionario `os.environ` de Python. Para obtener más información, visite <http://docs.python.org/library/os.html>.

Puede utilizar un código similar al siguiente para tener acceso a las claves y los parámetros:

```
import os
endpoint = os.environ['API_ENDPOINT']
```

Las propiedades de entorno también pueden proporcionar información a un marco de trabajo. Por ejemplo, puede crear una propiedad llamada `DJANGO_SETTINGS_MODULE` para configurar Django de forma que utilice un módulo de configuración específico. En función del entorno, el valor podría ser `development.settings`, `production.settings`, etc.

Para obtener más información, consulte [Propiedades del entorno y otras opciones del software](#).

Espacios de nombres de configuración de Python

Puede usar un [archivo de configuración](#) para definir opciones de configuración y realizar otras tareas de configuración en las instancias durante las implementaciones. Las opciones de configuración se pueden definir a través del servicio de Elastic Beanstalk o la plataforma que utilice y están organizadas por espacios de nombres.

La plataforma Python define opciones en los espacios de nombres `aws:elasticbeanstalk:environment:proxy`, `aws:elasticbeanstalk:environment:proxy:staticfiles` y `aws:elasticbeanstalk:container:python`.

El siguiente archivo de configuración de ejemplo especifica los valores de las opciones de configuración para crear una propiedad de entorno denominada `DJANGO_SETTINGS_MODULE`,

elija el servidor proxy Apache, especifique dos opciones de archivos estáticos que asignan un directorio denominado `statichtml` a la ruta `/html` y un directorio denominado `staticimages` a la ruta `/images`, así como opciones de configuración adicionales del espacio de nombres [aws:elasticbeanstalk:container:python](#). Este espacio de nombres contiene opciones que permiten especificar la ubicación del script WSGI del código fuente y el número de subprocesos y procesos que se van a ejecutar en WSGI.

```
option_settings:
  aws:elasticbeanstalk:application:environment:
    DJANGO_SETTINGS_MODULE: production.settings
  aws:elasticbeanstalk:environment:proxy:
    ProxyServer: apache
  aws:elasticbeanstalk:environment:proxy:staticfiles:
    /html: statichtml
    /images: staticimages
  aws:elasticbeanstalk:container:python:
    WSGIPath: ebdjango.wsgi:application
    NumProcesses: 3
    NumThreads: 20
```

Notas

- Si está utilizando una versión de la plataforma Python AMI de Amazon Linux (anterior a Amazon Linux 2), reemplace el valor `WSGIPath` por `ebdjango/wsgi.py`. El valor del ejemplo funciona con el servidor WSGI de Gunicorn, que no es compatible con las versiones de la plataforma de la AMI de Amazon Linux.
- Además, estas versiones antiguas de la plataforma utilizan un espacio de nombres diferente para configurar archivos estáticos: `aws:elasticbeanstalk:container:python:staticfiles`. Tiene los mismos nombres de opción y semántica que el espacio de nombres de archivo estático estándar.

Los archivos de configuración también admiten varias claves para [modificar aún más el software en las instancias del entorno](#). En este ejemplo, se utiliza la clave `packages` para instalar Memcached con yum y [container commands](#) para ejecutar comandos que configuren el servidor durante la implementación:

```
packages:
```

```
yum:
  libmemcached-devel: '0.31'

container_commands:
  collectstatic:
    command: "django-admin.py collectstatic --noinput"
  01syncdb:
    command: "django-admin.py syncdb --noinput"
    leader_only: true
  02migrate:
    command: "django-admin.py migrate"
    leader_only: true
  03wsgipass:
    command: 'echo "WSGIProcessAuthorization On" >> ../wsgi.conf'
  99customize:
    command: "scripts/customize.sh"
```

Elastic Beanstalk cuenta con numerosas opciones de configuración para personalizar el entorno. Además de los archivos de configuración, también puede definir opciones en la consola, configuraciones guardadas, la CLI de EB o la AWS CLI. Para obtener más información, consulte [Opciones de configuración](#).

Configuración del servidor WSGI con un archivo Procfile

Puede añadir un Procfile a su paquete de código fuente para especificar y configurar el servidor WSGI de su aplicación. En el ejemplo siguiente se utiliza Procfile para especificar uWSGI como servidor y configurarlo.

Example Procfile

```
web: uwsgi --http :8000 --wsgi-file application.py --master --processes 4 --threads 2
```

En el ejemplo siguiente se utiliza Procfile para configurar Gunicorn, el servidor WSGI predeterminado.

Example Procfile

```
web: gunicorn --bind :8000 --workers 3 --threads 2 project.wsgi:application
```

Notas

- Si configura cualquier servidor WSGI que no sea Gunicorn, asegúrese de especificarlo también como una dependencia de la aplicación, de modo que esté instalado en las instancias de su entorno. Para obtener más información sobre la especificación de dependencias, consulte [the section called “Especificación de dependencias”](#).
- El puerto predeterminado para el servidor WSGI es 8000. Si especifica un número de puerto diferente en el comando `Procfile`, establezca también el `PORT` [propiedad del entorno](#) en este número de puerto.

Cuando se utiliza un `Procfile`, este anula las opciones de espacio de nombres de `aws:elasticbeanstalk:container:python` que se establecen mediante archivos de configuración.

Para obtener más información sobre el uso de `Procfile`, expanda la sección `Buildfile` y `Procfile` en [the section called “Ampliación de plataformas Linux”](#).

Especificar dependencias mediante un archivo de requisitos

Una aplicación de Python típica tiene dependencias de otros paquetes de Python de terceros. Con la plataforma Python de Elastic Beanstalk, tiene algunas formas de especificar paquetes Python de los que depende su aplicación.

Utilice `pip` y `requirements.txt`

La herramienta estándar para instalar paquetes de Python es `pip`. Esta tiene una característica que permite especificar todos los paquetes necesarios (así como las versiones) en un único archivo de requisitos. Para obtener más información sobre el archivo de requisitos, consulte [Requirements File Format](#) en el sitio web de documentación de `pip`.

Cree un archivo llamado `requirements.txt` y sitúelo en el directorio principal del paquete de código fuente. A continuación se muestra un ejemplo del archivo `requirements.txt` para Django.

```
Django==2.2
mysqlclient==2.0.3
```

En el entorno de desarrollo, puede utilizar el comando `pip freeze` para generar el archivo de requisitos.

```
~/my-app$ pip freeze > requirements.txt
```

Para asegurarse de que el archivo de requisitos solo contiene los paquetes que se utilizan realmente en la aplicación, utilice un [entorno virtual](#) que solo tenga instalado esos paquetes. Fuera del entorno virtual, la salida de `pip freeze` incluirá todos los paquetes de `pip` instalados en el equipo de desarrollo, incluidos aquellos que venían con el sistema operativo.

Note

En las versiones de la plataforma Python AMI de Amazon Linux, Elastic Beanstalk no admite de forma nativa Pipenv o Pipfiles. Si utiliza Pipenv para administrar las dependencias de la aplicación, ejecute el siguiente comando para generar un archivo `requirements.txt`.

```
~/my-app$ pipenv lock -r > requirements.txt
```

Para obtener más información, consulte [Generación de un requirements.txt](#) en la documentación de Pipenv.

Use Pipenv y **Pipfile**

Pipenv es una herramienta de empaquetado de Python moderna. Combina la instalación de paquetes con la creación y gestión de un archivo de dependencia y un `virtualenv` para su aplicación. Para obtener más información, consulte [Pipenv: Python Dev Workflow for Humans](#).

Pipenv mantiene dos archivos:

- `Pipfile`: este archivo contiene varios tipos de dependencias y requisitos.
- `Pipfile.lock`: este archivo contiene una instantánea de la versión que permite compilaciones deterministas.

Puede crear estos archivos en su entorno de desarrollo e incluirlos en el directorio de nivel superior del paquete de origen que se implementa en Elastic Beanstalk. Para obtener más información sobre estos dos archivos, visite [Example Pipfile and Pipfile.lock](#).

El siguiente ejemplo utiliza Pipenv para instalar Django y el marco REST de Django. Estos comandos crean los archivos `Pipfile` y `Pipfile.lock`.


```
~/my-app$ pipenv install django
~/my-app$ pipenv install djangorestframework
```

Prioridad

Si incluye más de uno de los archivos de requisitos descritos en este tema, Elastic Beanstalk utiliza solo uno de ellos. La siguiente lista muestra la prioridad, en orden descendente.

1. `requirements.txt`
2. `Pipfile.lock`
3. `Pipfile`

Note

A partir de la versión de la plataforma Amazon Linux 2 del 7 de marzo de 2023, si proporciona más de uno de estos archivos, Elastic Beanstalk emitirá un mensaje en la consola donde se indicará cuál de los archivos de dependencia se utilizó durante la implementación.

En los siguientes pasos se describe la lógica que sigue Elastic Beanstalk para instalar las dependencias al implementar una instancia.

- Si hay un archivo `requirements.txt`, utilizamos el comando `pip install -r requirements.txt`.
- A partir del lanzamiento de la plataforma Amazon Linux 2 del 7 de marzo de 2023, si no hay ningún archivo `requirements.txt`, pero hay un `Pipfile.lock`, utilizamos el comando `pipenv sync`. Antes de ese lanzamiento, usábamos `pipenv install --ignore-pipfile`.
- Si no hay ni un archivo `requirements.txt` ni un `Pipfile.lock`, pero hay un `Pipfile`, entonces usamos el comando `pipenv install --skip-lock`.
- Si no se encuentra ninguno de los tres archivos de requisitos, no instalamos ninguna de las dependencias de la aplicación.

Implementación de una aplicación Flask en Elastic Beanstalk

Flask es un marco de aplicaciones web de código abierto para Python. Este tutorial explica el proceso de generación de una aplicación de Flask e implementarla en un AWS Elastic Beanstalk entorno.

En este tutorial, hará lo siguiente:

- [Configuración de un entorno virtual Python con Flask](#)
- [Creación de una aplicación Flask](#)
- [Implementación del sitio con la CLI de EB](#)
- [Limpieza](#)

Requisitos previos

Este tutorial asume que tiene conocimiento de las operaciones básicas de Elastic Beanstalk y de la consola de Elastic Beanstalk. Si aún no lo ha hecho, siga las instrucciones que se indican en [Introducción a Elastic Beanstalk](#) para lanzar su primer entorno de Elastic Beanstalk.

Para seguir los procedimientos de esta guía, necesitará un shell o un terminal de línea de comando donde pueda ejecutar los comandos. Los comandos aparecen en listas y van precedidos del símbolo del sistema (\$) y del nombre del directorio actual, si es aplicable.

```
~/eb-project$ this is a command  
this is output
```

En Linux y macOS, puede utilizar el administrador de shell y paquetes preferido. En Windows, puede [instalar el subsistema de Windows para Linux para](#) obtener una versión de Ubuntu y Bash integrada en Windows.

Flask requiere Python 3.7 o posterior. En este tutorial, vamos a utilizar Python 3.7 y la versión de la plataforma Elastic Beanstalk correspondiente. Instale Python; para ello, siga las instrucciones de [Configuración del entorno de desarrollo de Python](#).

La instalación del marco de [Flask](#) forma parte del tutorial.

En este tutorial también se utiliza la interfaz de línea de comandos de Elastic Beanstalk (CLI de EB). Para obtener detalles sobre la instalación y configuración de la CLI de EB, consulte [Instalación de la CLI de EB](#) y [Configuración de la CLI de EB](#).

Configuración de un entorno virtual Python con Flask

Cree un directorio de proyecto y un entorno virtual para su aplicación e instale Flask.

Para configurar el entorno del proyecto

1. Cree el directorio del proyecto.

```
~$ mkdir eb-flask
~$ cd eb-flask
```

2. Cree y active un entorno virtual llamado `virt`:

```
~/eb-flask$ virtualenv virt
~$ source virt/bin/activate
(virt) ~/eb-flask$
```

Verá `(virt)` delante del símbolo del sistema, lo que indica que se encuentra en un entorno virtual. Utilice el entorno virtual para el resto de este tutorial.

3. Instale Flask con `pip install`:

```
(virt)~/eb-flask$ pip install flask==2.0.3
```

4. Consulte las bibliotecas instaladas con `pip freeze`:

```
(virt)~/eb-flask$ pip freeze
click==8.1.1
Flask==2.0.3
itsdangerous==2.1.2
Jinja2==3.1.1
MarkupSafe==2.1.1
Werkzeug==2.1.0
```

Este comando enumera todos los paquetes instalados en su entorno virtual. Debido a que se encuentra en un entorno virtual, no se muestran los paquetes instalados globalmente como la CLI de EB.

5. Guarde la salida de `pip freeze` en un archivo llamado `requirements.txt`.

```
(virt)~/eb-flask$ pip freeze > requirements.txt
```

Este archivo indica a Elastic Beanstalk que instale las bibliotecas durante la implementación. Para obtener más información, consulte [Especificar dependencias mediante un archivo de requisitos](#).

Creación de una aplicación Flask

A continuación, vamos a crear una aplicación que implementará con Elastic Beanstalk. Crearemos el servicio web RESTful "Hello World".

En este directorio, cree un nuevo archivo de texto llamado `application.py` con el contenido siguiente:

Example `~/eb-flask/application.py`

```
from flask import Flask

# print a nice greeting.
def say_hello(username = "World"):
    return '<p>Hello %s!</p>\n' % username

# some bits of text for the page.
header_text = '''
    <html>\n<head> <title>EB Flask Test</title> </head>\n<body>'''
instructions = '''
    <p><em>Hint</em>: This is a RESTful web service! Append a username
    to the URL (for example: <code>/Thelonious</code>) to say hello to
    someone specific.</p>\n'''
home_link = '<p><a href="/">Back</a></p>\n'
footer_text = '</body>\n</html>'

# EB looks for an 'application' callable by default.
application = Flask(__name__)

# add a rule for the index page.
application.add_url_rule('/', 'index', (lambda: header_text +
    say_hello() + instructions + footer_text))

# add a rule when the page is accessed with a name appended to the site
# URL.
application.add_url_rule('/<username>', 'hello', (lambda username:
    header_text + say_hello(username) + home_link + footer_text))
```

```
# run the app.
if __name__ == "__main__":
    # Setting debug to True enables debug output. This line should be
    # removed before deploying a production app.
    application.debug = True
    application.run()
```

En este ejemplo, se imprime un saludo personalizado que varía en función de la ruta utilizada para obtener acceso al servicio.

Note

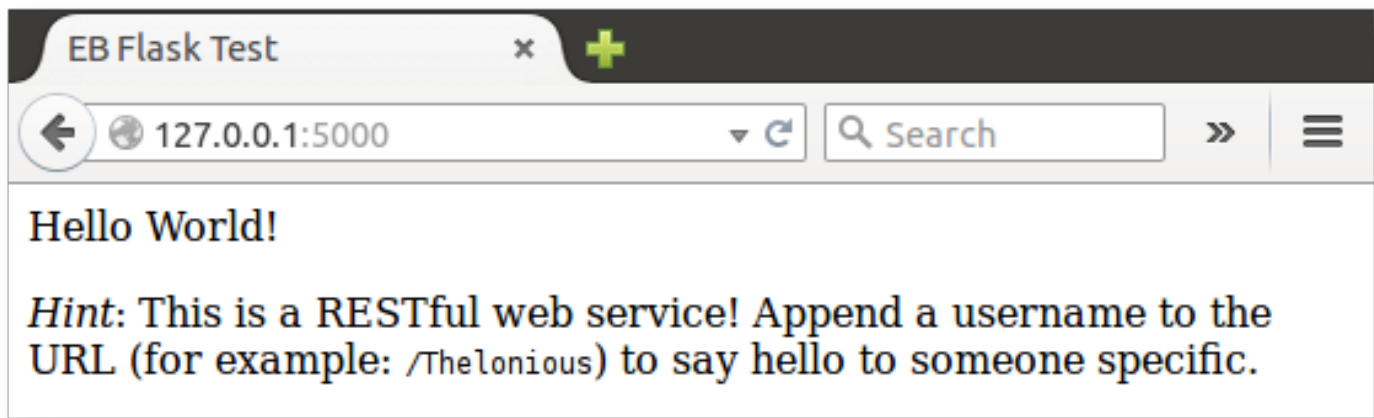
Si se agrega `application.debug = True` antes de ejecutar la aplicación, se habilitará la salida de depuración en caso de que se produzca algún error. Esta es una práctica recomendada cuando se realizan tareas de desarrollo. Sin embargo, las instrucciones de depuración deben eliminarse en el código de producción, ya que la salida de depuración puede desvelar aspectos internos de la aplicación.

Si utiliza `application.py` como nombre de archivo y proporciona un objeto `application` al que se pueda llamar (en este caso, el objeto Flask), Elastic Beanstalk podrá encontrar fácilmente el código de la aplicación.

Ejecute `application.py` con Python:

```
(virt) ~/eb-flask$ python application.py
* Serving Flask app "application" (lazy loading)
* Environment: production
  WARNING: Do not use the development server in a production environment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 313-155-123
```

Abra `http://127.0.0.1:5000/` en el navegador web. Debería aparecer la página de índice de la aplicación en ejecución:



Compruebe el registro del servidor para ver el resultado de la solicitud. Si desea detener el servidor web y volver al entorno virtual, escriba Ctrl+C.

Si aparece la salida de depuración, solucione los errores y asegúrese de que la aplicación está ejecutándose en local antes de configurarla para Elastic Beanstalk.

Implementación del sitio con la CLI de EB

Ha añadido todo lo que necesita para implementar su aplicación en Elastic Beanstalk. El directorio del proyecto debe ser similar al siguiente:

```
~/eb-flask/  
|-- virt  
|-- application.py  
`-- requirements.txt
```

Sin embargo, la carpeta `virt`, no es obligatoria para que la aplicación se ejecute en Elastic Beanstalk. Al realizar la implementación, Elastic Beanstalk crea un nuevo entorno virtual en las instancias de servidor e instala las bibliotecas indicadas en `requirements.txt`. Para minimizar el tamaño del paquete de código fuente que cargue durante la implementación, agregue un archivo [.ebignore](#) que le indique a la CLI de EB que omita la carpeta `virt`.

Example `~/eb-flask/.ebignore`

```
virt
```

A continuación, creará el entorno de la aplicación e implementará su aplicación configurada con Elastic Beanstalk.

Para crear un entorno e implementar la aplicación de Flask

1. Inicialice el repositorio de la CLI de EB con el comando `eb init`:

```
~/eb-flask$ eb init -p python-3.7 flask-tutorial --region us-east-2
Application flask-tutorial has been created.
```

Este comando crea una nueva aplicación llamada `flask-tutorial` y configura el repositorio local para crear entornos con la última versión de la plataforma Python 3.7.

2. (opcional) Ejecute de nuevo `eb init` para configurar un par de claves predeterminadas de forma que pueda conectarse a la instancia EC2 donde se ejecuta la aplicación con SSH:

```
~/eb-flask$ eb init
Do you want to set up SSH for your instances?
(y/n): y
Select a keypair.
1) my-keypair
2) [ Create new KeyPair ]
```

Seleccione un par de claves si tiene uno o siga las instrucciones para crear una nuevo par de claves. Si no ve el símbolo del sistema o más adelante necesita cambiar la configuración, ejecute `eb init -i`.

3. Cree un entorno e implemente la aplicación en él con `eb create`:

```
~/eb-flask$ eb create flask-env
```

El entorno tarda unos cinco minutos en crearse e incluye los recursos siguientes:

- Instancia de EC2: una máquina virtual de Amazon Elastic Compute Cloud (Amazon EC2) configurada para ejecutar aplicaciones web en la plataforma que elija.

Cada plataforma ejecuta un conjunto específico de software, archivos de configuración y scripts compatibles con una determinada versión de lenguaje, marco de trabajo y contenedor web (o una combinación de estos). La mayoría de las plataformas utilizan Apache o nginx como un proxy inverso que se sitúa delante de la aplicación web, reenvía las solicitudes a esta, administra los recursos estáticos y genera registros de acceso y errores.

- Instance security group (Grupo de seguridad de la instancia): grupo de seguridad de Amazon EC2 configurado para permitir el tráfico entrante en el puerto 80. Este recurso permite que el tráfico HTTP procedente del balanceador de carga llegue a la instancia EC2 en la que se ejecuta la aplicación web. De forma predeterminada, el tráfico no está permitido en otros puertos.
- Balanceador de carga: balanceador de carga de Elastic Load Balancing configurado para distribuir solicitudes a las instancias que se ejecutan en la aplicación. Los balanceadores de carga también permiten que las instancias no estén expuestas directamente a Internet.
- Grupo de seguridad del balanceador de carga: grupo de seguridad de Amazon EC2 configurado para permitir el tráfico entrante en el puerto 80. Este recurso permite que el tráfico HTTP procedente de Internet llegue al balanceador de carga. De forma predeterminada, el tráfico no está permitido en otros puertos.
- Grupo de Auto Scaling: grupo de Auto Scaling configurado para reemplazar una instancia si termina o deja de estar disponible.
- Bucket de Amazon S3: ubicación de almacenamiento para el código fuente, los registros y otros artefactos que se crean al utilizar Elastic Beanstalk.
- CloudWatch Alarmas de Amazon: dos CloudWatch alarmas que monitorean la carga de las instancias de su entorno y que se activan si la carga es demasiado alta o demasiado baja. Cuando se activa una alarma, en respuesta, el grupo de Auto Scaling aumenta o reduce los recursos.
- AWS CloudFormation pila: Elastic AWS CloudFormation Beanstalk se utiliza para lanzar los recursos de su entorno y propagar los cambios de configuración. Los recursos se definen en una plantilla que puede verse en la [consola de AWS CloudFormation](#).
- Nombre de dominio: nombre de dominio que direcciona el tráfico a la aplicación web con el formato *subdominio.región.elasticbeanstalk.com*.

Note

Para aumentar la seguridad de las aplicaciones de Elastic Beanstalk, el dominio `elasticbeanstalk.com` está registrado en la [lista de sufijos públicos \(PSL\)](#). Para mayor seguridad, se recomienda que utilice cookies con un prefijo `__Host-` en caso de que necesite configurar cookies confidenciales en el nombre de dominio predeterminado de sus aplicaciones de Elastic Beanstalk. Esta práctica le ayudará a defender su dominio contra intentos de falsificación de solicitudes entre sitios (CSRF). Para obtener más información, consulte la página de [configuración de cookies](#) en la red de desarrolladores de Mozilla.

Todos estos recursos los administra Elastic Beanstalk. Cuando termina su entorno, Elastic Beanstalk termina todos los recursos que este contiene.

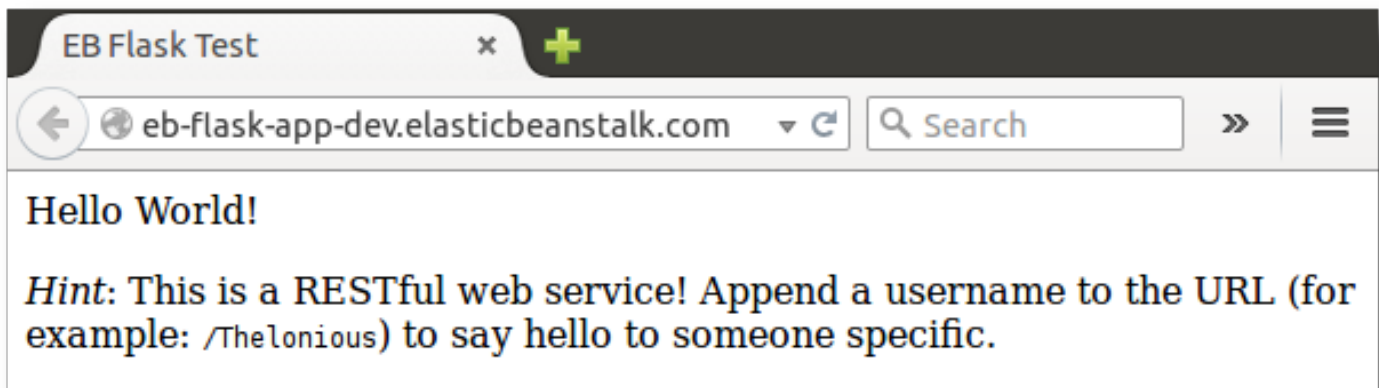
Note

El bucket de Amazon S3 que Elastic Beanstalk crea se comparte entre los entornos y no se elimina durante la terminación del entorno. Para obtener más información, consulte [Usar Elastic Beanstalk con Amazon S3](#).

Cuando finalice el proceso de creación del entorno, abra el sitio web con `eb open`:

```
~/eb-flask$ eb open
```

Se abrirá una ventana del navegador con el nombre de dominio creado para su aplicación. Debería ver el mismo sitio web de Flask que creó y probó en local.



Si no ve la aplicación en ejecución o aparece un mensaje de error, consulte [Solución de problemas de implementación](#) para intentar determinar la causa del error.


En caso de que sí vea la aplicación en ejecución (le felicitamos en ese caso), habrá implementado la primera aplicación de Flask con Elastic Beanstalk.

Limpieza

Cuando termine de trabajar con Elastic Beanstalk, puede terminar su entorno. [Elastic Beanstalk AWS cancela todos los recursos asociados al entorno, como las instancias de Amazon EC2, las instancias de bases de datos, los balanceadores de carga, los grupos de seguridad y las alarmas.](#)

Para cerrar el entorno de Elastic Beanstalk desde la consola

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

 Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. Elija Actions (Acciones) y, a continuación, Terminate Environment (Terminar el entorno).
4. Utilice el cuadro de diálogo en pantalla para confirmar la terminación del entorno.

Con Elastic Beanstalk, puede crear fácilmente un nuevo entorno para su aplicación en cualquier momento.

O, con la CLI de EB:

```
~/eb-flask$ eb terminate flask-env
```

Siguientes pasos

Para obtener más información acerca de Flask, visite flask.pocoo.org.

Si desea probar otra plataforma web de Python, consulte [Implementación de una aplicación Django en Elastic Beanstalk](#).

Implementación de una aplicación Django en Elastic Beanstalk

En este tutorial se explica la implementación de un sitio web de [Django](#) autogenerado y predeterminado en un entorno de AWS Elastic Beanstalk que ejecuta Python. En este tutorial se muestra cómo alojar una aplicación web de Python en la nube mediante un entorno de Elastic Beanstalk.

En este tutorial, hará lo siguiente:

- [Configuración de un entorno virtual de Python e instalación de Django](#)
- [Creación de un proyecto de Django](#)

- [Configure su aplicación Django para Elastic Beanstalk](#)
- [Implementación del sitio con la CLI de EB](#)
- [Actualizar la aplicación](#)
- [Eliminar recursos](#)

Requisitos previos

Para utilizar cualquier servicio de AWS, incluido Elastic Beanstalk, debe tener una cuenta de y credenciales de AWS. Para obtener más información y registrarse, visite <https://aws.amazon.com/>.

Para seguir este tutorial, debe tener todos los [requisitos previos comunes](#) de Python instalados, incluidos los siguientes paquetes:

- Python 3.7 o posterior
- pip
- virtualenv
- awsebcli

La instalación de la plataforma [Django](#) forma parte del tutorial.

Note

Para crear entornos con la CLI de EB, se requiere una [función de servicio](#). Puede crear una función de servicio creando un entorno en la consola de Elastic Beanstalk. Si no cuenta con ninguna función de servicio, el EB CLI intentará crear uno cuando ejecute `eb create`.

Configuración de un entorno virtual de Python e instalación de Django

Cree un entorno virtual con `virtualenv` y utilícelo para instalar Django y sus dependencias. Al utilizar un entorno virtual, puede determinar exactamente qué paquetes son necesarios para la aplicación a fin de que los paquetes necesarios se instalen en las instancias Amazon EC2 que ejecutan su aplicación.

Los siguientes pasos muestran los comandos que debe especificar para sistemas basados en Unix y Windows, que aparecen en pestañas independientes.

Para configurar el entorno virtual

1. Cree un entorno virtual llamado `eb-virt`.

Unix-based systems

```
~$ virtualenv ~/eb-virt
```

Windows

```
C:\> virtualenv %HOMEPATH%\eb-virt
```

2. Active el entorno virtual.

Unix-based systems

```
~$ source ~/eb-virt/bin/activate  
(eb-virt) ~$
```

Windows

```
C:\>%HOMEPATH%\eb-virt\Scripts\activate  
(eb-virt) C:\>
```

Verá `(eb-virt)` delante del símbolo del sistema, lo que indica que se encuentra en un entorno virtual.

Note

En el resto de estas instrucciones se muestra el símbolo del sistema de Linux en el directorio principal `~$`. En Windows, este directorio es `C:\Users\USERNAME>`, donde *USERNAME* es su nombre de inicio de sesión de Windows.

3. Use `pip` para instalar Django.

```
(eb-virt)~$ pip install django==2.2
```

Note

La versión de Django que instale debe ser compatible con la versión de Python en la configuración de Elastic Beanstalk Python que elija para implementar su aplicación. Para obtener información sobre la implementación, consulte [???](#) en este tema.

Para obtener más información sobre las versiones de plataforma de Python actuales, consulte [Python](#) en el documento Plataformas de AWS Elastic Beanstalk.

Para obtener información sobre la compatibilidad de versiones de Django con Python, consulte [¿Qué versión de Python puedo utilizar con Django?](#)

4. Para comprobar que Django está instalado, escriba lo siguiente.

```
(eb-virt)~$ pip freeze
Django==2.2
...
```

Este comando enumera todos los paquetes instalados en su entorno virtual. Más adelante utilizará el resultado de este comando para configurar su proyecto para usarlo con Elastic Beanstalk.

Creación de un proyecto de Django

Ahora está preparado para crear un proyecto de Django y ejecutarlo en su equipo, utilizando el entorno virtual.

Note

Este tutorial utiliza SQLite, que es un motor de base de datos incluido en Python. La base de datos se implementa con los archivos del proyecto. Para entornos de producción, se recomienda utilizar Amazon Relational Database Service (Amazon RDS) y separarlo del entorno. Para obtener más información, consulte [Agregar una instancia de base de datos de Amazon RDS a su entorno de aplicaciones Python](#).

Para generar una aplicación de Django

1. Active el entorno virtual.

Unix-based systems

```
~$ source ~/eb-virt/bin/activate
(eb-virt) ~$
```

Windows

```
C:\>%HOMEPATH%\eb-virt\Scripts\activate
(eb-virt) C:\>
```

Verá el prefijo (eb-virt) delante del símbolo del sistema, lo que indica que se encuentra en un entorno virtual.

Note

En el resto de estas instrucciones se muestra el símbolo del sistema de Linux ~\$ en el directorio principal y el directorio principal de Linux ~/. En Windows, estos directorios son C:\Users*USERNAME*>, donde *USERNAME* es su nombre de inicio de sesión de Windows.

- Utilice el comando `django-admin startproject ebdjango` para crear un proyecto de Django llamado ebdjango.

```
(eb-virt)~$ django-admin startproject ebdjango
```

Este comando crea un sitio de Django estándar denominado ebdjango con la siguiente estructura de directorios.

```
~/ebdjango
|-- ebdjango
|   |-- __init__.py
|   |-- settings.py
|   |-- urls.py
|   |-- wsgi.py
|-- manage.py
```

- Ejecute el sitio de Django localmente con `manage.py runserver`.

```
(eb-virt) ~$ cd ebdjango
```

```
(eb-virt) ~/ebdjango$ python manage.py runserver
```

4. En un navegador web, abra `http://127.0.0.1:8000/` para ver el sitio.
5. Compruebe el registro del servidor para ver el resultado de la solicitud. Para detener el servidor web y volver al entorno virtual, presione `Ctrl+C`.

```
Django version 2.2, using settings 'ebdjango.settings'  
Starting development server at http://127.0.0.1:8000/  
Quit the server with CONTROL-C.  
[07/Sep/2018 20:14:09] "GET / HTTP/1.1" 200 16348  
Ctrl+C
```

Configure su aplicación Django para Elastic Beanstalk

Ahora que ya tiene un sitio de Django en su equipo local, puede configurarlo para implementarlo con Elastic Beanstalk.

De forma predeterminada, Elastic Beanstalk busca un archivo llamado `application.py` para iniciar la aplicación. Como este archivo no existe en el proyecto de Django que ha creado, tendrá que realizar algunos ajustes en el entorno de su aplicación. También deberá configurar las variables de entorno para que puedan cargarse los módulos de la aplicación.

Para configurar su sitio para Elastic Beanstalk

1. Active el entorno virtual.

Unix-based systems

```
~/ebdjango$ source ~/eb-virt/bin/activate
```

Windows

```
C:\Users\USERNAME\ebdjango>%HOMEPATH%\eb-virt\Scripts\activate
```

2. Ejecute `pip freeze` y, a continuación, guarde la salida en un archivo llamado `requirements.txt`.

```
(eb-virt) ~/ebdjango$ pip freeze > requirements.txt
```

Elastic Beanstalk utiliza `requirements.txt` para determinar qué paquete se va a instalar en las instancias EC2 donde se ejecuta la aplicación.

3. Cree un directorio llamado `.ebextensions`.

```
(eb-virt) ~/ebdjango$ mkdir .ebextensions
```

4. En el directorio `.ebextensions`, añada un [archivo de configuración](#) llamado `django.config` con el siguiente texto.

Example `~/ebdjango/.ebextensions/django.config`

```
option_settings:  
  aws:elasticbeanstalk:container:python:  
    WSGIPath: ebdjango.wsgi:application
```

Esta configuración, `WSGIPath`, especifica la ubicación del script de WSGI que utiliza Elastic Beanstalk para iniciar su aplicación.

Note

Si está utilizando una versión de la plataforma Python AMI de Amazon Linux (anterior a Amazon Linux 2), reemplace el valor `WSGIPath` por `ebdjango/wsgi.py`. El valor del ejemplo funciona con el servidor WSGI de Gunicorn, que no es compatible con las versiones de la plataforma de la AMI de Amazon Linux.

5. Desactive el entorno virtual con el comando `deactivate`.

```
(eb-virt) ~/ebdjango$ deactivate
```

Vuelva a activar su entorno virtual siempre que necesite añadir paquetes a su aplicación o ejecutar su aplicación localmente.

Implementación del sitio con la CLI de EB

Ha añadido todo lo que necesita para implementar su aplicación en Elastic Beanstalk. El directorio del proyecto debe ser similar al siguiente.

```
~/ebdjango/  
|-- .ebextensions  
|   |-- django.config  
|-- ebdjango  
|   |-- __init__.py  
|   |-- settings.py  
|   |-- urls.py  
|   |-- wsgi.py  
|-- db.sqlite3  
|-- manage.py  
|-- requirements.txt
```

A continuación, creará el entorno de la aplicación e implementará su aplicación configurada con Elastic Beanstalk.

Inmediatamente después de la implementación, editará la configuración de Django para agregar al ajuste de Elastic Beanstalk de Django el nombre de dominio que asignó a su aplicación `ALLOWED_HOSTS`. A continuación, volverá a implementar la aplicación. Se trata de un requisito de seguridad de Django, diseñado para impedir los ataques de encabezado de Host HTTP. Para obtener más información, consulte [Host header validation](#).

Para crear un entorno e implementar la aplicación de Django

Note

El tutorial utiliza la CLI de EB como un mecanismo de implementación, pero también se puede utilizar la consola de Elastic Beanstalk para implementar un archivo ZIP que incluya el contenido del proyecto.

1. Inicialice el repositorio de la CLI de EB con el comando `eb init`.

```
~/ebdjango$ eb init -p python-3.7 django-tutorial  
Application django-tutorial has been created.
```

Este comando crea una aplicación llamada `django-tutorial`. También configura su repositorio local para crear entornos con la última versión de la plataforma Python 3.7.

- (Opcional) Ejecute de nuevo `eb init` para configurar un par de claves predeterminadas de forma que pueda usar SSH para conectarse a la instancia EC2 donde se ejecuta la aplicación.

```
~/ebdjango$ eb init
Do you want to set up SSH for your instances?
(y/n): y
Select a keypair.
1) my-keypair
2) [ Create new KeyPair ]
```

Seleccione un par de claves si ya tiene uno o siga las instrucciones para crear uno. Si no ve el símbolo del sistema o más adelante necesita cambiar la configuración, ejecute `eb init -i`.

- Cree un entorno e implemente la aplicación en él con `eb create`.

```
~/ebdjango$ eb create django-env
```

Note

Si recibe un mensaje de error "service role required" (es necesario el rol de servicio), ejecute `eb create` interactivamente (sin especificar un nombre de entorno) y la CLI de EB creará el rol.

Este comando crea un entorno de Elastic Beanstalk con equilibrio de carga denominado `django-env`. Se tarda alrededor de cinco minutos en crear el entorno. A medida que Elastic Beanstalk crea los recursos necesarios para ejecutar la aplicación, envía mensajes informativos que la CLI de EB transmite a su terminal.

- Cuando finalice el proceso de creación del entorno, busque el nombre de dominio de su nuevo entorno. Para ello, ejecute `eb status`.

```
~/ebdjango$ eb status
Environment details for: django-env
Application name: django-tutorial
...
CNAME: eb-django-app-dev.elasticbeanstalk.com
```

```
...
```

El nombre de dominio del entorno es el valor de la propiedad CNAME.

- Abra el archivo `settings.py` en el directorio `ebdjango`. Localice el ajuste `ALLOWED_HOSTS` y, a continuación, agregue el nombre de dominio de la aplicación que encontró en el paso anterior al valor del ajuste. Si no encuentra este ajuste en el archivo, agréguelo en una línea nueva.

```
...  
ALLOWED_HOSTS = ['eb-django-app-dev.elasticbeanstalk.com']
```

- Guarde el archivo y, a continuación, vuelva a implementar la aplicación. Para ello, ejecute `eb deploy`. Cuando ejecuta `eb deploy`, la CLI de EB empaqueta el contenido del directorio del proyecto y lo implementa en su entorno.

```
~/ebdjango$ eb deploy
```

Note

Si utiliza Git con el proyecto, consulte [Uso de la CLI de EB con Git](#).

- Cuando finalice el proceso de actualización del entorno, abra el sitio web con `eb open`.

```
~/ebdjango$ eb open
```

Se abre una ventana del navegador con el nombre de dominio creado para su aplicación. Debería ver el mismo sitio web de Django que creó y probó localmente.

Si no ve la aplicación en ejecución o aparece un mensaje de error, consulte [Solución de problemas de implementación](#) para intentar determinar la causa del error.

En caso de que sí vea la aplicación en ejecución (le felicitamos en ese caso), habrá implementado la primera aplicación de Django con Elastic Beanstalk.

Actualizar la aplicación

Ahora que tiene una aplicación en ejecución en Elastic Beanstalk, puede actualizar y volver a implementar su aplicación o su configuración; Elastic Beanstalk se encargará de la tarea de actualizar sus instancias e iniciar la nueva versión de la aplicación.

Para este ejemplo, vamos a habilitar la consola de administración de Django y configurar otros ajustes.

Modificar la configuración del sitio

De forma predeterminada, el sitio web de Django utiliza la zona horaria UTC para mostrar el tiempo. Puede cambiar esta configuración especificando una zona horaria en `settings.py`.

Para cambiar la zona horaria del sitio

1. Modifique el valor de `TIME_ZONE` en `settings.py`.

Example `~/ebdjango/ebdjango/settings.py`

```
...  
# Internationalization  
LANGUAGE_CODE = 'en-us'  
TIME_ZONE = 'US/Pacific'  
USE_I18N = True  
USE_L10N = True  
USE_TZ = True
```

Para obtener una lista de zonas horarias, visite [esta página](#).

2. Implemente la aplicación en su entorno Elastic Beanstalk.

```
~/ebdjango/$ eb deploy
```

Crear un administrador del sitio

Puede crear un administrador del sitio para su aplicación de Django que tenga acceso a la consola de administración directamente desde el sitio web. Los detalles de inicio de sesión del administrador se almacenan de forma segura en la imagen de la base de datos local incluida en el proyecto predeterminado que genera Django.

Para crear un administrador del sitio

1. Inicialice la base de datos local de la aplicación de Django.

```
(eb-virt) ~/ebdjango$ python manage.py migrate  
Operations to perform:
```

```
Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying sessions.0001_initial... OK
```

2. Ejecute `manage.py createsuperuser` para crear un administrador.

```
(eb-virt) ~/ebdjango$ python manage.py createsuperuser
Username: admin
Email address: me@mydomain.com
Password: *****
Password (again): *****
Superuser created successfully.
```

3. Para indicar a Django dónde debe almacenar los archivos estáticos, defina `STATIC_ROOT` en `settings.py`.

Example `~/ebdjango/ebdjango/settings.py`

```
# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/2.2/howto/static-files/
STATIC_URL = '/static/'
STATIC_ROOT = 'static'
```

4. Ejecute `manage.py collectstatic` para rellenar el directorio `static` con los recursos estáticos (JavaScript, CSS e imágenes) del sitio de administración.

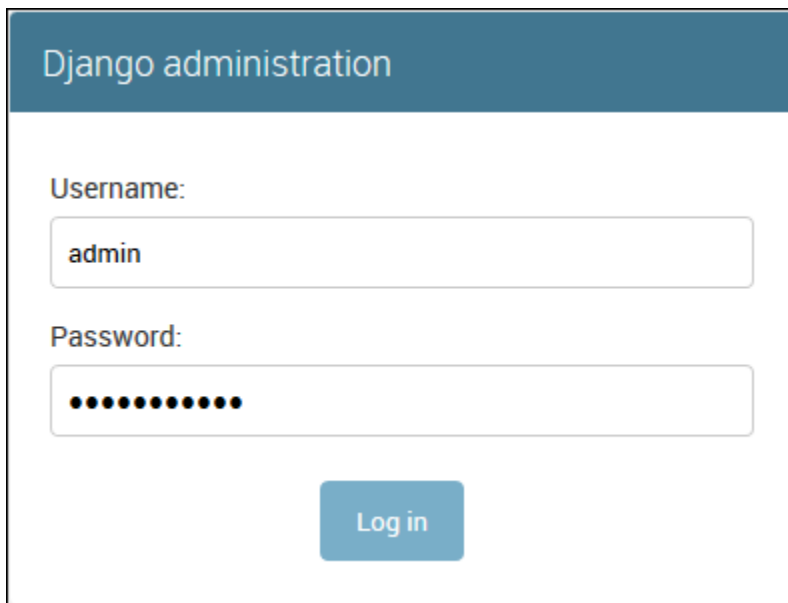
```
(eb-virt) ~/ebdjango$ python manage.py collectstatic
119 static files copied to ~/ebdjango/static
```

5. Implementación de la aplicación

```
~/ebdjango$ eb deploy
```

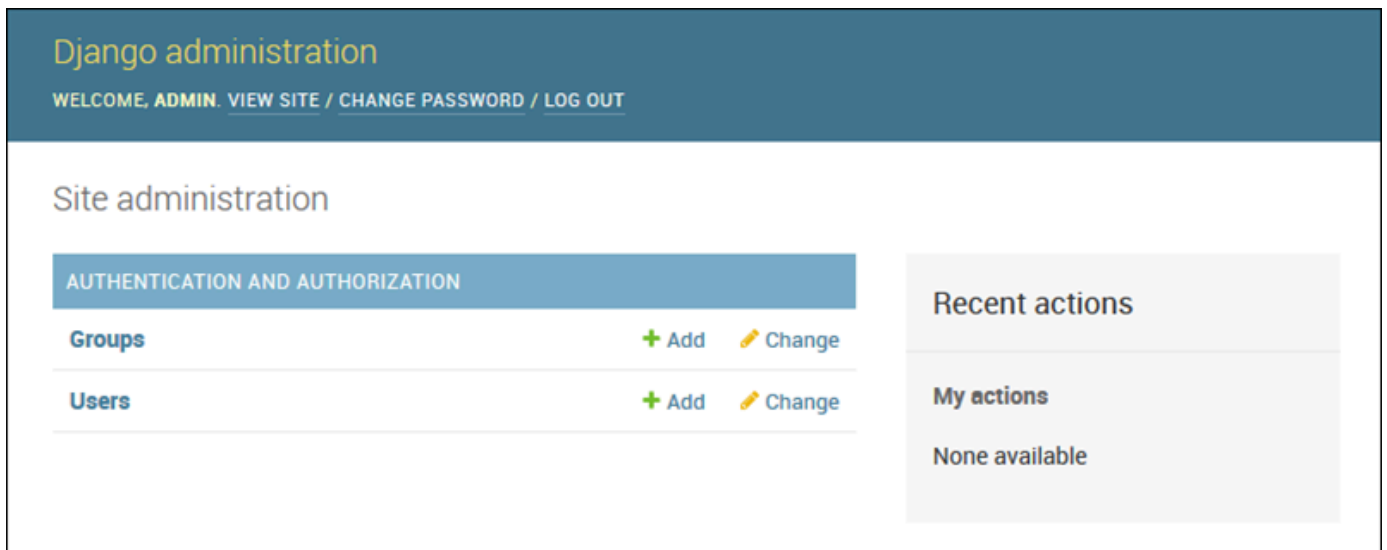
6. Consulte la consola de administración abriendo el sitio en el navegador y añadiendo `/admin/` a la dirección URL del sitio, como la siguiente.

```
http://django-env.p33kq46sfh.us-west-2.elasticbeanstalk.com/admin/
```



The screenshot shows the Django administration login interface. At the top, there is a dark blue header with the text "Django administration" in white. Below the header, there are two input fields: "Username:" with the text "admin" entered, and "Password:" with a masked password represented by ten black dots. A blue "Log in" button is positioned below the password field.

7. Inicie sesión con el nombre de usuario y la contraseña que ha configurado en el paso 2.



The screenshot displays the Django administration dashboard after a successful login. The top dark blue header contains the text "Django administration" in yellow, followed by "WELCOME, ADMIN." and navigation links: "VIEW SITE / CHANGE PASSWORD / LOG OUT". The main content area is titled "Site administration" and features a section for "AUTHENTICATION AND AUTHORIZATION" with a blue background. This section lists "Groups" and "Users", each with a "+ Add" link in green and a "Change" link with a pencil icon in yellow. To the right, there is a "Recent actions" section with a "My actions" link and the text "None available".

Puede utilizar un procedimiento similar de actualización/pruebas locales seguido de `eb deploy`. Elastic Beanstalk se encarga de la tarea de actualizar los servidores en directo, para que usted pueda centrarse en el desarrollo de la aplicación en lugar de en la administración del servidor.

Añadir un archivo de configuración de migración de la base de datos

Puede añadir comandos a su script `.ebextensions` que se ejecuten cuando se actualice el sitio. Esto le permite generar automáticamente migraciones de base de datos.

Para añadir un paso de migración cuando se implemente su aplicación

1. Cree un [archivo de configuración](#) denominado `db-migrate.config` con el contenido siguiente.

Example `~/ebdjango/.ebextensions/db-migrate.config`

```
container_commands:
  01_migrate:
    command: "source /var/app/venv/*/bin/activate && python3 manage.py migrate"
    leader_only: true
option_settings:
  aws:elasticbeanstalk:application:environment:
    DJANGO_SETTINGS_MODULE: ebdjango.settings
```

Este archivo de configuración activa el entorno virtual del servidor y ejecuta el comando `manage.py migrate` durante el proceso de implementación, antes de iniciar la aplicación. Como se ejecuta antes de iniciar la aplicación, deberá configurar la variable de entorno `DJANGO_SETTINGS_MODULE` explícitamente (normalmente `wsgi.py` se encarga de todo esto por usted durante el inicio). La especificación de `leader_only: true` en el comando garantiza que se ejecute solo una vez cuando realice la implementación en varias instancias.

2. Implementación de la aplicación

```
~/ebdjango$ eb deploy
```

Eliminar recursos

Para ahorrar horas de instancias y otros recursos de AWS entre las sesiones de implementación, termine el entorno de Elastic Beanstalk con `eb terminate`.

```
~/ebdjango$ eb terminate django-env
```

Este comando finaliza el entorno y todos los recursos de AWS que se ejecutan dentro de él. No elimina la aplicación. Al contrario, podrá crear más entornos con la misma configuración siempre que quiera ejecutando de nuevo `eb create`. Para obtener más información sobre los comandos de la CLI de EB, consulte [Administración de entornos de Elastic Beanstalk con la CLI de EB](#).

Si ha terminado de trabajar con la aplicación de ejemplo, puede eliminar también la carpeta del proyecto y el entorno virtual.

```
~$ rm -rf ~/eb-virt
~$ rm -rf ~/ebdjango
```

Pasos siguientes

Para obtener más información sobre Django, como un tutorial detallado, consulte [la documentación oficial](#).

Si desea probar otra plataforma web de Python, consulte [Implementación de una aplicación Flask en Elastic Beanstalk](#).

Agregar una instancia de base de datos de Amazon RDS a su entorno de aplicaciones Python

Puede utilizar una instancia de base de datos de Amazon Relational Database Service (Amazon RDS) para almacenar datos recopilados y modificados por su aplicación. La base de datos se puede adjuntar al entorno y administrarse con Elastic Beanstalk, o bien, crearse como desacoplada y administrarse de forma externa a través de otro servicio. En este tema se proporcionan instrucciones para crear un Amazon RDS mediante la consola de Elastic Beanstalk. La base de datos se acoplará al entorno y se administrará con Elastic Beanstalk. Para obtener más información sobre el uso de Elastic Beanstalk con Amazon RDS, consulte [Añadir una base de datos al entorno de Elastic Beanstalk](#).

Secciones

- [Añadir una instancia de base de datos al entorno](#)
- [Descargar un controlador](#)
- [Conexión a una base de datos](#)

Añadir una instancia de base de datos al entorno

Para agregar una instancia de base de datos al entorno

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Database (Base de datos), seleccione Edit (Editar).
5. Seleccione un motor de base de datos y escriba el nombre de usuario y la contraseña.
6. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

Se tarda alrededor de 10 minutos en añadir una instancia de base de datos. Cuando se haya completado la actualización del entorno, el nombre de host de la instancia de base de datos y otra información de conexión estarán disponibles para su aplicación mediante las siguientes propiedades de entorno

Nombre de la propiedad	Descripción	Valor de la propiedad
RDS_HOSTNAME	El nombre de host de la instancia de base de datos.	En la pestaña Connectivity & security (Conectividad y seguridad) de la consola de Amazon RDS: Endpoint (Punto de enlace).
RDS_PORT	El puerto en donde la instancia de base de datos acepta las conexiones. El valor predeterminado varía	En la pestaña Connectivity & security (Conectividad y seguridad) de la consola de Amazon RDS: Port (Puerto).

Nombre de la propiedad	Descripción	Valor de la propiedad
	dependiendo del motor de base de datos.	
RDS_DB_NAME	El nombre de la base de dato, ebdb .	En la pestaña Configuration (Configuración) de la consola de Amazon RDS: DB Name (Nombre de la base de datos).
RDS_USERNAME	El nombre de usuario que ha configurado para la base de datos.	En la pestaña Configuration (Configuración) de la consola de Amazon RDS: Master username (Nombre de usuario maestro).
RDS_PASSWORD	La contraseña que ha configurado para la base de datos.	No está disponible como referencia en la consola de Amazon RDS.

Para obtener más información sobre la configuración de una instancia de bases de datos acopladas a un entorno de Elastic Beanstalk, consulte [Añadir una base de datos al entorno de Elastic Beanstalk](#).

Descargar un controlador

Añada el controlador de base de datos al [archivo de requisitos](#) de su proyecto.

Example requirements.txt: Django con MySQL

```
Django==2.2
mysqlclient==2.0.3
```

Paquetes de controladores comunes para Python

- MySQL – mysqlclient
- PostgreSQL – psycopg2
- Oracle: cx_Oracle

- SQL Server – adodbapi

Para obtener más información, consulte [Python DatabaseInterfaces](#) y [Django 2.2 - bases de datos compatibles](#).

Conexión a una base de datos

Elastic Beanstalk proporciona información de conexión para instancias de base de datos asociadas en propiedades de entorno. Use `os.environ['VARIABLE']` para leer las propiedades y configurar una conexión con la base de datos.

Example Archivo de configuración de Django: DATABASES Dictionary

```
import os

if 'RDS_HOSTNAME' in os.environ:
    DATABASES = {
        'default': {
            'ENGINE': 'django.db.backends.mysql',
            'NAME': os.environ['RDS_DB_NAME'],
            'USER': os.environ['RDS_USERNAME'],
            'PASSWORD': os.environ['RDS_PASSWORD'],
            'HOST': os.environ['RDS_HOSTNAME'],
            'PORT': os.environ['RDS_PORT'],
        }
    }
```

Herramientas y recursos de Python

Existen varios lugares a los que puede acudir para obtener ayuda adicional con el desarrollo de sus aplicaciones Python:

Recurso	Descripción
Boto (el SDK para Python de AWS)	Instalar Boto mediante GitHub.
Foro de desarrollo de Python	Publique sus preguntas y obtenga comentarios.
Centro para desarrolladores de Python	Un único lugar para obtener código de ejemplo, documentación, herramientas y recursos adicionales.

Creación e implementación de aplicaciones Ruby en Elastic Beanstalk

AWS Elastic Beanstalk para Ruby facilita la implementación, administración y escalado de sus aplicaciones web Ruby que utilizan Amazon Web Services. Elastic Beanstalk está disponible para todo aquel que desee desarrollar o alojar una aplicación web con Ruby. En esta sección se proporcionan instrucciones paso a paso para implementar una aplicación de ejemplo en Elastic Beanstalk mediante la interfaz de línea de comandos de Elastic Beanstalk (CLI de EB) y, a continuación, actualizar la aplicación para que use los marcos de aplicaciones web [Rails](#) y [Sinatra](#).

En los temas de este capítulo se supone que tiene conocimientos de los entornos de Elastic Beanstalk. Si no ha usado Elastic Beanstalk antes, pruebe el [tutorial de introducción](#) para conocer los conceptos básicos.

Temas

- [Configuración del entorno de desarrollo de Ruby](#)
- [Uso de la plataforma Elastic Beanstalk Ruby](#)
- [Implementación de una aplicación de rieles en Elastic Beanstalk](#)
- [Implementación de una aplicación sinatra en Elastic Beanstalk](#)
- [Incorporar una instancia de base de datos de Amazon RDS a un entorno de aplicaciones Ruby](#)

Configuración del entorno de desarrollo de Ruby

Configure el entorno de desarrollo de Ruby para probar la aplicación localmente antes de implementarla en AWS Elastic Beanstalk. En este tema, se explican los pasos de configuración del entorno de desarrollo y se incluyen enlaces a páginas de instalación donde encontrará herramientas útiles.

Para seguir los procedimientos de esta guía, necesitará un shell o un terminal de línea de comando donde pueda ejecutar los comandos. Los comandos aparecen en listas y van precedidos del símbolo del sistema (\$) y del nombre del directorio actual, si es aplicable.

```
~/eb-project$ this is a command  
this is output
```

En Linux y macOS, puede utilizar el administrador de shell y paquetes preferido. En Windows, puede [instalar el subsistema de Windows para Linux para](#) obtener una versión de Ubuntu y Bash integrada en Windows.

Para ver procedimientos de configuración y herramientas comunes que se utilizan en todos los lenguajes, consulte [Configuración de su máquina de desarrollo para su uso con Elastic Beanstalk](#).

Secciones

- [Instalación de Ruby](#)
- [Instalación del AWS SDK para Ruby](#)
- [Instalación de un IDE o un editor de texto](#)

Instalación de Ruby

Instale GCC si no tiene un compilador de C. En Ubuntu, use apt.

```
~$ sudo apt install gcc
```

En Amazon Linux, use yum.

```
~$ sudo yum install gcc
```

Instale RVM para administrar las instalaciones del lenguaje Ruby en su equipo. Utilice los comandos en rvm.io para obtener las claves de proyecto y ejecutar el script de instalación.

```
~$ gpg2 --recv-keys key1 key2  
~$ curl -sSL https://get.rvm.io | bash -s stable
```

Este script instala RVM en una carpeta denominada `.rvm` en el directorio de usuario y modifica el perfil de shell para cargar un script de instalación cada vez que abra un nuevo terminal. Cargue el script de forma manual para comenzar.

```
~$ source ~/.rvm/scripts/rvm
```

Utilice `rvm get head` para obtener la última versión.

```
~$ rvm get head
```

Consulte las versiones disponibles de Ruby.

```
~$ rvm list known
# MRI Rubies
...
[ruby-]2.6[.8]
[ruby-]2.7[.4]
[ruby-]3[.0.2]
...
```

Consulte [Ruby](#) en el documento AWS Elastic Beanstalk Platforms (Plataformas de AEBlong) para encontrar la versión más reciente de Ruby disponible en una plataforma de Elastic Beanstalk. Instale dicha versión.

```
~$ rvm install 3.0.2
Searching for binary rubies, this might take some time.
Found remote file https://rubies.travis-ci.org/ubuntu/20.04/x86_64/ruby-3.0.2.tar.bz2
Checking requirements for ubuntu.
Updating system..
...
Requirements installation successful.
ruby-3.0.2 - #configure
ruby-3.0.2 - #download
...
```

Pruebe la instalación de Ruby.

```
~$ ruby --version
ruby 3.0.2p107 (2021-07-07 revision 0db68f0233) [x86_64-linux]
```

Instalación del AWS SDK para Ruby

Si necesita administrar AWS los recursos desde su aplicación, instale el AWS SDK for Ruby. Por ejemplo, con el SDK para Ruby, puede utilizar Amazon DynamoDB (DynamoDB) para almacenar información de usuario y sesión sin crear una base de datos relacional.

Instale el SDK para Ruby y sus dependencias con el comando gem.

```
$ gem install aws-sdk
```

Visite la [página de inicio de AWS SDK for Ruby](#) si desea obtener más información e instrucciones de instalación.

Instalación de un IDE o un editor de texto

Los entornos de desarrollo integrado (IDE) cuentan con una amplia gama de características que facilitan el desarrollo de aplicaciones. Si no ha utilizado un IDE para el desarrollo de Ruby, pruebe Aptana RubyMine y compruebe cuál funciona mejor para usted.

- [Instalar Aptana](#)
- [RubyMine](#)

Note

Un IDE podría agregar archivos a la carpeta de proyectos que tal vez no quiera confirmar en el control de código fuente. Para evitar confirmar estos archivos en el control de código fuente, utilice `.gitignore` o la característica equivalente de la herramienta de control de código fuente.

Si simplemente quiere comenzar a escribir el código y no necesita todas las características de un IDE, considere la posibilidad de [instalar Sublime Text](#).

Uso de la plataforma Elastic Beanstalk Ruby

La plataforma Ruby de AWS Elastic Beanstalk es un conjunto de [configuraciones de entorno](#) para las aplicaciones web de Ruby que se pueden ejecutar detrás de un servidor proxy NGINX en un servidor de aplicaciones Puma. Cada ramificación de la plataforma corresponde a una versión de Ruby. Si utiliza RubyGems, puede [incluir un archivo Gemfile](#) en el paquete de código fuente para instalar paquetes durante la implementación.

Configuración del servidor de aplicaciones

Elastic Beanstalk instala el servidor de aplicaciones Puma en función de la ramificación de la plataforma Ruby que elija al crear su entorno. Para obtener más información acerca de los componentes que se proporcionan con las versiones de plataforma Ruby, consulte [Plataformas admitidas](#) en la guía Plataformas de AWS Elastic Beanstalk.

Puede configurar la aplicación con su propio servidor Puma proporcionado. Esto proporciona la opción de usar una versión de Puma que no sea la preinstalada con la ramificación de la plataforma Ruby. También puede configurar la aplicación para que utilice un servidor de aplicaciones diferente, como Passenger. Para ello, debe incluir y personalizar un Gemfile en su implementación. También es necesario configurar un Procfile para iniciar el servidor de aplicaciones. Para obtener más información, consulte [Configuración del proceso de la aplicación con un Procfile](#).

Otras opciones de configuración

Elastic Beanstalk proporciona [opciones de configuración](#) que puede utilizar para personalizar el software que se ejecuta en las instancias de Amazon Elastic Compute Cloud (Amazon EC2) en su entorno de Elastic Beanstalk. Puede configurar las variables de entorno que necesita la aplicación, habilitar la rotación de logs en Amazon S3 y asignar carpetas del código fuente de la aplicación que contengan archivos estáticos a rutas proporcionadas por el servidor proxy. La plataforma también preestablece algunas variables de entorno comunes relacionadas con Rails y Rack para facilitar la detección y el uso.

En la consola de Elastic Beanstalk hay opciones de configuración disponibles para [modificar la configuración de un entorno en ejecución](#). Para evitar perder la configuración del entorno cuando lo termina, puede usar las [configuraciones guardadas](#) para guardar la configuración y aplicarla posteriormente a otro entorno.

Para guardar la configuración en el código fuente, puede incluir [archivos de configuración](#). Los valores de configuración de los archivos de configuración se aplican cada vez que crea un entorno o que implementa la aplicación. También puede usar archivos de configuración para instalar paquetes, ejecutar scripts y llevar a cabo otras operaciones de personalización de instancias durante las implementaciones.

La configuración aplicada en la consola de Elastic Beanstalk anula la misma configuración en los archivos de configuración, si existe. Esto le permite tener la configuración predeterminada en los archivos de configuración y anularla con la configuración específica del entorno en la consola. Para obtener más información acerca de la prioridad y otros métodos para cambiar valores de configuración, consulte [Opciones de configuración](#).

Para obtener más información sobre las diversas formas en las que puede ampliar una plataforma Elastic Beanstalk basada en Linux, consulte [the section called “Ampliación de plataformas Linux”](#).

Configuración del entorno de Ruby

Puede utilizar la consola de Elastic Beanstalk para habilitar la rotación de registros en Amazon S3 y configurar variables que la aplicación pueda leer desde el entorno.

Para tener acceso a las opciones de configuración de software del entorno

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Actualizaciones, supervisión y registro, seleccione Editar.

Log Options (Opciones de registro)

La sección Log Options (Opciones de registro) tiene dos valores:

- Instance profile (Perfil de instancia): especifica el perfil de instancia que tiene permiso para obtener acceso al bucket de Amazon S3 asociado con la aplicación.
- Enable log file rotation to Amazon S3 (Habilitar la rotación de archivos de registro para Amazon S3): especifica si los archivos de registro de las instancias de Amazon EC2 de la aplicación se copian en el bucket de Amazon S3 asociado a la aplicación.

Archivos estáticos

Para mejorar el rendimiento, puede utilizar la sección Static files (Archivos estáticos) para configurar el servidor proxy para entregar archivos estáticos de servidor (por ejemplo, HTML o imágenes) desde un conjunto de directorios dentro de su aplicación web. Para cada directorio, se establece la ruta virtual para la asignación de directorios. Cuando el servidor proxy recibe una solicitud de un archivo en la ruta especificada, proporciona directamente el archivo en lugar de direccionar la solicitud a la aplicación.

Para obtener más información sobre la configuración de archivos estáticos mediante archivos de configuración o la consola de Elastic Beanstalk, consulte [the section called “Archivos estáticos”](#).

De forma predeterminada, el servidor proxy en un entorno Ruby se configura de modo tal que sirva los archivos estáticos como se indica a continuación:

- Archivos en la carpeta `public` se sirven desde la ruta `/public` y la raíz del dominio (ruta `/`).
- Archivos en la subcarpeta `public/assets` se sirven desde la ruta `/assets`.

En los siguientes ejemplos se ilustra el funcionamiento de la configuración predeterminada:

- Por ejemplo, si el código fuente de la aplicación contiene un archivo llamado `logo.png` en una carpeta denominada `public`, el servidor proxy lo proporciona a los usuarios desde `subdomain.elasticbeanstalk.com/public/logo.png` y `subdomain.elasticbeanstalk.com/logo.png`.
- Si el código fuente de la aplicación contiene un archivo llamado `logo.png` en una carpeta denominada `assets` dentro de la carpeta `public`, el servidor proxy lo proporciona desde `subdomain.elasticbeanstalk.com/assets/logo.png`.

Puede configurar más asignaciones para los archivos estáticos. Para obtener más información, consulte [Espacios de nombres de la configuración de Ruby](#) más adelante en este tema.

Note

En el caso de las versiones de la plataforma anteriores a la versión 3.3.7 de Ruby 2.7 AL2, la configuración predeterminada del servidor proxy nginx de Elastic Beanstalk no es compatible con que se sirvan archivos estáticos desde la raíz del dominio (`subdomain.elasticbeanstalk.com/`). Esta versión de la plataforma se lanzó el 21 de octubre de 2021. Para obtener más información, consulte [Nuevas versiones de la plataforma: Ruby](#) en la Notas de la versión de AWS Elastic Beanstalk.

Propiedades del entorno

La sección Environment Properties (Propiedades de entorno) le permite especificar opciones de configuración del entorno en las instancias de Amazon EC2 que ejecutan la aplicación. Las propiedades del entorno se pasan como pares de clave-valor a la aplicación.

La plataforma Ruby define las siguientes propiedades de configuración del entorno:

- `BUNDLE_WITHOUT`: lista de grupos separados por dos puntos que se van a omitir al [instalar dependencias](#) desde un [archivo Gemfile](#).
- `BUNDLER_DEPLOYMENT_MODE`: establézcalo en `true` (opción por defecto) para instalar dependencias en el [modo de implementación](#) mediante Bundler. Cámbielo a `false` para ejecutar `bundle install` en modo de desarrollador.

Note

Esta propiedad de entorno no está definida en las ramas de la plataforma Ruby AMI de Amazon Linux (antes de Amazon Linux 2).

- `RAILS_SKIP_ASSET_COMPILATION`: establézcalo en `true` para omitir la ejecución de [rake assets:precompile](#) durante la implementación.
- `RAILS_SKIP_MIGRATIONS`: establézcalo en `true` para omitir la ejecución de [rake db:migrate](#) durante la implementación.
- `RACK_ENV`: especifique la fase del entorno de Rack. Por ejemplo, `development`, `production` o `test`.

Dentro del entorno de Ruby que se ejecuta en Elastic Beanstalk, es posible acceder a las variables de entorno mediante el objeto `ENV`. Por ejemplo, podría leer una propiedad denominada `API_ENDPOINT` en una variable con el siguiente código:

```
endpoint = ENV['API_ENDPOINT']
```

Para obtener más información, consulte [Propiedades del entorno y otras opciones del software](#).

Espacios de nombres de la configuración de Ruby

Puede usar un [archivo de configuración](#) para definir opciones de configuración y realizar otras tareas de configuración en las instancias durante las implementaciones. Las opciones de configuración se pueden definir a través del servicio de Elastic Beanstalk o la plataforma que utilice y están organizadas por espacios de nombres.

Puede usar el espacio de nombres

`aws:elasticbeanstalk:environment:proxy:staticfiles` para configurar el proxy

del entorno para que sirva archivos estáticos. Defina mapeos de rutas virtuales a directorios de aplicaciones.

La plataforma Ruby no define ningún espacio de nombres específico de la plataforma. En su lugar, define propiedades para el entorno para opciones comunes de Rails y Rack.

El siguiente archivo de configuración especifica una opción de archivos estáticos que asigna un directorio denominado `staticimages` a la ruta `/images`, establece cada una de las propiedades de entorno definidas por la plataforma y establece una propiedad de entorno adicional llamada `LOGGING`.

Example `.ebextensions/ruby-settings.config`

```
option_settings:
  aws:elasticbeanstalk:environment:proxy:staticfiles:
    /images: staticimages
  aws:elasticbeanstalk:application:environment:
    BUNDLE_WITHOUT: test
    BUNDLER_DEPLOYMENT_MODE: true
    RACK_ENV: development
    RAILS_SKIP_ASSET_COMPILATION: true
    RAILS_SKIP_MIGRATIONS: true
    LOGGING: debug
```

Note

La propiedad del entorno `BUNDLER_DEPLOYMENT_MODE` y el espacio de nombres `aws:elasticbeanstalk:environment:proxy:staticfiles` no están definidos en las ramas de la plataforma Ruby de la AMI de Amazon Linux (anterior a Amazon Linux 2).

Elastic Beanstalk cuenta con numerosas opciones de configuración para personalizar el entorno. Además de los archivos de configuración, también puede definir opciones en la consola, configuraciones guardadas, la CLI de EB o la AWS CLI. Para obtener más información, consulte [Opciones de configuración](#).

Instalación de paquetes con un archivo Gemfile

Utilice un archivo `Gemfile` en la raíz del código fuente del proyecto si desea utilizar RubyGems para instalar los paquetes que necesita la aplicación.

Example Archivo Gemfile

```
source "https://rubygems.org"  
gem 'sinatra'  
gem 'json'  
gem 'rack-parser'
```

Cuando hay un archivo `Gemfile`, Elastic Beanstalk ejecuta `bundle install` para instalar dependencias. Para obtener más información, consulte las páginas [Gemfile](#) y [Bundler](#) en el sitio web de Bundler.io.

Note

Puede usar una versión diferente de Puma además de la predeterminada que viene preinstalada con la plataforma Ruby. Para ello, incluya una entrada en un `Gemfile` que especifique la versión. También puede especificar un servidor de aplicaciones diferente, como Passenger, mediante un `Gemfile` personalizado.

Para ambos casos, es necesario configurar un `Procfile` para iniciar el servidor de aplicaciones.

Para obtener más información, consulte [Configuración del proceso de la aplicación con un Procfile](#).

Configuración del proceso de la aplicación con un Procfile

Para especificar el comando que inicia su aplicación Ruby, incluya un archivo llamado `Procfile` en la raíz de su paquete de código fuente.

Note

Elastic Beanstalk no admite esta función en las ramas de la plataforma Ruby AMI de Amazon Linux (anterior a Amazon Linux 2). Las ramas de plataforma con nombres que contengan con Puma o con Passenger, independientemente de sus versiones de Ruby, preceden a Amazon Linux 2 y no admiten la característica `Procfile`.

Para obtener más información sobre cómo escribir y utilizar un `Procfile`, expanda la sección `Buildfile` y `Procfile` en [the section called “Ampliación de plataformas Linux”](#).

Cuando no se proporciona un `Procfile`, Elastic Beanstalk genera el siguiente archivo predeterminado, que asume que se está utilizando el servidor de aplicaciones Puma previamente instalado.

```
web: puma -C /opt/elasticbeanstalk/config/private/pumaconf.rb
```

Si desea utilizar un servidor Puma propio proporcionado por usted, puede instalarlo usando un [Gemfile](#). El siguiente `Procfile` de ejemplo muestra cómo iniciarlo.

Example Procfile

```
web: bundle exec puma -C /opt/elasticbeanstalk/config/private/pumaconf.rb
```

Si desea utilizar el servidor de aplicaciones Passenger, utilice los siguientes archivos de ejemplo para configurar su entorno Ruby para instalar y utilizar Passenger.

1. Utilice este archivo de ejemplo para instalar Passenger.

Example Archivo Gemfile

```
source 'https://rubygems.org'  
gem 'passenger'
```

2. Utilice este archivo de ejemplo para indicar a Elastic Beanstalk que inicie Passenger.

Example Procfile

```
web: bundle exec passenger start /var/app/current --socket /var/run/puma/my_app.sock
```

Note

No es necesario cambiar nada en la configuración del servidor proxy nginx para usar Passenger. Para utilizar otros servidores de aplicaciones, es posible que necesite personalizar la configuración de nginx para reenviar correctamente las solicitudes a la aplicación.

Implementación de una aplicación de rieles en Elastic Beanstalk

Rails es un marco de código abierto model-view-controller (MVC) para Ruby. Este tutorial explica el proceso de generar una aplicación de Rails y desplegarla en un AWS Elastic Beanstalk entorno.

Secciones

- [Requisitos previos](#)
- [Lance un entorno de Elastic Beanstalk](#)
- [Instalación de Rails y generación de un sitio web](#)
- [Configuración de los ajustes de Rails](#)
- [Implementar la aplicación](#)
- [Limpieza](#)
- [Siguiendo pasos](#)

Requisitos previos

Conocimiento básico de Elastic Beanstalk

Este tutorial asume que tiene conocimiento de las operaciones básicas de Elastic Beanstalk y de la consola de Elastic Beanstalk. Si aún no lo ha hecho, siga las instrucciones que se indican en [Introducción a Elastic Beanstalk](#) para lanzar su primer entorno de Elastic Beanstalk.

Línea de comandos

Para seguir los procedimientos de esta guía, necesitará un shell o un terminal de línea de comando donde pueda ejecutar los comandos. Los comandos aparecen en listas y van precedidos del símbolo del sistema (\$) y del nombre del directorio actual, si es aplicable.

```
~/eb-project$ this is a command  
this is output
```

En Linux y macOS, puede utilizar el administrador de shell y paquetes preferido. En Windows, puede [instalar el subsistema de Windows para Linux](#) y obtener una versión de Ubuntu y Bash integrada en Windows.

Dependencias de Rails

El marco de Rails 6.1.4.1 cuenta con las siguientes dependencias. Asegúrese de que las tiene todas instaladas.

- Ruby 2.5.0 o posterior: para obtener instrucciones de instalación, consulte [Configuración del entorno de desarrollo de Ruby](#).

En este tutorial utilizamos Ruby 3.0.2 y la versión correspondiente de la plataforma de Elastic Beanstalk.

- Node.js: para obtener instrucciones de instalación, consulte [Instalación de Node.js con un gestor de paquetes](#).
- Yarn: para obtener instrucciones de instalación, consulte la sección [Instalación](#) en el sitio web de Yarn.

Lance un entorno de Elastic Beanstalk

Utilice la consola de Elastic Beanstalk para crear un entorno de Elastic Beanstalk. Elija la plataforma Ruby y acepte la configuración predeterminada y el código de muestra.

Para lanzar un entorno (consola)

1. [Abra la consola de Elastic Beanstalk mediante este enlace preconfigurado: console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=Tutorials&EnvironmentType=LoadBalanced](https://console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=Tutorials&EnvironmentType=LoadBalanced)
2. En Platform (Plataforma), seleccione la plataforma y la ramificación de plataforma que coincidan con el idioma utilizado por la aplicación.
3. En Application code (Código de aplicación), seleccione Sample application (Aplicación de muestra).
4. Elija Review and launch (Revisar y lanzar).
5. Revise las opciones disponibles. Elija la opción disponible que quiera usar y, cuando esté listo, elija Create app (Crear aplicación).

El entorno tarda unos cinco minutos en crearse e incluye los recursos siguientes:

- Instancia de EC2: una máquina virtual de Amazon Elastic Compute Cloud (Amazon EC2) configurada para ejecutar aplicaciones web en la plataforma que elija.

Cada plataforma ejecuta un conjunto específico de software, archivos de configuración y scripts compatibles con una determinada versión de lenguaje, marco de trabajo y contenedor web (o una combinación de estos). La mayoría de las plataformas utilizan Apache o nginx como un proxy inverso que se sitúa delante de la aplicación web, reenvía las solicitudes a esta, administra los recursos estáticos y genera registros de acceso y errores.

- Instance security group (Grupo de seguridad de la instancia): grupo de seguridad de Amazon EC2 configurado para permitir el tráfico entrante en el puerto 80. Este recurso permite que el tráfico HTTP procedente del balanceador de carga llegue a la instancia EC2 en la que se ejecuta la aplicación web. De forma predeterminada, el tráfico no está permitido en otros puertos.
- Balanceador de carga: balanceador de carga de Elastic Load Balancing configurado para distribuir solicitudes a las instancias que se ejecutan en la aplicación. Los balanceadores de carga también permiten que las instancias no estén expuestas directamente a Internet.
- Grupo de seguridad del balanceador de carga: grupo de seguridad de Amazon EC2 configurado para permitir el tráfico entrante en el puerto 80. Este recurso permite que el tráfico HTTP procedente de Internet llegue al balanceador de carga. De forma predeterminada, el tráfico no está permitido en otros puertos.
- Grupo de Auto Scaling: grupo de Auto Scaling configurado para reemplazar una instancia si termina o deja de estar disponible.
- Bucket de Amazon S3: ubicación de almacenamiento para el código fuente, los registros y otros artefactos que se crean al utilizar Elastic Beanstalk.
- CloudWatch Alarmas de Amazon: dos CloudWatch alarmas que monitorean la carga de las instancias de su entorno y que se activan si la carga es demasiado alta o demasiado baja. Cuando se activa una alarma, en respuesta, el grupo de Auto Scaling aumenta o reduce los recursos.
- AWS CloudFormation pila: Elastic AWS CloudFormation Beanstalk se utiliza para lanzar los recursos de su entorno y propagar los cambios de configuración. Los recursos se definen en una plantilla que puede verse en la [consola de AWS CloudFormation](#).
- Nombre de dominio: nombre de dominio que direcciona el tráfico a la aplicación web con el formato *subdominio.región.elasticbeanstalk.com*.

Note

Para aumentar la seguridad de las aplicaciones de Elastic Beanstalk, el dominio `elasticbeanstalk.com` está registrado en la [lista de sufijos públicos \(PSL\)](#). Para mayor seguridad, se recomienda que utilice cookies con un prefijo `__Host-` - en caso de que necesite configurar cookies confidenciales en el nombre de dominio predeterminado de

sus aplicaciones de Elastic Beanstalk. Esta práctica le ayudará a defender su dominio contra intentos de falsificación de solicitudes entre sitios (CSRF). Para obtener más información, consulte la página de [configuración de cookies](#) en la red de desarrolladores de Mozilla.

Todos estos recursos los administra Elastic Beanstalk. Cuando termina su entorno, Elastic Beanstalk termina todos los recursos que este contiene.

Note

El bucket de Amazon S3 que Elastic Beanstalk crea se comparte entre los entornos y no se elimina durante la terminación del entorno. Para obtener más información, consulte [Usar Elastic Beanstalk con Amazon S3](#).

Instalación de Rails y generación de un sitio web

Instale Rails y sus dependencias con el comando `gem`.

```
~$ gem install rails
Fetching: concurrent-ruby-1.1.9.gem
Successfully installed concurrent-ruby-1.1.9
Fetching: rack-2.2.3.gem
Successfully installed rack-2.2.3
...
```

Pruebe la instalación de Rails.

```
~$ rails --version
Rails 6.1.4.1
```

Utilice `rails new` con el nombre de la aplicación para crear un nuevo proyecto de Rails.

```
~$ rails new ~/eb-rails
```

Rails crea un directorio con el nombre especificado, genera todos los archivos necesarios para ejecutar localmente un proyecto de ejemplo y, a continuación, ejecuta `bundler` para instalar todas las dependencias (Gems) definidas en el `Gemfile` del proyecto.

Note

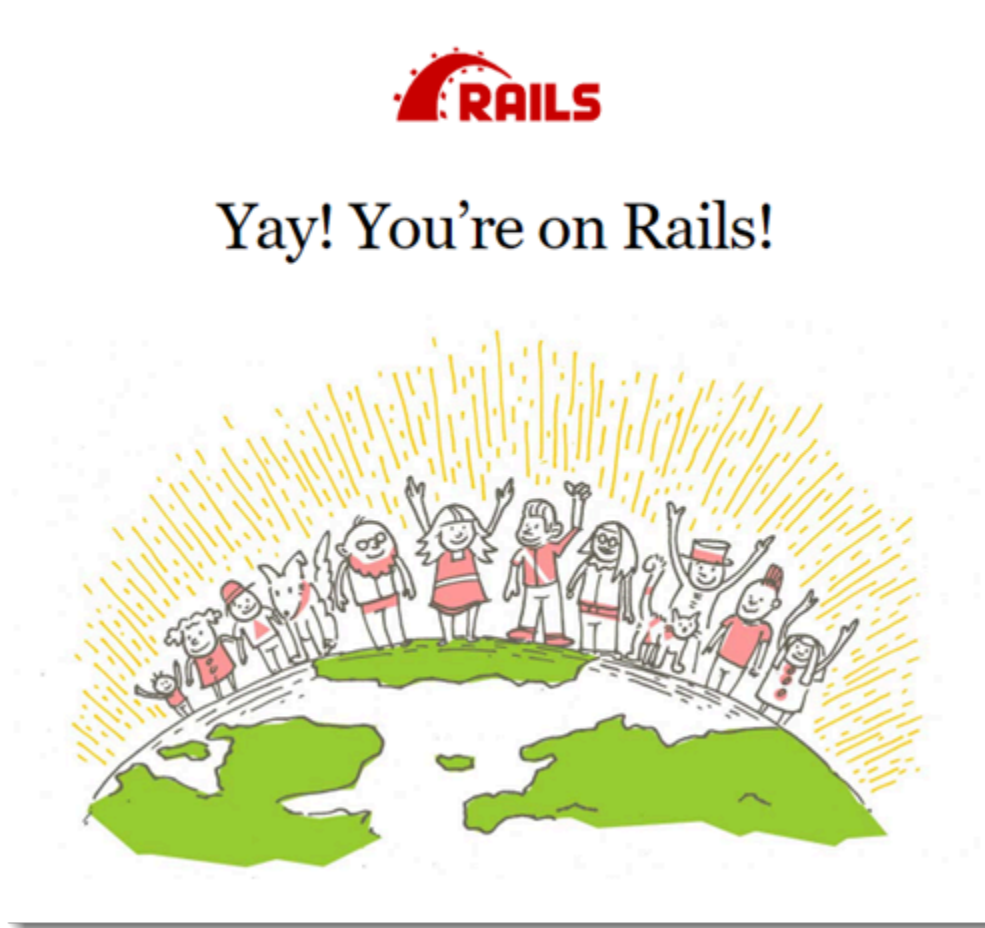
Este proceso instala la última versión de Puma para el proyecto. Esta versión puede ser diferente de la versión que Elastic Beanstalk proporciona en la versión de la plataforma Ruby de su entorno. Para ver las versiones de Puma proporcionadas por Elastic Beanstalk, consulte el [Historial de la plataforma Ruby](#) en la AWS Elastic Beanstalk Guía de plataformas. Para obtener más información acerca de la versión más reciente de Puma, consulte el sitio web de [Puma.io](#). Si hay una discrepancia entre las dos versiones de Puma, utilice una de las opciones siguientes:

- Uso de la versión Puma instalada por el comando `rails new` anterior. En este caso, debe agregar un `Procfile` para que la plataforma utilice la versión del servidor Puma propia que usted proporcionó. Para obtener más información, consulte [Configuración del proceso de la aplicación con un Procfile](#).
- Actualice la versión de Puma para que sea consistente con la versión preinstalada en la versión de la plataforma Ruby de su entorno. Para hacerlo, modifique la versión de Puma en [Gemfile](#) ubicada en la raíz del directorio fuente del proyecto. A continuación, ejecute `bundle update`. Para obtener más información, consulte la página [de actualización de Bundle](#) en el sitio web de Bundler.io.

Pruebe su instalación de Rails ejecutando el proyecto predeterminado localmente.

```
~$ cd eb-rails
~/eb-rails$ rails server
=> Booting Puma
=> Rails 6.1.4.1 application starting in development
=> Run `bin/rails server --help` for more startup options
Puma starting in single mode...
* Puma version: 5.5.2 (ruby 3.0.2-p107) ("Zawgyi")
* Min threads: 5
* Max threads: 5
* Environment: development
*           PID: 77857
* Listening on http://127.0.0.1:3000
* Listening on http://[::]:3000
Use Ctrl-C to stop
...
```

Abra `http://localhost:3000` en un navegador web para ver el proyecto predeterminado en acción.



En esta página solo es visible en el modo de desarrollo. Agregue contenido a la página de inicio de la aplicación para admitir la implementación de producción en Elastic Beanstalk. Use `rails generate` para crear un controlador, una ruta y una vista para la página de bienvenida.

```
~/eb-rails$ rails generate controller WelcomePage welcome
  create  app/controllers/welcome_page_controller.rb
  route  get 'welcome_page/welcome'
  invoke erb
  create  app/views/welcome_page
  create  app/views/welcome_page/welcome.html.erb
  invoke test_unit
  create  test/controllers/welcome_page_controller_test.rb
  invoke helper
  create  app/helpers/welcome_page_helper.rb
  invoke test_unit
  invoke assets
```

```
invoke coffee
create app/assets/javascripts/welcome_page.coffee
invoke scss
create app/assets/stylesheets/welcome_page.scss.
```

Con esto dispondrá de todo lo necesario para obtener acceso a la página en `/welcome_page/welcome`. Sin embargo, antes de publicar los cambios, cambie el contenido de la vista y añada una ruta para que esta página aparezca en el nivel superior del sitio.

Utilice un editor de texto para modificar el contenido de `app/views/welcome_page/welcome.html.erb`. En este ejemplo, usará `cat` para sobrescribir el contenido del archivo existente.

Example `app/views/welcome_page/welcome.html.erb`

```
<h1>Welcome!</h1>
<p>This is the front page of my first Rails application on Elastic Beanstalk.</p>
```

Por último, añada la siguiente ruta a `config/routes.rb`:

Example `config/routes.rb`

```
Rails.application.routes.draw do
  get 'welcome_page/welcome'
  root 'welcome_page#welcome'
```

Esto indica a Rails que dirija las solicitudes a la raíz del sitio web al método de bienvenida del controlador de la página de bienvenida, que presenta el contenido en la vista de bienvenida (`welcome.html.erb`).

Para que Elastic Beanstalk implemente correctamente la aplicación en la plataforma Ruby, es necesario actualizar `Gemfile.lock`. Es posible que algunas de las dependencias de `Gemfile.lock` sean específicas de la plataforma. Por lo tanto, es necesario agregar **platform ruby** a `Gemfile.lock` de modo que todas las dependencias requeridas se instalen con la implementación.

Example

```
~/eb-rails$ bundle lock --add-platform ruby
Fetching gem metadata from https://rubygems.org/.....
```

```
Resolving dependencies...  
Writing lockfile to /Users/janedoe/EBDPT/RubyApps/eb-rails-doc-app/Gemfile.lock
```

Configuración de los ajustes de Rails

Utilice la consola de Elastic Beanstalk para configurar Rails con propiedades de entorno. Establezca la propiedad del entorno de `SECRET_KEY_BASE` en una cadena de hasta 256 caracteres alfanuméricos.

Rails utiliza esta propiedad para crear claves. Por lo tanto, debe mantenerla en secreto y no almacenarla en el control de código fuente en texto no cifrado. En su lugar, debe incorporarla al código de Rails de su entorno mediante una propiedad de entorno.

Para configurar las propiedades del entorno en la consola de Elastic Beanstalk

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Actualizaciones, supervisión y registro, seleccione Editar.
5. Desplácese hacia abajo hasta Propiedades del entorno.
6. Seleccione Agregar propiedad de entorno.
7. Introduzca los pares Nombre y Valor de la propiedad.
8. Si necesita agregar más variables, repita Paso 6 y Paso 7.
9. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

Ahora ya está listo para implementar el sitio en su entorno.

Implementar la aplicación

Cree un [paquete de código fuente](#) que contenga los archivos creados por Rails. El comando siguiente crea un paquete de código fuente denominado `rails-default.zip`.

```
~/eb-rails$ zip ../rails-default.zip -r * .[^.]*
```

Cargue el paquete de origen en Elastic Beanstalk para implementar Rails en su entorno.

Para implementar un paquete de código fuente

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En la página de información general del entorno, elija Upload and deploy (Cargar e implementar).
4. Utilice el cuadro de diálogo en pantalla para cargar el paquete de código fuente.
5. Elija Deploy (Implementar).
6. Cuando se complete la implementación, seleccione la URL del para abrir el sitio web en una nueva pestaña.

Limpieza

Cuando termine de trabajar con Elastic Beanstalk, puede terminar su entorno. [Elastic Beanstalk AWS cancela todos los recursos asociados al entorno, como las instancias de Amazon EC2, las instancias de bases de datos, los balanceadores de carga, los grupos de seguridad y las alarmas.](#)

Para cerrar el entorno de Elastic Beanstalk desde la consola

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. Elija Actions (Acciones) y, a continuación, Terminate Environment (Terminar el entorno).
4. Utilice el cuadro de diálogo en pantalla para confirmar la terminación del entorno.

Con Elastic Beanstalk, puede crear fácilmente un nuevo entorno para su aplicación en cualquier momento.

Siguientes pasos

Para obtener más información acerca de Rails, visite rubyonrails.org.

A medida que siga desarrollando la aplicación, es probable que desee contar con un mecanismo que le permita administrar los entornos e implementar la aplicación sin tener que crear manualmente un archivo .zip y cargarlo en la consola de Elastic Beanstalk. La interfaz de [línea de comandos de Elastic Beanstalk](#) (EB CLI easy-to-use) proporciona comandos para crear, configurar e implementar aplicaciones en entornos de Elastic Beanstalk desde la línea de comandos.

Por último, si pretende utilizar la aplicación en un entorno de producción, querrá [configurar un nombre de dominio personalizado](#) para el entorno y [habilitar HTTPS](#) para las conexiones seguras.

Implementación de una aplicación Sinatra en Elastic Beanstalk

En esta guía, se explica cómo implementar una sencilla aplicación web [Sinatra](#) en AWS Elastic Beanstalk.

Requisitos previos

Este tutorial asume que tiene conocimiento de las operaciones básicas de Elastic Beanstalk y de la consola de Elastic Beanstalk. Si aún no lo ha hecho, siga las instrucciones que se indican en [Introducción a Elastic Beanstalk](#) para lanzar su primer entorno de Elastic Beanstalk.

Para seguir los procedimientos de esta guía, necesitará un shell o un terminal de línea de comando donde pueda ejecutar los comandos. Los comandos aparecen en listas y van precedidos del símbolo del sistema (\$) y del nombre del directorio actual, si es aplicable.

```
~/eb-project$ this is a command  
this is output
```

En Linux y macOS, puede utilizar el administrador de shell y paquetes preferido. En Windows, puede [instalar el subsistema de Windows para Linux para](#) obtener una versión de Ubuntu y Bash integrada en Windows.

Sinatra 2.1.0 requiere la versión Ruby 2.3.0 o posterior. En este tutorial utilizamos Ruby 3.0.2 y la versión correspondiente de la plataforma de Elastic Beanstalk. Instale Ruby; para ello, siga las instrucciones de [Configuración del entorno de desarrollo de Ruby](#).

Lance un entorno de Elastic Beanstalk

Utilice la consola de Elastic Beanstalk para crear un entorno de Elastic Beanstalk. Elija la plataforma Ruby y acepte la configuración predeterminada y el código de muestra.

Para lanzar un entorno (consola)

1. [Abra la consola de Elastic Beanstalk mediante este enlace preconfigurado:](https://console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=Tutorials&EnvironmentType=LoadBalanced)
console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=Tutorials&EnvironmentType=LoadBalanced
2. En Platform (Plataforma), seleccione la plataforma y la ramificación de plataforma que coincidan con el idioma utilizado por la aplicación.
3. En Application code (Código de aplicación), seleccione Sample application (Aplicación de muestra).
4. Elija Review and launch (Revisar y lanzar).
5. Revise las opciones disponibles. Elija la opción disponible que quiera usar y, cuando esté listo, elija Create app (Crear aplicación).

El entorno tarda unos cinco minutos en crearse e incluye los recursos siguientes:

- Instancia de EC2: una máquina virtual de Amazon Elastic Compute Cloud (Amazon EC2) configurada para ejecutar aplicaciones web en la plataforma que elija.

Cada plataforma ejecuta un conjunto específico de software, archivos de configuración y scripts compatibles con una determinada versión de lenguaje, marco de trabajo y contenedor web (o una combinación de estos). La mayoría de las plataformas utilizan Apache o nginx como un proxy inverso que se sitúa delante de la aplicación web, reenvía las solicitudes a esta, administra los recursos estáticos y genera registros de acceso y errores.

- Instance security group (Grupo de seguridad de la instancia): grupo de seguridad de Amazon EC2 configurado para permitir el tráfico entrante en el puerto 80. Este recurso permite que el tráfico HTTP procedente del balanceador de carga llegue a la instancia EC2 en la que se ejecuta la aplicación web. De forma predeterminada, el tráfico no está permitido en otros puertos.

- **Balanceador de carga:** balanceador de carga de Elastic Load Balancing configurado para distribuir solicitudes a las instancias que se ejecutan en la aplicación. Los balanceadores de carga también permiten que las instancias no estén expuestas directamente a Internet.
- **Grupo de seguridad del balanceador de carga:** grupo de seguridad de Amazon EC2 configurado para permitir el tráfico entrante en el puerto 80. Este recurso permite que el tráfico HTTP procedente de Internet llegue al balanceador de carga. De forma predeterminada, el tráfico no está permitido en otros puertos.
- **Grupo de Auto Scaling:** grupo de Auto Scaling configurado para reemplazar una instancia si termina o deja de estar disponible.
- **Bucket de Amazon S3:** ubicación de almacenamiento para el código fuente, los registros y otros artefactos que se crean al utilizar Elastic Beanstalk.
- **CloudWatch Alarmas de Amazon:** dos CloudWatch alarmas que monitorean la carga de las instancias de su entorno y que se activan si la carga es demasiado alta o demasiado baja. Cuando se activa una alarma, en respuesta, el grupo de Auto Scaling aumenta o reduce los recursos.
- **AWS CloudFormation pila:** Elastic AWS CloudFormation Beanstalk se utiliza para lanzar los recursos de su entorno y propagar los cambios de configuración. Los recursos se definen en una plantilla que puede verse en la [consola de AWS CloudFormation](#).
- **Nombre de dominio:** nombre de dominio que direcciona el tráfico a la aplicación web con el formato *subdominio.región.elasticbeanstalk.com*.

Note

Para aumentar la seguridad de las aplicaciones de Elastic Beanstalk, el dominio elasticbeanstalk.com está registrado en la [lista de sufijos públicos \(PSL\)](#). Para mayor seguridad, se recomienda que utilice cookies con un prefijo `__Host-` en caso de que necesite configurar cookies confidenciales en el nombre de dominio predeterminado de sus aplicaciones de Elastic Beanstalk. Esta práctica le ayudará a defender su dominio contra intentos de falsificación de solicitudes entre sitios (CSRF). Para obtener más información, consulte la página de [configuración de cookies](#) en la red de desarrolladores de Mozilla.

Todos estos recursos los administra Elastic Beanstalk. Cuando termina su entorno, Elastic Beanstalk termina todos los recursos que este contiene.

Note

El bucket de Amazon S3 que Elastic Beanstalk crea se comparte entre los entornos y no se elimina durante la terminación del entorno. Para obtener más información, consulte [Usar Elastic Beanstalk con Amazon S3](#).

Escribir un sitio web de Sinatra básico

Para crear e implementar una aplicación de Sinatra

1. Cree un archivo de configuración llamado `config.ru` con el siguiente contenido.

Example `config.ru`

```
require './helloworld'  
run Sinatra::Application
```

2. Cree un archivo de código Ruby llamado `helloworld.rb` con el siguiente contenido.

Example `helloworld.rb`

```
require 'sinatra'  
get '/' do  
  "Hello World!"  
end
```

3. Cree un archivo Gemfile con el siguiente contenido.

Example Archivo Gemfile

```
source 'https://rubygems.org'  
gem 'sinatra'  
gem 'puma'
```

4. Ejecutar la instalación del paquete para generar el `Gemfile.lock`

Example

```
~/eb-sinatra$ bundle install  
Fetching gem metadata from https://rubygems.org/....
```

```
Resolving dependencies...
Using bundler 2.2.22
Using rack 2.2.3
...
```

5. Para que Elastic Beanstalk implemente correctamente la aplicación en la plataforma Ruby, es necesario actualizar `Gemfile.lock`. Es posible que algunas de las dependencias de `Gemfile.lock` sean específicas de la plataforma. Por lo tanto, es necesario agregar **platform ruby** a `Gemfile.lock` de modo que todas las dependencias requeridas se instalen con la implementación.

Example

```
~/eb-sinatra$ bundle lock --add-platform ruby
Fetching gem metadata from https://rubygems.org/....
Resolving dependencies...
Writing lockfile to /Users/janedoe/EBDPT/RubyApps/eb-sinatra/Gemfile.lock
```

6. Cree un archivo Procfile con el siguiente contenido:

Example Procfile

```
web: bundle exec puma -C /opt/elasticbeanstalk/config/private/pumaconf.rb
```

Implementar la aplicación

Cree un [paquete de código fuente](#) que contenga los archivos de código fuente. El comando siguiente crea un paquete de código fuente denominado `sinatra-default.zip`.

```
~/eb-sinatra$ zip ../sinatra-default.zip -r * .[^.]*
```

Cargue el paquete de origen en Elastic Beanstalk para implementar Sinatra en su entorno.

Para implementar un paquete de código fuente

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En la página de información general del entorno, elija Upload and deploy (Cargar e implementar).
4. Utilice el cuadro de diálogo en pantalla para cargar el paquete de código fuente.
5. Elija Deploy (Implementar).
6. Cuando se complete la implementación, seleccione la URL del para abrir el sitio web en una nueva pestaña.

Limpieza

Cuando termine de trabajar con Elastic Beanstalk, puede terminar su entorno. [Elastic Beanstalk AWS cancela todos los recursos asociados al entorno, como las instancias de Amazon EC2, las instancias de bases de datos, los balanceadores de carga, los grupos de seguridad y las alarmas.](#)

Para cerrar el entorno de Elastic Beanstalk desde la consola

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. Elija Actions (Acciones) y, a continuación, Terminate Environment (Terminar el entorno).
4. Utilice el cuadro de diálogo en pantalla para confirmar la terminación del entorno.

Con Elastic Beanstalk, puede crear fácilmente un nuevo entorno para su aplicación en cualquier momento.

Siguientes pasos

Para obtener más información acerca de Sinatra, visite sinatrarb.com.

A medida que siga desarrollando la aplicación, es probable que desee contar con un mecanismo que le permita administrar los entornos e implementar la aplicación sin tener que crear manualmente un archivo .zip y cargarlo en la consola de Elastic Beanstalk. La interfaz de [línea de comandos de Elastic Beanstalk](#) (EB CLI easy-to-use) proporciona comandos para crear, configurar e implementar aplicaciones en entornos de Elastic Beanstalk desde la línea de comandos.

Por último, si pretende utilizar la aplicación en un entorno de producción, querrá [configurar un nombre de dominio personalizado](#) para el entorno y [habilitar HTTPS](#) para las conexiones seguras.

Incorporar una instancia de base de datos de Amazon RDS a un entorno de aplicaciones Ruby

Puede utilizar una instancia de base de datos de Amazon Relational Database Service (Amazon RDS) para almacenar datos recopilados y modificados por su aplicación. La base de datos se puede adjuntar al entorno y administrarse con Elastic Beanstalk, o bien, crearse como desacoplada y administrarse de forma externa a través de otro servicio. En este tema se proporcionan instrucciones para crear un Amazon RDS mediante la consola de Elastic Beanstalk. La base de datos se acoplará al entorno y se administrará con Elastic Beanstalk. Para obtener más información sobre el uso de Elastic Beanstalk con Amazon RDS, consulte [Añadir una base de datos al entorno de Elastic Beanstalk](#).

Secciones

- [Añadir una instancia de base de datos al entorno](#)
- [Descarga de un adaptador](#)
- [Conexión a una base de datos](#)

Añadir una instancia de base de datos al entorno

Para agregar una instancia de base de datos al entorno

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Database (Base de datos), seleccione Edit (Editar).
5. Seleccione un motor de base de datos y escriba el nombre de usuario y la contraseña.
6. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

Se tarda alrededor de 10 minutos en añadir una instancia de base de datos. Cuando se haya completado la actualización del entorno, el nombre de host de la instancia de base de datos y otra información de conexión estarán disponibles para su aplicación mediante las siguientes propiedades de entorno

Nombre de la propiedad	Descripción	Valor de la propiedad
RDS_HOSTNAME	El nombre de host de la instancia de base de datos.	En la pestaña Connectivity & security (Conectividad y seguridad) de la consola de Amazon RDS: Endpoint (Punto de enlace).
RDS_PORT	El puerto en donde la instancia de base de datos acepta las conexiones. El valor predeterminado varía dependiendo del motor de base de datos.	En la pestaña Connectivity & security (Conectividad y seguridad) de la consola de Amazon RDS: Port (Puerto).
RDS_DB_NAME	El nombre de la base de dato, ebdb .	En la pestaña Configuration (Configuración) de la consola de Amazon RDS: DB Name (Nombre de la base de datos).

Nombre de la propiedad	Descripción	Valor de la propiedad
RDS_USERNAME	El nombre de usuario que ha configurado para la base de datos.	En la pestaña Configuration (Configuración) de la consola de Amazon RDS: Master username (Nombre de usuario maestro).
RDS_PASSWORD	La contraseña que ha configurado para la base de datos.	No está disponible como referencia en la consola de Amazon RDS.

Para obtener más información sobre la configuración de una instancia de bases de datos acopladas a un entorno de Elastic Beanstalk, consulte [Añadir una base de datos al entorno de Elastic Beanstalk](#).

Descarga de un adaptador

Añada el adaptador de base de datos al [archivo gem](#) del proyecto.

Example Gemfile: Rails con MySQL

```
source 'https://rubygems.org'
gem 'puma'
gem 'rails', '~> 6.1.4', '>= 6.1.4.1'
gem 'mysql2'
```

Archivos gem de adaptadores comunes para Ruby

- MySQL – [mysql2](#)
- PostgreSQL – [pg](#)
- Oracle: [activerecord-oracle_enhanced-adapter](#)
- SQL Server – [activerecord-sqlserver-adapter](#)

Conexión a una base de datos

Elastic Beanstalk proporciona información de conexión para instancias de base de datos asociadas en propiedades de entorno. Use `ENV['VARIABLE']` para leer las propiedades y configurar una conexión con la base de datos.

Example `config/database.yml`: configuración de la base de datos de Ruby en Rails (MySQL)

```
production:
  adapter: mysql2
  encoding: utf8
  database: <%= ENV['RDS_DB_NAME'] %>
  username: <%= ENV['RDS_USERNAME'] %>
  password: <%= ENV['RDS_PASSWORD'] %>
  host: <%= ENV['RDS_HOSTNAME'] %>
  port: <%= ENV['RDS_PORT'] %>
```

Tutoriales y ejemplos

Los tutoriales específicos del lenguaje y el marco se encuentran repartidos en la Guía para AWS Elastic Beanstalk desarrolladores. A medida que se publiquen, se irán agregando a esta lista tutoriales nuevos o actualizados. Las actualizaciones más recientes aparecen en primer lugar.

Estos tutoriales están dirigidos a usuarios con un nivel intermedio y es posible que no contengan instrucciones básicas, como el proceso de registro en AWS. Si es la primera vez que utiliza AWS Elastic Beanstalk, [consulte el tutorial de introducción para poner](#) en marcha su primer entorno de Elastic Beanstalk.

- Ruby on Rails: [Implementación de una aplicación de rieles en Elastic Beanstalk](#)
- Ruby y Sinatra: [Implementación de una aplicación sinatra en Elastic Beanstalk](#)
- Configuración de PHP y MySQL HA Configuration: [Implementación de una aplicación PHP de alta disponibilidad con una base de datos externa de Amazon RDS en Elastic Beanstalk](#)
- PHP y Laravel: [Implementación de una aplicación de Laravel en Elastic Beanstalk](#)
- PHP y CakePHP: [Implementación de una aplicación de CakePHP en Elastic Beanstalk](#)
- Configuración de PHP y Drupal HA: [Implementación de un sitio web de Drupal de alta disponibilidad con una base de datos externa de Amazon RDS en Elastic Beanstalk](#)
- Configuración de PHP y HA - WordPress [Implementación de un WordPress sitio web de alta disponibilidad con una base de datos externa de Amazon RDS en Elastic Beanstalk](#)
- Configuración de Node.js con DynamoDB HA: [Implementación de una aplicación Node.js con DynamoDB en Elastic Beanstalk](#)
- ASP.NET Core: [Tutorial: Implementación de una aplicación ASP.NET Core con Elastic Beanstalk](#)
- Python y Flask: [Implementación de una aplicación Flask en Elastic Beanstalk](#)
- Python y Django: [Implementación de una aplicación Django en Elastic Beanstalk](#)
- Node.js y Express: [Implementación de una aplicación Express en Elastic Beanstalk](#)
- Docker, PHP y nginx: [Entornos de Docker administrados por ECS con la consola de Elastic Beanstalk](#)

Puede descargar las aplicaciones de ejemplo que se utilizan en Elastic Beanstalk para crear un entorno sin necesidad de proporcionar un paquete de código fuente a través de los siguientes enlaces:

- Docker: [docker.zip](#)
- Docker multicontenedor: [2.zip docker-multicontainer-v](#)
- Docker preconfigurado (Glassfish): [1.zip docker-glassfish-v](#)
- Go: [go.zip](#)
- Corretto: [corretto.zip](#)
- Tomcat: [tomcat.zip](#)
- .NET Core en Linux: [.zip dotnet-core-linux](#)
- .NET Core: [dotnet-asp-windows .zip](#)
- Node.js: [nodejs.zip](#)
- PHP: [php.zip](#)
- Python: [python.zip](#)
- Ruby: [ruby.zip](#)

Las aplicaciones de ejemplo más complejas que muestran el uso de marcos web, bibliotecas y herramientas adicionales están disponibles como proyectos de código abierto en GitHub:

- [Equilibrio de carga WordPress \(tutorial\)](#): archivos de configuración para instalarlo de WordPress forma segura y ejecutarlo en un entorno de Elastic Beanstalk con equilibrio de carga.
- [Load Balanced Drupal \(tutorial\)](#): archivos de configuración e instrucciones para instalar Drupal de forma segura y ejecutarlo en un entorno de Elastic Beanstalk con balanceo de carga.
- [Scorekeep](#): API web RESTful que utiliza el marco Spring y proporciona una interfaz AWS SDK for Java para crear y administrar usuarios, sesiones y juegos. La API está empaquetada con una aplicación web de Angular 1.5 que utiliza la API a través de HTTP. Incluye sucursales que muestran la integración con Amazon Cognito y Amazon AWS X-Ray Relational Database Service.

La aplicación utiliza características de la plataforma de Java SE para descargar dependencias y compilarse en la instancia, lo que minimiza el tamaño del paquete de código fuente. La aplicación incluye también archivos de configuración nginx que invalidan la configuración predeterminada para servir la aplicación web front-end estáticamente en el puerto 80 a través del proxy y dirigir las solicitudes dirigidas a las rutas incluidas bajo `/api` a la API que se ejecuta en `localhost:5000`.

- [¿Tiene serpientes?](#) - Aplicación Tomcat que muestra el uso de RDS en una aplicación web Java EE en Elastic Beanstalk. El proyecto ilustra el uso de archivos de configuración de Elastic Beanstalk, Servlets, JSP, Simple Tag Support, Tag Files, JDBC, SQL, Log4J, Bootstrap y Jackson.

- [Locust Load Generator](#): en este proyecto, se muestra el uso de las características de la plataforma de Java SE para instalar y ejecutar [Locust](#), una herramienta de generación de cargas escrita en Python. El proyecto incluye archivos de configuración que instalan y configuran Locust, un script de compilación que configura una tabla de DynamoDB y un archivo Procfile que ejecuta Locust.
- [Compartir Your Thoughts \(tutorial\)](#): aplicación PHP que ilustra el uso de MySQL en Amazon RDS, Composer y los archivos de configuración.
- [Un nuevo inicio \(tutorial\)](#): aplicación de ejemplo de Node.js que muestra el uso de DynamoDB, el SDK JavaScript para Node.js, AWS la administración de paquetes de npm y los archivos de configuración.

Administración y configuración de aplicaciones de Elastic Beanstalk

El primer paso para usar AWS Elastic Beanstalk es crear una aplicación que represente a la aplicación web de AWS. En Elastic Beanstalk, una aplicación actúa como si fuera un contenedor para el entorno que ejecuta la aplicación web, así como las versiones del código fuente, las configuraciones guardadas, los registros y otros artefactos de la aplicación web que se crean con Elastic Beanstalk.

Para crear una aplicación

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Applications (Aplicaciones), y, a continuación, seleccione Create a new application (Crear una aplicación).
3. Utilice el formulario en pantalla para proporcionar un nombre de aplicación.
4. Si lo desea, proporcione una descripción y añada las claves y los valores de las etiquetas.
5. Seleccione Create (Crear).

Elastic Beanstalk

Create new application

Application information

Application Name

Maximum length of 100 characters, not including forward slash (/).

Description

Tags

Apply up to 50 tags. You can use tags to group and filter your resources. A tag is a key-value pair. The key must be unique within the resource and is case-sensitive. [Learn more](#)

Key	Value	
<input type="text"/>	<input type="text"/>	<input type="button" value="Remove tag"/>

50 remaining

Después de crear la aplicación, la consola le pedirá que cree un entorno para ella. Para obtener información detallada sobre todas las opciones disponibles, consulte [Creación de un entorno de Elastic Beanstalk](#).

Si ya no necesita una aplicación, puede eliminarla.

Warning

Cuando se elimina una aplicación, se terminan todos los entornos asociados, así como todas las versiones y las configuraciones guardadas que pertenecen a la aplicación.

Para eliminar una aplicación

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Applications (Aplicaciones), y, a continuación, seleccione la aplicación en la lista.
3. Elija Actions (Acciones) y, a continuación, elija Delete application (Eliminar aplicación).

Temas

- [Consola de administración de aplicaciones Elastic Beanstalk](#)
- [Administración de versiones de la aplicación](#)
- [Cree el paquete de código fuente de la aplicación.](#)
- [Etiquetar recursos de la aplicación Elastic Beanstalk](#)

Consola de administración de aplicaciones Elastic Beanstalk

Puede utilizar la consola de AWS Elastic Beanstalk para administrar las aplicaciones y sus versiones así como las configuraciones guardadas.

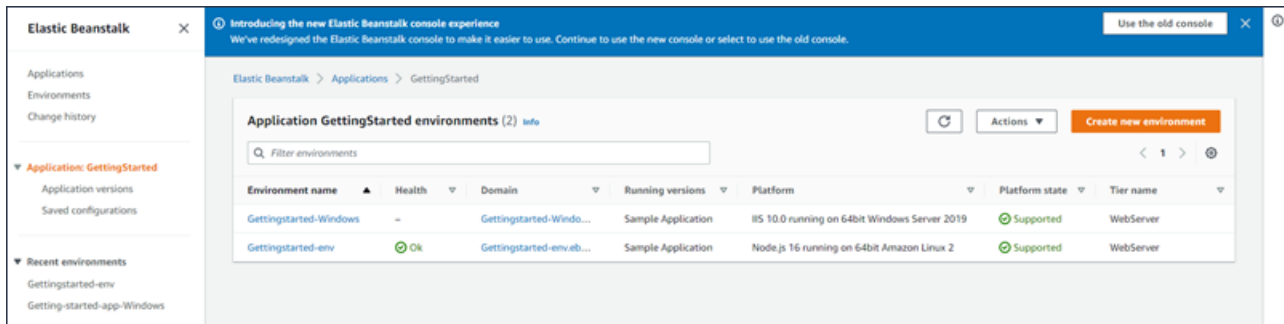
Para obtener acceso a la consola de administración de aplicaciones

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Applications (Aplicaciones) y, a continuación, elija el nombre de la aplicación de la lista.

Note

Si tiene muchas aplicaciones, utilice la barra de búsqueda para filtrar la lista de aplicaciones.

La página de descripción general de la aplicación muestra una lista con una descripción general de todos los entornos asociados a la aplicación.



3. Tienes varias maneras de continuar:

- a. Elija el menú desplegable Actions (Acciones) y, a continuación, elija una de las acciones de administración de aplicaciones. Para iniciar un entorno en esta aplicación, puede elegir directamente Create a new environment (Crear un nuevo entorno). Para obtener más información, consulte [the section called “Creación de entornos”](#).
- b. Elija un nombre de entorno para ir a la [consola de administración del entorno](#) de dicho entorno, en la que puede desea configurar, monitorizar o administrar el entorno.
- c. Seleccione Application versions (Versiones de la aplicación) siguiendo el nombre de la aplicación en el panel de navegación para ver y administrar las versiones de la aplicación.

Una versión de aplicación es una versión cargada del código de aplicación. Puede cargar nuevas versiones, implementar una versión existente en cualquiera de los entornos de la aplicación o eliminar versiones antiguas. Para obtener más información, consulte [Administración de versiones de la aplicación](#).

- d. Seleccione Saved configurations (Configuraciones guardadas) siguiendo el nombre de la aplicación en el panel de navegación para ver y administrar las configuraciones guardadas en entornos en ejecución.

Una configuración guardada es una colección de ajustes que se puede utilizar para restaurar la configuración de un entorno a un estado anterior o crear un entorno con la misma configuración. Para obtener más información, consulte [Uso de configuraciones guardadas de Elastic Beanstalk](#).

Administración de versiones de la aplicación

Elastic Beanstalk crea una versión de la aplicación cada vez que se carga el código fuente. Esto suele ocurrir cuando se crea un entorno o se carga e implementa código utilizando la [consola de administración del entorno](#) o la [CLI de EB](#). Elastic Beanstalk elimina estas versiones de la aplicación

según la política del ciclo de vida de la aplicación y cuándo elimine la aplicación. Para obtener información acerca de la política de ciclo de vida de la aplicación, consulte [Configuración del ciclo de vida de la versión de la aplicación](#).


Para cargar un paquete fuente sin implementarlo puede hacerlo en la [consola de administración de aplicaciones](#) o con el comando de la CLI de EB [eb appversion](#). Elastic Beanstalk almacena paquetes de origen en Amazon Simple Storage Service (Amazon S3) y no los elimina automáticamente.

Puede aplicar etiquetas a una versión de la aplicación cuando la cree y editar etiquetas de las versiones de la aplicación existentes. Para obtener más información, consulte [Etiquetado de versiones de la aplicación](#).

Para crear una nueva versión de la aplicación


También puede crear una nueva versión de la aplicación mediante la CLI de EB. Para obtener más información, consulte [eb appversion](#) en el capítulo Comandos de la CLI de EB.

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Applications (Aplicaciones) y, a continuación, elija el nombre de la aplicación de la lista.

 Note

Si tiene muchas aplicaciones, utilice la barra de búsqueda para filtrar la lista de aplicaciones.

3. En el panel de navegación, busque el nombre de la aplicación y elija Application versions (Versiones de la aplicación).
4. Seleccione Upload (Cargar). Utilice el formulario en pantalla para cargar el [paquete de código fuente](#) de la aplicación.


 Note

El límite de tamaño de archivo del paquete de código fuente es 500 MB.

5. Si lo desea, proporcione una breve descripción y añada las claves y los valores de etiqueta.
6. Seleccione Upload (Cargar).

El archivo especificado está vinculado con la aplicación. Puede implementar la versión de la aplicación en un entorno nuevo o existente.

Con el paso del tiempo, la aplicación puede acumular muchas versiones de la aplicación. Para ahorrar espacio de almacenamiento y evitar que se alcance la [cuota de versiones de la aplicación](#), es recomendable eliminar las versiones de la aplicación que ya no necesite.


 Note

La eliminación de una versión de la aplicación no afecta a los entornos que ejecutan actualmente esa versión.

Para eliminar una versión de la aplicación

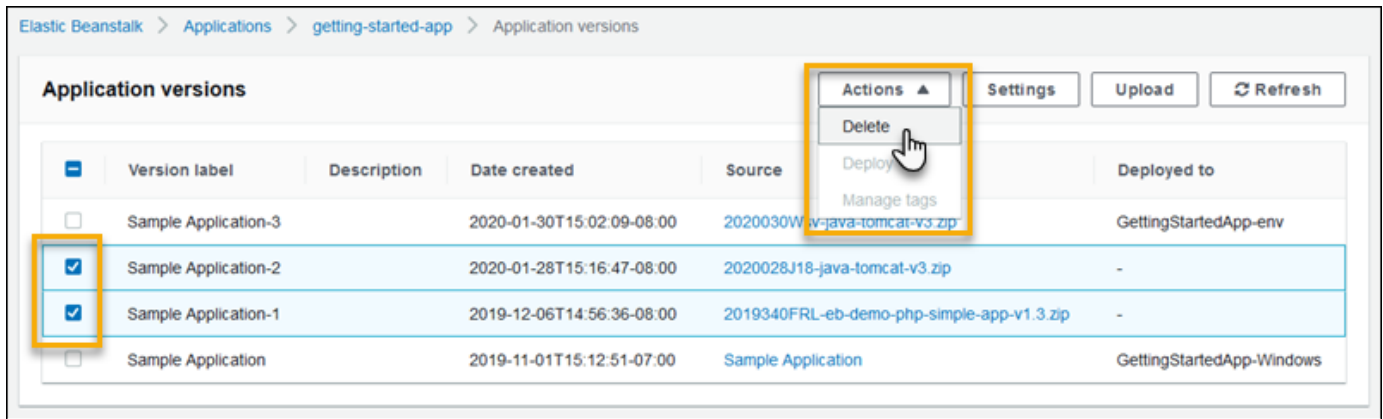
También puede eliminar una versión de la aplicación con la CLI de EB. Para obtener más información, consulte [eb appversion](#) en el capítulo Comandos de la CLI de EB.

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Applications (Aplicaciones) y, a continuación, elija el nombre de la aplicación de la lista.

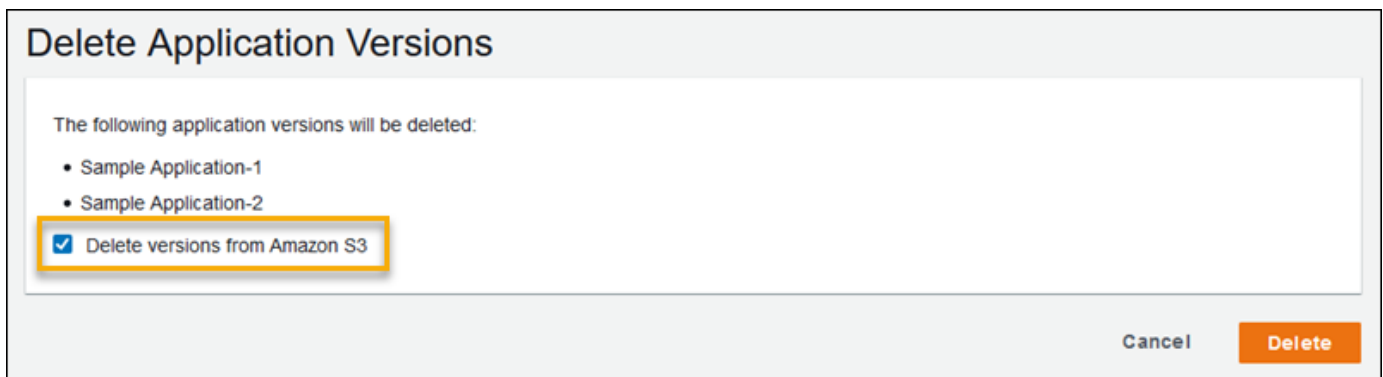
 Note

Si tiene muchas aplicaciones, utilice la barra de búsqueda para filtrar la lista de aplicaciones.

3. En el panel de navegación, busque el nombre de la aplicación y elija Application versions (Versiones de la aplicación).
4. Seleccione una o varias versiones de la aplicación que desee eliminar.



5. Seleccione Actions (Acciones) y, a continuación, seleccione Delete (Eliminar).
6. (Opcional) Para dejar el paquete de origen de la aplicación para estas versiones de aplicación en el bucket de Amazon Simple Storage Service (Amazon S3), desactive la casilla Eliminar versiones de Amazon S3.



7. Elija Eliminar (Delete).

También puede configurar Elastic Beanstalk para eliminar versiones antiguas automáticamente mediante la configuración del ciclo de vida de la versión de la aplicación. Si configura estas opciones del ciclo de vida, estas se aplican cuando se crean nuevas versiones de la aplicación. Por ejemplo, si establece un máximo de 25 versiones de la aplicación, Elastic Beanstalk elimina la versión más antigua cuando se carga la versión número 26. Si configura una antigüedad máxima de 90 días, las versiones con más de 90 días se eliminarán cuando se cargue una nueva versión. Para obtener más información, consulte [the section called “Ciclo de vida de la versión”](#).

Si no elige eliminar el paquete de origen de Amazon S3, Elastic Beanstalk seguirá eliminando la versión de sus registros. Sin embargo, el paquete de origen se deja en el [bucket de almacenamiento de Elastic Beanstalk](#). La cuota de versión de la aplicación sólo se aplica a las versiones de las pistas de Elastic Beanstalk. Por lo tanto, puede eliminar versiones para no sobrepasar la cuota, pero mantener todos los paquetes de código fuente de Amazon S3.

Note

El límite de versiones de la aplicación no se aplica a los paquetes de código fuente, pero podría seguir generando cargos de Amazon S3 y retener información personal más allá el momento en que la necesite. Elastic Beanstalk nunca elimina automáticamente los paquetes de origen. Debe eliminar los paquetes de código fuente cuando ya no los necesite.

Configuración del ciclo de vida de la versión de la aplicación

Cada vez que carga una nueva versión de su aplicación con la consola de Elastic Beanstalk o la CLI de EB, Elastic Beanstalk crea una [versión de la aplicación](#). Si no elimina las versiones que ya no utiliza, acabará superando la [cuota de versiones de aplicación](#) y no podrá crear nuevas versiones de dicha aplicación.

Puede evitar que esto ocurra aplicando una política de ciclo de vida de versiones de la aplicación a sus aplicaciones. Una política de ciclo de vida indica a Elastic Beanstalk que elimine las versiones de la aplicación que sean antiguas o cuando el número total de versiones de una aplicación supere un número especificado.

Elastic Beanstalk aplica la política de ciclo de vida de una aplicación cada vez que se crea una nueva versión de la aplicación y elimina hasta 100 versiones cada vez que se aplica la política de ciclo de vida. Elastic Beanstalk elimina las versiones antiguas después de crear la nueva versión. Además, la nueva versión no computa para el número máximo de versiones definido en la política.

Elastic Beanstalk no elimina las versiones de la aplicación que se están usando actualmente en un entorno ni las versiones de la aplicación implementadas en entornos que se terminaron menos de diez semanas antes de que se activara la política.

La cuota de versiones de la aplicación se aplica a todas las aplicaciones de una región. Si tiene varias aplicaciones, configure cada aplicación con la política de ciclo de vida correspondiente para evitar superar la cuota. Por ejemplo, si tiene 10 aplicaciones en una región y la cuota es de 1000 versiones de la aplicación, considere la posibilidad de establecer una política de ciclo de vida con una cuota de 99 versiones de la aplicación para todas las aplicaciones, o establezca otros valores en cada aplicación, siempre y cuando el total sea inferior a 1000 versiones de la aplicación. Elastic Beanstalk solo aplica la política si la versión de la aplicación se crea correctamente, por lo que si ya ha alcanzado el límite, debe eliminar algunas versiones manualmente antes de crear una nueva versión.

De forma predeterminada, Elastic Beanstalk deja el [paquete de origen](#) de la versión de la aplicación en Amazon S3 para evitar que se pierdan datos. Puede eliminar el paquete de código fuente para ahorrar espacio.

Puede establecer la configuración del ciclo de vida a través de la CLI de Elastic Beanstalk y las API. Consulte [eb appversion](#), [CreateApplication](#) (con el parámetro `ResourceLifecycleConfig`) y [UpdateApplicationResourceLifecycle](#) para obtener información detallada.

Configuración de las opciones del ciclo de vida de la aplicación en la consola

Puede especificar la configuración del ciclo de vida en la consola de Elastic Beanstalk.

Para especificar la configuración del ciclo de vida de la aplicación


1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Applications (Aplicaciones) y, a continuación, elija el nombre de la aplicación de la lista.

Note

Si tiene muchas aplicaciones, utilice la barra de búsqueda para filtrar la lista de aplicaciones.

3. En el panel de navegación, busque el nombre de la aplicación y elija Application versions (Versiones de la aplicación).
4. Elija Settings.
5. Utilice el formulario en pantalla para establecer la configuración del ciclo de vida de la aplicación.
6. Seleccione Save.

Application version lifecycle settings ✕

Configure a lifecycle policy to limit the number of application versions to retain for future deployments. This policy will not delete application versions that are currently deployed or are in the process of being created. [Learn more](#) 

Lifecycle policy

Enable

Lifecycle rule

Set the application versions limit by total count

200 Application Versions

Set the application versions limit by age

180 days

Retention

Delete source bundle from S3

Service role

Cancel Save

En la página Settings (Configuración), puede realizar lo siguiente:

- Establecer la configuración del ciclo de vida en función del recuento total de versiones de la aplicación o de la antigüedad de las versiones de la aplicación.
- Especificar si desea eliminar el paquete de código fuente de S3 cuando se elimine la versión de la aplicación.
- Especificar el rol del servicio en el que se elimina la versión de la aplicación. Para incluir todos los permisos necesarios para la eliminación de la versión, elija el rol de servicio de `aws-elasticbeanstalk-service-role` predeterminado, denominada Elastic Beanstalk u otro rol de servicio mediante las políticas de servicios administrados de Elastic Beanstalk. Para obtener más información, consulte [Administración de roles de servicio de Elastic Beanstalk](#).

Etiquetado de versiones de la aplicación

Puede aplicar etiquetas a las versiones de la aplicación AWS Elastic Beanstalk. Las etiquetas son pares de clave-valor asociadas a los recursos de AWS. Para obtener información sobre el etiquetado de recursos de Elastic Beanstalk, los casos de uso, las restricciones de las claves y los valores de las etiquetas y los tipos de recursos admitidos, consulte [Etiquetar recursos de la aplicación Elastic Beanstalk](#).

Puede especificar etiquetas al crear una versión de la aplicación. En una versión de la aplicación, puede añadir o eliminar etiquetas, y actualizar los valores de etiquetas existentes. Puede añadir hasta 50 etiquetas a cada versión de la versión de la aplicación.

Adición de etiquetas durante la creación de la versión de la aplicación

Cuando se utiliza la consola de Elastic Beanstalk para [crear un entorno](#), y elige cargar una versión de su código de la aplicación, puede especificar las claves y los valores de las etiquetas que se asociarán a la nueva versión de la aplicación.

También puede utilizar la consola de Elastic Beanstalk para [cargar una versión de la aplicación](#) sin usarla inmediatamente en un entorno. Puede especificar claves y valores de las etiquetas al cargar una versión de la aplicación.

Con la AWS CLI u otros clientes basados en API, agregue etiquetas con el parámetro `--tags` en el comando [create-application-version](#).

```
$ aws elasticbeanstalk create-application-version \  
  --tags Key=mytag1,Value=value1 Key=mytag2,Value=value2 \  
  --application-name my-app --version-label v1
```

Cuando se utiliza la CLI de EB para crear o actualizar un entorno, se crea una versión de la aplicación a partir del código que implemente. No hay un modo directo de etiquetar una versión de la aplicación durante su creación a través de la CLI de EB. Consulte la siguiente sección para obtener más información acerca de la adición de etiquetas a una versión de la aplicación existente.

Administración de etiquetas de una versión de la aplicación existente

Puede agregar, actualizar y eliminar etiquetas en una versión de aplicación de Elastic Beanstalk existente.

Para administrar las etiquetas de una versión de una aplicación mediante la consola de Elastic Beanstalk

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Applications (Aplicaciones) y, a continuación, elija el nombre de la aplicación de la lista.

Note

Si tiene muchas aplicaciones, utilice la barra de búsqueda para filtrar la lista de aplicaciones.

3. En el panel de navegación, busque el nombre de la aplicación y elija Application versions (Versiones de la aplicación).
4. Seleccione la versión de la aplicación que desea administrar.
5. Elija Actions (Acciones) y, a continuación, elija Manage Tags (Administrar etiquetas).
6. Utilice el formulario en pantalla para añadir, actualizar o eliminar etiquetas.
7. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

Si utiliza la CLI de EB para actualizar su versión de la aplicación, utilice [eb tags](#) para añadir, actualizar, eliminar o enumerar etiquetas.

Por ejemplo, el siguiente comando muestra las etiquetas en una versión de la aplicación.

```
~/workspace/my-app$ eb tags --list --resource "arn:aws:elasticbeanstalk:us-east-2:my-account-id:applicationversion/my-app/my-version"
```

El siguiente comando actualiza la etiqueta mytag1 y elimina la etiqueta mytag2.

```
~/workspace/my-app$ eb tags --update mytag1=newvalue --delete mytag2 \
  --resource "arn:aws:elasticbeanstalk:us-east-2:my-account-id:applicationversion/my-app/my-version"
```

Para obtener una lista de las opciones y más ejemplos, consulte [eb tags](#).

Con la AWS CLI u otros clientes basados en API, utilice el comando [list-tags-for-resource](#) para enumerar las etiquetas de una versión de la aplicación.


```
$ aws elasticbeanstalk list-tags-for-resource --resource-arn  
"arn:aws:elasticbeanstalk:us-east-2:my-account-id:applicationversion/my-app/my-  
version"
```

Utilice el comando [update-tags-for-resource](#) para añadir, actualizar o eliminar etiquetas en una versión de la aplicación.

```
$ aws elasticbeanstalk update-tags-for-resource \  
--tags-to-add Key=mytag1,Value=newvalue --tags-to-remove mytag2 \  
--resource-arn "arn:aws:elasticbeanstalk:us-east-2:my-account-  
id:applicationversion/my-app/my-version"
```

Especifique las etiquetas que desea añadir y las que desea actualizar en el parámetro `--tags-to-add` de `update-tags-for-resource`. Se añade una etiqueta no existente y se actualiza el valor de una etiqueta existente.

Note

Para utilizar algunos de los comandos de la CLI de EB y los comandos de la AWS CLI con una versión de la aplicación Elastic Beanstalk, necesita el ARN de la versión de la aplicación. Puede recuperar el ARN mediante el siguiente comando.

```
$ aws elasticbeanstalk describe-application-versions --application-name my-app  
--version-label my-version
```

Cree el paquete de código fuente de la aplicación.

Cuando utilice la consola de AWS Elastic Beanstalk para implementar una nueva aplicación o la versión de esta, necesitará cargar un paquete de código fuente. El paquete de código fuente debe cumplir los siguientes requisitos:

- Debe constar de un único archivo ZIP o WAR (puede incluir varios archivos WAR dentro del archivo ZIP).
- No puede superar los 500 MB.
- No puede incluir una carpeta principal o un directorio de nivel superior (sí puede incluir subdirectorios).

Si desea implementar una aplicación de trabajo que procese tareas periódicas en segundo plano, el paquete de código fuente de la aplicación también debe incluir un archivo `cron.yaml`. Para obtener más información, consulte [Tareas periódicas](#).

Si implementará la aplicación con la interfaz de línea de comandos de Elastic Beanstalk (CLI de EB), el AWS Toolkit for Eclipse o el AWS Toolkit for Visual Studio, el archivo ZIP o WAR se estructurará correctamente de manera automática. Para obtener más información, consulte [Uso de la interfaz de línea de comandos de Elastic Beanstalk \(CLI de EB\)](#), [Creación e implementación de aplicaciones Java en Elastic Beanstalk](#) y [El AWS Toolkit for Visual Studio](#).

Secciones

- [Creación de un paquete de código fuente desde la línea de comandos](#)
- [Creación de un paquete de código fuente con Git](#)
- [Comprimir archivos en Mac OS X Finder o en el Explorador de Windows](#)
- [Creación de un paquete de código fuente para una aplicación .NET](#)
- [Comprobación del paquete de código fuente](#)

Creación de un paquete de código fuente desde la línea de comandos

Cree un paquete de código fuente con el comando `zip`. Para incluir archivos y carpetas ocultos, utilice un patrón como el siguiente.

```
~/myapp$ zip ../myapp.zip -r * .[^.]*
  adding: app.js (deflated 63%)
  adding: index.js (deflated 44%)
  adding: manual.js (deflated 64%)
  adding: package.json (deflated 40%)
  adding: restify.js (deflated 85%)
  adding: .ebextensions/ (stored 0%)
  adding: .ebextensions/xray.config (stored 0%)
```

De este modo se garantiza que los [archivos de configuración](#) de Elastic Beanstalk y otros archivos y carpetas que comiencen con un punto se incluyan en el archivo.

Para aplicaciones web de Tomcat, utilice `jar` para crear un archivo web.

```
~/myapp$ jar -cvf myapp.war .
```

Los comandos anteriores incluyen archivos ocultos que pueden aumentar el tamaño del paquete de código fuente innecesariamente. Si desea más control, use un patrón de archivos más detallado o [cree su paquete de código fuente con Git](#).

Creación de un paquete de código fuente con Git

Si va a utilizar Git para administrar el código fuente de la aplicación, use el comando `git archive` para crear el paquete de código fuente.

```
$ git archive -v -o myapp.zip --format=zip HEAD
```

`git archive` solo incluye los archivos que se almacenan en git y excluye los archivos ignorados y los archivos de git. Esto ayuda a mantener el tamaño más pequeño posible del paquete de código fuente. Para obtener más información, vaya a la [página del manual de git-archive](#).

Comprimir archivos en Mac OS X Finder o en el Explorador de Windows

Cuando cree un archivo ZIP en Mac OS X Finder o el Explorador de Windows, asegúrese de comprimir los archivos y subcarpetas, y no la carpeta principal donde están incluidos.

Note

La interfaz gráfica de usuario (GUI) de Mac OS X y los sistemas operativos basados en Linux no muestran los archivos y carpetas cuyos nombres empiezan por punto (.). Utilice la línea de comando en lugar de la GUI para comprimir la aplicación si el archivo ZIP debe incluir una carpeta oculta; por ejemplo, `.ebextensions`. Si desea obtener información sobre los procedimientos de línea de comando para crear un archivo ZIP en Mac OS X o un sistema operativo basado en Linux, consulte [Creación de un paquete de código fuente desde la línea de comandos](#).

Example

Supongamos que tiene una carpeta de proyectos de Python llamada `myapp` que contiene los siguientes archivos y subcarpetas:

```
myapplication.py
README.md
static/
static/css
```

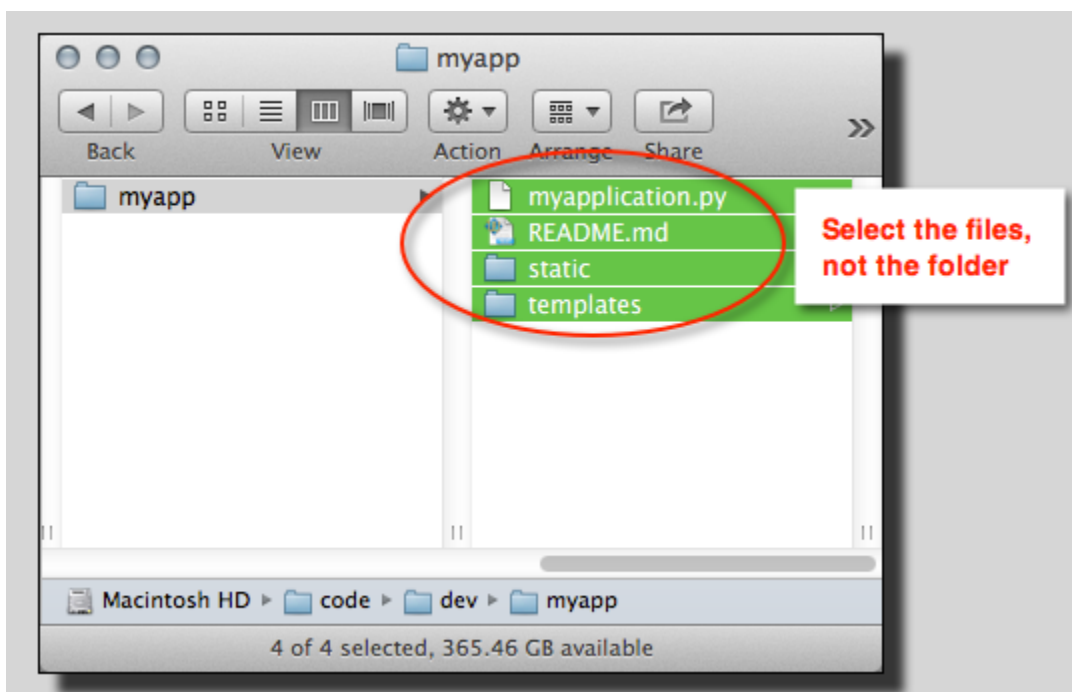
```
static/css/styles.css
static/img
static/img/favicon.ico
static/img/logo.png
templates/
templates/base.html
templates/index.html
```

Tal y como se indicó anteriormente en la lista de requisitos, el paquete de código fuente no puede comprimirse sin una carpeta principal, ya que su estructura al descomprimirse no incluye un directorio de nivel superior adicional. En este ejemplo, no debería crearse ninguna carpeta `myapp` al descomprimir los archivos (o, en la línea de comando, no debería agregarse ningún segmento `myapp` en las rutas de archivo).

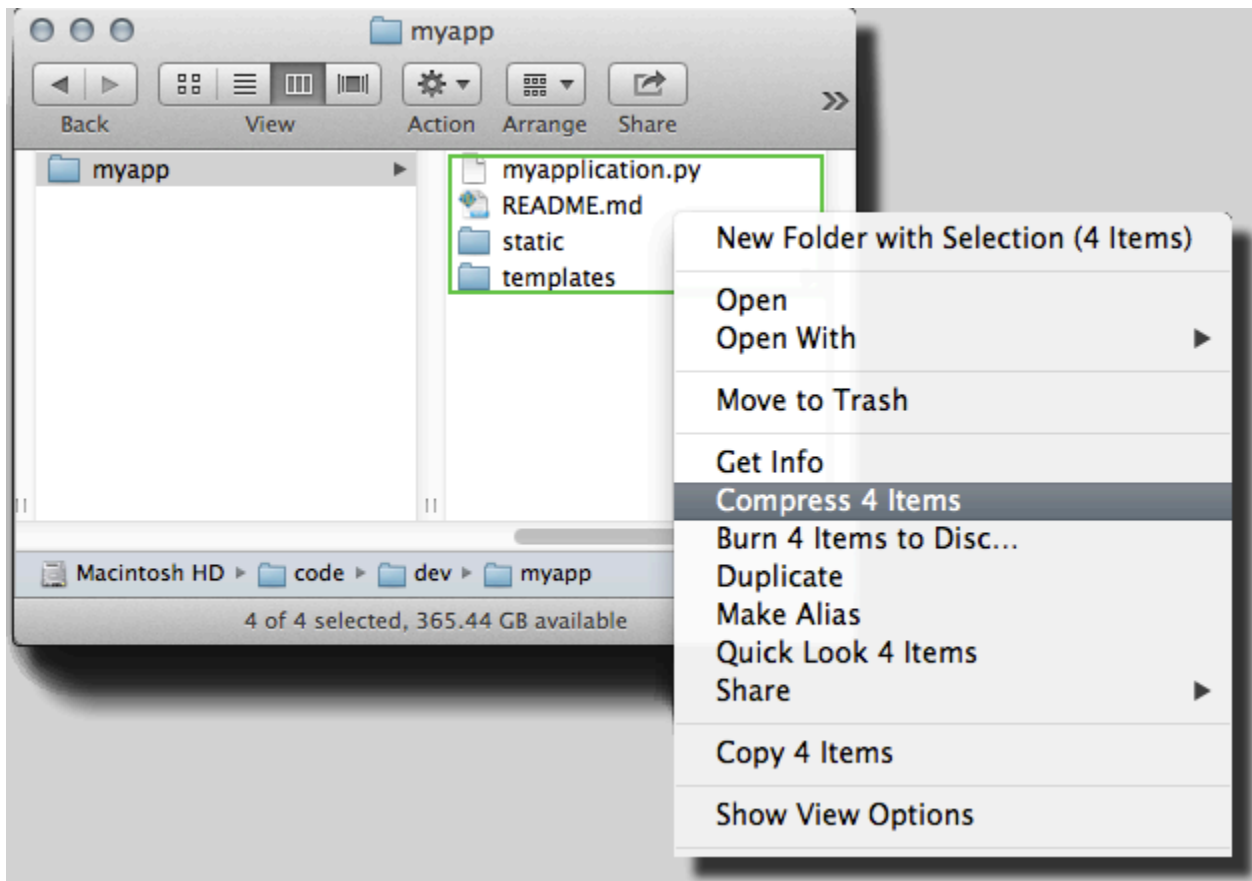
La estructura de este archivo de ejemplo se va a utilizar en todo el tema para ilustrar la compresión de archivos.

Para comprimir archivos en Mac OS X Finder

1. Abra la carpeta de proyecto de nivel superior y seleccione todos los archivos y subcarpetas que contiene. No seleccione la carpeta de nivel superior.

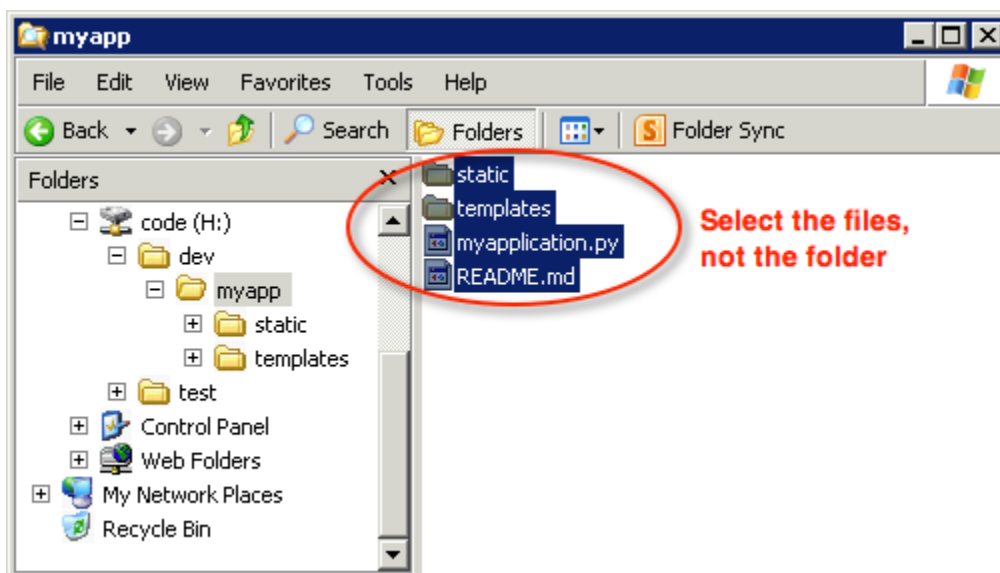


2. Haga clic con el botón derecho en los archivos seleccionados y seleccione Compress (Comprimir) X items (elementos), donde X es el número de archivos y subcarpetas que se han seleccionado.

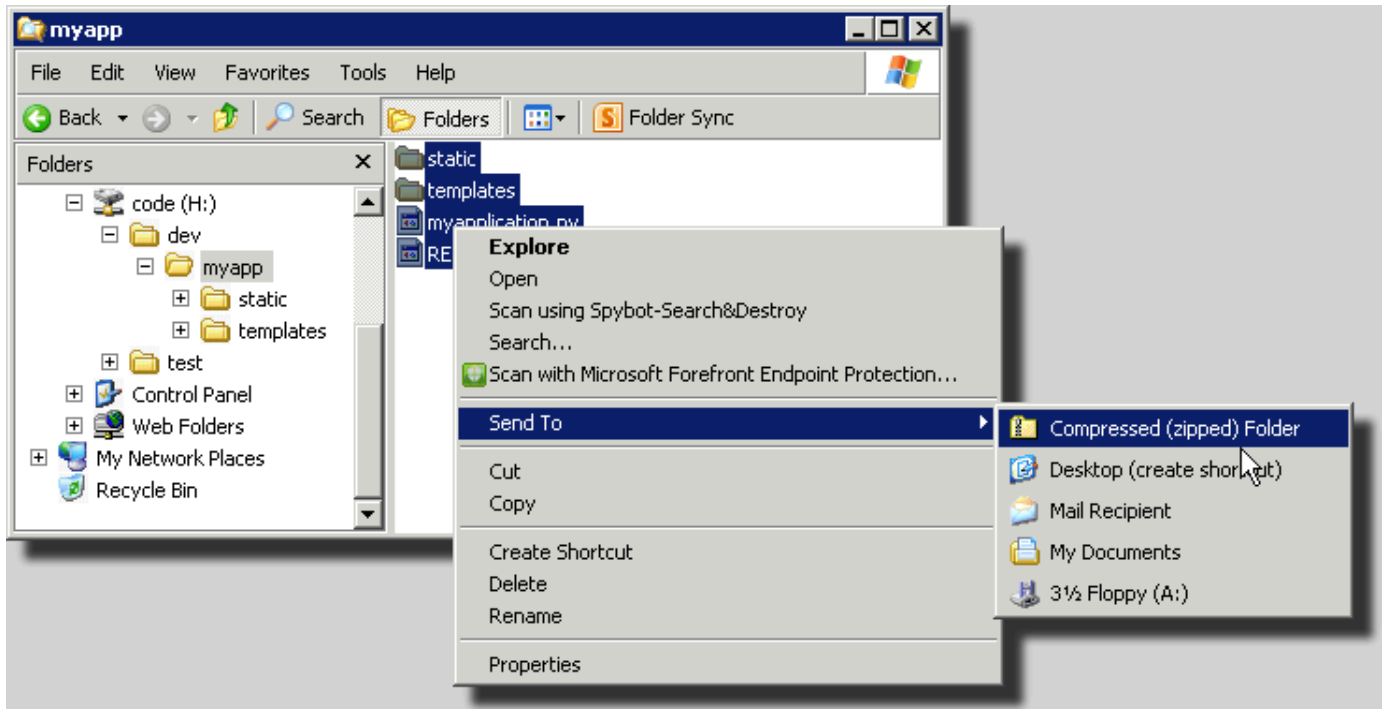


Para comprimir archivos en el Explorador de Windows

1. Abra la carpeta de proyecto de nivel superior y seleccione todos los archivos y subcarpetas que contiene. No seleccione la carpeta de nivel superior.



- Haga clic con el botón derecho en los archivos seleccionados, seleccione Enviar a y haga clic en Carpeta comprimida (en zip).



Creación de un paquete de código fuente para una aplicación .NET

Si utiliza Visual Studio, puede utilizar la herramienta de implementación incluida en AWS Toolkit for Visual Studio para implementar su aplicación .NET en Elastic Beanstalk. Para obtener más información, consulte [Implementación de aplicaciones de Elastic Beanstalk en .NET con la herramienta de implementación](#).

Si necesita crear manualmente un paquete de código fuente para la aplicación .NET, no basta con crear un archivo ZIP que contenga el directorio del proyecto. Debe crear un paquete de implementación web para el proyecto que sea adecuado para implementarlo en Elastic Beanstalk. Existen varios métodos que puede utilizar para crear un paquete de implementación:

- Puede crear el paquete de implementación utilizando el asistente de Publicación web de Visual Studio. Para obtener más información, consulte [Cómo: Crear un paquete de implementación web en Visual Studio](#).

⚠ Important

Al crear el paquete de implementación web, debe iniciar el nombre del sitio con Default Web Site.

- Se tiene un proyecto .NET, puede crear el paquete de implementación utilizando el comando msbuild, tal y como se muestra en el siguiente ejemplo.

⚠ Important

El parámetro DeployIisAppPath debe empezar por Default Web Site.

```
C:/> msbuild <web_app>.csproj /t:Package /p:DeployIisAppPath="Default Web Site"
```

- Si tiene un proyecto de sitio web, puede utilizar la herramienta IIS Web Deploy para crear el paquete de implementación. Para obtener más información, consulte [Packaging and Restoring a Web site](#).

⚠ Important

El parámetro apphostconfig debe empezar por Default Web Site.

Si va a implementar varias aplicaciones o una aplicación de ASP.NET Core, incluya la carpeta .ebextensions en la raíz del paquete de código fuente, junto con los paquetes y el archivo de manifiesto de la aplicación:

```
~/workspace/source-bundle/  
|-- .ebextensions  
|   |-- environmentvariables.config  
|   |-- healthcheckurl.config  
|-- AspNetCore101HelloWorld.zip  
|-- AspNetCoreHelloWorld.zip  
|-- aws-windows-deployment-manifest.json  
|-- VS2015AspNetWebApiApp.zip
```

Comprobación del paquete de código fuente

Tal vez desee probar localmente el paquete de código fuente antes de cargarlo en Elastic Beanstalk. Como Elastic Beanstalk utiliza la línea de comando básicamente para extraer los archivos, es mejor que haga las pruebas desde la línea de comando que con una herramienta de GUI.

Para probar la extracción de archivos en Mac OS X o Linux

1. Abra una ventana del terminal (Mac OS X) o conéctese al servidor de Linux. Vaya al directorio que contiene el paquete de código fuente.
2. Utilice el comando `unzip` o `tar xf` para descomprimir el archivo.
3. Asegúrese de que los archivos descomprimidos están en la misma carpeta que el propio archivo y no en una nueva carpeta o directorio de nivel superior.

Note

Si utiliza Mac OS X Finder para descomprimir el archivo, se creará una nueva carpeta de nivel superior sin importar cómo esté estructurado el propio archivo. Para obtener mejores resultados, utilice la línea de comandos.

Para probar la extracción de archivos en Windows

1. Descargue o instale un programa que permita extraer archivos comprimidos a través de la línea de comandos. Por ejemplo, puede descargar el programa gratuito `unzip.exe` de <http://stahlforce.com/dev/index.php?tool=zipunzip>.
2. Si es necesario, copie el archivo ejecutable en el directorio que contiene el paquete de código fuente. Si ha instalado una herramienta para todo el sistema, puede omitir este paso.
3. Descomprima el archivo utilizando el comando apropiado. Si ha descargado `unzip.exe` con el enlace del paso 1, el comando será `unzip <archive-name>`.
4. Asegúrese de que los archivos descomprimidos están en la misma carpeta que el propio archivo y no en una nueva carpeta o directorio de nivel superior.

Etiquetar recursos de la aplicación Elastic Beanstalk

Puede aplicar etiquetas a los recursos de sus aplicaciones de AWS Elastic Beanstalk. Las etiquetas son pares de clave-valor asociadas a los recursos de AWS. Las etiquetas pueden ayudarle a

clasificar los recursos. Son especialmente útiles si administra muchos recursos como parte de varias aplicaciones de AWS.


Estas son algunas formas de usar el etiquetado con recursos de Elastic Beanstalk:

- **Etapas de implementación:** identifique los recursos asociados con diferentes etapas de su aplicación, como la de desarrollo, beta y producción.
- **Asignación de costo:** utilice los informes de asignación de costos para supervisar el uso de los recursos de AWS asociados a varias cuentas de gastos. Los informes incluyen recursos etiquetados y sin etiquetar, y agrupan los costos en función de las etiquetas. Para obtener información sobre el uso de etiquetas en los informes de asignación de costos, consulte [Uso de etiquetas de asignación de costos para informes de facturación personalizados](#) en la Guía del usuario de AWS Billing and Cost Management.
- **Control de acceso:** utilice las etiquetas para administrar los permisos a las solicitudes y los recursos. Por ejemplo, un usuario que solo puede crear y administrar entornos beta solo debe tener acceso a los recursos de la etapa beta. Para obtener más información, consulte [Uso de etiquetas para controlar el acceso a los recursos de Elastic Beanstalk](#).

Puede agregar hasta 50 etiquetas a cada recurso de . Los entornos son ligeramente diferentes: Elastic Beanstalk añade tres etiquetas del sistema predeterminadas a los entornos, y no puede editar o eliminar esas etiquetas. Además de las etiquetas predeterminadas, puede agregar hasta 47 etiquetas adicionales a cada entorno.

Se aplican las siguientes limitaciones en las claves y valores de las etiquetas:

- Las claves y los valores pueden contener letras, números, espacios en blanco y los siguientes símbolos: `_ . : / = + - @`
- Las claves pueden contener un máximo de 127 caracteres. Los valores pueden contener un máximo de 255 caracteres.

 Note

Estos límites de longitud son para los caracteres Unicode en UTF-8. Para otras codificaciones multibyte, los límites podrían ser inferiores.

- Las claves distinguen entre mayúsculas y minúsculas.
- Las claves no pueden empezar por `aws :` o `elasticbeanstalk:.`

Propagación de etiquetas para plantillas de lanzamiento

Elastic Beanstalk ofrece una opción para permitir la propagación de etiquetas de entorno en plantillas de lanzamiento. Esta opción proporciona compatibilidad continua para el control de acceso basado en etiquetas (TBAC) con las plantillas de lanzamiento.

Note

Las configuraciones de lanzamiento se están eliminando gradualmente y se están reemplazando por plantillas de lanzamiento. Para obtener más información, consulte [Configuraciones de lanzamiento](#) en la Guía del usuario de Amazon EC2 Auto Scaling.

Para evitar el tiempo de inactividad de instancias de EC2 en ejecución, AWS CloudFormation no propaga las etiquetas a plantillas de lanzamiento existentes. Si existe un caso de uso que requiera etiquetas para los recursos de su entorno, puede habilitar Elastic Beanstalk para que cree plantillas de lanzamiento con etiquetas para estos recursos. Para ello, establezca la opción `LaunchTemplateTagPropagationEnabled` del espacio de nombres de [aws:autoscaling:launchconfiguration](#) como `true`. El valor predeterminado es `false`.

El siguiente ejemplo de [archivo de configuración](#) permite la propagación de etiquetas en plantillas de lanzamiento.

```
option_settings:
  aws:autoscaling:launchconfiguration:
    LaunchTemplateTagPropagationEnabled: true
```

Elastic Beanstalk solo puede propagar etiquetas en plantillas de lanzamiento para los siguientes recursos:

- Volúmenes de EBS
- Instancias EC2
- Interfaces de red de EC2
- Plantillas de lanzamiento de AWS CloudFormation que definan un recurso

Esta restricción se debe a que CloudFormation solo permite etiquetas en la creación de plantillas para recursos específicos. Para obtener más información, consulte [TagSpecification](#) en la Guía del usuario de AWS CloudFormation.

Important

- Cambiar el valor de esta opción de `false` a `true` para un entorno existente puede suponer un cambio sustancial para las etiquetas que ya existían anteriormente.
- Cuando se habilita esta característica, la propagación de las etiquetas requerirá la sustitución de EC2, lo que puede provocar un tiempo de inactividad. Puede habilitar actualizaciones escalonadas para aplicar los cambios de configuración por lotes y evitar el tiempo de inactividad durante el proceso de actualización. Para obtener más información, consulte [Cambios de configuración](#).

Para obtener más información sobre las plantillas de lanzamiento, consulte lo siguiente:

- [Launch templates](#) en la Guía del usuario de Amazon EC2 Auto Scaling
- [Trabajo con plantillas](#) en la Guía del usuario de AWS CloudFormation
- [Fragmentos de código de plantillas de Elastic Beanstalk](#) en la Guía del usuario de AWS CloudFormation

Recursos que se pueden etiquetar

A continuación, se muestran los tipos de recursos de Elastic Beanstalk que puede etiquetar y los enlaces a temas específicos acerca de la administración de etiquetas para cada uno de ellos:

- [Aplicaciones](#)
- [Entornos](#)
- [Versiones de la aplicación](#)
- [Configuraciones guardadas](#)
- [Versiones personalizadas de la plataforma](#)

Etiquetado de aplicaciones

Puede aplicar etiquetas a sus aplicaciones de AWS Elastic Beanstalk. Las etiquetas son pares de clave-valor asociadas a los recursos de AWS. Para obtener información sobre el etiquetado de recursos de Elastic Beanstalk, los casos de uso, las restricciones de las claves y los valores de

las etiquetas y los tipos de recursos admitidos, consulte [Etiquetar recursos de la aplicación Elastic Beanstalk](#).

Puede especificar etiquetas al crear una aplicación. En una aplicación existente, puede añadir o eliminar etiquetas, y actualizar los valores de etiquetas existentes. Puede añadir hasta 50 etiquetas a cada aplicación.

Adición de etiquetas durante la creación de aplicaciones

Cuando se utiliza la consola de Elastic Beanstalk para [crear una aplicación](#), puede especificar las claves y los valores de las etiquetas en el cuadro de diálogo Create New Application (Crear nueva aplicación).

Elastic Beanstalk

Create new application

Application information

Application Name

Maximum length of 100 characters, not including forward slash (/).

Description

Tags

Apply up to 50 tags. You can use tags to group and filter your resources. A tag is a key-value pair. The key must be unique within the resource and is case-sensitive. [Learn more](#)

Key	Value	
<input type="text"/>	<input type="text"/>	<input type="button" value="Remove tag"/>

50 remaining

Si utiliza la CLI de EB para crear una aplicación, use la opción `--tags` con [eb init](#) para añadir etiquetas.

```
~/workspace/my-app$ eb init --tags mytag1=value1,mytag2=value2
```

Con la AWS CLI u otros clientes basados en API, agregue etiquetas con el parámetro `--tags` en el comando [create-application](#).

```
$ aws elasticbeanstalk create-application \  
  --tags Key=mytag1,Value=value1 Key=mytag2,Value=value2 \  
  --application-name my-app --version-label v1
```

Administración de etiquetas de una aplicación existente

Puede agregar, actualizar y eliminar etiquetas en una aplicación de Elastic Beanstalk existente.

Para administrar las etiquetas de una aplicación en la consola de Elastic Beanstalk

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Applications (Aplicaciones) y, a continuación, elija el nombre de la aplicación de la lista.

Note

Si tiene muchas aplicaciones, utilice la barra de búsqueda para filtrar la lista de aplicaciones.

3. Elija Actions (Acciones) y, a continuación, elija Manage Tags (Administrar etiquetas).
4. Utilice el formulario en pantalla para añadir, actualizar o eliminar etiquetas.
5. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

Si utiliza la CLI de EB para actualizar su aplicación, utilice [eb tags](#) para añadir, actualizar, eliminar o enumerar etiquetas.

Por ejemplo, el siguiente comando muestra las etiquetas en una aplicación.

```
~/workspace/my-app$ eb tags --list --resource "arn:aws:elasticbeanstalk:us-east-2:my-account-id:application/my-app"
```

El siguiente comando actualiza la etiqueta mytag1 y elimina la etiqueta mytag2.

```
~/workspace/my-app$ eb tags --update mytag1=newvalue --delete mytag2 \
  --resource "arn:aws:elasticbeanstalk:us-east-2:my-account-id:application/my-app"
```

Para obtener una lista de las opciones y más ejemplos, consulte [eb tags](#).

Con la AWS CLI u otros clientes basados en API, utilice el comando [list-tags-for-resource](#) para enumerar las etiquetas de una aplicación.

```
$ aws elasticbeanstalk list-tags-for-resource --resource-arn
  "arn:aws:elasticbeanstalk:us-east-2:my-account-id:application/my-app"
```

Utilice el comando [update-tags-for-resource](#) para añadir, actualizar o eliminar etiquetas en una aplicación.

```
$ aws elasticbeanstalk update-tags-for-resource \
  --tags-to-add Key=mytag1,Value=newvalue --tags-to-remove mytag2 \
  --resource-arn "arn:aws:elasticbeanstalk:us-east-2:my-account-id:application/my-app"
```

Especifique las etiquetas que desea añadir y las que desea actualizar en el parámetro `--tags-to-add` de `update-tags-for-resource`. Se añade una etiqueta no existente y se actualiza el valor de una etiqueta existente.

Note

Para utilizar algunos de los comandos de la CLI de EB y los comandos de la AWS CLI con una aplicación Elastic Beanstalk, necesita el ARN de la aplicación. Puede recuperar el ARN mediante el siguiente comando.

```
$ aws elasticbeanstalk describe-applications --application-names my-app
```

Administración de entornos

AWS Elastic Beanstalk permite crear fácilmente nuevos entornos para su aplicación. Puede crear y administrar entornos distintos para desarrollo, pruebas y uso en producción, y puede [implementar cualquier versión](#) de la aplicación en cualquier entorno. Los entornos pueden mantenerse durante mucho tiempo o temporalmente. Cuando termine de usar un entorno, puede guardar su configuración para volver a crearlo más adelante.

Cuando desarrolle su aplicación, posiblemente la implementará en varios entornos diferentes para fines distintos. Elastic Beanstalk le permite [configurar cómo se realizan las implementaciones](#). Puede implementarla en todas las instancias de su entorno simultáneamente o dividir una implementación en lotes con implementaciones continuas.

Los [cambios de configuración](#) se procesan por separado de las implementaciones y tienen su propio alcance. Por ejemplo, si cambia el tipo de las instancias EC2 que ejecutan su aplicación, deben reemplazarse todas las instancias. Por otra parte, si modifica la configuración del balanceador de carga del entorno, ese cambio se puede realizar in situ sin que se interrumpa el servicio ni se reduzca la capacidad. También puede aplicar cambios de configuración que modifiquen las instancias de su entorno en lotes con [actualizaciones de configuración continuas](#).

Note

Modifique los recursos de su entorno solo mediante Elastic Beanstalk. Si modifica los recursos mediante la consola de otro servicio, los comandos de la CLI o los SDK, Elastic Beanstalk no podrá controlar de forma precisa el estado de los recursos y usted no podrá guardar la configuración ni volver a crear de forma fiable el entorno. Los cambios fuera de banda también causan problemas cuando se actualiza o se termina un entorno.

Cuando lanza un entorno, elige una versión de plataforma. Actualizamos las plataformas de forma periódica con nuevas versiones para ofrecer mejoras de rendimiento y nuevas características. Puede [actualizar su entorno a la última versión de la plataforma](#) en cualquier momento.

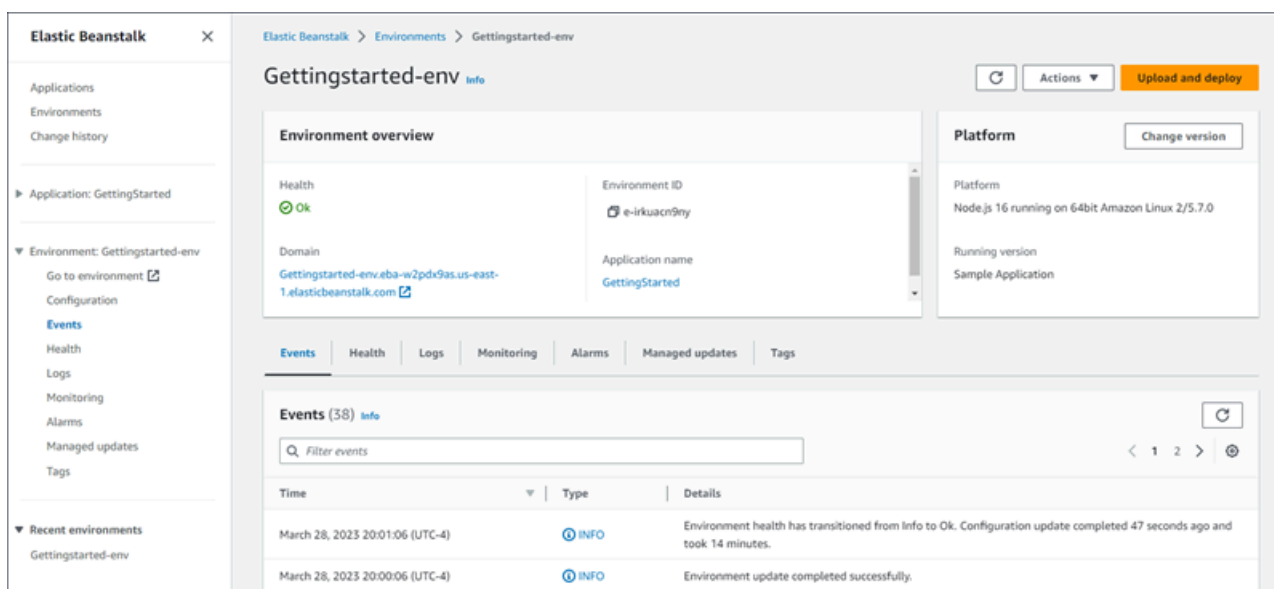
A medida que su aplicación se vaya haciendo más compleja, puede dividirla en varios componentes que se ejecuten en un entorno distinto. Para las cargas de trabajo de larga ejecución, puede lanzar [entornos de trabajo](#) que procesen los trabajos de una cola de Amazon Simple Queue Service (Amazon SQS).

Temas

- [Uso de la consola de administración del entorno de Elastic Beanstalk](#)
- [Creación de un entorno de Elastic Beanstalk](#)
- [Implementación de aplicaciones en entornos Elastic Beanstalk](#)
- [Cambios de configuración](#)
- [Actualización de la versión de la plataforma del entorno de Elastic Beanstalk](#)
- [Cancelación de actualizaciones de configuración del entorno e implementaciones de aplicaciones](#)
- [Volver a crear entornos de Elastic Beanstalk](#)
- [Tipos de entornos](#)
- [Entornos de trabajo de Elastic Beanstalk](#)
- [Creación de enlaces entre entornos de Elastic Beanstalk](#)

Uso de la consola de administración del entorno de Elastic Beanstalk

La consola de Elastic Beanstalk proporciona la página Environment overview (Información general del entorno) para que pueda administrar cada uno de los entornos de AWS Elastic Beanstalk. Desde la página Environment overview (Información general del entorno), puede administrar la configuración del entorno y realizar acciones comunes. Estas acciones incluyen reiniciar los servidores web que se ejecutan en el entorno, clonar el entorno o volver a crear el entorno desde cero.



The screenshot displays the AWS Elastic Beanstalk console interface. On the left, a navigation sidebar shows 'Applications', 'Environments', and 'Change history'. The main content area is titled 'Gettingstarted-env' and includes an 'Info' tab. The 'Environment overview' section shows 'Health' as 'Ok' (green checkmark), 'Environment ID' as 'e-irkuaacn9ny', 'Domain' as 'Gettingstarted-env.eba-w2pdx9as.us-east-1.elasticbeanstalk.com', and 'Application name' as 'GettingStarted'. The 'Platform' section shows 'Platform' as 'Node.js 16 running on 64bit Amazon Linux 2/5.7.0' and 'Running version' as 'Sample Application'. Below this, there are tabs for 'Events', 'Health', 'Logs', 'Monitoring', 'Alarms', 'Managed updates', and 'Tags'. The 'Events' tab is active, showing a list of events with columns for 'Time', 'Type', and 'Details'. Two events are visible, both of type 'INFO', indicating successful health transitions and updates.

Time	Type	Details
March 28, 2023 20:01:06 (UTC-4)	INFO	Environment health has transitioned from Info to Ok. Configuration update completed 47 seconds ago and took 14 minutes.
March 28, 2023 20:00:06 (UTC-4)	INFO	Environment update completed successfully.

Para acceder a la consola de administración del entorno

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

Verá la página Environment overview (Información general del entorno). El panel de navegación de la consola muestra el nombre de la aplicación a la que pertenece el entorno, con las páginas de administración de la aplicación relacionadas, y el nombre del entorno, con las páginas de administración del entorno.

Temas

- [Información general del entorno](#)
- [Acciones del entorno](#)
- [Eventos](#)
- [Estado](#)
- [Registros](#)
- [Supervisión](#)
- [Alarmas](#)
- [Managed Updates \(Actualizaciones administradas\)](#)
- [Etiquetas](#)
- [Configuración](#)

Información general del entorno

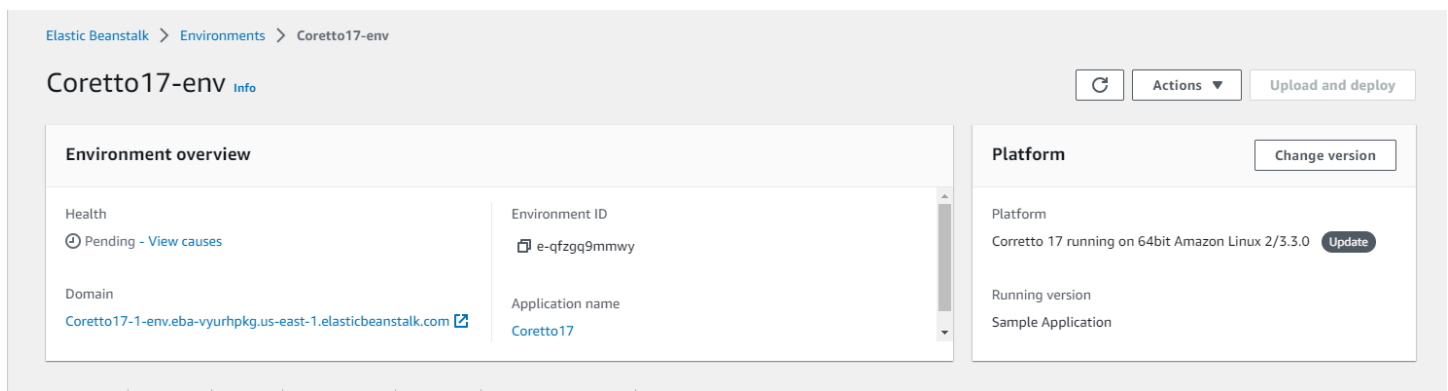
Para ver la página Environment overview (Información general del entorno), elija el nombre del entorno en el panel de navegación, si es el entorno actual. Como alternativa, puede dirigirse al entorno desde la página Applications (Aplicaciones) o desde la lista principal de entornos en la página Environments (Entornos).

El panel superior de la página de información general del entorno muestra información de nivel superior sobre el entorno. Esto incluye su nombre, dirección URL y estado actual junto con el nombre de la versión de la aplicación implementada actualmente y la versión de la plataforma en la que se ejecuta la aplicación. Debajo del panel de información general, podrá ver los eventos más recientes del entorno.

Elija Refresh (Actualizar) para actualizar la información que se muestra. La página de información general contiene la siguiente información y las siguientes opciones.

Estado

El estado general del entorno. Si el estado del entorno se deteriora, aparece el enlace Ver causas junto al estado del entorno. Seleccione este enlace para ver la pestaña Health con más detalles.



Dominio

El Domain (Dominio) del entorno, o la URL, se encuentra en la parte superior de la página Environment overview (Información general del entorno), debajo de Health (Estado) del entorno. Se trata de la dirección URL de la aplicación web que el entorno ejecuta.

ID del entorno

ID del entorno. Se trata de un ID interno que se genera cuando se crea el entorno.

Nombre de la aplicación

El nombre de la aplicación que se implementa y se ejecuta en el entorno.

Running version (Versión en ejecución)

El nombre de la versión de la aplicación que se implementa y se ejecuta en el entorno. Seleccione Upload and deploy (Cargar e implementar) para cargar un [paquete de código fuente](#) e implementarlo en el entorno. Esta opción crea una nueva versión de la aplicación.

Plataforma

El nombre de la versión de la plataforma que se ejecuta en el entorno. Normalmente este incluye la arquitectura, el sistema operativo (SO), el idioma y el servidor de aplicaciones (conjuntamente, la ramificación de la plataforma), con un número específico de versión de la plataforma.

Si la versión de su plataforma no es la más reciente que hay disponible, aparecerá una etiqueta de estado junto a ella en la sección Plataforma. La etiqueta Actualización indica que, aunque la versión de la plataforma es compatible, hay una versión más reciente disponible. La versión de la plataforma también puede etiquetarse como obsoleta o retirada. Seleccione Cambiar versión para actualizar la ramificación de la plataforma a una versión más reciente. Para obtener más información sobre los estados de una versión de la plataforma, consulte la sección Ramificación de la plataforma en [Glosario de plataformas Elastic Beanstalk](#).

The screenshot displays the AWS Elastic Beanstalk console for an environment named 'Coretto17-env'. At the top, there are navigation links for 'Elastic Beanstalk', 'Environments', and 'Coretto17-env'. Below the environment name, there are buttons for 'Actions' and 'Upload and deploy'. The main content is divided into two panels. The left panel, titled 'Environment overview', shows the 'Health' status as 'Ok' with a green checkmark, and the 'Domain' as 'Coretto17-1-env.eba-vyurhpkg.us-east-1.elasticbeanstalk.com'. The right panel, titled 'Platform', shows the current platform as 'Corretto 17 running on 64bit Amazon Linux 2/3.3.0' with an 'Update' button next to it. Below this, it shows the 'Running version' as 'Sample Application'.

Pestañas de información general del entorno

Las pestañas que aparecen en la parte inferior de la página incluyen información más detallada sobre el entorno y proporcionan acceso a características adicionales:

- **Events (Eventos):** muestra información o mensajes de error del servicio de Elastic Beanstalk y de otros servicios cuyos recursos usa este entorno.
- **Health (Estado):** muestra el estado y otros detalles de las instancias de Amazon EC2 que ejecutan la aplicación.
- **Logs (Registros):** recupere y descargue registros de Amazon EC2 en su entorno. Puede recuperar registros completos o actividades recientes. Los registros recuperados están disponibles durante 15 minutos.
- **Monitoring (Monitorización):** muestra las estadísticas del entorno, como la latencia media y el uso de CPU.

- **Alarms (Alarmas):** muestra las alarmas que se configuraron para las métricas del entorno. En esta página, puede agregar, modificar o eliminar alarmas.
- **Managed updates (Actualizaciones administradas):** muestra información sobre las actualizaciones administradas de la plataforma, próximas y completadas, y la sustitución de instancias.
- **Tags (Etiquetas):** muestra etiquetas de entorno y le permite administrarlas. Las etiquetas son pares clave-valor que se aplican a su entorno.

Note

El panel de navegación del lado izquierdo de la consola contiene enlaces con el mismo nombre que las pestañas. Al seleccionar cualquiera de estos enlaces, se mostrará el contenido de la pestaña correspondiente.

Acciones del entorno

La página de información general del entorno contiene un menú **Actions (Acciones)** que puede utilizar para realizar operaciones comunes en el entorno. Este menú se muestra en la parte derecha del encabezado del entorno junto a la opción **Create a new environment (Crear un entorno nuevo)**.

Note

Algunas acciones solo están disponibles bajo ciertas condiciones, quedando deshabilitadas hasta que se cumplan las condiciones adecuadas.

Load configuration (Configuración de carga)

Cargue una configuración guardada con anterioridad. Las configuraciones se guardan en su aplicación y cualquier entorno asociado puede cargarlas. Si ha realizado cambios en la configuración de su entorno, puede cargar una configuración guardada para deshacer los cambios. También puede cargar una configuración que haya guardado desde otro entorno en el que se ejecuta la misma aplicación para propagar los cambios de configuración entre ellos.

Save configuration (Guardar configuración)

Guarde la configuración actual del entorno en su aplicación. Antes de realizar cambios en la configuración del entorno, guarde la configuración actual para que pueda restaurarla más adelante si es necesario. También puede aplicar una configuración guardada cuando lanza un nuevo entorno.

Swap environment Domains (URLs) (Intercambiar los Dominios del entorno [URL])

Intercambie el CNAME del entorno actual con el de un nuevo entorno. Después de un intercambio de CNAME, todo el tráfico dirigido a la aplicación que usa la URL del entorno va a parar al nuevo entorno. Cuando esté listo para implementar una nueva versión de la aplicación, puede lanzar un entorno distinto en la nueva versión. Cuando el nuevo entorno esté listo para empezar a recibir solicitudes, realice un intercambio de CNAME para empezar a dirigir el tráfico al nuevo entorno. Hacer esto no interrumpe los servicios. Para obtener más información, consulte [Implementaciones Blue/Green \(azul/verde\) con Elastic Beanstalk](#).

Clone environment (Clonar entorno)

Lance un nuevo entorno con la misma configuración que el entorno que se está ejecutando actualmente.

Clone with latest platform (Clonar con la plataforma más reciente)

Clone su entorno actual con la versión más reciente de la plataforma Elastic Beanstalk en uso. Esta opción solo está disponible cuando existe una versión más reciente de la plataforma del entorno actual.

Abort current operation (Interrumpir operación actual)

Detiene una actualización del entorno en curso. Detener una operación puede hacer que algunas de las instancias del entorno tengan un estado diferente de las otras, en función de lo avanzada que esté la operación. Esta opción solo está disponible cuando se está actualizando el entorno.

Restart app servers (Reiniciar servidores de aplicaciones)

Reinicie el servidor web que se ejecuta en las instancias del entorno. Esta opción no termina ni reinicia ningún recurso de AWS. Si el entorno presenta un comportamiento extraño en respuesta a algunas solicitudes erróneas, la funcionalidad se puede restaurar temporalmente reiniciando el servidor de la aplicación mientras se soluciona la causa raíz.

Rebuild environment (Volver a crear un entorno)

Termina todos los recursos del entorno de ejecución y crea un nuevo entorno con la misma configuración. Esta operación tarda varios minutos, similar a la cantidad de tiempo necesario para implementar un nuevo entorno desde cero. Todas las instancias de Amazon RDS que se ejecutan en la capa de datos del entorno se eliminan durante la reconstrucción de un entorno. Si necesita los datos, cree una instantánea. Puede crear una instantánea manualmente [en la consola de RDS](#) o configurar la política de eliminación de la capa de datos para crear automáticamente una instantánea antes de eliminar la instancia. Esta es la configuración predeterminada cuando se crea una capa de datos.

Terminate Environment (Finalizar entorno)

Termina todos los recursos del entorno de ejecución y elimina el entorno de la aplicación. Si tiene una instancia de RDS que se ejecuta en una capa de datos y necesita retener sus datos, asegúrese de que la política de eliminación de base de datos esté configurada en cualquiera de los dos Snapshot o Retain. Para obtener más información, consulte [Ciclo de vida de la base de datos](#) en el capítulo Configuración de entornos de esta guía.

Restore Environment (Restaurar entorno)

Si el entorno se ha terminado en la última hora, puede restablecerlo desde esta página. Al cabo de una hora, puede [restaurarlo desde la página de información general de la aplicación](#).

Eventos

La pestaña Events (Eventos) muestra la secuencia de eventos del entorno. Elastic Beanstalk muestra mensajes de eventos siempre que interactúa con el entorno y cuando se crea o modifica algún recurso del entorno.

The screenshot shows the AWS Elastic Beanstalk Events console. At the top, there are navigation tabs: Events (selected), Health, Logs, Monitoring, Alarms, Managed updates, and Tags. Below the tabs, the 'Events (68)' section is visible, with a search bar labeled 'Filter events' and a refresh button. The main content is a table of events with columns for Time, Type, and Details.

Time	Type	Details
December 22, 2022 16:30:28 (UTC-5)	INFO	Completed swapping CNAMEs for environments 'Coretto17-env' and 'Coretto17-1-env'.
December 22, 2022 16:30:28 (UTC-5)	INFO	'Coretto17-1-env.eba-vyurhpkg.us-east-1.elasticbeanstalk.com' now points to 'awseb-AWSEB-189U97E0IFODU-547333735.us-east-1.elb.amazonaws.com'.
December 22, 2022 16:30:25 (UTC-5)	INFO	Swapping CNAMEs for environments 'Coretto17-env' and 'Coretto17-1-env'.
December 22, 2022 16:30:25 (UTC-5)	INFO	swapEnvironmentCNAMEs is starting.
December 21, 2022 22:51:51 (UTC-5)	INFO	Environment health has transitioned from Pending to Ok.
December 21, 2022 22:50:51 (UTC-5)	INFO	Environment health has transitioned from Ok to Pending.
December 21, 2022 22:48:51 (UTC-5)	INFO	Environment health has transitioned from Pending to Ok. Initialization completed 53 seconds ago and took 3 minutes.
December 21, 2022 22:48:09 (UTC-5)	INFO	Successfully launched environment: Coretto17-env
December 21, 2022 22:48:09 (UTC-5)	INFO	Application available at Coretto17-env.eba-vyurhpkg.us-east-1.elasticbeanstalk.com.
December 21, 2022 22:47:52 (UTC-5)	INFO	Instance deployment completed successfully.

Para obtener más información, consulte [Visualización de la transmisión de eventos de un entorno de Elastic Beanstalk](#).

Estado

Cuando se habilita el monitoreo de estado mejorado, la página muestra información en tiempo real del estado de las instancias. El panel Overall health (Estado general) muestra los datos del estado como un promedio de todas las instancias del entorno combinadas. El panel Enhanced instance health (Estado mejorado de las instancias) muestra información en tiempo real del estado de cada instancia individual del entorno. La monitorización de estado mejorada permite a Elastic Beanstalk monitorizar de cerca los recursos del entorno para que pueda evaluar el estado de la aplicación con mayor precisión.

Cuando se habilita la monitorización de estado mejorada, esta página muestra información sobre las solicitudes atendidas por las instancias del entorno y las métricas del sistema operativo, incluida la latencia, la carga y el uso de la CPU.

Instance ID	Status	Running time	Deployment ID	Requests/sec	2xx Responses	P99 Latency	P90 Latency	P75 Latency	P50 Latency
i-04a22cd25ba...	Ok	January 10, 20...	1	1	1	-	-	-	-
i-0b1530c3cab...	Ok	January 8, 202...	1	1	1	0.001	0.001	0.001	0.001

Para obtener más información, consulte [Informes y monitorización de estado mejorados](#).

Registros

La página Logs (Registros) le permite recuperar los registros de las instancias EC2 en su entorno. Cuando solicita registros, Elastic Beanstalk envía un comando a las instancias, que cargan los registros en su bucket de almacenamiento de Elastic Beanstalk en Amazon S3. Cuando solicita registros en esta página, Elastic Beanstalk los elimina automáticamente de Amazon S3 después de 15 minutos.

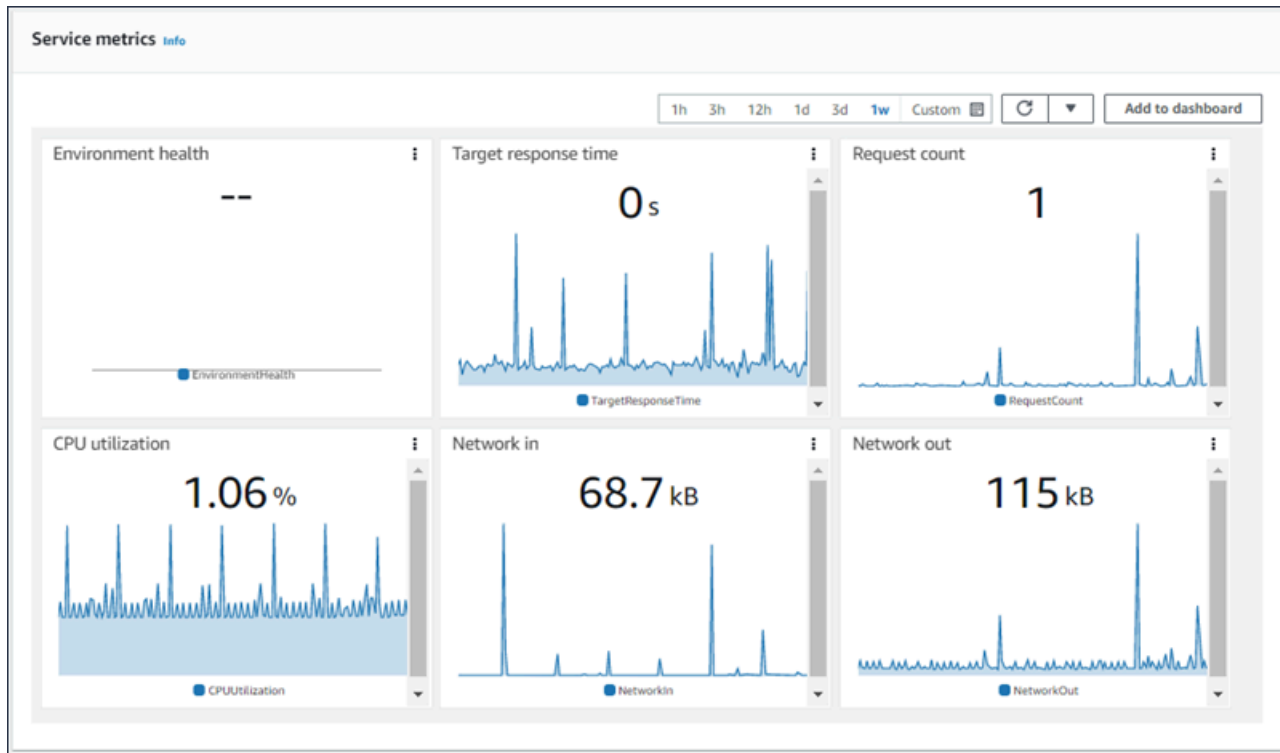
También puede configurar las instancias de su entorno para que carguen los registros en Amazon S3 para su almacenamiento permanente una vez que se hayan rotado localmente.

Log file	Time	EC2 instance	Type
Download	January 11, 2023 22:00:23 (UTC-5)	i-0b1530c3cabd58083	bundle
Download	January 11, 2023 22:00:23 (UTC-5)	i-04a22cd25ba2f7c4e	bundle
Download	January 11, 2023 22:01:04 (UTC-5)	i-04a22cd25ba2f7c4e	tail
Download	January 11, 2023 22:01:04 (UTC-5)	i-0b1530c3cabd58083	tail

Para obtener más información, consulte [Visualización de registros de instancias de Amazon EC2 en su entorno de Elastic Beanstalk](#).

Supervisión

La página Monitoring (Monitorización) muestra información general sobre el estado del entorno. Esta información incluye el conjunto predeterminado de métricas que proporcionan Elastic Load Balancing y Amazon EC2, así como gráficos que muestran cómo ha cambiado el estado del entorno a lo largo del tiempo.



Para obtener más información, consulte [Monitoreo del estado del entorno en la consola de administración de AWS](#).

Alarmas

La página Existing alarms (Alarmas existentes) muestra información sobre las alarmas que ha configurado en el entorno. Puede utilizar las opciones de la página para crear o eliminar alarmas.

The screenshot shows the 'Alarms (2)' page in the AWS console. It includes a search bar for filtering alarms, a 'Delete alarm' button, and a 'Create a new alarm' button. Below these is a table listing the configured alarms.

Alarm name	Description	Metric name	Period	Change state after	Notify when state chan...	Notify when state change
Test2	Test2	CPUUtilization	1 minute	5 minutes	-	arn:aws:sns:us-east-1:164
TestAlarm	Test	CPUUtilization	1 minute	5 minutes	-	arn:aws:sns:us-east-1:164

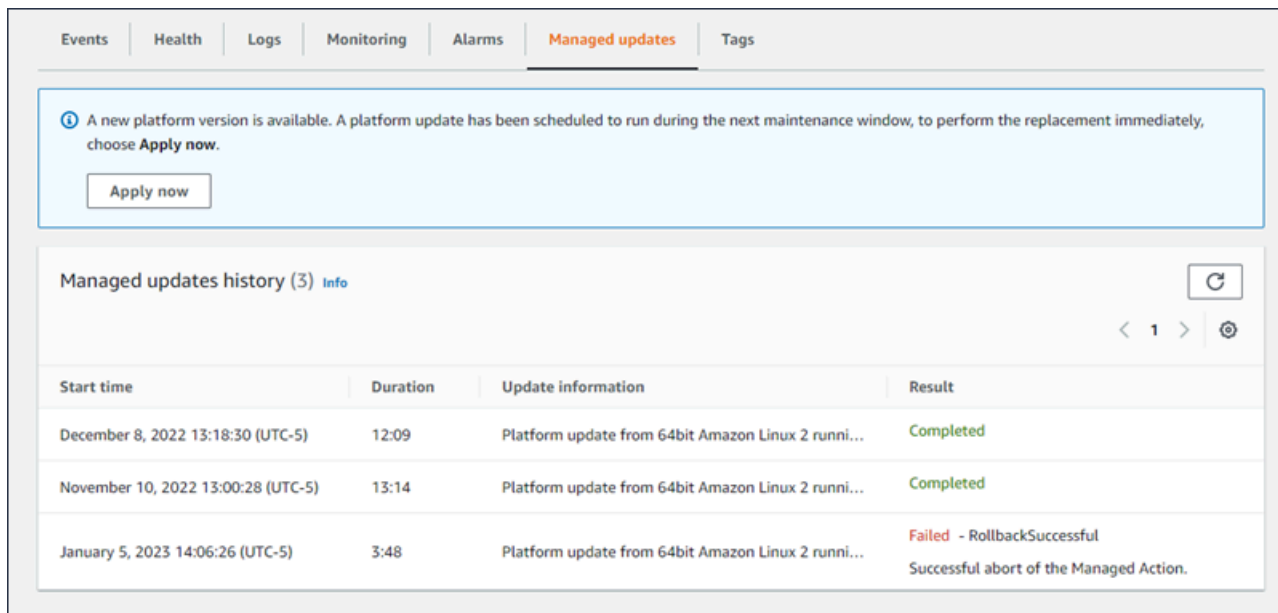
Para obtener más información, consulte [Administrar alarmas](#).

Managed Updates (Actualizaciones administradas)

La página Managed updates overview (Información general de actualizaciones administradas) muestra información sobre las actualizaciones de plataforma administradas próximas y completadas, así como sobre la sustitución de instancias.

La característica de actualización administrada permite configurar el entorno para que se actualice de manera automática a la última versión de la plataforma durante el periodo de mantenimiento semanal elegido. Entre las versiones de la plataforma, puede elegir que su entorno reemplace todas sus instancias de Amazon EC2 durante el periodo de mantenimiento. Esto puede aliviar los problemas que se producen cuando la aplicación se ejecuta durante periodos de tiempo prolongados.

Para obtener más información, consulte [Actualizaciones de plataforma administradas](#).



The screenshot displays the AWS Managed Updates overview page. At the top, there are navigation tabs: Events, Health, Logs, Monitoring, Alarms, Managed updates (selected), and Tags. A notification box at the top left states: "A new platform version is available. A platform update has been scheduled to run during the next maintenance window, to perform the replacement immediately, choose **Apply now**." Below this is an "Apply now" button. The main section is titled "Managed updates history (3) info" and contains a table with the following data:

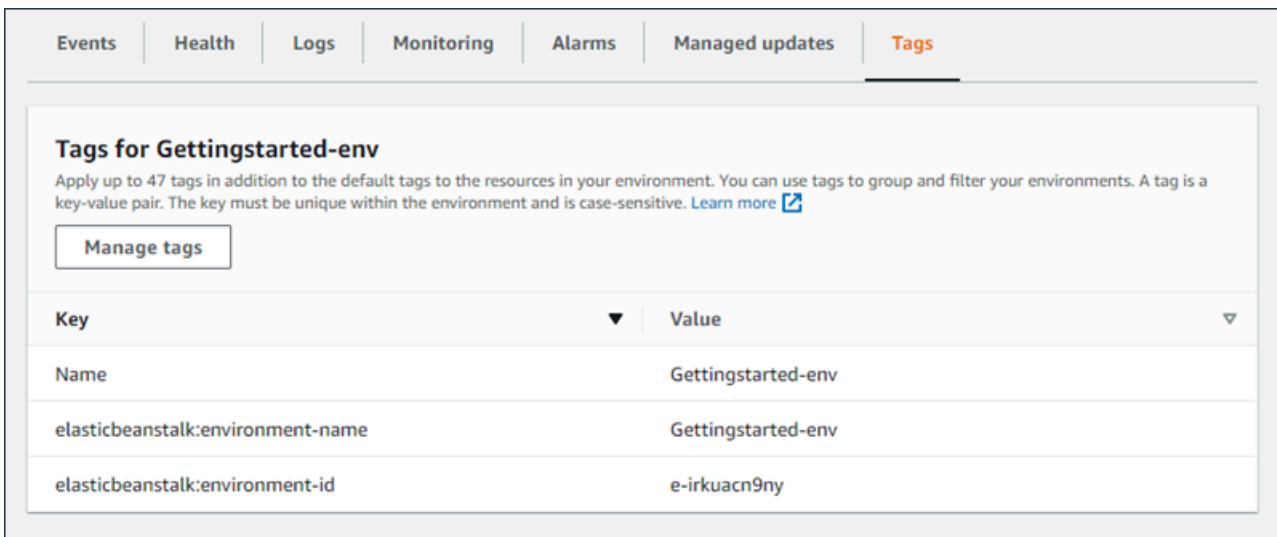
Start time	Duration	Update information	Result
December 8, 2022 13:18:30 (UTC-5)	12:09	Platform update from 64bit Amazon Linux 2 runni...	Completed
November 10, 2022 13:00:28 (UTC-5)	13:14	Platform update from 64bit Amazon Linux 2 runni...	Completed
January 5, 2023 14:06:26 (UTC-5)	3:48	Platform update from 64bit Amazon Linux 2 runni...	Failed - RollbackSuccessful Successful abort of the Managed Action.

Para obtener más información, consulte [Actualizaciones de plataforma administradas](#).

Etiquetas

La página Tags (Etiquetas) muestra las etiquetas que Elastic Beanstalk aplicó al entorno cuando lo creó y las etiquetas que ha añadido usted. Puede añadir, editar y eliminar etiquetas personalizadas. No puede editar ni eliminar las etiquetas que aplicó Elastic Beanstalk.

Las etiquetas del entorno se aplican a todos los recursos que Elastic Beanstalk crea para respaldar su aplicación.



The screenshot shows the 'Tags' page in the AWS Elastic Beanstalk console for an environment named 'Gettingstarted-env'. The page has a navigation bar with tabs for 'Events', 'Health', 'Logs', 'Monitoring', 'Alarms', 'Managed updates', and 'Tags'. Below the navigation bar, there is a section titled 'Tags for Gettingstarted-env' with a brief explanation of tags and a 'Manage tags' button. Below this is a table with two columns: 'Key' and 'Value'. The table contains three rows of tags.

Key	Value
Name	Gettingstarted-env
elasticbeanstalk:environment-name	Gettingstarted-env
elasticbeanstalk:environment-id	e-irkuacn9ny

Para obtener más información, consulte [Etiquetado de recursos en sus entornos de Elastic Beanstalk](#).

Configuración

La página Configuration (Configuración) muestra la configuración actual del entorno y los recursos, incluida la configuración de las instancias de Amazon EC2, un equilibrador de carga, las notificaciones y el monitoreo de estado. Utilice la configuración de esta página para personalizar el comportamiento de su entorno durante las implementaciones, habilitar características adicionales y modificar el tipo de instancia y otros ajustes que eligió durante la creación del entorno.

Elastic Beanstalk > Environments > Gettingstarteda-env > Configuration

Configuration Info

Service access Info

Configure the service role and EC2 instance profile that Elastic Beanstalk uses to manage your environment. Choose an EC2 key pair to securely log in to your EC2 instances.

Service role
arn:aws:iam::164656829171:role/aws-elasticbeanstalk-service-role

Instance traffic and scaling Info

Customize the capacity and scaling for your environment's instances. Select security groups to control instance traffic. Configure the software that runs on your environment's instances by setting platform-specific options.

Instances

IMDSv1
Deactivated

Capacity

Environment type	Fleet composition	On-demand base
Load balanced	On-Demand instances	0
On-demand above base	Processor type	Instance types
70	x86_64	t2.micro,t2.small

Load balancer

Load balancer type
application

Networking, database, and tags Info

Configure VPC settings, and subnets for your environment's EC2 instances and load balancer. Set up an Amazon RDS database that's integrated with your environment.

Network

Load balancer visibility	Load balancer subnets
public	—

Database

Has coupled database
false

Updates, monitoring, and logging Info

Define when and how Elastic Beanstalk deploys changes to your environment. Manage your application's monitoring and logging settings, instances, and other environment resources.

Updates

Managed updates	Update batch size	Deployment batch size
Deactivated	1	100
Deployment batch size type	Command timeout	Deployment policy
Percentage	600	AllAtOnce
Health threshold	Ignore health check	Instance replacement
Ok	false	false
Minimum capacity	Notifications email	
0	—	

Para obtener más información, consulte [Configuración de entornos de Elastic Beanstalk](#).

Creación de un entorno de Elastic Beanstalk

Un entorno AWS Elastic Beanstalk es un conjunto de recursos de AWS que ejecutan una versión de la aplicación. Puede implementar varios entornos cuando necesite ejecutar varias versiones de una aplicación. Por ejemplo, es posible que tenga entornos de desarrollo, integración y producción.

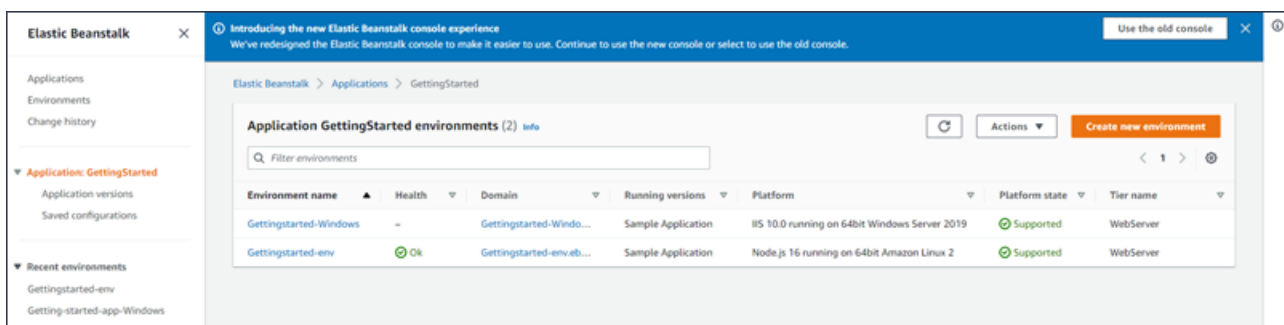
El siguiente procedimiento inicia un nuevo entorno que ejecuta la aplicación predeterminada. Estos pasos se simplifican para que su entorno funcione rápidamente, utilizando los valores de opción predeterminados. Para obtener instrucciones detalladas con descripciones de las muchas opciones que puede utilizar para configurar los recursos que Elastic Beanstalk implementa en su nombre, consulte [El asistente de creación de nuevo entorno](#).

Notas

- Para obtener instrucciones sobre cómo crear y administrar entornos con la CLI de EB, consulte [Administración de entornos de Elastic Beanstalk con la CLI de EB](#).
- La creación de un entorno requiere los permisos de la política administrada de acceso completo de Elastic Beanstalk. Para obtener más información, consulte [Política de usuario de Elastic Beanstalk](#).

Para lanzar un entorno con una aplicación de ejemplo (consola)

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Applications (Aplicaciones) y, a continuación, elija el nombre de una aplicación existente de la lista o [cree una](#).
3. En la página de información general de la aplicación, seleccione Create new environment (Crear un nuevo entorno).



Esto abre el asistente de Create environment (Creación de entorno). El asistente proporciona un conjunto de pasos para crear un entorno nuevo.

Step 1
Configure environment

Step 2
Configure service access

Step 3 - optional
Configure instance traffic and scaling

Step 4 - optional
Set up networking, database, and tags

Step 5 - optional
Configure updates, monitoring, and logging

Step 6
Review

Configure environment [Info](#)

Environment tier [Info](#)

Amazon Elastic Beanstalk has two types of environment tiers to support different types of web applications.

- Web server environment**
Run a website, web application, or web API that serves HTTP requests. [Learn more](#)
- Worker environment**
Run a worker application that processes long-running workloads on demand or performs tasks on a schedule. [Learn more](#)

Application information [Info](#)

Application name
GettingStarted
Maximum length of 100 characters.

▶ **Application tags (optional)**

Environment information [Info](#)

Choose the name, subdomain and description for your environment. These cannot be changed later.

Environment name
GettingStarted-env
Must be from 4 to 40 characters in length. The name can contain only letters, numbers, and hyphens. It can't start or end with a hyphen. This name must be unique within a region in your account.

Domain name
Leave blank for autogenerated value .us-east-1.elasticbeanstalk.com [Check availability](#)

Environment description

Platform [Info](#)

Platform type

- Managed platform**
Platforms published and maintained by Amazon Elastic Beanstalk. [Learn more](#)
- Custom platform**
Platforms created and owned by you. This option is unavailable if you have no platforms.

Platform
Choose a platform ▼


Platform branch
Choose a platform branch ▼

Platform version
Choose a platform version ▼

Application code [Info](#)


- Sample application**
- Existing version**
Application versions that you have uploaded.
Sample Application ▼
- Upload your code**
Upload a source bundle from your computer or copy one from Amazon S3.

4. Para el [environmet tier](#) (nivel de entorno), elija Web server environment (Entorno del servidor web) o Worker environment (Entorno de trabajo). Una vez creado, no se puede cambiar la capa del entorno.

 Note

La [plataforma .NET en Windows Server](#) no admite la capa del entorno de trabajo.

5. En Plataforma, seleccione la plataforma y la ramificación de la plataforma que coincidan con el lenguaje utilizado por la aplicación.

 Note

Elastic Beanstalk admite varias [versiones](#) para la mayoría de las plataformas que se indican. De forma predeterminada, la consola selecciona la versión recomendada para la plataforma y la ramificación de la plataforma que elija. Si la aplicación requiere una versión diferente, puede seleccionarla aquí. Para obtener más información acerca de las versiones compatibles de la plataforma, consulte [the section called “Plataformas admitidas”](#).

6. En Application code (Código de aplicación), seleccione Sample application (Aplicación de muestra).
7. Para los Configuration presets (Modelos de configuración preestablecidos), elija Single instance (Instancia individual).
8. Elija Next (Siguiente).
9. Aparece la página Configurar acceso a servicio.

Configure service access [Info](#)

Service access
IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profiles allow Elastic Beanstalk to create and manage your environment. Both the IAM role and instance profile must be attached to IAM managed policies that contain the required permissions. [Learn more](#)

Service role

Create and use new service role

Use an existing service role

Existing service roles
Choose an existing IAM role for Elastic Beanstalk to assume as a service role. The existing IAM role must have the required IAM managed policies.

aws-elasticbeanstalk-service-role

EC2 key pair
Select an EC2 key pair to securely log in to your EC2 instances. [Learn more](#)

Choose a key pair

EC2 instance profile
Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.

aws-elasticbeanstalk-ec2-role

10. Seleccione Crear un rol de servicio para Rol de servicio.

11. A continuación, nos centraremos en la lista desplegable Perfil de instancia de EC2. Los valores que aparecen en esta lista desplegable pueden variar, en función de si su cuenta ha creado previamente un entorno nuevo.

Seleccione una de las siguientes opciones, dependiendo de los valores que aparezcan en la lista.

- Si `aws-elasticbeanstalk-ec2-role` aparece en la lista desplegable, selecciónelo en la lista desplegable Perfil de instancia de EC2.
- Si aparece otro valor en la lista y es el perfil de instancia de EC2 predeterminado destinado a sus entornos, selecciónelo en la lista desplegable Perfil de instancia de EC2.
- Si en la lista desplegable Perfil de instancia de EC2 no aparecen valores para seleccionar, amplíe el procedimiento siguiente, Crear un rol de IAM para el perfil de instancia de EC2.

Siga los pasos de Crear un rol de IAM para el perfil de instancia de EC2 para crear un rol de IAM que pueda seleccionar posteriormente para el Perfil de instancia de EC2. Luego, vuelva a este paso.

Ahora que ha creado un rol de IAM y ha actualizado la lista, aparece como opción en la lista desplegable. Seleccione el rol de IAM que acaba de crear en la lista desplegable Perfil de instancia de EC2.

12. Seleccione Skip to review (Saltar a revisión) en la página Configure service access (Configurar el acceso al servicio).

Se seleccionarán los valores predeterminados para este paso y se omitirán los pasos opcionales.

13. En la página Review (Revisar), se muestra un resumen de todas sus elecciones.

Para personalizar aún más su entorno, elija Edit (Editar) junto al paso que incluye los elementos que desee configurar. Solo puede establecer las siguientes opciones durante la creación del entorno:

- Environment name
- Nombre de dominio
- Versión de la plataforma
- Procesador
- VPC
- datos y búsqueda

Puede modificar los siguientes ajustes después de crear el entorno, pero requieren que se aprovisionen nuevas instancias u otros recursos, lo que puede llevar algún tiempo:

- Tipo de instancia, volumen raíz, par de claves y rol de AWS Identity and Access Management (IAM)
- Base de datos interna de Amazon RDS
- Balanceador de carga

Para obtener más información sobre los ajustes disponibles, consulte [El asistente de creación de nuevo entorno](#).

14. Elija Submit (Enviar) en la parte inferior de la página para iniciar la creación del nuevo entorno.

Crear un rol de IAM para el perfil de instancia de EC2

Configure service access [Info](#)

Service access
IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profiles allow Elastic Beanstalk to create and manage your environment. Both the IAM role and instance profile must be attached to IAM managed policies that contain the required permissions. [Learn more](#)

Service role

Create and use new service role
 Use an existing service role

Existing service roles
Choose an existing IAM role for Elastic Beanstalk to assume as a service role. The existing IAM role must have the required IAM managed policies.

aws-elasticbeanstalk-service-role

EC2 key pair
Select an EC2 key pair to securely log in to your EC2 instances. [Learn more](#)

Choose a key pair

EC2 instance profile
Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.

aws-elasticbeanstalk-ec2-role

[View permission details](#)

Cancel Skip to review Previous **Next**

Para crear un rol de IAM para la selección del perfil de instancia de EC2


1. Seleccione Ver detalles de permisos. Esta opción aparece en la lista desplegable Perfil de instancia de EC2.

Aparece una ventana modal titulada Ver permisos del perfil de instancia. En esta ventana se muestran los perfiles administrados que deberá asociar al nuevo perfil de instancia de EC2 que cree. También proporciona un enlace para ejecutar la consola de IAM.

2. Seleccione el enlace Consola de IAM que aparece en la parte superior de la ventana.
3. En el panel de navegación de la consola de IAM, elija Roles.
4. Elija Create role (Crear rol).
5. En Tipo de entidad de confianza, seleccione Servicio de AWS.
6. En Use case (Caso de uso), elija EC2.
7. Elija Next (Siguiente).
8. Asocie las políticas administradas adecuadas. Desplácese por la ventana modal Ver permisos del perfil de instancia para ver las políticas administradas. Las políticas también aparecen aquí:

- `AWSElasticBeanstalkWebTier`
- `AWSElasticBeanstalkWorkerTier`
- `AWSElasticBeanstalkMulticontainerDocker`

9. Elija Next (Siguiente).
10. Escriba un nombre para el rol.
11. (Opcional) Añada etiquetas al rol.
12. Elija Create role (Crear rol).
13. Vuelva a la ventana de la consola de Elastic Beanstalk que está abierta.
14. Cierre la ventana modal Ver permisos del perfil de instancia.

 Important

No cierre la página del navegador donde se muestra la consola de Elastic Beanstalk.

15. Seleccione



(actualizar), junto a la lista desplegable Perfil de instancia de EC2.

Esto actualiza la lista desplegable, para que el rol que acaba de crear aparezca en ella.

Mientras Elastic Beanstalk crea su entorno, se le redirige a la [consola de Elastic Beanstalk](#). Una vez que el estado del entorno pase a verde, elija la URL junto al nombre del entorno para ver la aplicación en ejecución. Esta URL es, por lo general, accesible desde Internet, a menos que configure su entorno para utilizar una [VPC personalizada con un balanceador de carga interno](#).

Temas

- [El asistente de creación de nuevo entorno](#)
- [Clonación de un entorno de Elastic Beanstalk](#)
- [Terminación de un entorno de Elastic Beanstalk](#)
- [Creación de entornos de Elastic Beanstalk con la CLI de AWS](#)
- [Creación de entornos de Elastic Beanstalk con la API](#)
- [Construcción de una URL de lanzamiento inmediato](#)
- [Creación y actualización de grupos de entornos de Elastic Beanstalk](#)

El asistente de creación de nuevo entorno

En [Creación de un entorno de Elastic Beanstalk](#) le mostramos cómo abrir el asistente Create environment (Creación de entorno) y crear un entorno de forma rápida. Elija Create environment (Crear entorno) para lanzar un entorno con el nombre de entorno predeterminado, el dominio generado de forma automática, código de aplicación de ejemplo y la configuración recomendada.

En este tema se describe el asistente Create environment (Creación de entorno) y las formas en las que puede utilizarlo para configurar el entorno que desea crear.

Página del asistente

El asistente Create environment (Creación de entorno) proporciona un conjunto de pasos para la creación de un entorno nuevo.

Step 1
Configure environment

Step 2
Configure service access

Step 3 - optional
Configure instance traffic and scaling

Step 4 - optional
Set up networking, database, and tags

Step 5 - optional
Configure updates, monitoring, and logging

Step 6
Review

Configure environment [Info](#)

Environment tier [Info](#)

Amazon Elastic Beanstalk has two types of environment tiers to support different types of web applications.

- Web server environment**
Run a website, web application, or web API that serves HTTP requests. [Learn more](#)
- Worker environment**
Run a worker application that processes long-running workloads on demand or performs tasks on a schedule. [Learn more](#)

Application information [Info](#)

Application name
GettingStarted
Maximum length of 100 characters.

▶ **Application tags (optional)**

Environment information [Info](#)

Choose the name, subdomain and description for your environment. These cannot be changed later.

Environment name
GettingStarted-env
Must be from 4 to 40 characters in length. The name can contain only letters, numbers, and hyphens. It can't start or end with a hyphen. This name must be unique within a region in your account.

Domain name
 .us-east-1.elasticbeanstalk.com

Environment description

Platform [Info](#)

Platform type

- Managed platform**
Platforms published and maintained by Amazon Elastic Beanstalk. [Learn more](#)
- Custom platform**
Platforms created and owned by you. This option is unavailable if you have no platforms.

Platform

Platform branch

Platform version

Application code [Info](#)

- Sample application**
- Existing version**
Application versions that you have uploaded.
- Upload your code**
Upload a source bundle from your computer or copy one from Amazon S3.

El asistente de creación de nuevo entorno

Capa del entorno

Para el environment tier (nivel de entorno), elija Web server environment (Entorno del servidor web) o Worker environment (Entorno de empleado) del [nivel de entorno](#). Una vez creado, no se puede cambiar la capa del entorno.

Environment tier [Info](#)

Amazon Elastic Beanstalk has two types of environment tiers to support different types of web applications.

- Web server environment**
Run a website, web application, or web API that serves HTTP requests. [Learn more](#)
- Worker environment**
Run a worker application that processes long-running workloads on demand or performs tasks on a schedule. [Learn more](#)

Note

La [plataforma .NET en Windows Server](#) no admite la capa del entorno de trabajo.

Información sobre la aplicación

Si inició el asistente mediante Create new environment (Crear nuevo entorno) desde la página Application overview (Información general de la aplicación), el Application name (Nombre de la aplicación) viene predefinido. De lo contrario, escriba un nombre para la aplicación. Si lo desea, agregue [etiquetas de la aplicación](#).

Application information [Info](#)

Application name

GettingStarted

Maximum length of 100 characters.

▼ **Application tags (optional)**

Apply up to 50 tags. You can use tags to group and filter your resources. A tag is a key-value pair. The key must be unique within the resource and is case-sensitive. [Learn more](#)

No tags associated with the resource.

Add new tag

You can add 50 more tags.

Información sobre el entorno

Establezca el nombre y el dominio del entorno, y cree una descripción para el entorno. Tenga en cuenta que no se puede cambiar esta configuración del entorno tras crearlo.

Environment information [Info](#)

Choose the name, subdomain and description for your environment. These cannot be changed later.

Environment name

Must be from 4 to 40 characters in length. The name can contain only letters, numbers, and hyphens. It can't start or end with a hyphen. This name must be unique within a region in your account.

Domain name

 .us-east-1.elasticbeanstalk.com

Environment description

- **Name (Nombre):** introduzca un nombre para el entorno. El formulario proporciona un nombre generado.
- **Domain (Dominio):** (entornos de servidor web) introduzca un nombre de dominio único para el entorno. El nombre predeterminado es el nombre del entorno. Puede introducir otro nombre de dominio. Elastic Beanstalk utiliza este nombre para crear un CNAME único para el entorno. Para comprobar si el nombre del dominio que desea usar está disponible, elija **Check Availability** (Comprobar disponibilidad).
- **Description (Descripción):** escriba una descripción para este entorno.

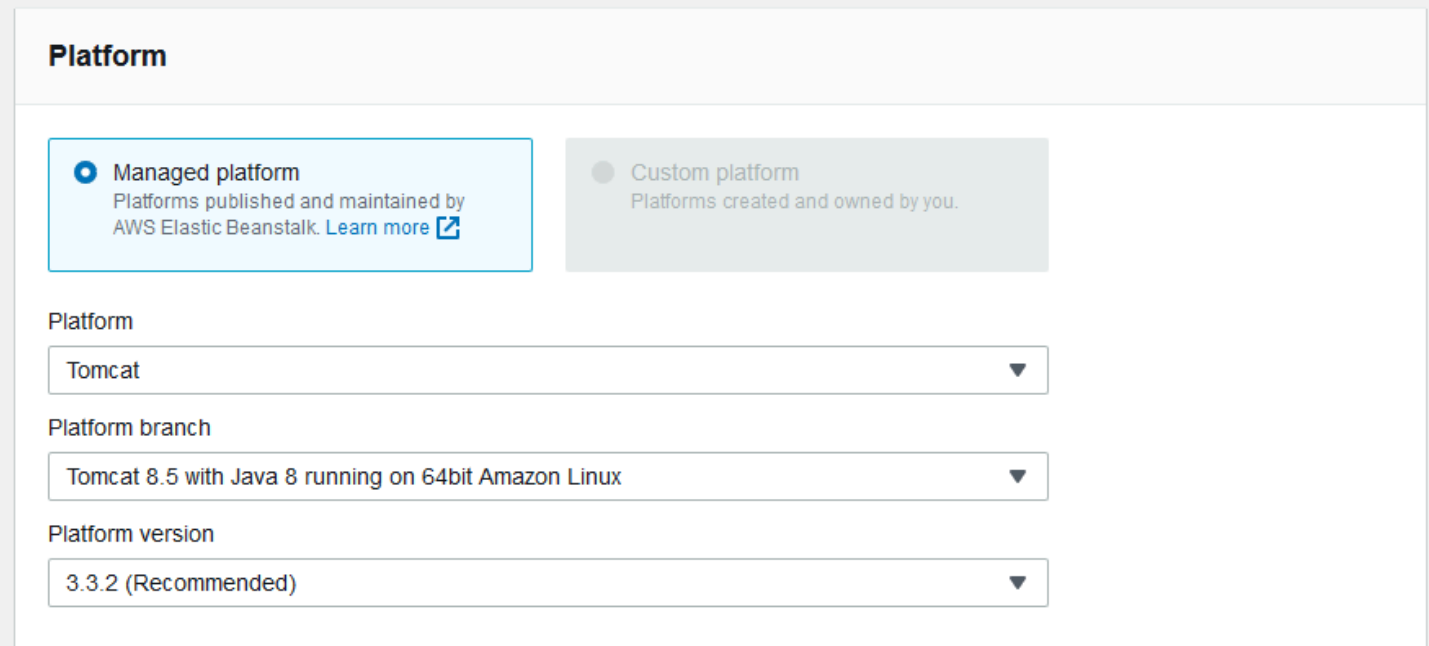
Selección de una plataforma para el nuevo entorno

Puede crear un nuevo entorno desde dos tipos de plataformas:

- Plataforma administrada
- Plataforma personalizada

Plataforma administrada

En la mayoría de los casos utilizará una plataforma administrada de Elastic Beanstalk para su nuevo entorno. Cuando se inicia el nuevo asistente de entorno, este selecciona la opción **Managed platform** (Plataforma administrada) de forma predeterminada.



Platform

Managed platform
Platforms published and maintained by AWS Elastic Beanstalk. [Learn more](#)

Custom platform
Platforms created and owned by you.

Platform
Tomcat

Platform branch
Tomcat 8.5 with Java 8 running on 64bit Amazon Linux

Platform version
3.3.2 (Recommended)

Seleccione una plataforma, una ramificación de plataforma dentro de esa plataforma y una versión de plataforma específica en la ramificación. Cuando selecciona una ramificación de plataforma, la versión recomendada dentro de la ramificación se selecciona de forma predeterminada. Además, puede seleccionar cualquier versión de plataforma que haya utilizado anteriormente.

Note

Para un entorno de producción, le recomendamos que elija una versión de plataforma en una ramificación de plataforma compatible. Para obtener más información sobre los estados de ramificación de plataforma, consulte la definición de Ramificación de la plataforma en [the section called “Glosario de plataformas”](#).

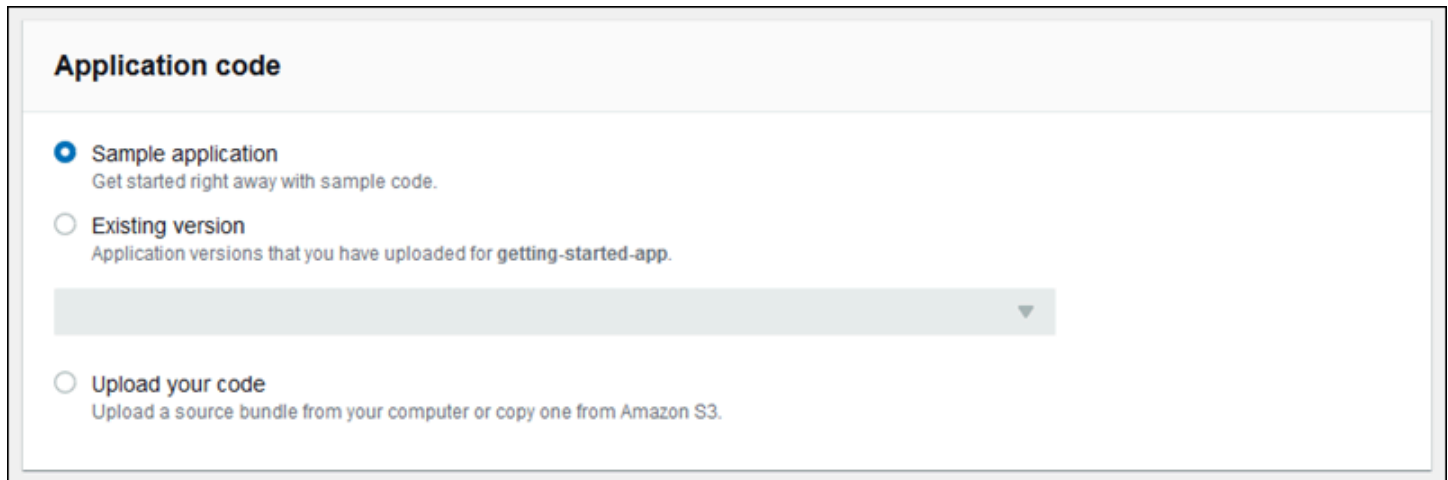
Plataforma personalizada

Si una plataforma estándar no satisface sus necesidades, puede crear un nuevo entorno a partir de una plataforma personalizada. Para especificar una plataforma personalizada, elija la opción **Custom platform** (Plataforma personalizada) y, a continuación, seleccione una de las plataformas

personalizadas disponibles. Si no hay plataformas personalizadas disponibles, esta opción aparece atenuada.

Especificación del código de la aplicación

Ahora que ha seleccionado la plataforma que va a utilizar, el siguiente paso consiste en proporcionar el código de la aplicación.



Application code

Sample application
Get started right away with sample code.

Existing version
Application versions that you have uploaded for `getting-started-app`.

Upload your code
Upload a source bundle from your computer or copy one from Amazon S3.

Dispone de varias opciones para hacerlo:

- Puede utilizar la aplicación de ejemplo que Elastic Beanstalk proporciona para cada plataforma.
- Puede utilizar código que ya haya implementado en Elastic Beanstalk. Elija Existing version (Versión existente) y la aplicación en la sección Application code (Código de aplicación).
- Puede cargar código nuevo. Seleccione Upload your code (Cargar el código) y, a continuación, elija Upload (Cargar). Puede cargar código de aplicación nuevo de un archivo local, o puede especificar la URL del bucket de Amazon S3 que contiene el código de la aplicación.

Note

En función de la versión de la plataforma seleccionada, puede cargar la aplicación en un [paquete de código fuente](#) con formato ZIP, un [archivo WAR](#) o una [configuración de Docker en texto sin formato](#). El límite de tamaño de archivo es 500 MB.

Cuando elige cargar nuevo código, también puede proporcionar etiquetas que asociar con el código. Para obtener más información sobre cómo etiquetar una versión de aplicación, consulte [the section called “Etiquetado de versiones de la aplicación”](#).

Application code

Sample application
Get started right away with sample code.

Existing version
Application versions that you have uploaded for `getting-started-app`.

Upload your code
Upload a source bundle from your computer or copy one from Amazon S3.

▼ Source code origin

(Maximum size 512 MB)

Local file

Public S3 URL

File name : **java-tomcat-v3.zip**

File successfully uploaded

Version label
Unique name for this version of your application code.

▼ Application code tags

Apply up to 50 tags. You can use tags to group and filter your resources. A tag is a key-value pair. The key must be unique within the resource and is case-sensitive. [Learn more](#)

Key	Value	
<input type="text"/>	<input type="text"/>	<input type="button" value="Remove tag"/>

50 remaining

Para la creación rápida del entorno mediante las opciones de configuración predeterminadas, ahora puede elegir **Create environment** (Crear entorno). Elija **Configure more options** (Configurar

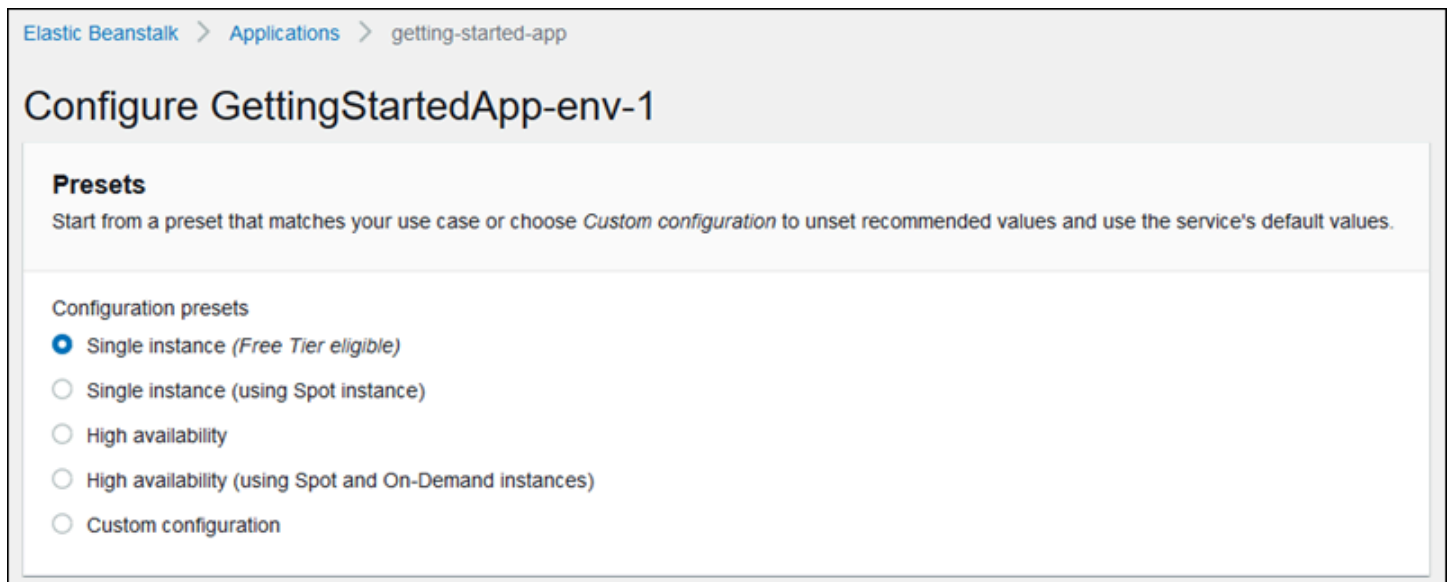
más opciones) para tener acceso a cambios de configuración adicionales, que se describen en las secciones siguientes.

Página de configuración del asistente

Al elegir *Configure more options* (Configurar más opciones), el asistente muestra la página *Configure* (Configurar) . En esta página puede seleccionar un ajuste preestablecido de configuración, cambiar la versión de la plataforma que desea que utilice su entorno o realizar opciones de configuración específicas para el nuevo entorno.

Elegir una configuración preestablecida

En la sección *Presets* (Ajustes preestablecidos) de la página, Elastic Beanstalk proporciona varios modelos de configuración preestablecidos para diferentes casos de uso. Cada modelo preestablecido incluye valores recomendados para varias [opciones de configuración](#).



Los ajustes preestablecidos de *High availability* (alta disponibilidad) incluyen un balanceador de carga y están recomendados para entornos de producción. Seleccione esta opción si desea un entorno con balanceo de carga que pueda ejecutar varias instancias para ofrecer una alta disponibilidad y capacidad de ampliación en respuesta a la carga. Los ajustes preestablecidos de *Single instance* (instancia única) están recomendados principalmente para entornos de desarrollo. Dos de los ajustes preestablecidos habilitan las solicitudes de instancias de spot. Para obtener más información sobre la configuración de capacidad de Elastic Beanstalk, consulte [Grupo de Auto Scaling](#).

El tercer ajuste preestablecido, Custom configuration (Configuración personalizada), elimina todos los valores recomendados, excepto la configuración del rol y utiliza los valores predeterminados de la API. Seleccione esta opción si va a implementar un paquete de código fuente con [archivos de configuración](#) que establecen opciones de configuración. El modelo Custom configuration (Configuración personalizada) también se selecciona automáticamente si modifica los modelo de configuración preestablecidos Low cost (Bajo costo) o High availability (Alta disponibilidad).

Personalización de la configuración

Además de (o en lugar de) elegir un modelo de configuración preestablecido, puede ajustar las [opciones de configuración](#) del entorno. El asistente de Configure (Configuración) muestra varias categorías de configuración. Cada categoría de configuración muestra un resumen de valores para un grupo de opciones de configuración. Elija Edit (Editar) para editar este grupo de opciones.

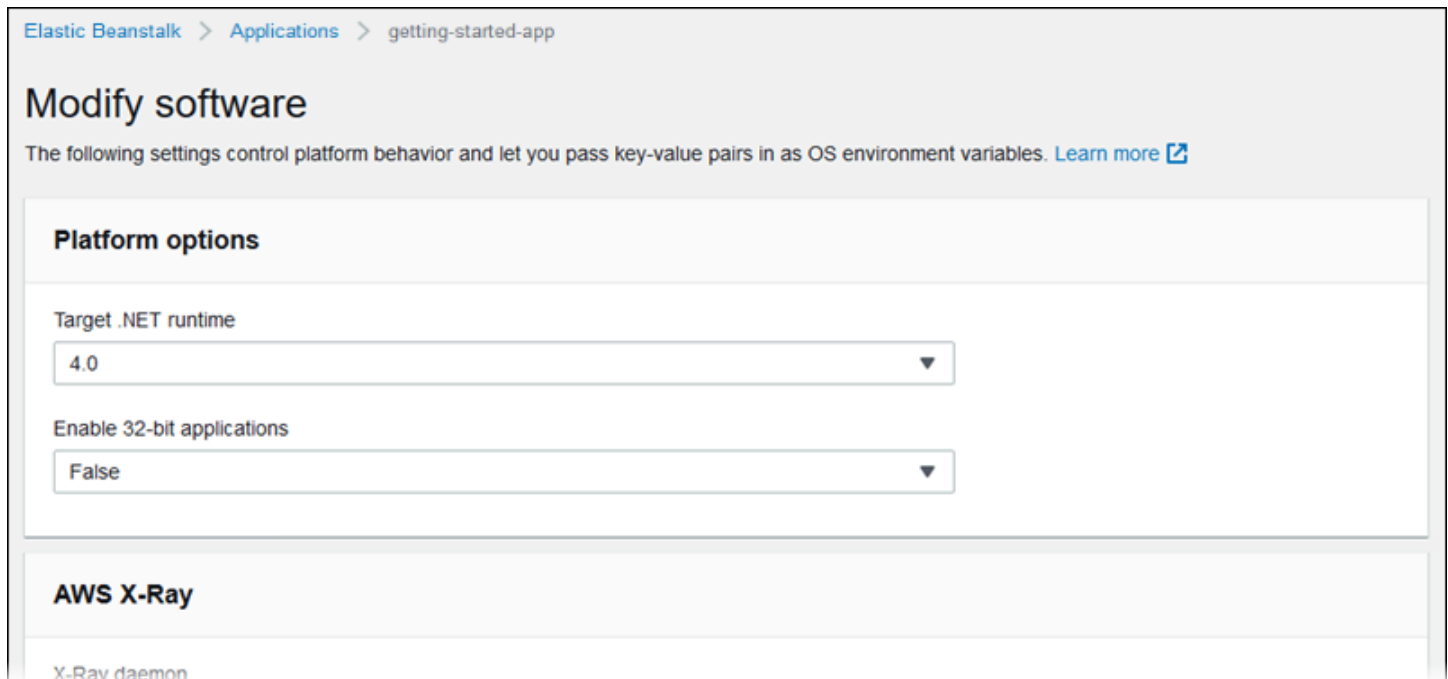
Categorías de configuración

- [Configuración del software](#)
- [Instancias](#)
- [Capacidad](#)
- [Balanceador de carga](#)
- [Actualizaciones e implementaciones continuas](#)
- [Seguridad](#)
- [Supervisión](#)
- [Managed Updates \(Actualizaciones administradas\)](#)
- [Notificaciones](#)
- [Red](#)
- [Base de datos](#)
- [Etiquetas](#)
- [Entorno de trabajo](#)

Configuración del software

Utilice la página de configuración Modify software (Modificar software) para configurar el software en las instancias de Amazon Elastic Compute Cloud (Amazon EC2) que ejecutan la aplicación. Puede configurar las propiedades del entorno, la depuración de AWS X-Ray, el almacenamiento y el

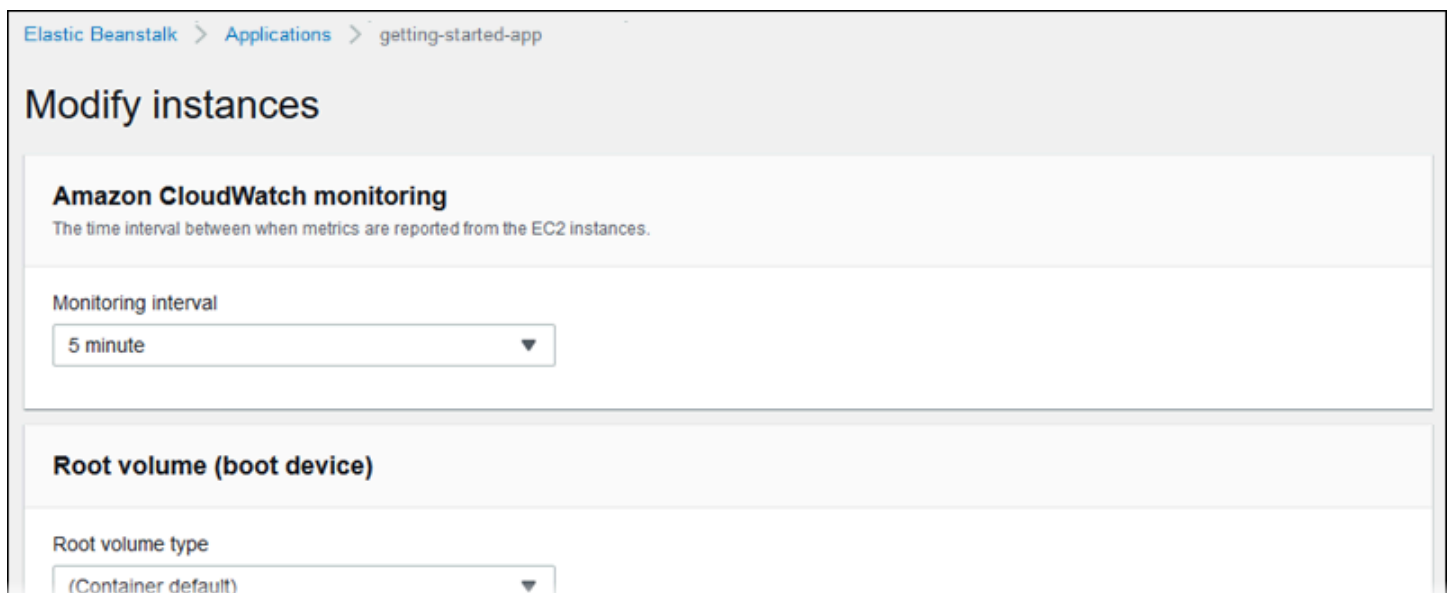
streaming de registros de instancias y la configuración específica de la plataforma. Para obtener más información, consulte [the section called “Propiedades del entorno y configuración de software”](#).



The screenshot shows the 'Modify software' page in the AWS Elastic Beanstalk console. The breadcrumb trail is 'Elastic Beanstalk > Applications > getting-started-app'. The page title is 'Modify software'. Below the title is a subtitle: 'The following settings control platform behavior and let you pass key-value pairs in as OS environment variables. [Learn more](#)'. The page is divided into two main sections: 'Platform options' and 'AWS X-Ray'. Under 'Platform options', there are two dropdown menus: 'Target .NET runtime' set to '4.0' and 'Enable 32-bit applications' set to 'False'. Under 'AWS X-Ray', there is a section for 'X-Ray daemon'.

Instancias

Utilice la página de configuración Modify instances (Modificar instancias) para configurar las instancias de Amazon EC2 que ejecutan la aplicación. Para obtener más información, consulte [the section called “Instancias de Amazon EC2”](#).



The screenshot shows the 'Modify instances' page in the AWS Elastic Beanstalk console. The breadcrumb trail is 'Elastic Beanstalk > Applications > getting-started-app'. The page title is 'Modify instances'. Below the title is a section for 'Amazon CloudWatch monitoring' with the subtitle 'The time interval between when metrics are reported from the EC2 instances.' There is a dropdown menu for 'Monitoring interval' set to '5 minute'. Below this is a section for 'Root volume (boot device)' with a dropdown menu for 'Root volume type' set to '(Container default)'.

Capacidad

Utilice la página de configuración **Modify capacity** (Modificar capacidad) para configurar la capacidad de cómputo de su entorno y las opciones del **Auto Scaling group** (Grupo de Auto Scaling) para optimizar el número y el tipo de instancias que va a utilizar. También puede cambiar la capacidad de su entorno en función de desencadenadores o de una programación.

Un entorno con balanceo de carga puede ejecutar varias instancias a fin de ofrecer un alto nivel de disponibilidad y evitar el tiempo de inactividad durante las actualizaciones de configuración y las implementaciones. En un entorno con balanceo de carga, el nombre de dominio se asigna al balanceador de carga. En un entorno de una sola instancia, se asigna a una dirección IP elástica de la instancia.

Warning

Un entorno de una sola instancia no está preparado para producción. Si la instancia se vuelve inestable durante la implementación o Elastic Beanstalk termina y reinicia la instancia durante una actualización de la configuración, es posible que la aplicación no esté disponible durante algún tiempo. Utilice entornos de una sola instancia para desarrollo, pruebas o ensayos. Utilice entornos con balanceo de carga para producción.

Para obtener más información sobre la configuración de capacidad del entorno, consulte [the section called “Grupo de Auto Scaling”](#) y [the section called “Instancias de Amazon EC2”](#).

Elastic Beanstalk > Applications > getting-started-app


Modify capacity

Configure the compute capacity of your environment and Auto Scaling settings to optimize the number of instances used.

Auto Scaling Group

Environment type
Load balanced ▼

Instances
Min 1 ▼
Max 2 ▼

Fleet composition
Choose a mix of On-Demand and Spot Instances with multiple instance types. Spot Instances are automatically launched at the lowest available price. [Learn more](#) 

On-Demand instances
 Combine purchase options and instances

Maximum spot price
The maximum price per instance-hour, in USD, that you're willing to pay for a Spot Instance. Setting a custom price limits your chances to fulfill your target capacity using [Spot Instances](#).

Balanceador de carga

Utilice la página de configuración Modify load balancer (Modificar balanceador de carga) para seleccionar un tipo de balanceador de carga y configurar sus opciones. En un entorno con balanceo de carga, el balanceador de carga del entorno es el punto de entrada de todo el tráfico dirigido a la aplicación. Elastic Beanstalk admite varios tipos de balanceadores de carga. De forma predeterminada, la consola de Elastic Beanstalk crea un Application Load Balancer y lo configura para servir el tráfico HTTP en el puerto 80.

Note

Solo puede seleccionar el tipo de balanceador de carga del entorno durante la creación del entorno.

Para obtener más información sobre los tipos de balanceador de carga y sus opciones, consulte [the section called “Balanceador de carga”](#) y [the section called “HTTPS”](#).

Elastic Beanstalk > Applications > getting-started-app

Modify load balancer

Application Load Balancer

Application layer load balancer—routing HTTP and HTTPS traffic based on protocol, port, and route to environment processes.

Classic Load Balancer

Previous generation — HTTP, HTTPS, and TCP

Network Load Balancer

Ultra-high performance and static IP addresses for your application.

Application Load Balancer

You can specify listeners for your load balancer. Each listener routes incoming client traffic on a specified port using a specified protocol to your environment processes. By default, we've configured your load balancer with a standard web server on port 80.

Actions ▾

Add listener

<input type="checkbox"/>	Port	Protocol	SSL certificate	Enabled
<input type="checkbox"/>	80	HTTP	--	<input checked="" type="checkbox"/>

Processes

For each environment process, you can specify the protocol and port that the load balancer uses to route requests to the process. You can

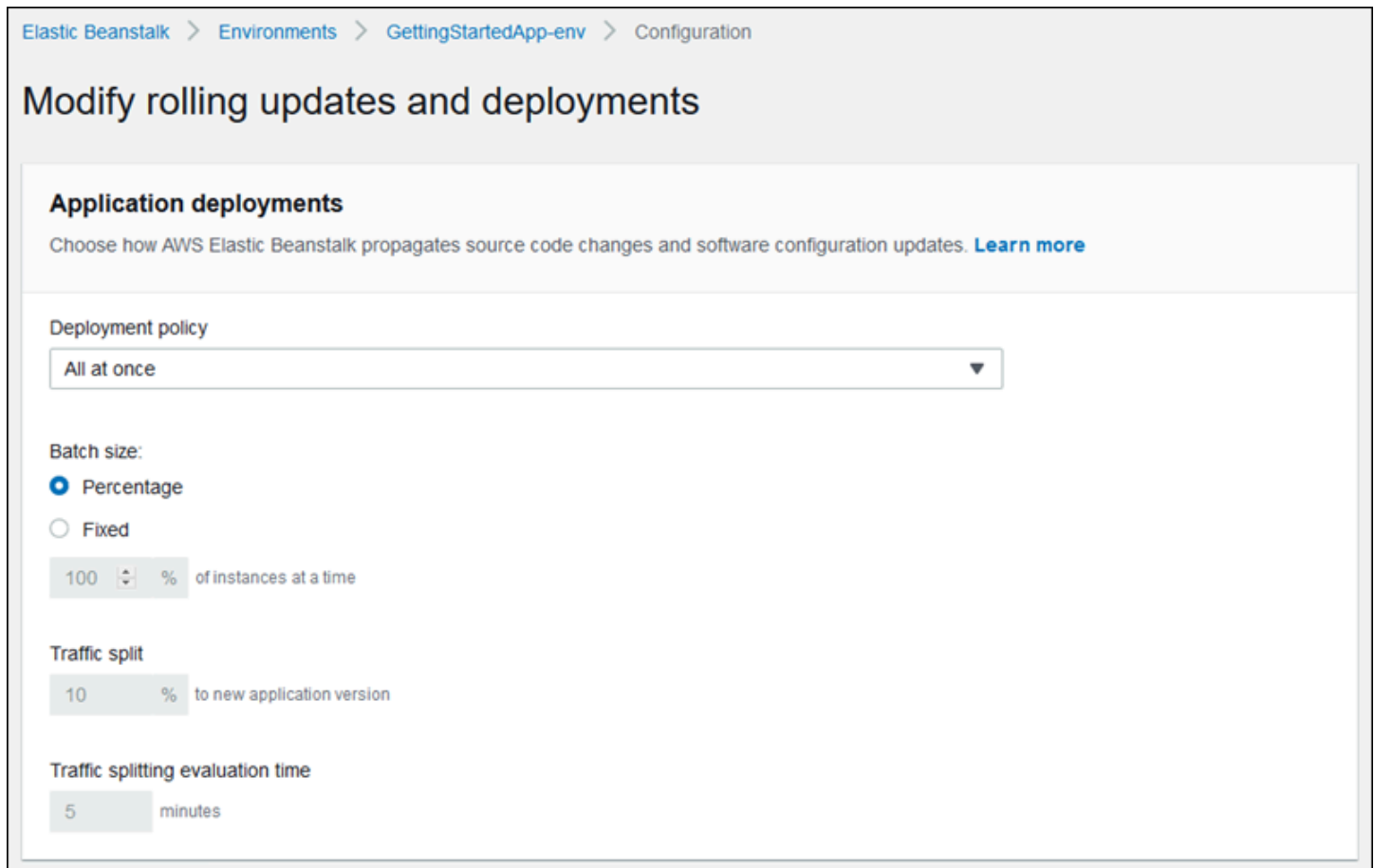
Note

La opción Equilibrador de carga clásico (CLB) está deshabilitada en el asistente de la consola Crear entorno. Si ya tiene un entorno configurado con un Equilibrador de carga clásico, puede crear otro nuevo [clonando el entorno existente](#) mediante la consola de Elastic Beanstalk o la [CLI de EB](#). También tiene la opción de utilizar la [CLI de EB](#) o la [AWS CLI](#) para crear un nuevo entorno configurado con un Equilibrador de carga clásico. Estas herramientas de línea de comandos crearán un nuevo entorno con un CLB incluso si aún no existe ninguno en su cuenta.

Actualizaciones e implementaciones continuas

Utilice la página de configuración **Modify rolling updates and deployments** (Modificar actualizaciones e implementaciones continuas) para configurar cómo Elastic Beanstalk procesa las implementaciones de aplicaciones y las actualizaciones de configuración para su entorno.

Las implementaciones de aplicaciones se producen cuando carga un paquete de origen de aplicaciones actualizado y lo implementa en su entorno. Para obtener más información acerca de la configuración de implementaciones, consulte [the section called “Opciones de implementación”](#).



The screenshot shows the AWS Elastic Beanstalk console interface for configuring application deployments. The breadcrumb navigation at the top reads: Elastic Beanstalk > Environments > GettingStartedApp-env > Configuration. The main heading is "Modify rolling updates and deployments". Below this, there is a section titled "Application deployments" with a sub-heading: "Choose how AWS Elastic Beanstalk propagates source code changes and software configuration updates. [Learn more](#)".

The configuration options are as follows:

- Deployment policy:** A dropdown menu currently set to "All at once".
- Batch size:** Radio buttons for "Percentage" (selected) and "Fixed". Below the "Percentage" option is a numeric input field set to "100" followed by "% of instances at a time".
- Traffic split:** A numeric input field set to "10" followed by "% to new application version".
- Traffic splitting evaluation time:** A numeric input field set to "5" followed by "minutes".

Para los cambios de configuración que modifican la [configuración de lanzamiento](#) o la [configuración de VPC](#) es necesario terminar todas las instancias del entorno y reemplazarlas. Para obtener más información sobre cómo establecer el tipo de actualización y otras opciones, consulte [the section called “Cambios de configuración”](#).

Configuration updates

Changes to virtual machine settings and VPC configuration trigger rolling updates to replace the instances in your environment without downtime. [Learn more](#)

Rolling update type

Rolling based on Health

Batch size

1

The maximum number of instances to replace in each phase of the update.

Minimum capacity

1

The minimum number of instances to keep in service at all times.

Pause time

hh:mm:ss

Pause the update for up to an hour between each batch.

Seguridad

Utilice la página [Configure service access](#) (Configurar el acceso al servicio) para establecer las opciones de seguridad del servicio y de la instancia.

Para obtener una descripción de los conceptos de seguridad de Elastic Beanstalk, consulte [Permisos](#).

La primera vez que se crea un entorno en la consola de Elastic Beanstalk, se debe crear un perfil de instancia de EC2 con un conjunto de permisos predeterminados. Si en la lista desplegable Perfil de instancia de EC2 no aparecen valores para seleccionar, amplíe el procedimiento siguiente. Proporciona pasos para crear un rol que puede seleccionar posteriormente para el Perfil de instancia de EC2.


Crear un rol de IAM para el perfil de instancia de EC2

Para crear un rol de IAM para la selección del perfil de instancia de EC2

1. Seleccione [Ver detalles de permisos](#). Esta opción aparece en la lista desplegable Perfil de instancia de EC2.

Aparece una ventana modal titulada Ver permisos del perfil de instancia. En esta ventana se muestran los perfiles administrados que deberá asociar al nuevo perfil de instancia de EC2 que cree. También proporciona un enlace para ejecutar la consola de IAM.

2. Seleccione el enlace Consola de IAM que aparece en la parte superior de la ventana.
3. En el panel de navegación de la consola de IAM, elija Roles.
4. Elija Create role (Crear rol).
5. En Tipo de entidad de confianza, seleccione Servicio de AWS.
6. En Use case (Caso de uso), elija EC2.
7. Elija Next (Siguiente).
8. Asocie las políticas administradas adecuadas. Desplácese por la ventana modal Ver permisos del perfil de instancia para ver las políticas administradas. Las políticas también aparecen aquí:
 - AWSElasticBeanstalkWebTier
 - AWSElasticBeanstalkWorkerTier
 - AWSElasticBeanstalkMulticontainerDocker
9. Elija Next (Siguiente).
10. Escriba un nombre para el rol.
11. (Opcional) Añada etiquetas al rol.
12. Elija Create role (Crear rol).
13. Vuelva a la ventana de la consola de Elastic Beanstalk que está abierta.
14. Cierre la ventana modal Ver permisos del perfil de instancia.

 Important

No cierre la página del navegador donde se muestra la consola de Elastic Beanstalk.

15. Seleccione



(actualizar), junto a la lista desplegable Perfil de instancia de EC2.

Esto actualiza la lista desplegable, para que el rol que acaba de crear aparezca en ella.

Configure service access Info

Service access
IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profiles allow Elastic Beanstalk to create and manage your environment. Both the IAM role and instance profile must be attached to IAM managed policies that contain the required permissions. [Learn more](#)

Service role

Create and use new service role

Use an existing service role

Existing service roles
Choose an existing IAM role for Elastic Beanstalk to assume as a service role. The existing IAM role must have the required IAM managed policies.

aws-elasticbeanstalk-service-role

EC2 key pair
Select an EC2 key pair to securely log in to your EC2 instances. [Learn more](#)

Choose a key pair

EC2 instance profile
Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.

aws-elasticbeanstalk-ec2-role

Supervisión

Utilice la página [Modify monitoring](#) (Modificar configuración de supervisión) para configurar informes de mantenimiento, reglas de supervisión y transmisión de eventos de mantenimiento. Para más detalles, consulte [the section called “Habilitación de informes de estado avanzados”](#), [the section called “Reglas de informes de estado avanzados”](#) y [the section called “Streaming del estado del entorno”](#).



Elastic Beanstalk > Applications > getting-started-app

Modify monitoring

Health reporting

Enhanced health reporting provides free real-time application and operating system monitoring of the instances and other resources in your environment. The **EnvironmentHealth** custom metric is provided free with enhanced health reporting. Additional charges apply for each custom metric. For more information, see [Amazon CloudWatch Pricing](#).

System

Enhanced

Basic

CloudWatch Custom Metrics - Instance

Choose metrics

Managed Updates (Actualizaciones administradas)

Utilice la página de configuración **Modify managed updates** (Modificar actualizaciones administradas) para configurar actualizaciones administradas de la plataforma. Puede decidir si desea activarlas, establecer la programación y configurar otras propiedades. Para obtener más información, consulte [the section called “Managed Updates \(Actualizaciones administradas\)”](#).

Elastic Beanstalk > Applications > getting-started-app

Modify managed updates

Managed platform updates

Enable managed platform updates to apply platform updates automatically during a weekly maintenance window that you choose. Your application stays available during the update process.

Managed updates

Enabled

Weekly update window

Tuesday at 12 : 00 UTC

Any available managed updates will run between Tuesday, 4:00 AM and Tuesday, 6:00 AM (-0800 GMT).

Update level

Minor and patch

Instance replacement

If enabled, an instance replacement will be scheduled if no other updates are available.

Enabled

Cancel Save

Notificaciones

Utilice la página Modify notifications (Modificar configuración de notificaciones) para especificar una dirección de correo electrónico para recibir [notificaciones de correo electrónico](#) de eventos importantes de su entorno.

Elastic Beanstalk > Applications > getting-started-app

Modify notifications

Email notifications

Enter an email address to receive email notifications for important events from your environment. [Learn more](#)

Email

Red

Si ha creado una [VPC personalizada](#), la página Modify network (Modificar configuración de red) para configurar el entorno y utilizarlo. Si no elige una VPC, Elastic Beanstalk utiliza la VPC y las subredes predeterminadas.

Elastic Beanstalk > Applications > getting-started-app

Modify network

Virtual private cloud (VPC)

VPC
Launch your environment in a custom VPC instead of the default VPC. You can create a VPC and subnets in the VPC management console. [Learn more](#)

vpc-0f9c96ae77f3c49c1 (172.31.0.0/16) | private-public

[Create custom VPC](#)

Load balancer settings

Assign your load balancer to a subnet in each Availability Zone (AZ) in which your application runs. For a publicly accessible application, set **Visibility** to **Public** and choose public subnets.

Visibility
Make your load balancer internal if your application serves requests only from connected VPCs. Public load balancers serve requests from the Internet.

Public

Load balancer subnets

Base de datos


Utilice la página de configuración **Modify database** (Modificar base de datos) para agregar una base de datos de Amazon Relational Database Service (Amazon RDS) a su entorno para las tareas de desarrollo y pruebas. Elastic Beanstalk proporciona información de conexión a sus instancias mediante la configuración de propiedades de entorno para el nombre de host, el nombre de usuario, la contraseña, el nombre de tabla y el puerto de la base de datos.

Para obtener más información, consulte [the section called “Base de datos”](#).

Elastic Beanstalk > Applications > getting-started-app

Modify database

Add an Amazon RDS SQL database to your environment for development and testing. AWS Elastic Beanstalk provides connection information to your instances by setting environment properties for the database hostname, username, password, table name, and port. When you add a database to your environment, its lifecycle is tied to your environment's.

For production environments, you can configure your instances to connect to a database. [Learn more](#) 

Restore a snapshot

Restore an existing snapshot in your account, or create a new database.

Snapshot

None



Database settings

Choose an engine and instance type for your environment's database.

Engine

mysql



Engine version

Etiquetas

Utilice la página de configuración Modify tags (Modificar etiquetas) para agregar [etiquetas](#) a los recursos del entorno. Para obtener más información sobre el etiquetado del entorno, consulte [Etiquetado de recursos en sus entornos de Elastic Beanstalk](#).

Elastic Beanstalk > Applications > getting-started-app

Modify tags

Apply up to 50 tags to the resources in your environment in addition to the default tags.

Key	Value	
mytag1	value1	Remove

Add tag

49 remaining

Cancel Save

Entorno de trabajo

Si está creando un entorno de nivel del trabajador, utilice la página *Modify worker configuration page* (Modificar configuración de la página del trabajador) para configurar el entorno del trabajador. El daemon del trabajador extrae elementos de una cola de Amazon Simple Queue Service (Amazon SQS) de instancias de su entorno y los transmite como mensajes de publicación a su aplicación de trabajo. Puede elegir la cola de Amazon SQS que lee el daemon del trabajador (generada automáticamente o existente). También puede configurar los mensajes que el daemon del trabajador envía a la aplicación.

Para obtener más información, consulte [the section called “Entornos de trabajo”](#).

Elastic Beanstalk > Applications > getting-started-app

Modify worker

You can create a new Amazon SQS queue for your worker application or pull work items from an existing queue. The worker daemon on the instances in your environment pulls an item from the queue and relays it in the body of a POST request to a local HTTP path relative to localhost.

Queue

Worker queue

Autogenerated queue

SQS queue from which to read work items.

Messages

HTTP path

Clonación de un entorno de Elastic Beanstalk

Puede utilizar un entorno de Elastic Beanstalk existente como punto de partida para un nuevo entorno clonando el entorno existente. Por ejemplo, es posible que desee crear un clon para utilizar una versión más reciente de la ramificación de la plataforma que utiliza la plataforma del entorno original. Elastic Beanstalk configura el clon con la configuración del entorno utilizada por el entorno original. Al clonar un entorno existente en lugar de crear uno nuevo, no tiene que configurar manualmente los ajustes de las opciones, las variables de entorno y otros ajustes que haya creado con el servicio Elastic Beanstalk. Elastic Beanstalk también crea una copia AWS de cualquier recurso asociado al entorno original.

Es importante tener en cuenta las siguientes situaciones:

- Durante el proceso de clonación, Elastic Beanstalk no copia los datos de Amazon RDS al clon.
- Elastic Beanstalk no incluye ningún cambio sin administrar en los recursos del clon. Los cambios en los recursos de AWS que realiza con herramientas diferentes de la consola de Elastic Beanstalk, las herramientas de línea de comandos o la API se consideran cambios sin administrar.
- Los grupos de seguridad para la entrada se consideran cambios no gestionados. Los entornos clonados de Elastic Beanstalk no transfieren los grupos de seguridad para su entrada, lo que deja el entorno abierto a todo el tráfico de Internet. Deberá restablecer los grupos de seguridad de entrada para el entorno clonado.

Solo puede clonar un entorno en una versión de plataforma diferente de la misma ramificación de plataforma. No se garantiza que una ramificación de la plataforma diferente sea compatible. Para usar una ramificación de la plataforma diferente, debe crear manualmente un nuevo entorno, implementar el código de la aplicación y realizar los cambios necesarios en el código y las opciones para asegurarse de que la aplicación funcione correctamente en la nueva ramificación de la plataforma.

AWS consola de administración

Important

Los entornos clonados de Elastic Beanstalk no transfieren los grupos de seguridad para su entrada, lo que deja el entorno abierto a todo el tráfico de Internet. Deberá restablecer los grupos de seguridad de entrada para el entorno clonado.

Para ver los recursos que no se pueden clonar, compruebe el estado de desviación de la configuración de su entorno. Para obtener más información, consulte [Detectar desviaciones en una CloudFormation pila completa](#) en la Guía del AWS CloudFormation usuario.

Para clonar un entorno

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En la página de información general del entorno, seleccione Acciones.
4. Seleccione Clonar entorno.
5. En la página Clone Environment (Clonar entorno), revise la información de la sección Original Environment (Entorno original) para verificar que ha elegido el entorno del que desea crear un clon.
6. En la sección Nuevo entorno, puede cambiar opcionalmente los valores Nombre de entorno, URL de entorno, Descripción, Versión de la plataforma y Rol de servicio que Elastic Beanstalk define automáticamente en función del entorno original.

Note

Si la versión de la plataforma utilizada en el entorno original no es la recomendada para su uso en la ramificación de la plataforma, se le advierte que se recomienda una versión de plataforma diferente. Elija Platform version (Versión de la plataforma) y podrá ver la versión de plataforma recomendada en la lista, por ejemplo, 3.3.2 (Recommended) (3.3.2 [Recomendado]).

7. Cuando esté preparado, elija Clone (Clonar).

Interfaz de línea de comandos de Elastic Beanstalk (CLI de EB)

Important

Los entornos clonados de Elastic Beanstalk no transfieren los grupos de seguridad para su entrada, lo que deja el entorno abierto a todo el tráfico de Internet. Deberá restablecer los grupos de seguridad de entrada para el entorno clonado.

Para ver los recursos que no se pueden clonar, compruebe el estado de desviación de la configuración de su entorno. Para obtener más información, consulte [Detectar desviaciones en una CloudFormation pila completa](#) en la Guía del AWS CloudFormation usuario.

Utilice el comando `eb clone` para clonar un entorno en ejecución, como se indica a continuación.

```
~/workspace/my-app$ eb clone my-env1
Enter name for Environment Clone
(default is my-env1-clone): my-env2
Enter DNS CNAME prefix
(default is my-env1-clone): my-env2
```

Puede especificar el nombre del entorno de origen en el comando `clone` o no especificarlo para que se clone el entorno predeterminado de la carpeta del proyecto actual. La CLI de EB le pide que introduzca un nombre y un prefijo DNS para el nuevo entorno.

De forma predeterminada, `eb clone` crea el nuevo entorno con la última versión disponible de la plataforma del entorno de origen. Para forzar a la CLI de EB a que use la misma versión, aunque haya una versión más reciente disponible, use la opción `--exact`.

```
~/workspace/my-app$ eb clone --exact
```

Para obtener más información acerca de este comando, consulte [eb clone](#).

Terminación de un entorno de Elastic Beanstalk

Puede terminar un entorno de AWS Elastic Beanstalk en ejecución con la consola de Elastic Beanstalk. Al hacer esto, evitará incurrir en cargos por recursos de AWS que no utilice.

Note

Siempre podrá lanzar más adelante un nuevo entorno con la misma versión.

Si tiene datos de un entorno que desea conservar, establezca la política de eliminación de base de datos en `Retain` antes de terminar el entorno. Esto mantiene la base de datos en funcionamiento fuera de Elastic Beanstalk. Después de esto, cualquier entorno de Elastic Beanstalk debe conectarse a él como una base de datos externa. Si desea hacer una copia de seguridad de los datos sin mantener la base de datos en funcionamiento, configure la política de eliminación para que tome una instantánea de la base de datos antes de terminar el entorno. Para obtener más información, consulte [Ciclo de vida de la base de datos](#) en el capítulo Configuración de entornos de esta guía.

Elastic Beanstalk podría no terminar el entorno correctamente. Un motivo habitual es que el grupo de seguridad de otro entorno tenga una dependencia del grupo de seguridad del entorno que se intenta terminar. Para obtener instrucciones sobre como evitar este problema, consulte [Grupos de seguridad](#) en página Instancias EC2 de esta guía.

Important

Si termina un entorno, también debe eliminar todas las asignaciones de CNAME que ha creado, ya que otros clientes pueden reutilizar un nombre de host disponible. Asegúrese de eliminar los registros de DNS que apuntan a su entorno terminado para evitar que se cree una entrada de DNS colgante. Una entrada de DNS colgante puede exponer el tráfico de Internet destinado a su dominio a vulnerabilidades de seguridad. También puede presentar otros riesgos.

Para más información, consulte [Protección contra registros de delegación colgantes en Route 53](#) en la Guía para desarrolladores de Amazon Route 53. También puede obtener más

información sobre las entradas de DNS colgantes en [Protecciones de dominio mejoradas para solicitudes de Amazon CloudFront](#) en el Blog de seguridad de AWS.

Consola de Elastic Beanstalk

Para terminar un entorno

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. Elija Actions (Acciones) y, a continuación, Terminate Environment (Terminar el entorno).
4. Utilice el cuadro de diálogo en pantalla para confirmar la terminación del entorno.

Note

Cuando termine su entorno, el CNAME asociado con el entorno terminado se libera y queda a disposición de cualquier persona que desee utilizarlo.

Elastic Beanstalk tarda unos minutos en terminar los recursos de AWS que se ejecutan en el entorno.

AWS CLI

Para terminar un entorno

- Ejecute el siguiente comando.

```
$ aws elasticbeanstalk terminate-environment --environment-name my-env
```


API

Para terminar un entorno

- Llame a `TerminateEnvironment` con el siguiente parámetro:

`EnvironmentName = SampleAppEnv`

```
https://elasticbeanstalk.us-west-2.amazon.com/?EnvironmentName=SampleAppEnv
&Operation=TerminateEnvironment
&AuthParams
```

Creación de entornos de Elastic Beanstalk con la CLI de AWS

[Para obtener más información sobre los AWS CLI comandos de Elastic Beanstalk, consulte la Referencia de comandos.AWS CLI](#)

1. Compruebe si el CNAME del entorno está disponible.

```
$ aws elasticbeanstalk check-dns-availability --cname-prefix my-cname
{
  "Available": true,
  "FullyQualifiedCNAME": "my-cname.elasticbeanstalk.com"
}
```

2. Asegúrese de que la versión de la aplicación existe.

```
$ aws elasticbeanstalk describe-application-versions --application-name my-app --
version-label v1
```

Si no tiene una versión de la aplicación para el origen aún, créelo. Por ejemplo, el siguiente comando crea una versión de la aplicación de un paquete de código fuente en Amazon Simple Storage Service (Amazon S3).

```
$ aws elasticbeanstalk create-application-version --application-name my-app --
version-label v1 --source-bundle S3Bucket=DOC-EXAMPLE-BUCKET,S3Key=my-source-
bundle.zip
```

3. Cree una plantilla de configuración para la aplicación.

```
$ aws elasticbeanstalk create-configuration-template --application-name my-app --  
template-name v1 --solution-stack-name "64bit Amazon Linux 2015.03 v2.0.0 running  
Ruby 2.2 (Passenger Standalone)"
```

4. Cree el entorno.

```
$ aws elasticbeanstalk create-environment --cname-prefix my-cname --application-  
name my-app --template-name v1 --version-label v1 --environment-name v1clone --  
option-settings file://options.txt
```

Los valores de las opciones se definen en el archivo options.txt:

```
[  
  {  
    "Namespace": "aws:autoscaling:launchconfiguration",  
    "OptionName": "IamInstanceProfile",  
    "Value": "aws-elasticbeanstalk-ec2-role"  
  }  
]
```

El valor de la opción anterior define el perfil de instancia de IAM. Puede especificar el ARN o el nombre de perfil.

5. Determine si el nuevo entorno se muestra como Green and Ready.

```
$ aws elasticbeanstalk describe-environments --environment-names my-env
```

Si el nuevo entorno no se muestra como Green and Ready, debe decidir si desea reintentar la operación o dejar el entorno en su estado actual para su investigación. Asegúrese de terminar el entorno una vez que haya finalizado y de eliminar los recursos sin utilizar.

Note

Puede ajustar el periodo de tiempo de espera si el entorno no se lanza en un plazo razonable.

Creación de entornos de Elastic Beanstalk con la API

1. Llame a `CheckDNSAvailability` con el siguiente parámetro:

- `CNAMEPrefix = SampleApp`

Example

```
https://elasticbeanstalk.us-east-2.amazonaws.com/?CNAMEPrefix=sampleapplication
&Operation=CheckDNSAvailability
&AuthParams
```

2. Llame a `DescribeApplicationVersions` con los siguientes parámetros:

- `ApplicationName = SampleApp`
- `VersionLabel = Version2`

Example

```
https://elasticbeanstalk.us-east-2.amazonaws.com/?ApplicationName=SampleApp
&VersionLabel=Version2
&Operation=DescribeApplicationVersions
&AuthParams
```

3. Llame a `CreateConfigurationTemplate` con los siguientes parámetros:

- `ApplicationName = SampleApp`
- `TemplateName = MyConfigTemplate`
- `SolutionStackName = 64bit%20Amazon%20Linux%202015.03%20v2.0.0%20running%20Ruby%202.2%20(Passenger%20Standalone)`

Example

```
https://elasticbeanstalk.us-east-2.amazonaws.com/?ApplicationName=SampleApp
&TemplateName=MyConfigTemplate
&Operation=CreateConfigurationTemplate
&SolutionStackName=64bit%20Amazon%20Linux%202015.03%20v2.0.0%20running%20Ruby
%202.2%20(Passenger%20Standalone)
```

```
&AuthParams
```

4. Llame a `CreateEnvironment` con uno de los siguientes conjuntos de parámetros.
 - a. Utilice lo siguiente para una capa de entorno de servidor web:
 - `EnvironmentName = SampleAppEnv2`
 - `VersionLabel = Version2`
 - `Description = description`
 - `TemplateName = MyConfigTemplate`
 - `ApplicationName = SampleApp`
 - `CNAMEPrefix = sampleapplication`
 - `OptionSettings.member.1.Namespace = aws:autoscaling:launchconfiguration`
 - `OptionSettings.member.1.OptionName = IamInstanceProfile`
 - `OptionSettings.member.1.Value = aws-elasticbeanstalk-ec2-role`

Example

```
https://elasticbeanstalk.us-east-2.amazonaws.com/?ApplicationName=SampleApp
&VersionLabel=Version2
&EnvironmentName=SampleAppEnv2
&TemplateName=MyConfigTemplate
&CNAMEPrefix=sampleapplication
&Description=description
&Operation=CreateEnvironment
&OptionSettings.member.1.Namespace=aws%3Aautoscaling%3Alaunchconfiguration
&OptionSettings.member.1.OptionName=IamInstanceProfile
&OptionSettings.member.1.Value=aws-elasticbeanstalk-ec2-role
&AuthParams
```

- b. Utilice lo siguiente para una capa de entorno de trabajo:
 - `EnvironmentName = SampleAppEnv2`
 - `VersionLabel = Version2`
 - `Description = description`
 - `TemplateName = MyConfigTemplate`

- `ApplicationName = SampleApp`
- `Tier = Worker`
- `OptionSettings.member.1.Namespace = aws:autoscaling:launchconfiguration`
- `OptionSettings.member.1.OptionName = IamInstanceProfile`
- `OptionSettings.member.1.Value = aws-elasticbeanstalk-ec2-role`
- `OptionSettings.member.2.Namespace = aws:elasticbeanstalk:sqs`
- `OptionSettings.member.2.OptionName = WorkerQueueURL`
- `OptionSettings.member.2.Value = sqs.elasticbeanstalk.us-east-2.amazonaws.com`
- `OptionSettings.member.3.Namespace = aws:elasticbeanstalk:sqs`
- `OptionSettings.member.3.OptionName = HttpPath`
- `OptionSettings.member.3.Value = /`
- `OptionSettings.member.4.Namespace = aws:elasticbeanstalk:sqs`
- `OptionSettings.member.4.OptionName = MimeType`
- `OptionSettings.member.4.Value = application/json`
- `OptionSettings.member.5.Namespace = aws:elasticbeanstalk:sqs`
- `OptionSettings.member.5.OptionName = HttpConnections`
- `OptionSettings.member.5.Value = 75`
- `OptionSettings.member.6.Namespace = aws:elasticbeanstalk:sqs`
- `OptionSettings.member.6.OptionName = ConnectTimeout`
- `OptionSettings.member.6.Value = 10`
- `OptionSettings.member.7.Namespace = aws:elasticbeanstalk:sqs`
- `OptionSettings.member.7.OptionName = InactivityTimeout`
- `OptionSettings.member.7.Value = 10`
- `OptionSettings.member.8.Namespace = aws:elasticbeanstalk:sqs`
- `OptionSettings.member.8.OptionName = VisibilityTimeout`
- `OptionSettings.member.8.Value = 60`
- `OptionSettings.member.9.Namespace = aws:elasticbeanstalk:sqs`
- `OptionSettings.member.9.OptionName = RetentionPeriod`

- `OptionSettings.member.9.Value = 345600`

Example

```
https://elasticbeanstalk.us-east-2.amazonaws.com/?ApplicationName=SampleApp
&VersionLabel=Version2
&EnvironmentName=SampleAppEnv2
&TemplateName=MyConfigTemplate
&Description=description
&Tier=Worker
&Operation=CreateEnvironment
&OptionSettings.member.1.Namespace=aws%3Aautoscaling%3Alaunchconfiguration
&OptionSettings.member.1.OptionName=IamInstanceProfile
&OptionSettings.member.1.Value=aws-elasticbeanstalk-ec2-role
&OptionSettings.member.2.Namespace=aws%3Aelasticbeanstalk%3Asqs
&OptionSettings.member.2.OptionName=WorkerQueueURL
&OptionSettings.member.2.Value=sqs.elasticbeanstalk.us-east-2.amazonaws.com
&OptionSettings.member.3.Namespace=aws%3elasticbeanstalk%3sqs
&OptionSettings.member.3.OptionName=HttpPath
&OptionSettings.member.3.Value=%2F
&OptionSettings.member.4.Namespace=aws%3Aelasticbeanstalk%3Asqs
&OptionSettings.member.4.OptionName=MimeType
&OptionSettings.member.4.Value=application%2Fjson
&OptionSettings.member.5.Namespace=aws%3Aelasticbeanstalk%3Asqs
&OptionSettings.member.5.OptionName=HttpConnections
&OptionSettings.member.5.Value=75
&OptionSettings.member.6.Namespace=aws%3Aelasticbeanstalk%3Asqs
&OptionSettings.member.6.OptionName=ConnectTimeout
&OptionSettings.member.6.Value=10
&OptionSettings.member.7.Namespace=aws%3Aelasticbeanstalk%3Asqs
&OptionSettings.member.7.OptionName=InactivityTimeout
&OptionSettings.member.7.Value=10
&OptionSettings.member.8.Namespace=aws%3Aelasticbeanstalk%3Asqs
&OptionSettings.member.8.OptionName=VisibilityTimeout
&OptionSettings.member.8.Value=60
&OptionSettings.member.9.Namespace=aws%3Aelasticbeanstalk%3Asqs
&OptionSettings.member.9.OptionName=RetentionPeriod
&OptionSettings.member.9.Value=345600
&AuthParams
```

Construcción de una URL de lanzamiento inmediato

Puede crear una URL personalizada para que cualquier persona pueda implementar y ejecutar rápidamente una aplicación web predeterminada en AWS Elastic Beanstalk. Esta dirección URL se denomina "URL de lanzamiento inmediato". Es posible que necesite una URL de lanzamiento inmediato, por ejemplo, para mostrar una aplicación web que está diseñada para ejecutarse en Elastic Beanstalk. Con las URL de lanzamiento inmediato, puede utilizar parámetros para proporcionar por adelantado la información necesaria al asistente de creación de la aplicación. Luego de agregar esta información al asistente, en unos pocos pasos, cualquier persona puede utilizar el enlace de la URL para lanzar un entorno de Elastic Beanstalk con el código fuente de la aplicación web. Esto significa que los usuarios no tienen que cargar o especificar manualmente la ubicación del paquete de origen de la aplicación. Tampoco tienen que proporcionar ninguna otra información al asistente.

Las URL de lanzamiento inmediato proporcionan a Elastic Beanstalk la información mínima necesaria para crear una aplicación: el nombre de la aplicación, la pila de soluciones, el tipo de instancia y el tipo de entorno. Elastic Beanstalk utiliza los valores predeterminados de otros detalles de configuración que no se especifican explícitamente en la URL de lanzamiento inmediato personalizada.

Las URL de lanzamiento inmediato utilizan la sintaxis estándar de las URL. Para obtener más información, consulte [RFC 3986 - Uniform Resource Identifier \(URI\): Generic Syntax](#).

Parámetros de la URL

La URL debe incluir los siguientes parámetros, que distinguen entre mayúsculas y minúsculas:

- **región:** especifique una AWS región. Para obtener una lista de las regiones que admite Elastic Beanstalk, consulte [Puntos de enlace y cuotas de AWS Elastic Beanstalk](#) en Referencia general de AWS.
- **applicationName** (Nombre de la aplicación): especifique el nombre de la aplicación. Elastic Beanstalk muestra el nombre de la aplicación en la consola de Elastic Beanstalk para distinguirlo de otras aplicaciones. De forma predeterminada, el nombre de aplicación también sirve de base para el nombre del entorno y la URL del entorno.
- **platform** (plataforma): especifique la versión de la plataforma que se va a utilizar para el entorno. Utilice uno de los siguientes métodos; a continuación, codifique como URL su elección:
 - Especifique un ARN de plataforma sin una versión. Elastic Beanstalk selecciona la última versión de la plataforma de la versión principal de la plataforma correspondiente. Por ejemplo, para

seleccionar la última versión de la plataforma Python 3.6, especifique `Python 3.6 running on 64bit Amazon Linux`.

- Especifique el nombre de la plataforma. Elastic Beanstalk selecciona la última versión del tiempo de ejecución del lenguaje más reciente de la plataforma (por ejemplo: Python).

Para obtener una descripción de todas las plataformas disponibles y sus versiones, consulte [Plataformas compatibles con Elastic Beanstalk](#).

Puede utilizar la [AWS Command Line Interface](#) (AWS CLI) para obtener una lista de todas las versiones de plataforma disponibles con sus respectivos ARN. El comando `list-platform-versions` muestra información detallada sobre todas las versiones de plataforma disponibles. Utilice el argumento `--filters` para reducir la lista. Por ejemplo, puede examinar la lista para mostrar únicamente las versiones de plataforma de un lenguaje específico.

En el siguiente ejemplo se consultan todas las versiones de plataforma de Python y envía el resultado a través de una serie de comandos. El resultado es una lista de los ARN de versiones de plataforma (sin la cola `/version`), en un formato legible para los usuarios, sin codificación URL.

```
$ aws elasticbeanstalk list-platform-versions --filters
  'Type="PlatformName",Operator="contains",Values="Python"' | grep PlatformArn | awk -
-F '""' '{print $4}' | awk -F '/' '{print $2}'
Preconfigured Docker - Python 3.4 running on 64bit Debian
Preconfigured Docker - Python 3.4 running on 64bit Debian
Python 2.6 running on 32bit Amazon Linux
Python 2.6 running on 32bit Amazon Linux 2014.03
...
Python 3.6 running on 64bit Amazon Linux
```

En el siguiente ejemplo se agrega un comando Perl al último ejemplo, para codificar la salida como URL.

```
$ aws elasticbeanstalk list-platform-versions --filters
  'Type="PlatformName",Operator="contains",Values="Python"' | grep PlatformArn | awk
-F '""' '{print $4}' | awk -F '/' '{print $2}' | perl -MURI::Escape -ne 'chomp;print
uri_escape($_), "\n"'
Preconfigured%20Docker%20-%20Python%203.4%20running%20on%2064bit%20Debian
Preconfigured%20Docker%20-%20Python%203.4%20running%20on%2064bit%20Debian
Python%202.6%20running%20on%2032bit%20Amazon%20Linux
Python%202.6%20running%20on%2032bit%20Amazon%20Linux%202014.03
...
```



```
Python%203.6%20running%20on%2064bit%20Amazon%20Linux
```

Las URL de lanzamiento inmediato pueden contener los siguientes parámetros. Si no incluye los parámetros opcionales en la URL de lanzamiento inmediato, Elastic Beanstalk utiliza los valores predeterminados para crear y ejecutar la aplicación. Si no incluye el `sourceBundleUrl` parámetro, Elastic Beanstalk utiliza la aplicación de ejemplo predeterminada para la plataforma especificada.

- `sourceBundleUrl`— Especifique la ubicación del paquete fuente de la aplicación web en formato URL. Por ejemplo, si ha cargado el paquete de código fuente en un bucket de Amazon S3, puede especificar el valor del `sourceBundleUrl` parámetro como `https://mybucket.s3.amazonaws.com/myobject`.

Note

Puede especificar el valor del `sourceBundleUrl` parámetro como una URL HTTP, pero el navegador web del usuario convertirá los caracteres según sea necesario aplicando la codificación HTML de la URL.

- `environmentType`: especifique si el entorno es escalable y tiene balanceo de carga o es solo una instancia. Para obtener más información, consulte [Tipos de entornos](#). Puede especificar `LoadBalancing` o `SingleInstance` como valor del parámetro.
- `tierName`: especifique si el entorno es compatible con una aplicación web que procese solicitudes web o que ejecute trabajos en segundo plano. Para obtener más información, consulte [Entornos de trabajo de Elastic Beanstalk](#). Puede especificar `WebServer` o `Worker`,
- `instanceType`: especifique un servidor con las características más adecuadas para su aplicación (entre ellas, el tamaño de la memoria y la potencia de la CPU). Para obtener más información sobre las familias y los tipos de instancias de Amazon EC2, consulte Tipos de [instancias](#) en la Guía del usuario de Amazon EC2 o [Tipos de instancias](#) en la Guía del usuario de Amazon EC2. Para obtener más información sobre los tipos de instancias disponibles en todas las regiones, consulte [Tipos de instancias disponibles](#) en la Guía del usuario de Amazon EC2 o [Tipos de instancias disponibles](#) en la Guía del usuario de Amazon EC2.
- `withVpc`: especifique si va a crear el entorno en una Amazon VPC. Puede especificar `true` o `false`. Para obtener más información sobre el uso de Elastic Beanstalk con Amazon VPC, consulte [Uso de Elastic Beanstalk con Amazon VPC](#).

- `withRds`: especifique si va a crear una instancia de base de datos de Amazon RDS con este entorno. Para obtener más información, consulte [Uso de Elastic Beanstalk con Amazon RDS](#). Puede especificar `true` o `false`.
- `rdsDBEngine`: especifique el motor de base de datos que desee usar con las instancias de Amazon EC2 de este entorno. Puede especificar `mysql`, `oracle-se1`, `sqlserver-ex`, `sqlserver-web` o `sqlserver-se`. El valor predeterminado es `mysql`.
- `RDSdb AllocatedStorage`: especifique el tamaño de almacenamiento de la base de datos asignado en gigabytes (GB). Puede especificar los valores siguientes:
 - MySQL: de 5 a 1024. El valor predeterminado es 5.
 - Oracle: de 10 a 1024. El valor predeterminado es 10.
 - Microsoft SQL Server Express Edition – 30.
 - Microsoft SQL Server Web Edition – 30.
 - Microsoft SQL Server Standard Edition – 200.
- `RdsDbInstanceClass`: especifique el tipo de instancia de base de datos. El valor predeterminado es `db.t2.micro` (`db.m1.large` para un entorno que no se ejecuta en una Amazon VPC). Para ver una lista de las clases de instancia de base de datos compatibles con Amazon RDS, consulte [Clase de instancia de base de datos](#) en la Guía del usuario de Amazon Relational Database Service.
- `rdsMultiAZDatabase`: especifique si Elastic Beanstalk debe crear la instancia de base de datos en varias zonas de disponibilidad. Puede especificar `true` o `false`. Para obtener más información sobre las implementaciones en varias zonas de disponibilidad con Amazon RDS, consulte [Regiones y zonas de disponibilidad](#) en la Guía del usuario de Amazon Relational Database Service.
- `RdsDb DeletionPolicy`: especifique si desea eliminar o hacer una instantánea de la instancia de base de datos al finalizar el entorno. Puede especificar `Delete` o `Snapshot`.

Ejemplo

A continuación, se muestra un ejemplo de una URL de lanzamiento inmediato. Después de crear su propia URL, puede enviársela a los usuarios. Por ejemplo, puede incrustar la URL en una página web o en el material de formación. Si los usuarios crean una aplicación utilizando la URL de lanzamiento inmediato, el asistente de creación de aplicaciones de Elastic Beanstalk no necesitará ningún otro dato.

```
https://console.aws.amazon.com/elasticbeanstalk/home?region=us-west-2#/
newApplication?applicationName=YourCompanySampleApp&platform=PHP%207.3%20running
%20on%2064bit%20Amazon%20Linux&sourceBundleUrl=http://s3.amazonaws.com/mybucket/
myobject&environmentType=SingleInstance&tierName=WebServer&instanceType=m1.small&withVpc=true&
```

Cuando los usuarios elijan una URL de lanzamiento inmediato, Elastic Beanstalk mostrará una página similar a la siguiente.



Create a web app

Create a new application and environment with a sample application or your own code. By creating an environment, you allow AWS Elastic Beanstalk to manage AWS resources and permissions on your behalf. [Learn more](#)

Application information

Application name

Up to 100 Unicode characters, not including forward slash (/).

Environment information

Choose the name, subdomain, and description for your environment. These cannot be changed later.

Environment name

Domain

Description

Base configuration

Tier Web Server ([Choose tier](#))

Platform Preconfigured platform

Platforms published and maintained by AWS Elastic Beanstalk.

Custom platform ^{NEW}

Platforms created and owned by you. [Learn more](#)

Application code Sample application

Get started right away with sample code.

Upload your code

Upload a source bundle from your computer or copy one from Amazon S3.

ZIP or WAR

Para utilizar la URL de lanzamiento inmediato

1. Elija la URL de lanzamiento inmediato.
2. Luego de que se abra la consola de Elastic Beanstalk, en la página Create a web app (Crear una aplicación web), elija Review and launch (Revisar y lanzar) para ver la configuración que Elastic Beanstalk utiliza para crear la aplicación y lanzar el entorno en el que se ejecuta.
3. En la página Configure (Configurar), elija Create app (Crear aplicación) para crear la aplicación.

Creación y actualización de grupos de entornos de Elastic Beanstalk

Con la AWS Elastic Beanstalk Compose Environments API, puede crear y actualizar grupos de entornos de Elastic Beanstalk en una sola aplicación. Cada entorno del grupo puede ejecutar un componente independiente de una aplicación que tenga una arquitectura orientada a servicios. La API Compose Environments toma una lista de las versiones de la aplicación y un nombre de grupo opcional. Elastic Beanstalk crea un entorno para cada versión de la aplicación o, si los entornos ya existen, implementa las versiones de la aplicación en ellos.

Cree enlaces entre los entornos de Elastic Beanstalk para definir un entorno como una dependencia de otro. Cuando se genera un grupo de entornos con la API Compose Environments, Elastic Beanstalk solamente crea entornos dependientes una vez que sus dependencias están listas y en ejecución. Para obtener más información acerca de los enlaces del entorno, consulte [Creación de enlaces entre entornos de Elastic Beanstalk](#).

La API Compose Environments utiliza un [manifiesto de entorno](#) para guardar los detalles de la configuración que comparten los grupos de entornos. Cada aplicación integrante debe tener un archivo de configuración `env.yaml` en el paquete de código fuente que especifique los parámetros utilizados para crear el entorno.

Compose Environments requiere que se especifique `EnvironmentName` y `SolutionStack` en el manifiesto del entorno de cada aplicación integrante.

Puede usar la Compose Environments API con la interfaz de línea de comandos de Elastic Beanstalk (EB CLI) AWS CLI, el o un SDK. Consulte [Administración de múltiples entornos Elastic Beanstalk como un grupo con la CLI de EB](#) para obtener instrucciones sobre la CLI de EB.

Uso de la API **Compose Environments**

Por ejemplo, puede crear una aplicación llamada `Media Library` que permita a los usuarios cargar y administrar las imágenes y los vídeos guardados en Amazon Simple Storage Service (Amazon

S3). La aplicación tendrá un entorno front-end, `front`, donde se ejecutará una aplicación web que permitirá a los usuarios cargar y descargar archivos individuales, consultar la biblioteca e iniciar trabajos de procesamiento por lotes.

En lugar de procesar los trabajos directamente, la aplicación front-end los agregará a una cola de Amazon SQS. El segundo entorno, `worker`, extraerá los trabajos de la cola y los procesará. `worker` utilizará un tipo de instancia G2 que tenga una GPU de alto rendimiento, mientras que `front` puede ejecutarse en un tipo de instancia genérica más rentable.

La carpeta del proyecto, `Media Library`, debe organizarse en diferentes directorios para cada componente, donde cada directorio tendrá un archivo de definición del entorno (`env.yaml`) con su código fuente:

```
~/workspace/media-library
|-- front
|   `-- env.yaml
`-- worker
     `-- env.yaml
```

A continuación se muestra el archivo `env.yaml` de cada aplicación integrante.

~/workspace/media-library/front/env.yaml

```
EnvironmentName: front+
EnvironmentLinks:
  "WORKERQUEUE" : "worker+"
AWSConfigurationTemplateVersion: 1.1.0.0
EnvironmentTier:
  Name: WebServer
  Type: Standard
SolutionStack: 64bit Amazon Linux 2015.09 v2.0.4 running Java 8
OptionSettings:
  aws:autoscaling:launchconfiguration:
    InstanceType: m4.large
```

~/workspace/media-library/worker/env.yaml

```
EnvironmentName: worker+
AWSConfigurationTemplateVersion: 1.1.0.0
EnvironmentTier:
  Name: Worker
```

```
Type: SQS/HTTP
SolutionStack: 64bit Amazon Linux 2015.09 v2.0.4 running Java 8
OptionSettings:
  aws:autoscaling:launchconfiguration:
    InstanceType: g2.2xlarge
```

Después de [crear una versión de la aplicación](#) para los componentes de aplicación front-end (front-v1) y de trabajo (worker-v1), llame a la API Compose Environments con los nombres de las versiones. En este ejemplo, utilizamos la AWS CLI para llamar a la API.

```
# Create application versions for each component:
~$ aws elasticbeanstalk create-application-version --application-name media-
library --version-label front-v1 --process --source-bundle S3Bucket="DOC-EXAMPLE-
BUCKET",S3Key="front-v1.zip"
{
  "ApplicationVersion": {
    "ApplicationName": "media-library",
    "VersionLabel": "front-v1",
    "Description": "",
    "DateCreated": "2015-11-03T23:01:25.412Z",
    "DateUpdated": "2015-11-03T23:01:25.412Z",
    "SourceBundle": {
      "S3Bucket": "DOC-EXAMPLE-BUCKET",
      "S3Key": "front-v1.zip"
    }
  }
}
~$ aws elasticbeanstalk create-application-version --application-name media-
library --version-label worker-v1 --process --source-bundle S3Bucket="DOC-EXAMPLE-
BUCKET",S3Key="worker-v1.zip"
{
  "ApplicationVersion": {
    "ApplicationName": "media-library",
    "VersionLabel": "worker-v1",
    "Description": "",
    "DateCreated": "2015-11-03T23:01:48.151Z",
    "DateUpdated": "2015-11-03T23:01:48.151Z",
    "SourceBundle": {
      "S3Bucket": "DOC-EXAMPLE-BUCKET",
      "S3Key": "worker-v1.zip"
    }
  }
}
```

```
# Create environments:
~$ aws elasticbeanstalk compose-environments --application-name media-library --group-name dev --version-labels front-v1 worker-v1
```

La tercera llamada crea dos entornos: `front-dev` y `worker-dev`. La API crea los nombres de los entornos concatenando el valor de `EnvironmentName` especificado en el archivo `env.yaml` con la opción `group name` especificada en la llamada a `Compose Environments`, separados por un guion. La longitud total de estas dos opciones y el guion no debe superar el máximo permitido para el nombre del entorno, que es de 23 caracteres.

La aplicación que se ejecuta en el entorno `front-dev` puede obtener acceso al nombre de la cola de Amazon SQS asociada al entorno `worker-dev` a través de la variable `WORKERQUEUE`. Para obtener más información acerca de los enlaces del entorno, consulte [Creación de enlaces entre entornos de Elastic Beanstalk](#).

Implementación de aplicaciones en entornos Elastic Beanstalk

Puede utilizar la consola de AWS Elastic Beanstalk con el objetivo de cargar un paquete de [código fuente](#) actualizado e implementarlo en el entorno de Elastic Beanstalk o para volver a implementar una versión que se cargó anteriormente.

Cada implementación se identifica a través de un ID de implementación. Los ID de implementación empiezan a partir de 1 y van aumentando en una unidad con cada implementación y cada cambio de configuración de la instancia. Si habilita los [informes de estado avanzados](#), Elastic Beanstalk muestra el ID de implementación en la [consola de estado](#) y en la [CLI de EB](#) cuando notifica el estado de la instancia. El ID de implementación le ayuda a determinar el estado del entorno cuando se produce un error en una actualización continua.

Elastic Beanstalk proporciona varias políticas y configuraciones de implementación. Para obtener información detallada sobre la configuración de una política y opciones adicionales, consulte [the section called "Opciones de implementación"](#). En la tabla siguiente se enumeran las políticas y los tipos de entornos que las admiten.

Políticas de implementación admitidas

Política de implementación	Entornos con balanceo de carga	Entornos de una sola instancia	Entornos de Windows Server heredados†
Todo a la vez	✓ Sí	✓ Sí	✓ Sí

Política de implementación	Entornos con balanceo de carga	Entornos de una sola instancia	Entornos de Windows Server heredados†
Continua	✓ Sí	× No	✓ Sí
Continua con un lote adicional	✓ Sí	× No	× No
Inmutable	✓ Sí	✓ Sí	× No
División de tráfico	✓ Sí (Balanceador de carga de aplicaciones).	× No	× No

†En esta tabla, un entorno Windows Server heredado es un entorno basado en una [configuración de plataforma de Windows Server](#) que utiliza una versión de IIS anterior a la IIS 8.5.

Warning

Algunas políticas reemplazan todas las instancias durante la implementación o actualización. Esto provoca la pérdida de todos los [saldos de ráfagas de Amazon EC2](#) acumulados. Sucede en los siguientes casos:

- Actualizaciones de plataforma administradas con reemplazo de instancias habilitado
- Actualizaciones inmutables
- Implementaciones con actualizaciones inmutables o división de tráfico habilitada

Elección de una política de implementación

Elegir la política de implementación adecuada para su aplicación es un balance de algunas consideraciones, y depende de sus necesidades particulares. La página [the section called “Opciones de implementación”](#) tiene más información sobre cada política y una descripción detallada del funcionamiento de algunas de ellas.



En la lista siguiente se proporciona información resumida sobre las diferentes políticas de implementación y se agregan consideraciones relacionadas.

- **Todo a la vez:** el método de implementación más rápido. Adecuado si puede aceptar una breve pérdida de servicio y si las implementaciones rápidas son importantes para usted. Con este método, Elastic Beanstalk implementa la nueva versión de la aplicación en cada instancia. A continuación, es posible que el proxy web o el servidor de aplicaciones necesite reiniciar. Como resultado, es posible que la aplicación no esté disponible para los usuarios (o tenga poca disponibilidad) durante un corto período de tiempo.
- **Continua:** evita el tiempo de inactividad y minimiza la disponibilidad reducida, a un costo de más tiempo de implementación. Adecuado si usted no puede aceptar cualquier período de servicio perdido por completo. Con este método, la aplicación se implementa en el entorno una serie de instancias a la vez. La mayor parte del ancho de banda se conserva durante toda la implementación.
- **Continua con lote adicional:** evita cualquier disponibilidad reducida, a un costo de un tiempo de implementación aún mayor en comparación con el método Continua. Adecuado si debe mantener el mismo ancho de banda durante toda la implementación. Con este método, Elastic Beanstalk inicia un lote adicional de instancias y, a continuación, realiza una implementación sucesiva. El inicio del lote adicional lleva tiempo y garantiza que se conserve el mismo ancho de banda durante toda la implementación.
- **Inmutable:** método de implementación más lento, que garantiza que la nueva versión de la aplicación siempre se implementa en instancias nuevas, en lugar de actualizar instancias existentes. También tiene la ventaja adicional de una restauración rápida y segura en caso de que falle la implementación. Con este método, Elastic Beanstalk realiza una [actualización inmutable](#) para implementar la aplicación. En una actualización inmutable, se lanza un segundo grupo de Auto Scaling en el entorno y la nueva versión administra el tráfico junto con la versión antigua hasta que las nuevas instancias superan las comprobaciones de estado.
- **División de tráfico:** método de implementación de pruebas de canary. Adecuado si desea probar el estado de la nueva versión de la aplicación utilizando una parte del tráfico entrante, mientras mantiene el resto del tráfico servido por la versión anterior de la aplicación.

En la tabla siguiente se comparan las propiedades de los métodos de implementación.

Métodos de implementación

Método	Repercusión de la implementación con errores	Tiempo de implementación	Sin inactividad	Sin cambios de DNS	Proceso de restauración	El código se implementa en
Todo a la vez	Tiempo de inactividad.	⊕	X No	✓ Sí	Nueva implementación manual	Instancias existentes
Continua	Un lote fuera de servicio y los lotes implementados correctamente antes del error ejecutan la nueva versión de la aplicación.	⊕	☹ ✓ Sí	✓ Sí	Nueva implementación manual	Instancias existentes
Continua con un lote adicional	Mínima si se produce un error en el primer lote; de lo contrario, similar a Rolling (Continua).	⊕	☹ ✓ Sí	✓ Sí	Nueva implementación manual	Instancias nuevas y existentes
Inmutables	Mínima	⊕	☹ ✓ Sí	✓ Sí	Terminar nuevas instancias	Instancias nuevas
División de tráfico	Porcentaje de tráfico del cliente dirigido a la nueva versión afectada temporalmente	⊕	☹ ✓ Sí	✓ Sí	Redireccionar tráfico y finalizar nuevas instancias	Instancias nuevas

Método	Repercusión de la implementación con errores	Tiempo de implementación	Sin inactividad	Sin cambios de DNS	Proceso de restauración	El código se implementa en
Azul/verde	Mínima		 ✓ Sí	× No	Intercambio de URL	Instancias nuevas

† Varía en función del tamaño del lote.

†† Varía en función de la configuración de la opción de tiempo de evaluación.

Implementación de una nueva versión de la aplicación

Puede realizar las implementaciones desde el panel del entorno.

Para implementar una nueva versión de aplicación en un entorno de Elastic Beanstalk

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. Elija Upload and Deploy (Cargar e implementar).
4. Utilice el formulario en pantalla para cargar el paquete de código fuente de la aplicación.
5. Elija Implementar.

Nueva implementación de una versión anterior

También puede implementar una versión de la aplicación cargada anteriormente en cualquiera de los entornos desde la página de versiones de la aplicación.

Para implementar una versión de la aplicación existente en un entorno existente

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Applications (Aplicaciones) y, a continuación, elija el nombre de la aplicación de la lista.

Note

Si tiene muchas aplicaciones, utilice la barra de búsqueda para filtrar la lista de aplicaciones.

3. En el panel de navegación, busque el nombre de la aplicación y elija Application versions (Versiones de la aplicación).
4. Seleccione la versión de la aplicación que desee implementar.
5. Elija Actions (Acciones) y a continuación seleccione Deploy (Implementar).
6. Seleccione un entorno y elija Deploy (Implementar).

Otras formas de implementar la aplicación

Si realiza implementaciones con frecuencia, considere la posibilidad de utilizar la [interfaz de línea de comandos de Elastic Beanstalk](#) (CLI de EB) para administrar los entornos. La CLI de EB crea un repositorio junto con su código fuente. También puede crear un paquete fuente, cargarlo en Elastic Beanstalk e implementarlo con un solo comando.

En el caso de las implementaciones que dependen de los cambios de configuración de recursos o de una versión nueva que no se puede ejecutar junto con la versión antigua, puede lanzar un entorno nuevo con la versión nueva y realizar un intercambio de CNAME en una [implementación "blue/green"](#) (azul/verde).

Políticas y ajustes de implementación

AWS Elastic Beanstalk dispone de varias opciones para definir cómo se procesarán las [implementaciones](#), tales como las políticas de implementación (Todo a la vez, Continua, Continua con lote adicional, Inmutable y División de tráfico) y opciones que le permiten configurar el tamaño de los lotes y el comportamiento de las comprobaciones de estado durante las implementaciones. De forma predeterminada, su entorno utiliza implementaciones de tipo "todo a la vez". Si ha creado el


entorno con la CLI de EB y es un entorno escalable (no especificó la opción `--single`), este utiliza implementaciones continuas.

Con implementaciones continuas, Elastic Beanstalk divide las instancias de Amazon EC2 del entorno en lotes e implementa la nueva versión de la aplicación en un lote cada vez. Deja el resto de las instancias en el entorno ejecutando la versión anterior de la aplicación. Durante la implementación continua, algunas instancias atienden solicitudes con la versión antigua de la aplicación, mientras que las instancias de los lotes completados atienden otras solicitudes con la nueva versión. Para obtener más información, consulte [the section called “Funcionamiento de las implementaciones continuas”](#).

Para mantener toda la capacidad durante las implementaciones, puede configurar el entorno para lanzar un nuevo lote de instancias antes de que alguna de estas instancias se quede fuera de servicio. Esta opción se conoce como implementación continua con un lote adicional. Cuando la implementación se completa, Elastic Beanstalk termina el lote adicional de instancias.

Las implementaciones inmutables ejecutan una [actualización inmutable](#) para lanzar un conjunto completo de instancias nuevas con la versión nueva de la aplicación en otro grupo de Auto Scaling, junto con las instancias que ejecutan la versión antigua. Las implementaciones inmutables pueden impedir que se produzcan los problemas causados por las implementaciones continuas realizadas parcialmente. Si las nuevas instancias no superan las comprobaciones de estado, Elastic Beanstalk las termina y deja las instancias originales tal y como estaban.

Las implementaciones de división de tráfico le permiten realizar pruebas de canary controlados como parte de la implementación de la aplicación. En una implementación de división de tráfico, Elastic Beanstalk lanza un conjunto completo de instancias nuevas al igual que durante una implementación inmutable. A continuación, reenvía un porcentaje especificado del tráfico de cliente entrante a la nueva versión de la aplicación durante un período de evaluación especificado. Si las nuevas instancias permanecen en buen estado, Elastic Beanstalk les reenvía todo el tráfico a ellas y termina las antiguas. Si las nuevas instancias no pasan las comprobaciones de estado, o si decide anular la implementación, Elastic Beanstalk devuelve el tráfico a las instancias antiguas y termina las nuevas. Nunca hay interrupción del servicio. Para obtener más información, consulte [the section called “Funcionamiento de las implementaciones de división de tráfico”](#).

 Warning

Algunas políticas reemplazan todas las instancias durante la implementación o actualización. Esto provoca la pérdida de todos los [saldos de ráfagas de Amazon EC2](#) acumulados. Sucede en los siguientes casos:

- Actualizaciones de plataforma administradas con reemplazo de instancias habilitado
- Actualizaciones inmutables
- Implementaciones con actualizaciones inmutables o división de tráfico habilitada

Si la aplicación no supera todas las comprobaciones de estado, pero funciona correctamente con un nivel de corrección inferior, puede permitir que las instancias superen las comprobaciones de estado con un nivel de corrección inferior (por ejemplo, Warning) modificando la opción Healthy threshold (Umbral de buen estado). Si se producen errores en las implementaciones por no superar las comprobaciones de estado y necesita aplicar forzosamente la actualización con independencia del estado que tengan, establezca la opción Ignore health check (ignorar comprobación de buen estado).

Si especifica un tamaño de lote en las actualizaciones continuas, Elastic Beanstalk utilizará también ese valor para los reinicios continuos de las aplicaciones. Utilice los reinicios continuos cuando necesite reiniciar el servidor proxy y el servidor de aplicaciones que se ejecutan en las instancias del entorno sin tiempo de inactividad.

Configuración de la implementación de aplicaciones

En la [consola de administración del entorno](#), habilite y configure implementaciones de versiones basadas en lotes. Para ello, edite Updates and Deployments (Actualizaciones e implementaciones) en la página Configuration (Configuración) del entorno.

Para configurar las implementaciones (consola)

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).

4. En la categoría de configuración Rolling updates and deployments (Actualizaciones acumulativas e implementaciones), elija Edit (Editar).
5. En la sección Application Deployments (Implementaciones de aplicación), elija Deployment policy (Política de implementación), la configuración por lotes y las opciones de comprobación de estado.
6. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

La sección Application Deployments (Implementaciones de aplicaciones) de la página Rolling updates and deployments (Actualizaciones e implementaciones continuas) contiene las siguientes opciones para las implementaciones de aplicaciones:

- Deployment policy (Política de implementación). Seleccione una de las siguientes opciones de implementación:
 - All at once (Todo a la vez): implemente la nueva versión en todas las instancias a la vez. Todas las instancias del entorno se quedarán fuera de servicio durante un breve periodo de tiempo, mientras tiene lugar la implementación.
 - Rolling (Continua): implemente la nueva versión por lotes. Cada lote se queda fuera de servicio durante la fase de implementación, lo que reduce la capacidad del entorno en el número de instancias que tenga el lote.
 - Rolling with additional batch (Continua con un lote adicional): implemente la nueva versión por lotes, pero lance primero un nuevo lote de instancias para asegurarse de que cuenta con toda la capacidad durante el proceso de implementación.
 - Immutable (Inmutable): implemente la nueva versión en un nuevo grupo de instancias a través de una [actualización inmutable](#).
 - Traffic splitting (División de tráfico): implemente la nueva versión en un nuevo grupo de instancias y divida temporalmente el tráfico de cliente entrante entre la versión de la aplicación existente y la nueva.

Para las políticas de implementación Rolling (Continua) y Rolling with additional batch (Continua con lote adicional) puede configurar:

- Batch size (Tamaño del lote): el tamaño del conjunto de instancias que se deben implementar en cada lote.

Elija Percentage (Porcentaje) para configurar un porcentaje del número total de instancias EC2 del grupo de Auto Scaling (hasta el 100%) o elija Fixed (Fijo) para configurar un número fijo de instancias (hasta el número máximo de instancias de la configuración de Auto Scaling de su entorno).

Para la política de implementación Traffic splitting (División de tráfico) puede configurar lo siguiente:

- Traffic split (Dividir tráfico): el porcentaje inicial del tráfico de cliente entrante que Elastic Beanstalk pasa a instancias de entorno que ejecutan la nueva versión de la aplicación que está implementando.
- Traffic splitting evaluation time (Tiempo de evaluación de división de tráfico): el período de tiempo, en minutos, que Elastic Beanstalk espera después de una implementación inicial en buen estado antes de proceder a cambiar todo el tráfico de cliente entrante a la nueva versión de la aplicación que está implementando.

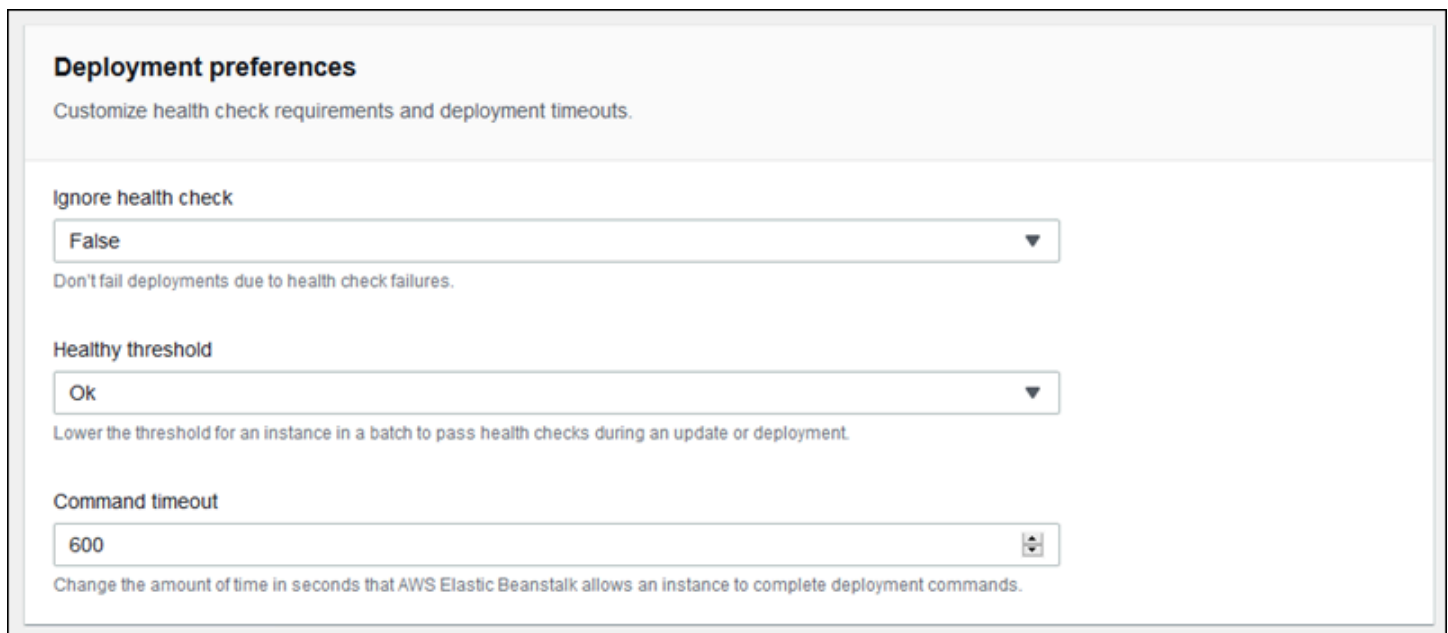
The screenshot shows the AWS Elastic Beanstalk console interface for configuring rolling updates and deployments. The breadcrumb navigation at the top reads: Elastic Beanstalk > Environments > GettingStartedApp-env > Configuration. The main heading is "Modify rolling updates and deployments". Below this, there is a section titled "Application deployments" with a sub-heading: "Choose how AWS Elastic Beanstalk propagates source code changes and software configuration updates. [Learn more](#)".

The configuration options are as follows:

- Deployment policy:** A dropdown menu currently set to "All at once".
- Batch size:** Two radio buttons are present: "Percentage" (which is selected) and "Fixed". Below the "Percentage" option is a numeric input field set to "100" followed by a percentage sign and the text "of instances at a time".
- Traffic split:** A numeric input field set to "10" followed by a percentage sign and the text "to new application version".
- Traffic splitting evaluation time:** A numeric input field set to "5" followed by the text "minutes".

La sección Deployment preferences (Preferencias de implementación) contiene opciones relacionadas con las comprobaciones de estado.

- Ignore health check (omitir comprobación de estado): impide que se revierta una implementación cuando un lote no es capaz de adquirir un estado correcto durante el periodo de tiempo especificado en Command timeout.
- Healthy threshold (Umbral de buen estado): reduce el umbral en el que una instancia se considera que está en buen estado durante las implementaciones continuas, las actualizaciones continuas y las actualizaciones inmutables.
- Command timeout (Tiempo de espera del comando): número de segundos que debe esperarse a que una instancia adquiera un estado correcto antes de cancelar la implementación o, si se ha establecido Ignore health check (Omitir la comprobación de estado), antes de continuar con el lote siguiente.



The screenshot shows the 'Deployment preferences' section in the AWS console. It has a title 'Deployment preferences' and a subtitle 'Customize health check requirements and deployment timeouts.' Below this, there are three settings:

- Ignore health check:** A dropdown menu currently set to 'False'. Below it is the text: 'Don't fail deployments due to health check failures.'
- Healthy threshold:** A dropdown menu currently set to 'Ok'. Below it is the text: 'Lower the threshold for an instance in a batch to pass health checks during an update or deployment.'
- Command timeout:** A text input field containing '600' and a spinner icon. Below it is the text: 'Change the amount of time in seconds that AWS Elastic Beanstalk allows an instance to complete deployment commands.'

Funcionamiento de las implementaciones continuas

Cuando se procesa un lote, Elastic Beanstalk desconecta todas las instancias del lote del balanceador de carga, implementa la nueva versión de la aplicación y vuelve a conectar las instancias. Si habilita el [vaciado de conexiones](#), Elastic Beanstalk vacía las conexiones existentes de las instancias EC2 de cada lote antes de comenzar la implementación.

Cuando las instancias de un lote vuelven a conectarse con el balanceador de carga, Elastic Load Balancing espera hasta que superan un número mínimo de comprobaciones de estado de Elastic

Load Balanceador (el valor de Healthy check count threshold (Umbral de comprobación de buen estado)) y después reanuda el tráfico dirigido a ellas. Si no se ha configurado [health check URL](#), esto puede producirse con mucha rapidez, ya que una instancia superará la comprobación de estado tan pronto como pueda aceptar una conexión TCP. Si se ha configurado una URL de comprobación de estado, el balanceador de carga no direcciona el tráfico a las instancias cargadas hasta que devuelven el código de estado 200 OK en respuesta a una solicitud HTTP GET enviada a la URL de comprobación de estado.

Elastic Beanstalk espera hasta que todas las instancias de un lote tienen un estado correcto antes de pasar al siguiente. Con [los informes de estado básicos](#), el estado de la instancia depende del estado de comprobación de estado de Elastic Load Balancing. Cuando todas las instancias de un lote superan suficientes comprobaciones de estado como para que Elastic Load Balancing considere que tiene un estado correcto, el lote se ha completado. Si están habilitados los [informes de estado avanzados](#), Elastic Beanstalk tiene en cuenta otros factores, como el resultado de las solicitudes entrantes. Con los informes de estado avanzados, todas las instancias deben superar 12 comprobaciones de estado consecutivas con el estado [OK](#) en un plazo de dos minutos en el caso de los entornos de servidor web y 18 comprobaciones en un plazo de tres minutos en el caso de los entornos de trabajo.

Si un lote de instancias no adopta un estado correcto durante el [tiempo de espera del comando](#), la implementación no se realiza correctamente. Después de un error en la implementación, [compruebe el estado de las instancias de su entorno](#) para obtener información sobre la causa del error. A continuación, realice otra implementación con una versión correcta o corregida de la aplicación que desea restaurar.

Si se produce un error en la implementación después de que uno o varios lotes se han completado correctamente, los lotes completados ejecutarán la nueva versión de la aplicación, mientras que los lotes pendientes seguirán ejecutando la versión antigua. Puede identificar la versión que se ejecuta en las instancias del entorno en la [página de estado](#) de la consola. En esta página, se muestra el ID de la implementación de la última implementación que se ejecutó en cada instancia del entorno. Si termina las instancias de una implementación con errores, Elastic Beanstalk las reemplazará por instancias con la versión de la aplicación que se utilizó en la última implementación correcta.

Funcionamiento de las implementaciones de división de tráfico

Las implementaciones de división de tráfico le permiten realizar pruebas de canary controlados. Dirija un poco de tráfico de cliente entrante a la nueva versión de la aplicación para verificar el estado de la aplicación antes de confirmar la nueva versión y dirigir todo el tráfico a ella.

Durante una implementación de división de tráfico, Elastic Beanstalk crea un nuevo conjunto de instancias en un grupo temporal de Auto Scaling independiente. A continuación, Elastic Beanstalk indica al balanceador de carga que dirija un determinado porcentaje del tráfico entrante de su entorno a las nuevas instancias. A continuación, durante un período de tiempo configurado, Elastic Beanstalk realiza un seguimiento del estado del nuevo conjunto de instancias. Si todo está bien, Elastic Beanstalk desplaza el tráfico restante a las nuevas instancias y las asocia al grupo Auto Scaling original del entorno, reemplazando las instancias antiguas. A continuación, Elastic Beanstalk limpia: finaliza las instancias antiguas y elimina el grupo temporal Auto Scaling.

Note

La capacidad del entorno no cambia durante una implementación de división del tráfico. Elastic Beanstalk inicia el mismo número de instancias en el grupo temporal de Auto Scaling que en el grupo Auto Scaling original en el momento en que se inicia la implementación. A continuación, mantiene un número constante de instancias en ambos grupos Auto Scaling durante la duración de la implementación. Tenga en cuenta este hecho al configurar el tiempo de evaluación de división de tráfico del entorno.

Deshacer la implementación a la versión anterior de la aplicación es rápido y no afecta el servicio al tráfico del cliente. Si las nuevas instancias no pasan las comprobaciones de estado, o si decide anular la implementación, Elastic Beanstalk devuelve el tráfico a las instancias antiguas y termina las nuevas. Puede anular cualquier implementación mediante la página de descripción general del entorno de la consola de Elastic Beanstalk y eligiendo **Abort current operation** (Anular operación actual) en **Environment actions** (Acciones de entorno). También puede llamar a la API [AbortEnvironmentUpdate](#) o al comando equivalente de la AWS CLI.

Las implementaciones de división del tráfico requieren un balanceador de carga de aplicaciones. Elastic Beanstalk utiliza este tipo de balanceador de carga de forma predeterminada al crear su entorno mediante la consola de Elastic Beanstalk o la CLI de EB.

Espacios de nombres de opciones de implementación

Puede utilizar las [opciones de configuración](#) del espacio de nombres [aws:elasticbeanstalk:command](#) para configurar sus implementaciones. Si elige la política de división de tráfico, hay opciones adicionales disponibles para esta política en el espacio de nombres [aws:elasticbeanstalk:trafficsplitting](#).

Utilice la opción `DeploymentPolicy` para establecer el tipo de implementación. Se admiten los siguientes valores:

- `AllAtOnce`: deshabilita las implementaciones continuas e implementa siempre todas las instancias a la vez.
- `Rolling`: habilita las implementaciones continuas estándar.
- `RollingWithAdditionalBatch`: lanza un lote adicional de instancias, antes de comenzar la implementación, para mantener la capacidad completa.
- `Immutable`: realiza una [actualización inmutable](#) de cada implementación.
- `TrafficSplitting`: realiza implementaciones de división de tráfico para probar con canary las implementaciones de aplicaciones.

Cuando habilite las implementaciones continuas, defina las opciones `BatchSize` y `BatchSizeType` para configurar el tamaño de cada lote. Por ejemplo, para implementar el 25 % de todas las instancias en cada lote, especifique las siguientes opciones y valores.

Example `.ebextensions/rolling-updates.config`

```
option_settings:
  aws:elasticbeanstalk:command:
    DeploymentPolicy: Rolling
    BatchSizeType: Percentage
    BatchSize: 25
```

Para realizar la implementación en cinco instancias en cada lote, independientemente del número de instancias en ejecución, y para crear un lote adicional de cinco instancias que ejecuten la versión nueva antes de dejar ninguna instancia fuera de servicio, especifique las siguientes opciones y valores.

Example `.ebextensions/rolling-additionalbatch.config`

```
option_settings:
  aws:elasticbeanstalk:command:
    DeploymentPolicy: RollingWithAdditionalBatch
    BatchSizeType: Fixed
    BatchSize: 5
```

Para ejecutar una actualización inmutable de cada implementación con el umbral de comprobación de estado Warning (Advertencia) y continuar con la implementación incluso si las instancias de un lote no superan las comprobaciones de estado durante un periodo de espera de 15 minutos, especifique las siguientes opciones y valores.

Example `.ebextensions/immutable-ignorehealth.config`

```
option_settings:
  aws:elasticbeanstalk:command:
    DeploymentPolicy: Immutable
    HealthCheckSuccessThreshold: Warning
    IgnoreHealthCheck: true
    Timeout: "900"
```

Para realizar implementaciones de división de tráfico, reenviando el 15 por ciento del tráfico de cliente a la nueva versión de la aplicación y evaluando el estado durante 10 minutos, especifique las opciones y valores siguientes.

Example `.ebextensions/traffic-splitting.config`

```
option_settings:
  aws:elasticbeanstalk:command:
    DeploymentPolicy: TrafficSplitting
  aws:elasticbeanstalk:trafficsplitting:
    NewVersionPercent: "15"
    EvaluationTime: "10"
```

La CLI de EB y la consola de Elastic Beanstalk aplican los valores recomendados a las opciones anteriores. Debe eliminar estos ajustes si desea usar archivos de configuración para configurar los. Para obtener más información, consulte [Valores recomendados](#).

Implementaciones Blue/Green (azul/verde) con Elastic Beanstalk

Debido a que AWS Elastic Beanstalk ejecuta una actualización in situ cuando se actualizan las versiones de la aplicación, esta última puede dejar de estar disponible para los usuarios durante un breve periodo de tiempo. Para evitarlo, realice una implementación azul/verde. Para hacerlo, implemente la nueva versión en un entorno distinto y, a continuación, se intercambian los CNAME de los dos entornos para redirigir el tráfico instantáneamente a la nueva versión.

Si desea actualizar un entorno a una versión de plataforma incompatible, también necesitará una implementación azul/verde. Para obtener más información, consulte [the section called “Actualizaciones de la plataforma”](#).

Las implementaciones azul/verde requieren que el entorno se ejecute con independencia de la base de datos, si es que la aplicación usa una. Si su entorno incluye una base de datos que Elastic Beanstalk creó en su nombre, la base de datos y la conexión del entorno no se conservan a menos que realice acciones específicas. Si tiene una base de datos que desea retener, utilice una de las opciones del ciclo de vida de la base de datos de Elastic Beanstalk. Puede elegir la opción Retain (Retener) para mantener la base de datos y el entorno en funcionamiento después de desacoplar la base de datos. Para obtener más información, consulte [Ciclo de vida de la base de datos](#) en el capítulo Configuración de entornos de esta guía.

Para obtener instrucciones sobre cómo configurar la aplicación para conectarse a una instancia de base de datos externa de Amazon RDS (no administrada por Elastic Beanstalk), consulte [Uso de Elastic Beanstalk con Amazon RDS](#).

Para realizar una implementación "blue/green"

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. [Clone el entorno actual](#) o lance un nuevo entorno que ejecute la versión de la plataforma que desee.
3. [Implemente la nueva versión de la aplicación](#) en el nuevo entorno.
4. Pruebe la nueva versión en el nuevo entorno.
5. En la página de información general del entorno, elija Actions (Acciones) y, a continuación, seleccione Swap environment URLs (Intercambiar las URL del entorno).
6. En Environment name (Nombre del entorno), seleccione el entorno actual.

Elastic Beanstalk > Environments > GettingStartedApp-env

Swap environment URLs

When you swap an environment's URL with another environment's URL, you can deploy versions with no downtime. [Learn more](#)

⚠ Swapping the environment URL will modify the Route 53 DNS configuration, which may take a few minutes. Your application will continue to run while the changes are propagated.

Environment details

Environment name:
staging-env

Environment URL:
staging-env.bx7dx222kw.us-east-2.elasticbeanstalk.com

Select an environment to swap

Environment name:
prod-env (e-2mwwbhpfc)

Environment URL:
prod-env.bx7dx222kw.us-east-2.elasticbeanstalk.com

Cancel Swap

7. Elija Swap (Intercambiar).

Elastic Beanstalk intercambia los registros CNAME del entorno antiguo y el nuevo, lo que redirige el tráfico desde la versión anterior a la nueva.

Una vez que Elastic Beanstalk complete la operación de intercambio, asegúrese de que el nuevo entorno responde cuando intenta conectarse a la URL del antiguo entorno. Sin embargo, no termine el entorno antiguo hasta que los cambios de DNS se hayan propagado y los antiguos registros DNS hayan caducado. Los servidores DNS no siempre borran los registros antiguos de la caché en función del período de vida (TTL) establecido en los registros DNS.

Cambios de configuración

Cuando modifica los valores de las opciones de configuración en la sección Configuration (Configuración) de la [consola de administración del entorno](#), AWS Elastic Beanstalk propaga el cambio a todos los recursos afectados. Estos recursos incluyen el balanceador de carga que distribuye el tráfico entre las instancias de Amazon EC2 que ejecutan la aplicación, el grupo de Auto Scaling que administra dichas instancias y las propias instancias EC2.

Se pueden aplicar muchos cambios de configuración a un entorno en ejecución sin reemplazar las instancias existentes. Por ejemplo, la configuración de una [URL de comprobación de estado](#) activa una actualización del entorno para modificar la configuración del balanceador de carga, pero no causa ningún tiempo de inactividad, ya que las instancias que ejecutan su aplicación siguen sirviendo solicitudes mientras se propaga la actualización.

Para los cambios de configuración que modifican la [configuración de lanzamiento](#) o la [configuración de VPC](#) es necesario terminar todas las instancias del entorno y reemplazarlas. Por ejemplo, cuando cambia el tipo de instancia o la configuración de la clave SSH de su entorno, las instancias EC2 deben terminarse y reemplazarse. Elastic Beanstalk proporciona varias políticas que determinan cómo se realiza este reemplazo.

- **Rolling updates (Actualizaciones continuas):** Elastic Beanstalk aplica los cambios de configuración en lotes, manteniendo un número mínimo de instancias en ejecución y sirviendo tráfico en todo momento. Este enfoque evita el tiempo de inactividad durante el proceso de actualización. Para obtener más información, consulte [Actualizaciones continuas](#).
- **Immutable updates (Actualizaciones inmutables):** Elastic Beanstalk lanza un grupo de Auto Scaling temporal fuera del entorno con un conjunto independiente de instancias que se ejecutan con la nueva configuración. A continuación, Elastic Beanstalk coloca estas instancias detrás del balanceador de carga del entorno. Las instancias antiguas y nuevas sirven tráfico hasta que las nuevas instancias pasan las comprobaciones de estado. En ese momento, Elastic Beanstalk mueve las nuevas instancias al grupo de Auto Scaling del entorno y finaliza el grupo temporal y las instancias antiguas. Para obtener más información, consulte [Actualizaciones inmutables](#).
- **Disabled (Desactivado):** Elastic Beanstalk no intenta evitar el tiempo de inactividad. Finaliza las instancias existentes de su entorno y las reemplaza por instancias nuevas que se ejecutan con la nueva configuración.

⚠ Warning

Algunas políticas reemplazan todas las instancias durante la implementación o actualización. Esto provoca la pérdida de todos los [saldos de ráfagas de Amazon EC2](#) acumulados. Sucede en los siguientes casos:

- Actualizaciones de plataforma administradas con reemplazo de instancias habilitado
- Actualizaciones inmutables
- Implementaciones con actualizaciones inmutables o división de tráfico habilitada

Tipos de actualización admitidos

Configuración de actualizaciones continuas	Entornos con balanceo de carga	Entornos de una sola instancia	Entornos de Windows Server heredados
Deshabilitado	✓ Sí	✓ Sí	✓ Sí
Continua en función del estado	✓ Sí	× No	✓ Sí
Continua en función del tiempo	✓ Sí	× No	✓ Sí
Inmutable	✓ Sí	✓ Sí	× No

† A efectos de esta tabla, un entorno de Windows Server heredado es un entorno basado en una [configuración de plataforma de Windows Server](#) que utiliza una versión de IIS anterior a IIS 8.5.

Temas

- [Actualizaciones de configuración del entorno continuas de Elastic Beanstalk](#)
- [Actualizaciones del entorno inmutables](#)

Actualizaciones de configuración del entorno continuas de Elastic Beanstalk

Cuando un [cambio de configuración requiere que se reemplacen las instancias](#), Elastic Beanstalk puede realizar la actualización en lotes para evitar tiempo de inactividad mientras se propaga el cambio. Durante una actualización continua, la capacidad solo se reduce por el tamaño de un solo lote, que puede configurar. Elastic Beanstalk deja un lote de instancias fuera de servicio, termina las instancias y después lanza un lote con la nueva configuración. Una vez que el nuevo lote empieza a servir solicitudes, Elastic Beanstalk pasa al siguiente lote.

Los lotes de actualizaciones de configuración continuas se pueden procesar periódicamente (actualización basada en el tiempo), con un retraso entre cada lote o en función del estado. En las actualizaciones continuas basadas en tiempo, puede configurar la cantidad de tiempo que Elastic Beanstalk debe esperar después de completar el lanzamiento de un lote de instancias antes de pasar al siguiente lote. Esta pausa permite que la aplicación arranque y empiece a atender solicitudes.

Con las actualizaciones continuas basadas en estado, Elastic Beanstalk espera a que las instancias de un lote superen las comprobaciones de estado antes de pasar al siguiente lote. El estado de una instancia lo determina el sistema de informes de estado, que puede ser básico o mejorado. Con el [estado básico](#), un lote se considera en buen estado en el momento en que todas las instancias que contiene superan las comprobaciones de estado de Elastic Load Balancing (ELB).

Con los [informes de estado mejorados](#), todas las instancias de un lote deben superar varias comprobaciones de estado consecutivas antes de que Elastic Beanstalk pase al siguiente lote. Además de las comprobaciones de estado de ELB, que únicamente comprueban las instancias, los informes de estado mejorados monitorizan los registros de la aplicación y el estado de otros recursos de su entorno. En un entorno de servidor web con informes de estado mejorados, todas las instancias deben superar 12 comprobaciones de estado a lo largo de dos minutos (18 comprobaciones de más de tres minutos para los entornos de trabajo). Si alguna instancia no supera una comprobación de estado, se restablece el contador.

Si un lote no pasa a tener un estado correcto durante el tipo de espera de la actualización continua (el valor predeterminado es de 30 minutos), la actualización se cancela. El tiempo de espera de la actualización continua es una [opción de configuración](#) disponible en el espacio de nombres [aws:autoscaling:updatepolicy:rollingupdate](#). Si la aplicación no supera las comprobaciones de estado con el estado Ok pero está estable en otro nivel, puede definir la opción `HealthCheckSuccessThreshold` en el espacio de nombres

[aws:elasticbeanstalk:healthreporting:system](#) para cambiar el nivel en el que Elastic Beanstalk considera que una instancia está en buen estado.

Si el proceso de actualización continua produce un error, Elastic Beanstalk inicia otra actualización continua para restaurar la configuración anterior. Una actualización continua puede producir un error debido a comprobaciones de estado no superadas o si el lanzamiento de nuevas instancias hace que se supere la cuota de su cuenta. Si, por ejemplo, alcanza la cuota de número de instancias de Amazon EC2, la actualización continua puede producir un error cuando intente aprovisionar un lote de nuevas instancias. En este caso, la restauración también produce un error.

Una restauración con error finaliza el proceso de actualización y deja su entorno en un estado Unhealthy. Los lotes sin procesar siguen ejecutando instancias con la configuración anterior, mientras que los lotes que se han ejecutado correctamente tienen la nueva configuración. Para corregir un entorno después de una restauración con error, primero resuelva el problema subyacente que ha causado el error de actualización y después inicie otra actualización del entorno.

Otro método consiste en implementar la nueva versión de la aplicación en un entorno diferente y después realizar un intercambio de CNAME para redirigir el tráfico sin tiempo de inactividad. Para obtener más información, consulte [Implementaciones Blue/Green \(azul/verde\) con Elastic Beanstalk](#).

Actualizaciones continuas frente a implementaciones continuas

Las actualizaciones continuas se producen al cambiar la configuración que requieren las nuevas instancias de Amazon EC2 que se van a aprovisionar para su entorno. Entre estos cambios se incluyen cambios en el grupo de Auto Scaling, como el tipo de instancia y la configuración del par de claves, y cambios en la configuración de la VPC. En una actualización continua, cada lote de instancias se termina antes de aprovisionar un nuevo lote para reemplazarlo.

Las [implementaciones continuas](#) se producen siempre que implementa su aplicación y generalmente se pueden realizar sin reemplazar las instancias en su entorno. Elastic Beanstalk deja cada lote fuera de servicio, implementa la nueva versión de la aplicación y, a continuación, lo vuelve a poner en servicio.

La excepción es si cambia la configuración que requiere que las instancias se reemplacen al implementar una nueva versión de la aplicación. Por ejemplo, si cambia el valor del [nombre de clave](#) en un [archivo de configuración](#) de su paquete de código fuente y lo implementa en el entorno, activa una actualización continua. En lugar de implementar la nueva versión de la aplicación en cada lote de instancias existentes, se aprovisiona un nuevo lote de instancias con la nueva configuración. En este caso, no se produce una implementación independiente, porque las nuevas instancias se obtienen con la nueva versión de la aplicación.

Cada vez que se aprovisionan nuevas instancias como parte de un entorno, hay una fase de implementación en la que el código fuente de la aplicación se implementa en las nuevas instancias y se aplican las opciones de configuración que modifican el sistema operativo o el software de las instancias. Las [opciones de comprobación de estado de la implementación](#) (Ignore health check [Omitir comprobación de estado], Healthy threshold [Umbral de buen estado] y Command timeout [Tiempo de espera de comando]) también se aplican a las actualizaciones continuas basadas en estado y a las aplicaciones inmutables durante la fase de implementación.

Configuración de actualizaciones continuas

Puede habilitar y configurar las actualizaciones continuas en la consola de Elastic Beanstalk.

Para habilitar las actualizaciones continuas

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Rolling updates and deployments (Actualizaciones acumulativas e implementaciones), elija Edit (Editar).
5. En la sección Configuration updates (Actualizaciones de configuración), para Rolling update type (Tipo de actualización continua), seleccione una de las opciones de Rolling (Continua).

Configuration updates

Changes to virtual machine settings and VPC configuration trigger rolling updates to replace the instances in your environment without downtime. [Learn more](#)

Rolling update type
Rolling based on Health

Batch size
1
The maximum number of instances to replace in each phase of the update.

Minimum capacity
1
The minimum number of instances to keep in service at all times.

Pause time
hh:mm:ss
Pause the update for up to an hour between each batch.

6. Seleccione los ajustes Batch size (Tamaño de lote), Minimum capacity (Capacidad mínima) y Pause time (Tiempo de pausa).
7. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

La sección Configuration Updates (Actualizaciones de configuración) de la página Rolling updates and deployments (Actualizaciones acumulativas e implementaciones) contiene las siguientes opciones para las actualizaciones continuas:

- Rolling update type (Tipo de actualización continua): Elastic Beanstalk espera una vez que termina de actualizar un lote de instancias antes de pasar al siguiente lote para permitir que dichas instancias finalicen el proceso de arranque y empiecen a servir tráfico. Puede elegir entre las siguientes opciones:
 - Rolling based on Health (Continua en función del estado): espere a que las instancias del lote actual estén en buen estado antes de poner las instancias en servicio e iniciar el siguiente lote.
 - Rolling based on Time (Continua en función del tiempo): especifique la cantidad de tiempo que se debe esperar entre el lanzamiento de nuevas instancias y su puesta en servicio antes de iniciar el siguiente lote.
 - Immutable (Inmutable): aplique el cambio de configuración a un nuevo grupo de instancias a través de una [actualización inmutable](#).

- **Batch size (Tamaño de lote):** el número de instancias que se van a reemplazar en cada lote, comprendido entre **1** y **10000**. De forma predeterminada, este valor es el tercio del tamaño mínimo del grupo de Auto Scaling redondeado al número entero superior más próximo.
- **Minimum capacity (Capacidad mínima):** el número mínimo de instancias que se mantienen en ejecución mientras se actualizan otras instancias, comprendido entre **0** y **9999**. El valor predeterminado es el tamaño mínimo del grupo de Auto Scaling o un tamaño inferior al tamaño máximo del grupo de Auto Scaling, el que sea menor.
- **Pause time (Pausa) (solo actualizaciones basadas en tiempo):** la cantidad de tiempo que se debe esperar después de que se actualice un lote antes de pasar al siguiente lote para que su aplicación pueda empezar a recibir tráfico. Entre 0 segundos y 1 hora.

El espacio de nombres `aws:autoscaling:updatepolicy:rollingupdate`

También puede utilizar las [opciones de configuración](#) del espacio de nombres [aws:autoscaling:updatepolicy:rollingupdate](#) para configurar actualizaciones continuas.

Utilice la opción `RollingUpdateEnabled` para habilitar las actualizaciones continuas y `RollingUpdateType` para elegir el tipo de actualización. Se admiten los siguientes valores para `RollingUpdateType`:

- **Health:** esperar a que las instancias del lote actual estén en buen estado antes de poner las instancias en servicio e iniciar el siguiente lote.
- **Time:** especifique la cantidad de tiempo que se debe esperar entre el lanzamiento de nuevas instancias y su puesta en servicio antes de iniciar el siguiente lote.
- **Immutable:** aplique el cambio de configuración a un nuevo grupo de instancias a través de una [actualización inmutable](#).

Cuando habilite las actualizaciones continuas, defina las opciones `MaxBatchSize` y `MinInstancesInService` para configurar el tamaño de cada lote. Para las actualizaciones continuas basadas en tiempo y estado, también puede configurar `PauseTime` y `Timeout`, respectivamente.

Por ejemplo, para lanzar hasta cinco instancias a la vez, manteniendo al menos dos instancias en servicio, y esperar cinco minutos y 30 segundos entre los lotes, especifique las siguientes opciones y valores.

Example .ebextensions/timebased.config

```
option_settings:
  aws:autoscaling:updatepolicy:rollingupdate:
    RollingUpdateEnabled: true
    MaxBatchSize: 5
    MinInstancesInService: 2
    RollingUpdateType: Time
    PauseTime: PT5M30S
```

Para habilitar las actualizaciones continuas basadas en estado, con un tiempo de espera de 45 minutos para cada lote, especifique las siguientes opciones y valores.

Example .ebextensions/healthbased.config

```
option_settings:
  aws:autoscaling:updatepolicy:rollingupdate:
    RollingUpdateEnabled: true
    MaxBatchSize: 5
    MinInstancesInService: 2
    RollingUpdateType: Health
    Timeout: PT45M
```

Los valores Timeout y PauseTime deben especificarse en un formato de [duración ISO8601](#): PT#H#M#S, donde cada # es el número de horas, minutos y segundos, respectivamente.

La CLI de EB y la consola de Elastic Beanstalk aplican los valores recomendados a las opciones anteriores. Debe eliminar estos ajustes si desea usar archivos de configuración para configurarlos. Para obtener más información, consulte [Valores recomendados](#).

Actualizaciones del entorno inmutables

Las actualizaciones inmutables del entorno son una alternativa a las [actualizaciones acumulativas](#). Las actualizaciones inmutables del entorno garantizan que los cambios de configuración que requieren sustituir instancias se apliquen de forma eficiente y segura. Si se produce un error en una actualización inmutable del entorno, el proceso de restauración solamente tiene que terminar un grupo de Auto Scaling. Por el contrario, en las actualizaciones continuas que experimentan errores, es necesario efectuar otra actualización continua para restaurar los cambios.

Para realizar una actualización inmutable del entorno, Elastic Beanstalk crea un segundo grupo temporal de Auto Scaling detrás del balanceador de carga del entorno con las nuevas instancias. En

primer lugar, Elastic Beanstalk lanza una única instancia con la nueva configuración del nuevo grupo. Esta instancia administra el tráfico junto con todas las instancias del grupo de Auto Scaling original que ejecutan la configuración anterior.

Cuando la primera instancia supera las comprobaciones de estado, Elastic Beanstalk lanza otras instancias con la nueva configuración hasta alcanzar el mismo número de instancias que se ejecutaban en el grupo de Auto Scaling original. Cuando todas las instancias nuevas superan las comprobaciones de estado, Elastic Beanstalk las transfiere al grupo de Auto Scaling original y termina el grupo de Auto Scaling temporal junto con las instancias antiguas.

Note

En una actualización inmutable del entorno, la capacidad del entorno se duplica durante un corto periodo de tiempo, cuando las instancias del nuevo grupo de Auto Scaling comienzan a emitir solicitudes y aún no se han terminado las instancias del grupo de Auto Scaling original. Si el entorno tiene muchas instancias o si usted tiene una [cuota de instancias bajo demanda](#) pequeña, asegúrese de que cuenta con la capacidad suficiente como para realizar una actualización inmutable del entorno. Si está rozando la cuota, considere la posibilidad de utilizar en su lugar una actualización continua.

Las actualizaciones inmutables requieren que los [informes de estado avanzados](#) evalúen el estado del entorno durante la actualización. Los informes de estado avanzados combinan las comprobaciones de estado del balanceador de carga estándar con la monitorización de instancias para asegurarse de que las instancias que ejecutan la nueva configuración [atienden las solicitudes correctamente](#).

También puede utilizar las actualizaciones inmutables para implementar nuevas versiones de la aplicación como alternativa a las implementaciones continuas. Cuando [configura Elastic Beanstalk para que utilice actualizaciones inmutables para la implementación de aplicaciones](#), reemplaza todas las instancias del entorno cada vez que se implementa una nueva versión de la aplicación. Si se produce un error en una implementación de aplicaciones inmutable, Elastic Beanstalk revierte los cambios de forma inmediata terminando el nuevo grupo de Auto Scaling. Esto impide que haya implementaciones parciales de flota, que sí pueden darse si las implementaciones continuas experimentan un error cuando algunos lotes ya están completados.

⚠ Warning

Algunas políticas reemplazan todas las instancias durante la implementación o actualización. Esto provoca la pérdida de todos los [saldos de ráfagas de Amazon EC2](#) acumulados. Sucede en los siguientes casos:

- Actualizaciones de plataforma administradas con reemplazo de instancias habilitado
- Actualizaciones inmutables
- Implementaciones con actualizaciones inmutables o división de tráfico habilitada

Si se produce un error en una actualización inmutable, las nuevas instancias cargan los [registros de paquete](#) en Amazon S3 antes de que Elastic Beanstalk las termine. Elastic Beanstalk deja los registros de las actualizaciones inmutables erróneas en Amazon S3 una hora antes de eliminarlos, en lugar de los 15 minutos estándar de los registros de paquete y los registros de finalización.

ℹ Note

Si utiliza actualizaciones inmutables con la implementación de versiones de la aplicación pero no con la configuración, puede producirse un error si intenta implementar una versión de la aplicación que contenga cambios de configuración que normalmente activarían una actualización continua (por ejemplo, configuraciones que cambian el tipo de instancia). Para evitar esto, efectúe el cambio de configuración en una actualización diferente o configure actualizaciones inmutables tanto para las implementaciones como para los cambios de configuración.

No se puede ejecutar una actualización inmutable junto con cambios de configuración de los recursos. Por ejemplo, no puede cambiar una [configuración que requiera la sustitución de instancias](#) al mismo tiempo que actualiza otras opciones o que ejecuta una implementación inmutable con archivos de configuración que cambian las opciones de ajuste u otros recursos del código fuente. Si intenta cambiar la configuración de recursos (por ejemplo, la configuración del balanceador de carga) al mismo tiempo que ejecuta una actualización inmutable, Elastic Beanstalk devolverá un error.

Si los cambios de configuración de los recursos no dependen de que se realicen modificaciones en el código fuente o en la configuración de la instancia, realícelos en dos actualizaciones. En caso de que sí dependan de estas modificaciones, realice en su lugar una [implementación "blue/green"](#).

Configuración de actualizaciones inmutables

Puede habilitar y configurar las actualizaciones inmutables en la consola de Elastic Beanstalk.

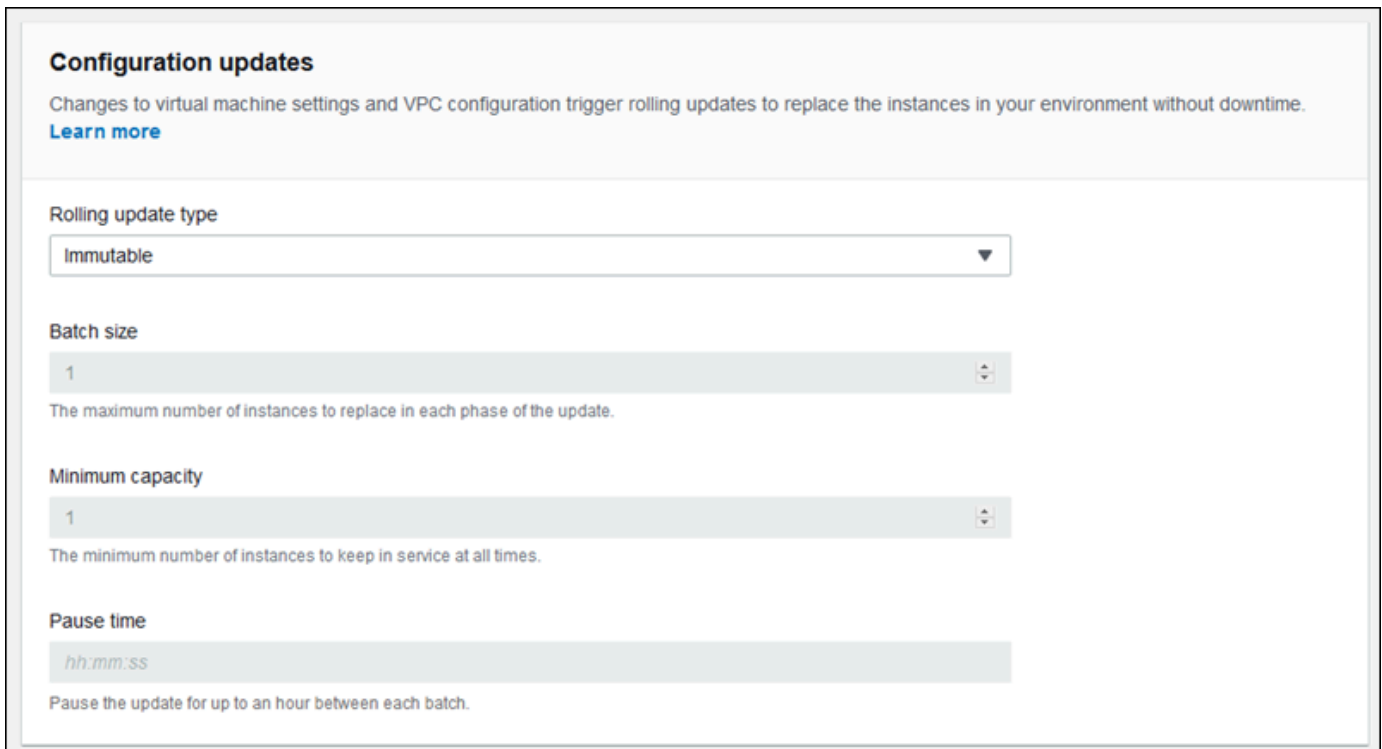
Para habilitar las actualizaciones inmutables (consola)

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Rolling updates and deployments (Actualizaciones acumulativas e implementaciones), elija Edit (Editar).
5. En la sección Configuration Updates (Actualizaciones de configuración), establezca Rolling update type (Tipo de actualización acumulativa) en Immutable (Inmutable).



Configuration updates

Changes to virtual machine settings and VPC configuration trigger rolling updates to replace the instances in your environment without downtime.
[Learn more](#)

Rolling update type
Immutable

Batch size
1
The maximum number of instances to replace in each phase of the update.

Minimum capacity
1
The minimum number of instances to keep in service at all times.

Pause time
hh:mm:ss
Pause the update for up to an hour between each batch.

6. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

El espacio de nombres aws:autoscaling:updatepolicy:rollingupdate

También puede utilizar las opciones del espacio de nombres `aws:autoscaling:updatepolicy:rollingupdate` para configurar actualizaciones inmutables. En el siguiente [archivo de configuración](#) de ejemplo, se habilitan actualizaciones inmutables para cambios de configuración.

Example `.ebextensions/immutable-updates.config`

```
option_settings:
  aws:autoscaling:updatepolicy:rollingupdate:
    RollingUpdateType: Immutable
```

En el siguiente ejemplo, se habilitan actualizaciones inmutables tanto para cambios de configuración como para implementaciones.

Example `.ebextensions/immutable-all.config`

```
option_settings:
  aws:autoscaling:updatepolicy:rollingupdate:
    RollingUpdateType: Immutable
  aws:elasticbeanstalk:command:
    DeploymentPolicy: Immutable
```

La CLI de EB y la consola de Elastic Beanstalk aplican los valores recomendados a las opciones anteriores. Debe eliminar estos ajustes si desea usar archivos de configuración para configurarlos. Para obtener más información, consulte [Valores recomendados](#).

Actualización de la versión de la plataforma del entorno de Elastic Beanstalk

Elastic Beanstalk publica con regularidad nuevas versiones de la plataforma para actualizar todas las [plataformas](#) basadas en Windows Server y Linux. Las nuevas versiones de la plataforma contienen actualizaciones de los componentes de software existentes y son compatibles con nuevas características y opciones de configuración. Para obtener más información sobre las plataformas y las versiones de la plataforma, consulte [Glosario de plataformas Elastic Beanstalk](#).

Puede utilizar la consola de Elastic Beanstalk o la CLI de EB para actualizar la versión de la plataforma del entorno. En función de la versión de la plataforma a la que desea actualizar, Elastic Beanstalk recomienda uno de los dos métodos para realizar actualizaciones en la plataforma.

- [Método 1: actualización de la versión de la plataforma del entorno](#). Le recomendamos este método cuando actualice a la versión más reciente de la plataforma dentro de una ramificación de la plataforma, con el mismo tiempo de ejecución, servidor web, servidor de aplicaciones y sistema operativo, y sin ningún cambio en la versión principal de la plataforma. Esta es la actualización de la plataforma más frecuente y rutinaria.
- [Método 2: realización de una implementación "blue/green"](#). Le recomendamos este método cuando actualice a una versión de plataforma de una ramificación de la plataforma diferente, con un tiempo de ejecución, un servidor web, un servidor de aplicaciones o un sistema operativo diferentes, o a una versión principal de la plataforma diferente. Este es un buen enfoque cuando desea aprovechar las nuevas capacidades de tiempo de ejecución o la última funcionalidad de Elastic Beanstalk, o cuando desea salir de una ramificación de la plataforma obsoleta o retirada.

[La migración desde una versión heredada de la plataforma](#) requiere una implementación blue/green (azul/verde), ya que estas versiones de plataforma no son compatibles con las versiones que se admiten actualmente.

[La migración de una aplicación Linux a Amazon Linux 2](#) requiere una implementación blue/green (azul/verde), ya que las versiones de la plataforma Amazon Linux 2 son incompatibles con las versiones anteriores de la plataforma de la AMI de Amazon Linux.

Para obtener más ayuda para elegir el mejor método de actualización de la plataforma, expanda la sección de la plataforma del entorno.

Docker

Utilice el [método 1](#) para realizar actualizaciones de la plataforma.

Multicontainer Docker

Utilice el [método 1](#) para realizar actualizaciones de la plataforma.

Preconfigured Docker

Analice los siguientes casos:

- Si va a migrar la aplicación a otra plataforma, por ejemplo, de Go 1.4 (Docker) a Go 1.11 o de Python 3.4 (Docker) a Python 3.6, utilice el [método 2](#).
- Si va a migrar la aplicación a una versión de contenedor de Docker diferente, por ejemplo, de Glassfish 4.1 (Docker) a Glassfish 5.0 (Docker), utilice el [método 2](#).
- Si va a actualizar a la versión de la plataforma más reciente sin cambios en la versión de contenedor o en la versión principal, utilice el [método 1](#).

Go

Utilice el [método 1](#) para realizar actualizaciones de la plataforma.

Java SE

Analice los siguientes casos:

- Si va a migrar la aplicación a una versión de tiempo de ejecución de Java diferente, por ejemplo, de Java 7 a Java 8, utilice el [método 2](#).
- Si va a actualizar a la versión de la plataforma más reciente sin cambios en la versión de tiempo de ejecución, utilice el [método 1](#).

Java con Tomcat

Analice los siguientes casos:

- Si va a migrar la aplicación a una versión de tiempo de ejecución de Java o a una versión de servidor de aplicaciones de Tomcat diferentes, por ejemplo, de Java 7 con Tomcat 7 a Java 8 con Tomcat 8.5, utilice el [método 2](#).
- Si va a migrar la aplicación a través de las versiones principales de la plataforma Java con Tomcat (v1.x.x, v2.x.x y v3.x.x), utilice el [método 2](#).
- Si va a actualizar a la versión de la plataforma más reciente sin cambios en la versión de tiempo de ejecución, la versión de servidor de aplicaciones o la versión principal, utilice el [método 1](#).

.NET en Windows Server con IIS

Analice los siguientes casos:

- Si va a migrar la aplicación a una versión del sistema operativo Windows diferente, por ejemplo, de Windows Server 2008 R2 a Windows Server 2016, utilice el [método 2](#).
- Si va a migrar la aplicación a través de las versiones principales de la plataforma Windows Server, consulte [Migración desde versiones principales anteriores de la plataforma Windows Server](#) y utilice el [método 2](#).
- Si la aplicación se está ejecutando actualmente en una plataforma Windows Server V2.x.x y va a actualizar a la versión de la plataforma más reciente, utilice el [método 1](#).

Note

[Las versiones de la plataforma Windows Server](#) anteriores a v2 no se han versionado semánticamente. Solo puede lanzar la versión más reciente de cada una de estas versiones principales de plataforma de Windows Server y, tras la actualización, no se puede restaurar una versión anterior.

Node.js

Utilice el [método 2](#) para realizar actualizaciones de plataforma.

PHP

Analice los siguientes casos:

- Si va a migrar la aplicación a una versión de tiempo de ejecución de PHP diferente, por ejemplo, de PHP 5.6 a PHP 7.2, utilice el [método 2](#).
- Si va a migrar la aplicación a través de las versiones principales de la plataforma PHP (v1.x.x y v2.x.x), utilice el [método 2](#).
- Si va a actualizar a la versión de la plataforma más reciente sin cambios en la versión de tiempo de ejecución o en la versión principal, utilice el [método 1](#).

Python

Analice los siguientes casos:

- Si va a migrar la aplicación a una versión de tiempo de ejecución de Python diferente, por ejemplo, de Python 2.7 a Python 3.6, utilice el [método 2](#).

- Si va a migrar la aplicación a través de las versiones principales de la plataforma Python (v1.x.x y v2.x.x), utilice el [método 2](#).
- Si va a actualizar a la versión de la plataforma más reciente sin cambios en la versión de tiempo de ejecución o en la versión principal, utilice el [método 1](#).

Ruby

Analice los siguientes casos:

- Si va a migrar la aplicación a una versión de tiempo de ejecución de Ruby o a una versión de servidor de aplicaciones diferentes, por ejemplo, de Ruby 2.3 con Puma a Ruby 2.6 con Puma, utilice el [método 2](#).
- Si va a migrar la aplicación a través de las versiones principales de la plataforma Ruby (v1.x.x y v2.x.x), utilice el [método 2](#).
- Si va a actualizar a la versión de la plataforma más reciente sin cambios en la versión de tiempo de ejecución, la versión de servidor de aplicaciones o la versión principal, utilice el [método 1](#).

Método 1: actualización de la versión de la plataforma del entorno

Utilice este método para actualizar a la versión más reciente de la ramificación de la plataforma del entorno. Si previamente creó un entorno utilizando una versión antigua de la plataforma o actualizó el entorno desde una versión anterior, también podrá utilizar este método para volver a una versión anterior de la plataforma, siempre que esté en la misma ramificación de la plataforma.

Para actualizar la versión de la plataforma del entorno, realice el siguiente procedimiento:

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

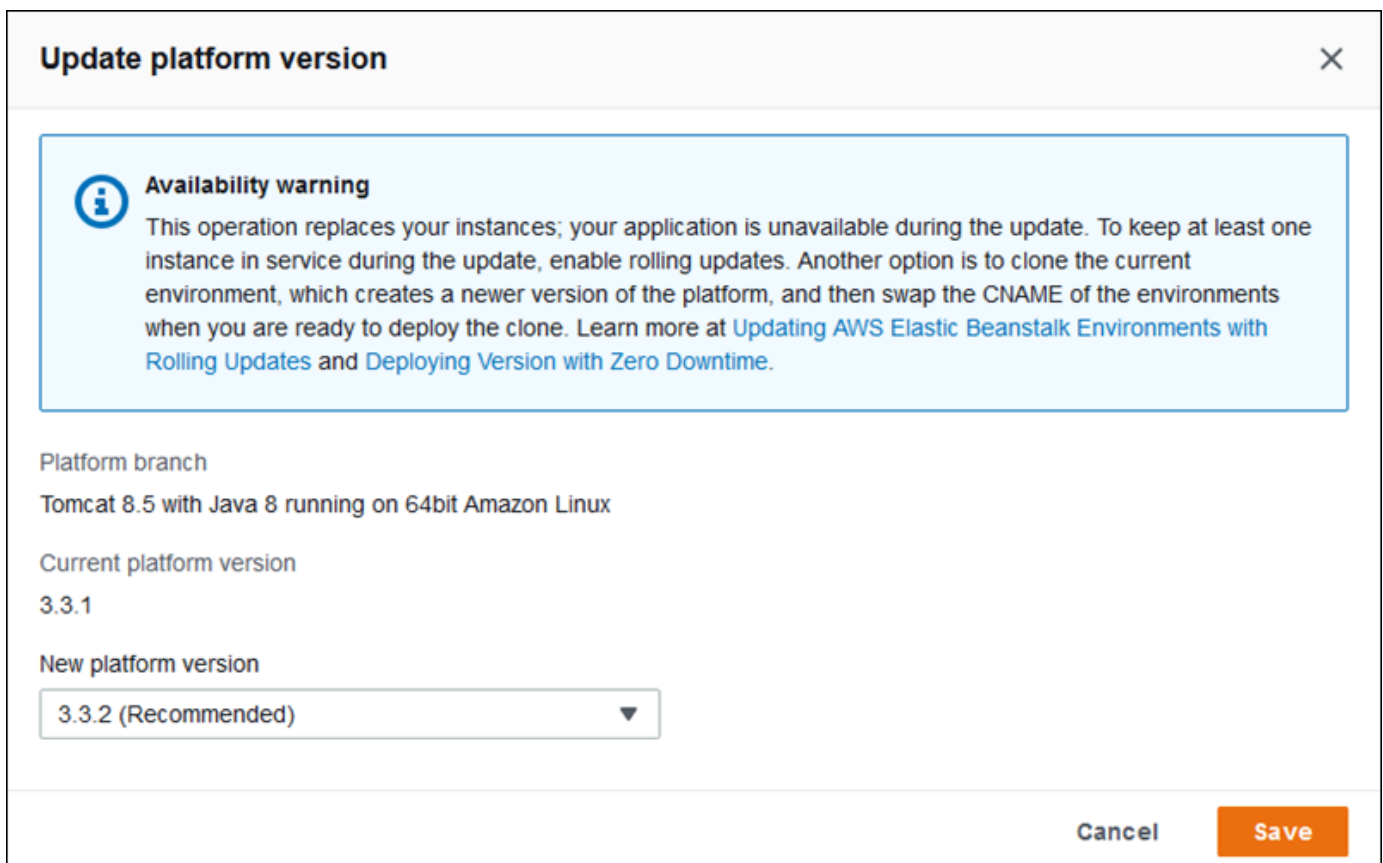
Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En la página de información general del entorno, en Platform (Plataforma), elija Change (Cambiar).



4. En el cuadro de diálogo Update platform version (Actualizar versión de plataforma) seleccione una versión de plataforma. La versión de la plataforma más reciente (recomendada) en la ramificación se selecciona automáticamente. Puede actualizar a cualquier versión que haya utilizado en el pasado.



5. Seleccione Save.

Para simplificar aún más las actualizaciones de la plataforma, Elastic Beanstalk puede administrarlas por usted. Puede configurar el entorno para que las actualizaciones de versiones secundarias y de parches se apliquen automáticamente durante un periodo de mantenimiento semanal que se puede configurar. Elastic Beanstalk aplica las actualizaciones administradas sin que se produzcan tiempos de inactividad ni se reduzca la capacidad y cancela la actualización de inmediato si las instancias que ejecutan la aplicación con la nueva versión no superan las comprobaciones de estado. Para obtener más información, consulte [Actualizaciones de plataforma administradas](#).

Método 2: realización de una implementación "blue/green"

Utilice este método para actualizar a una ramificación de la plataforma diferente con un tiempo de ejecución, servidor web, servidor de aplicaciones o sistema operativo diferentes, o a una versión principal de la plataforma diferente. Esto es normalmente necesario cuando desea aprovechar las nuevas capacidades de ejecución o la última funcionalidad de Elastic Beanstalk. También es necesario cuando migra fuera de una ramificación de la plataforma obsoleta o retirada.

Cuando migra a través de las versiones de la plataforma principal o a las versiones de la plataforma con las principales actualizaciones de componentes, existe una mayor probabilidad de que la aplicación, o algunos aspectos de ella, puedan no funcionar según lo previsto en la nueva versión de la plataforma, y podrían requerir cambios.

Antes de realizar la migración, actualice su equipo de desarrollo local a las versiones de tiempo de ejecución más recientes y otros componentes de la plataforma a los que tenga previsto migrar. Compruebe que su aplicación sigue funcionando según lo previsto, y realice cualquier corrección o cambio de código necesarios. A continuación, utilice el siguiente procedimiento de práctica recomendada para migrar de forma segura el entorno a una nueva versión de la plataforma.

Para migrar el entorno a una versión de la plataforma con las actualizaciones principales, realice el siguiente procedimiento:

1. [Cree un nuevo entorno](#), utilizando la nueva versión de plataforma de destino e implemente el código de la aplicación en ella. El nuevo entorno debe estar en la aplicación de Elastic Beanstalk que contiene el entorno que está migrando. No termine aún el entorno existente.
2. Utilice el nuevo entorno para migrar su aplicación. En particular:
 - Busque y corrija cualquier problema de compatibilidad de la aplicación no detectado durante la fase de desarrollo.
 - Asegúrese de que las personalizaciones que realiza su aplicación con [archivos de configuración](#) funcionen correctamente en el nuevo entorno. Estos podrían incluir la

configuración de opciones, los paquetes instalados adicionales, las políticas de seguridad personalizadas y archivos de configuración o script instalados en algunas instancias del entorno.

- Si la aplicación utiliza una imagen de máquina de Amazon (Amazon Machine Image, AMI) personalizada, cree una nueva AMI personalizada basada en una AMI de la nueva versión de la plataforma. Para obtener más información, consulte [Uso de una imagen de máquina de Amazon \(AMI\) personalizada](#). En concreto, es necesario si la aplicación utiliza la plataforma Windows Server con una AMI personalizada y va a migrar a una versión de la plataforma Windows Server V2. En ese caso, consulte también [Migración desde versiones principales anteriores de la plataforma Windows Server](#).

Recorra en iteración las pruebas y la implementación de las correcciones hasta que esté satisfecho con la aplicación en el nuevo entorno.

3. Convierta el nuevo entorno en su entorno de producción intercambiando su CNAME con el CNAME del entorno de producción existente. Para obtener más información, consulte [Implementaciones Blue/Green \(azul/verde\) con Elastic Beanstalk](#).
4. Cuando esté satisfecho con el estado de su nuevo entorno en la producción, termine el entorno antiguo. Para obtener más información, consulte [Terminación de un entorno de Elastic Beanstalk](#).

Actualizaciones de plataforma administradas

AWS Elastic Beanstalk publica de forma periódica [actualizaciones de las plataformas](#) para proporcionar correcciones, actualizaciones de software y nuevas características. Con las actualizaciones de plataforma administradas, puede configurar el entorno para que se actualice automáticamente a la última versión de una plataforma durante un [periodo de mantenimiento](#) programado. Su aplicación permanece en servicio durante el proceso de actualización, sin ningún tipo de reducción de la capacidad. Las actualizaciones administradas están disponibles en entornos de una sola instancia y en entornos con balanceo de carga.

Note

Esta característica no está disponible en las [versiones de la plataforma Windows Server](#) anteriores a la versión 2 (v2).

Puede configurar su entorno para que aplique automáticamente las [actualizaciones de versiones de parches](#) o las actualizaciones de versiones de parches y versiones secundarias. Las actualizaciones administradas de la plataforma no admiten las actualizaciones en ramificaciones de la plataforma (actualizaciones de diferentes versiones principales de componentes de la plataforma, como el sistema operativo, el tiempo de ejecución o los componentes de Elastic Beanstalk), ya que pueden introducir cambios incompatibles con versiones anteriores.

También puede configurar Elastic Beanstalk para que se sustituyan todas las instancias del entorno durante el periodo de mantenimiento, aunque no haya ninguna actualización de la plataforma disponible. La sustitución de todas las instancias de su entorno es útil si su aplicación encuentra errores o problemas de memoria cuando se ejecuta durante un periodo prolongado.

En los entornos que se crearon el 25 de noviembre de 2019 o con posterioridad a través de la consola de Elastic Beanstalk, de forma predeterminada y siempre que es posible, las actualizaciones administradas están habilitadas. Las actualizaciones administradas requieren que se habilite el [estado mejorado](#). El estado mejorado se habilita de forma predeterminada cuando se selecciona uno de los [valores preestablecidos de configuración](#) y se deshabilita cuando se selecciona una configuración personalizada. La consola no puede habilitar las actualizaciones administradas en las versiones anteriores de la plataforma que no admiten el estado mejorado o cuando el estado mejorado está deshabilitado. Si la consola permite realizar actualizaciones administradas en un nuevo entorno, el periodo de actualización semanal se establece en un día de la semana y un momento aleatorios. El valor de Update level (Nivel de actualización) está establecido en Minor and patch (Actualizaciones menores y parches) y la opción Instance replacement (Sustitución de instancias) está deshabilitada. Puede deshabilitar o volver a configurar las actualizaciones administradas antes de realizar el último paso de la creación del entorno.

En los entornos existentes, utilice la consola de Elastic Beanstalk en cualquier momento para configurar las actualizaciones administradas de la plataforma.

Important


Un número elevado de entornos de Beanstalk en una cuenta de AWS puede presentar un riesgo de problemas de limitación durante las actualizaciones administradas. La expresión número elevado hace referencia a una cantidad relativa que depende de la precisión con la que se programen las actualizaciones administradas para los entornos. Cuando hay más de 200 entornos estrechamente programados en una cuenta, es posible que se produzcan problemas de limitación. Sin embargo, una cantidad inferior también podría ser un problema.

Para equilibrar la carga de recursos para las actualizaciones administradas, le recomendamos que distribuya los periodos de mantenimiento programados para los entornos en una sola cuenta.

Además, considere una estrategia de varias cuentas. Para obtener más información, consulte [Organización del entorno de AWS mediante varias cuentas](#) en el sitio web de documentos técnicos y guías de AWS.

Para configurar las actualizaciones administradas de la plataforma

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

 Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría Managed Updates (Actualizaciones administradas), elija Edit (Editar).
5. Habilite o deshabilite Managed updates (Actualizaciones administradas).
6. Si las actualizaciones administradas están habilitadas, seleccione un periodo de mantenimiento y un valor de Update level (Nivel de actualización).
7. (Opcional) Seleccione Instance replacement (Reemplazo de instancia) para habilitar la sustitución semanal de instancias.

Elastic Beanstalk > Environments > GettingStartedApp-env > Configuration

Modify managed updates

Managed platform updates

Enable managed platform updates to apply platform updates automatically during a weekly maintenance window that you choose. Your application stays available during the update process.

Managed updates

Enabled

Weekly update window

Tuesday at 12 : 00 UTC

Any available managed updates will run between Tuesday, 4:00 AM and Tuesday, 6:00 AM (-0800 GMT).

Update level

Minor and patch

Instance replacement

If enabled, an instance replacement will be scheduled if no other updates are available.

Enabled

Cancel Continue Apply

8. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

Las actualizaciones de plataforma administradas utilizan los [informes de estado mejorados](#) para determinar si su aplicación tiene un funcionamiento lo suficiente bueno para considerar que la actualización de la plataforma se ha realizado correctamente. Para obtener instrucciones, consulte [Habilitación de informes de estado mejorado de Elastic Beanstalk](#).

Secciones

- [Permisos necesarios para realizar actualizaciones de plataforma administradas](#)
- [periodo de mantenimiento de actualizaciones administradas](#)
- [Actualizaciones de versiones secundarias y parches](#)
- [Actualizaciones del entorno inmutables](#)
- [Gestión de actualizaciones administradas](#)
- [Espacios de nombres de opciones de acciones administradas](#)

Permisos necesarios para realizar actualizaciones de plataforma administradas

Elastic Beanstalk necesita permiso para iniciar una actualización de plataforma en su nombre. Para obtener estos permisos, Elastic Beanstalk asume el rol de servicio de actualizaciones administradas. Cuando utiliza el [rol de servicio](#) predeterminado de su entorno, la consola de Elastic Beanstalk también lo utiliza como rol de servicio de actualizaciones administradas. La consola asigna la política administrada [AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy](#) a su rol de servicio. Esta política dispone de todos los permisos que Elastic Beanstalk necesita para realizar actualizaciones administradas de la plataforma.

Para obtener más información sobre otras formas de establecer el rol de servicio de actualizaciones administradas, consulte [the section called “Roles de servicio”](#).

Note

Si utiliza [archivos de configuración](#) para ampliar su entorno con recursos adicionales, es posible que necesite agregar permisos adicionales al rol del servicio de actualizaciones administradas de su entorno. Normalmente deberá agregar permisos cuando haga referencia a estos recursos por su nombre en otras secciones o archivos.

Si se produce un error en una actualización, puede encontrar el motivo del error en la página [Actualizaciones administradas](#).

periodo de mantenimiento de actualizaciones administradas

Cuando AWS lanza una nueva versión de la plataforma de su entorno, Elastic Beanstalk programa una actualización administrada de la plataforma durante el siguiente periodo de mantenimiento semanal. Los periodos de mantenimiento son de dos horas. Elastic Beanstalk inicia una actualización programada durante el periodo de mantenimiento. pero la actualización no se puede completar hasta que termina el periodo.

Note

En la mayoría de los casos, Elastic Beanstalk programa la actualización administrada para que se lleve a cabo durante el siguiente periodo de mantenimiento semanal. El sistema tiene en cuenta distintos aspectos sobre la seguridad de la actualización y la disponibilidad del servicio al programar las actualizaciones administradas. En casos excepcionales, es posible que una actualización no se programe para el primer periodo de mantenimiento que vaya a

tener lugar próximamente. Si esto ocurre, el sistema lo intenta de nuevo durante el siguiente periodo de mantenimiento. Para aplicar manualmente una actualización administrada, seleccione Apply now (Aplicar ahora), tal y como se indica en [Gestión de actualizaciones administradas](#) en esta página.

Actualizaciones de versiones secundarias y parches

Puede habilitar las actualizaciones de plataforma administradas para aplicar solo las actualizaciones de versiones de parches o para las actualizaciones de versiones secundarias y de parches. Las actualizaciones de versiones de parches proporcionan correcciones de errores y mejoras de desempeño, y pueden incluir cambios menores de configuración en el software de la instancia, scripts y opciones de configuración. Las actualizaciones de versiones secundarias proporcionan compatibilidad con las nuevas características de Elastic Beanstalk. No puede aplicar actualizaciones de versiones principales, ya que podrían introducir cambios incompatibles con versiones anteriores con las actualizaciones de plataforma administradas.

En un número de versión de la plataforma, el segundo número es la versión de actualización secundaria y el tercer número es la versión del parche. Por ejemplo, en una versión de plataforma 2.0.7, la versión secundaria es 0 y la versión de parche es 7.

Actualizaciones del entorno inmutables

Las actualizaciones de plataforma administradas realizan [actualizaciones del entorno inmutables](#) para actualizar su entorno a una nueva versión de la plataforma. Las actualizaciones inmutables actualizan el entorno sin poner fuera de servicio ninguna instancia ni modificar el entorno antes de confirmar que las instancias que ejecutan la nueva versión superan las comprobaciones de estado.

En una actualización inmutable, Elastic Beanstalk implementa el mismo número de instancias que se están ejecutando actualmente con la nueva versión de la plataforma. Las nuevas instancias empiezan a recibir solicitudes junto con aquellas que ejecutan la versión anterior. Si el nuevo conjunto de instancias supera todas las comprobaciones de estado, Elastic Beanstalk termina el conjunto anterior de instancias y deja solo las instancias con la nueva versión.

Las actualizaciones de plataforma administradas siempre realizan actualizaciones inmutables aunque se apliquen fuera del periodo de mantenimiento. Si cambia la versión de la plataforma desde el Dashboard (Panel), Elastic Beanstalk aplica la política de actualización que ha elegido para las actualizaciones de la configuración.

⚠ Warning

Algunas políticas reemplazan todas las instancias durante la implementación o actualización. Esto provoca la pérdida de todos los [saldos de ráfagas de Amazon EC2](#) acumulados. Sucede en los siguientes casos:

- Actualizaciones de plataforma administradas con reemplazo de instancias habilitado
- Actualizaciones inmutables
- Implementaciones con actualizaciones inmutables o división de tráfico habilitada

Gestión de actualizaciones administradas

La consola de Elastic Beanstalk muestra información detallada sobre las actualizaciones administradas en la página [Managed updates overview](#) (Información general sobre las actualizaciones administradas).

Para ver información sobre las actualizaciones administradas (consola)

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista [Regions](#) (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija [Environments](#) (Entornos) y, a continuación, elija el nombre del entorno en la lista.

i Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. Elija [Managed Updates](#) (Actualizaciones administradas).

En la sección [Información general sobre las actualizaciones](#) se proporciona información sobre las actualizaciones administradas programadas y pendientes. La sección [History](#) muestra las actualizaciones que se han realizado correctamente y los intentos que han producido un error.

Puede elegir aplicar una actualización programada inmediatamente, en lugar de esperar al periodo de mantenimiento.

Para aplicar inmediatamente una actualización administrada (consola)

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. Elija Managed Updates (Actualizaciones administradas).
4. Elija Apply now.
5. Compruebe los detalles de la actualización y, a continuación, elija Apply (Aplicar).

Cuando aplica una actualización de plataforma administrada fuera del periodo de mantenimiento, Elastic Beanstalk realiza una actualización inmutable. Si actualiza la plataforma del entorno desde el [Dashboard \(Panel\)](#) o con otro cliente, Elastic Beanstalk utiliza el tipo de actualización que ha seleccionado para los [cambios de configuración](#).

Si no dispone de una actualización administrada programada, es posible que su entorno ya esté ejecutando la versión más reciente. Otros motivos para no tener una actualización programada son los siguientes:

- Hay disponible una [versión secundaria](#), pero el entorno está configurado para que se apliquen automáticamente solo las actualizaciones de versiones de parches.
- Su entorno no se ha analizado desde que se lanzó la actualización. Por lo general, Elastic Beanstalk comprueba si hay actualizaciones cada hora.
- Hay una actualización pendiente o en curso.

Cuando comienza el periodo de mantenimiento o cuando elige Apply now (Aplicar ahora), las actualizaciones programadas entran en un estado pendiente antes de su ejecución.

Espacios de nombres de opciones de acciones administradas

Puede utilizar las [opciones de configuración](#) de los espacios de nombres [aws:elasticbeanstalk:managedactions](#) y

[aws:elasticbeanstalk:managedactions:platformupdate](#) para habilitar y configurar las actualizaciones de plataforma administradas.

La opción `ManagedActionsEnabled` activa las actualizaciones de plataforma administradas. Establezca esta opción en `true` para habilitar las actualizaciones de plataforma administradas y utilice el resto de las opciones para configurar el comportamiento de las actualizaciones.

Utilice `PreferredStartTime` para configurar el inicio del periodo de mantenimiento semanal en un formato de *día:hora:minuto*.

Establezca `UpdateLevel` en `minor` o `patch` para aplicar las actualizaciones de versiones secundarias y de parches o solo las actualizaciones de versiones de parches, respectivamente.

Cuando se habilitan las actualizaciones de plataforma administradas, puede activar la sustitución de instancias estableciendo la opción `InstanceRefreshEnabled` en `true`. Cuando esta opción está habilitada, Elastic Beanstalk ejecuta una actualización inmutable en su entorno cada semana, independientemente de si hay o no una nueva versión de la plataforma disponible.

El siguiente [archivo de configuración](#) de ejemplo habilita las actualizaciones de plataforma administradas para las actualizaciones de versiones de parches con un periodo de mantenimiento que empieza a las 9:00 a.m. UTC cada martes.

Example `.ebextensions/managed-platform-update.config`

```
option_settings:
  aws:elasticbeanstalk:managedactions:
    ManagedActionsEnabled: true
    PreferredStartTime: "Tue:09:00"
  aws:elasticbeanstalk:managedactions:platformupdate:
    UpdateLevel: patch
    InstanceRefreshEnabled: true
```

Migración de la aplicación desde una versión de plataforma heredada

Si tiene implementada una aplicación de Elastic Beanstalk que utiliza una versión de plataforma heredada, debe migrarla a un nuevo entorno en el que se utilice una versión de plataforma no heredada para que pueda obtener acceso a las nuevas características. Si no sabe con seguridad si la aplicación se ejecuta o no en una versión heredada de la plataforma, puede comprobarlo en la consola de Elastic Beanstalk. Para obtener instrucciones, consulte [Para comprobar si está utilizando una versión heredada de la plataforma](#).

¿Qué nuevas características faltan en las versiones heredadas de la plataforma?

Las plataformas heredadas no admiten las siguientes características:

- Archivos de configuración, tal y como se describe en el tema [Personalización avanzada de entornos con archivos de configuración \(.ebextensions\)](#).
- Comprobaciones de estado de ELB, tal y como se describe en el tema [Informes de estado básicos](#).
- Perfiles de instancias, tal y como se describe en el tema [Administración de perfiles de instancia de Elastic Beanstalk](#).
- VPC, tal y como se describe en el tema [Uso de Elastic Beanstalk con Amazon VPC](#).
- Niveles de datos, tal y como se describe en el tema [Añadir una base de datos al entorno de Elastic Beanstalk](#).
- Niveles de trabajo, tal y como se describe en el tema [Entornos de trabajo](#).
- Entornos con una única instancia, tal y como se describe en el tema [Tipos de entornos](#).
- Etiquetas, tal y como se describe en el tema [Etiquetado de recursos en sus entornos de Elastic Beanstalk](#).
- Actualizaciones continuas, tal y como se describe en el tema [Actualizaciones de configuración del entorno continuas de Elastic Beanstalk](#).

¿Por qué algunas versiones de la plataforma están marcadas como heredadas?

Algunas versiones de plataforma antiguas no admiten las últimas características de Elastic Beanstalk. Estas versiones llevan la marca (legacy) [heredada] en la página de información general del entorno de la consola de Elastic Beanstalk.

Para comprobar si está utilizando una versión heredada de la plataforma

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En la página de información general del entorno, vea el nombre de la plataforma .

Si aparece (legacy) (heredada) junto al nombre de la plataforma, la aplicación utiliza una versión heredada de la plataforma.

Para migrar la aplicación

1. Implemente la aplicación en un nuevo entorno. Para obtener instrucciones, consulte [Creación de un entorno de Elastic Beanstalk](#).
2. Si tiene una instancia de base de datos de Amazon RDS, actualice el grupo de seguridad de la base de datos para permitir el acceso al grupo de seguridad de EC2 en el nuevo entorno. Para obtener instrucciones sobre cómo encontrar el nombre del grupo de seguridad de EC2 a través de la consola de administración de AWS, consulte [Grupos de seguridad](#). Para obtener más información sobre la configuración del grupo de seguridad de EC2, consulte la sección "Autorizar acceso a la red a un grupo de seguridad de Amazon EC2" en [Trabajando con grupos de seguridad de base de datos](#) en la Guía del usuario de Amazon Relational Database Service.
3. Cambie la URL del entorno. Para obtener instrucciones, consulte [Implementaciones Blue/Green \(azul/verde\) con Elastic Beanstalk](#).
4. Termine el entorno antiguo. Para obtener instrucciones, consulte [Terminación de un entorno de Elastic Beanstalk](#).

Note

Si utiliza AWS Identity and Access Management (IAM), tendrá que actualizar sus políticas para incluir AWS CloudFormation y Amazon RDS (si procede). Para obtener más información, consulte [Uso de Elastic Beanstalk con AWS Identity and Access Management](#).

Migración de su aplicación de Linux de Elastic Beanstalk a Amazon Linux 2023 o Amazon Linux 2

En esta sección se describe cómo migrar su aplicación mediante una de las siguientes rutas de migración.

- Migración de una ramificación de la plataforma Amazon Linux 2 a una ramificación de la plataforma Amazon Linux 2023.

- Migración de una ramificación de la plataforma AMI de Amazon Linux (AL1) a una ramificación de la plataforma Amazon Linux 2023 (recomendada) o Amazon Linux 2.

Temas

- [Migración de Amazon Linux 2 a Amazon Linux 2023](#)
- [Migración de la AMI de Linux Amazon \(AL1\) a AL2 o AL2023](#)

Migración de Amazon Linux 2 a Amazon Linux 2023

En este tema se ofrece orientación para migrar su aplicación de una ramificación de la plataforma Amazon Linux 2 a una ramificación de la plataforma Amazon Linux 2023.

Diferencias y compatibilidad

Entre las plataformas AL2 y AL2023 de Elastic Beanstalk

Existe un alto grado de compatibilidad entre las plataformas Amazon Linux 2 y Amazon Linux 2023 de Elastic Beanstalk. Sin embargo, hay algunas diferencias a destacar:

- Instance Metadata Service Version 1 (IMDSv1): La configuración de la opción [DisableIMDSv1](#) establece `true` como valor predeterminado en las plataformas AL2023. El valor predeterminado es `false` en las plataformas AL2.
- Herramienta de instancia pkg-repo: la herramienta [pkg-repo](#) no está disponible para entornos que se ejecutan en plataformas AL2023. Sin embargo, puede aplicar manualmente las actualizaciones del paquete y del sistema operativo a una instancia de AL2023. Para obtener más información, consulte [Administración de paquetes y actualizaciones del sistema operativo](#) en la Guía del usuario de Amazon Linux 2023
- Configuración HTTPd de Apache: el archivo `httpd.conf` de Apache para las plataformas AL2023 tiene algunos ajustes de configuración diferentes a los de AL2:
 - Denegar el acceso a todo el sistema de archivos del servidor de forma predeterminada. Esta configuración se describe en Proteger los archivos del servidor de forma predeterminada en la página [Consejos de seguridad](#) del sitio web de Apache.
 - Impedir que los usuarios anulen las funciones de seguridad que se hayan configurado. La configuración deniega el acceso a la configuración `.htaccess` en todos los directorios, excepto en los habilitados específicamente. Esta configuración se describe en Proteger la configuración del sistema en la página [Consejos de seguridad](#) del sitio web de Apache. La página [Tutorial del](#)

[servidor HTTP Apache: archivos.htaccess](#) indica que esta configuración puede ayudar a mejorar el rendimiento.

- Denegar el acceso a los archivos con `.ht*` como patrón de nombre. Esta configuración impide que los clientes web visualicen los archivos `.htaccess` y `.htpasswd`.

Puede cambiar cualquiera de las opciones de configuración de entorno anteriores. Para obtener más información, consulte [Ampliación de las plataformas Linux de Elastic Beanstalk](#). Amplíe el tema Proxy inverso para ver la sección Configuración de HTTPD de Apache.

Entre los sistemas operativos Amazon Linux

Para obtener más información sobre las diferencias entre los sistemas operativos Amazon Linux 2 y Amazon Linux 2023, consulte [Comparing Amazon Linux 2 and Amazon Linux 2023](#) en la Guía del usuario de Amazon Linux 2023.

Para obtener más información sobre Amazon Linux 2023, consulte [What is Amazon Linux 2023?](#) en la Guía del usuario de Amazon Linux 2023.

Proceso de migración general

Cuando esté listo para ir a producción, Elastic Beanstalk necesitará una implementación blue/green para realizar la actualización. Los siguientes pasos generales corresponden a las prácticas recomendadas y se aconsejan para la migración con un procedimiento de implementación azul/verde.

Preparación para realizar una prueba de migración

Antes de implementar su aplicación y comenzar a realizar pruebas, revise la información de la sección anterior [Diferencias y compatibilidad](#). Consulte también la referencia citada en esa sección, [Comparing Amazon Linux 2 and Amazon Linux 2023](#) en la Guía del usuario de Amazon Linux 2023. Anote la información específica de este contenido que sea aplicable o pueda aplicarse a la configuración de su aplicación y ajustes.

Pasos de migración de alto nivel

1. Cree un nuevo entorno que esté basado en una ramificación de la plataforma AL2023.
2. Implemente su aplicación en el entorno de AL2023 de destino.

El entorno de producción existente permanecerá activo y no se verá afectado, mientras realiza pruebas y ajustes en el nuevo entorno.

3. Pruebe su aplicación a fondo en el nuevo entorno.
4. Cuando su entorno de AL2023 de destino esté listo para pasar a producción, intercambie los CNAME de los dos entornos para redirigir el tráfico al nuevo entorno de AL2023.

Prácticas recomendadas y pasos de migración más detallados

Para ver un procedimiento de implementación azul/verde más detallado, consulte [Implementaciones Blue/Green \(azul/verde\) con Elastic Beanstalk](#).

Para obtener una orientación más específica y pasos detallados de las mejores prácticas, consulte [Método azul/verde](#).

Más referencias para ayudarle a planificar la migración

Las siguientes referencias pueden ofrecer información adicional para planificar la migración.

- [Plataformas compatibles con Elastic Beanstalk](#) en Plataformas de AWS Elastic Beanstalk
- [Historial de la ramificación de la plataforma retirada](#)
- [the section called “Plataformas Linux”](#)
- [Preguntas frecuentes sobre la retirada de plataformas](#)

Migración de la AMI de Linux Amazon (AL1) a AL2 o AL2023

Si la aplicación de Elastic Beanstalk está basada en una ramificación de la plataforma de la AMI de Amazon Linux, utilice esta sección para aprender a migrar los entornos de su aplicación a Amazon Linux 2 o Amazon Linux 2023. Las ramificaciones de la plataforma de la generación anterior basadas en la [AMI de Amazon Linux](#) se ha retirado ya.

Se recomienda encarecidamente que migre a Amazon Linux 2023, ya que es más reciente que Amazon Linux 2. El sistema operativo Amazon Linux 2 alcanzará la finalización del soporte antes que Amazon Linux 2023, de modo que si migra a Amazon Linux 2023 se beneficiará de un período de soporte más prolongado.

Merece la pena señalar que existe un alto grado de compatibilidad entre las plataformas Amazon Linux 2 y Amazon Linux 2023 de Elastic Beanstalk. Aunque algunas áreas presentan diferencias: la opción predeterminada Servicio de metadatos de instancia, versión 1 (IMDSv1), la compatibilidad con la herramienta de instancias pkg-repo y algunas configuraciones de Apache HTTPd. Para obtener más información, consultar [Amazon Linux 2023](#)

Diferencias y compatibilidad

No se garantiza que las ramificaciones de la plataforma basadas en AL2023/AL2 sean retrocompatibles con su aplicación existente. También conviene saber que aunque el código de la aplicación se implemente correctamente en la nueva versión de la plataforma, es posible que se comporte o funcione de forma distinta debido a las diferencias del sistema operativo y el entorno de ejecución.

Aunque la AMI de Amazon Linux y AL2023/AL2 comparten el mismo kernel de Linux, difieren en los siguientes aspectos: el sistema de inicialización, las versiones de `libc`, la cadena de herramientas del compilador y diversos paquetes. Para obtener más información, consulte las [Preguntas frecuentes de Amazon Linux 2](#).

El servicio Elastic Beanstalk también ha actualizado versiones específicas de la plataforma del tiempo de ejecución, las herramientas de compilación y otras dependencias.

Por tanto, le recomendamos que se tome su tiempo, pruebe la aplicación a fondo en un entorno de desarrollo y realice los ajustes necesarios.

Proceso de migración general

Cuando esté listo para ir a producción, Elastic Beanstalk necesitará una implementación blue/green para realizar la actualización. Los siguientes pasos generales corresponden a las prácticas recomendadas y se aconsejan para la migración con un procedimiento de implementación azul/verde.

Preparación para realizar una prueba de migración

Antes de implementar su aplicación y comenzar a realizar pruebas, revise la información de [Consideraciones aplicables a todas las plataformas Linux](#), que aparece más adelante en este tema. Además, revise la información que sea aplicable a su plataforma en la sección [Consideraciones específicas de la plataforma](#) que sigue. Anote la información específica de este contenido que sea aplicable o pueda aplicarse a la configuración de su aplicación y ajustes.

Pasos de migración de alto nivel

1. Cree un nuevo entorno que esté basado en una ramificación de la plataforma AL2 o AL2023. Se recomienda que migre a una ramificación de la plataforma AL2023.
2. Implemente su aplicación en el entorno de AL2023/AL2 de destino.

El entorno de producción existente permanecerá activo y no se verá afectado, mientras realiza pruebas y ajustes en el nuevo entorno.

3. Pruebe su aplicación a fondo en el nuevo entorno.
4. Cuando su entorno de AL2023/AL2 de destino esté listo para pasar a producción, intercambie los CNAME de los dos entornos para redirigir el tráfico al nuevo entorno.

Prácticas recomendadas y pasos de migración más detallados

Para ver un procedimiento de implementación azul/verde más detallado, consulte [Implementaciones Blue/Green \(azul/verde\) con Elastic Beanstalk](#).

Para obtener una orientación más específica y pasos detallados de las mejores prácticas, consulte [Método azul/verde](#).

Más referencias para ayudarle a planificar la migración

Las siguientes referencias pueden ofrecer información adicional para planificar la migración.

- [Comparing Amazon Linux 2 and Amazon Linux 2023](#) Guía del usuario de Amazon Linux 2023.
- [What is Amazon Linux 2023?](#) en la Guía del usuario de Amazon Linux 2023
- [Plataformas compatibles con Elastic Beanstalk](#) en Plataformas de AWS Elastic Beanstalk
- [Historial de la ramificación de la plataforma retirada](#)
- [the section called “Plataformas Linux”](#)
- [Preguntas frecuentes sobre la retirada de plataformas](#)

Consideraciones aplicables a todas las plataformas Linux

En la tabla siguiente se detallan algunas consideraciones que debe tener en cuenta al planificar la migración de una aplicación a AL2023/AL2. Estas consideraciones se aplican a cualquiera de las plataformas Linux de Elastic Beanstalk, sin tener en cuenta los lenguajes de programación o los servidores de aplicaciones específicos.

Área	Cambios e información
Archivos de	En las plataformas AL2023/AL2, puede utilizar archivos de configuración como antes, y todas las secciones funcionan del mismo modo. Sin embargo, es posible

Área	Cambios e información
configuración	<p>que algunas opciones no funcionen igual que lo hacían en las plataformas anteriores de la AMI de Amazon Linux. Por ejemplo:</p> <ul style="list-style-type: none"> • Es posible que algunos de los paquetes de software que instale utilizando un archivo de configuración no estén disponibles en AL2023/AL2, o que sus nombres hayan cambiado. • Algunas opciones de configuración específicas de la plataforma han cambiado los espacios de nombres específicos de la plataforma por otros diferentes que no dependen de ninguna plataforma. • Los archivos de configuración del proxy que se encuentran en el directorio <code>.ebextensions/nginx</code> deben moverse al directorio de enlaces de plataforma <code>.platform/nginx</code>. Para obtener más información, expanda la sección Configuración del proxy inverso de the section called “Ampliación de plataformas Linux”. <p>Es conveniente utilizar enlaces de plataforma para ejecutar código personalizado en las instancias del entorno. Puede seguir utilizando comandos y comandos de contenedor en los archivos de configuración <code>.ebextensions</code>, pero no resulta fácil trabajar con ellos. Por ejemplo, escribir scripts de comando en un archivo YAML puede resultar engorroso y difícil de probar.</p> <p>Tendrá que seguir utilizando los archivos de configuración <code>.ebextensions</code> con cualquier script que necesite una referencia a un recurso de AWS CloudFormation.</p>
Enlaces de la plataforma	<p>Las plataformas AL2 incorporaron un nuevo mecanismo para ampliar la plataforma de su entorno agregando archivos ejecutables a directorios de enlaces de las instancias del entorno. Es posible que, en las versiones anteriores de la plataforma Linux, haya utilizado enlaces de plataforma personalizados. Aunque estos enlaces no estaban diseñados para plataformas administradas y no eran compatibles, podían resultar útiles en algunos casos. En las versiones de la plataforma AL2023/AL2, los enlaces de plataforma personalizados no funcionan. Debe migrar estos enlaces a los nuevos enlaces de plataforma. Para obtener más información, expanda la sección Enlaces de plataforma de the section called “Ampliación de plataformas Linux”.</p>

Área	Cambios e información
Servidores proxy compatibles	<p>Las versiones de la plataforma AL2023/AL2 admiten los mismos servidores proxy inversos que cada plataforma admitida en sus versiones de la plataforma de la AMI de Amazon Linux. Todas las versiones de la plataforma AL2023/AL2 utilizan nginx como servidor proxy inverso predeterminado, con la excepción de las plataformas ECS y Docker. Las plataformas Tomcat, Node.js, PHP y Python también admiten Apache HTTPD como alternativa. Todas las plataformas habilitan la configuración del servidor proxy de manera uniforme, como se describe en esta sección. Sin embargo, la configuración del servidor proxy es ligeramente diferente de lo que era en la AMI de Amazon Linux. Estas son las diferencias para todas las plataformas:</p> <ul style="list-style-type: none">• El servidor predeterminado es nginx: el servidor proxy predeterminado en todas las versiones de la plataforma AL2023/AL2 es nginx. En las versiones de la plataforma de la AMI de Amazon Linux de Tomcat, PHP y Python, el servidor proxy predeterminado era Apache HTTPD.• Espacio de nombres coherente: todas las versiones de la plataforma AL2023/AL2 usan el espacio de nombres <code>aws:elasticbeanstalk:environment:proxy</code> para configurar el servidor proxy. En las versiones de la plataforma de la AMI de Amazon Linux, esto era una decisión por plataforma y Node.js usaba un espacio de nombres diferente.• Ubicación de los archivos de configuración: debe colocar los archivos de configuración del proxy en los directorios <code>.platform/nginx</code> y <code>.platform/httpd</code> en todas las versiones de la plataforma AL2023/AL2. En las versiones de la plataforma de la AMI de Amazon Linux estas ubicaciones eran <code>.ebextensions/nginx</code> y <code>.ebextensions/httpd</code>, respectivamente. <p>Para obtener información sobre los cambios de configuración de proxy específicos de la plataforma, consulte the section called “Consideraciones específicas de la plataforma”. Para obtener información sobre la configuración del proxy en plataformas AL2023/AL2, expanda la sección Configuración del proxy inverso de the section called “Ampliación de plataformas Linux”.</p>

Área	Cambios e información
Cambios en la configuración del proxy	<p>Hay cambios de configuración del proxy que se aplican uniformemente a todas las plataformas, además de los cambios de configuración del proxy que son específicos de cada plataforma. Es importante referirse a ambos para configurar con precisión sus entornos.</p> <ul style="list-style-type: none"> • Todas las plataformas: amplíe la sección Configuración del proxy inverso en the section called “Ampliación de plataformas Linux”. • Específicos de la plataforma: consulte the section called “Consideraciones específicas de la plataforma”.
Perfil de instancia	<p>En las plataformas AL2023/AL2 es necesario configurar un perfil de instancia. Si se crea un entorno sin un perfil de instancia, este podría funcionar temporalmente; sin embargo, es posible que pronto se produjeran errores, cuando las acciones que requieren un perfil de instancia no pudieran ejecutarse correctamente. Para obtener más información, consulte the section called “Perfiles de instancias”.</p>
Estado mejorado	<p>Las versiones de la plataforma AL2023/AL2 habilitan el estado mejorado de manera predeterminada. Esto supone un cambio si no utiliza la consola de Elastic Beanstalk para crear sus entornos. De forma predeterminada, siempre que es posible, la consola habilita el estado mejorado, independientemente de la versión de la plataforma. Para obtener más información, consulte the section called “Informes y monitorización de estado mejorados”.</p>
AMI personalizada	<p>Si su entorno utiliza una AMI personalizada, cree una nueva AMI basada en AL2023/AL2 para el nuevo entorno que utilice una plataforma AL2023/AL2 de Elastic Beanstalk.</p>
Plataformas personalizadas	<p>Las AMI administradas de las versiones de la plataforma AL2023/AL2 no son compatibles con plataformas personalizadas.</p>

Consideraciones específicas de la plataforma

En esta sección se describen consideraciones de migración específicas de determinadas plataformas Linux de Elastic Beanstalk.

Docker

La familia de ramificaciones de la plataforma Docker basada en la AMI de Amazon Linux (AL1) incluye tres ramificaciones de plataforma. Se recomienda una ruta de migración distinta para cada una.

Ramificación de la plataforma a AL1	Ruta de migración a AL2023/AL2
<p>Multicontainer Docker administrado por Amazon ECS que se ejecuta en la AMI de Amazon Linux</p>	<p>Ramificaciones de la plataforma AL2023/AL2 de Docker basado en ECS</p> <p>La ramificación de la plataforma AL2023/AL2 de Docker basado en ECS ofrece una ruta de migración directa para los entornos que se ejecuten en la ramificación de la plataforma Docker multicontenedor AL1.</p> <ul style="list-style-type: none"> Al igual que la ramificación Docker multicontenedor AL1 anterior, las nuevas ramificaciones de la plataforma AL2023/AL2 utilizan Amazon ECS para coordinar la implementación de múltiples contenedores de Docker en un clúster de Amazon ECS dentro de un entorno de Elastic Beanstalk. Las nuevas ramificaciones de la plataforma AL2023/AL2 admiten todas las características de la ramificación de la plataforma Docker multicontenedor AL1 anterior. Las ramificaciones de la plataforma AL2023/AL2 también admiten el mismo archivo <code>DockerRun.aws.json v2</code>. <p>Para obtener más información sobre cómo migrar las aplicaciones que se ejecuten en la ramificación de la plataforma Docker multicontenedor Amazon Linux a una ramificación de la plataforma Amazon ECS sobre AL2023/AL2, consulte ???.</p>
<p>Docker que se ejecuta en la AMI de Amazon Linux</p>	<p>Ramificación de la plataforma Docker sobre AL2023/AL2</p> <p>Se recomienda migrar las aplicaciones que se ejecuten en entornos basados en Docker preconfigurado (GlassFish 5.0) o Docker sobre la AMI de Amazon Linux (AL1) a entornos que estén basados en las ramificaciones de la plataforma Docker sobre Amazon Linux 2 o Docker sobre AL2023.</p>

Ramificación de la plataforma a AL1	Ruta de migración a AL2023/AL2							
Docker preconfigurado (Glassfish 5.0) que se ejecuta en la AMI de Amazon Linux	<p>Si su entorno está basado en la ramificación de la plataforma del Docker preconfigurado (Glassfish 5.0), consulte the section called “Tutorial: GlassFish en Docker: camino a Amazon Linux 2023”.</p> <p>La siguiente tabla muestra información sobre la migración específica de la ramificación de la plataforma Docker sobre AL2023/AL2.</p> <table border="1" data-bbox="321 672 1507 1885"> <thead> <tr> <th data-bbox="321 672 490 751">Área</th> <th data-bbox="490 672 1507 751">Cambios e información</th> </tr> </thead> <tbody> <tr> <td data-bbox="321 751 490 1722">Almacenamiento</td> <td data-bbox="490 751 1507 1722"> <p>Elastic Beanstalk configura Docker de tal forma que use controladores de almacenamiento para almacenar los datos de contenidos y las imágenes de Docker. En la AMI de Amazon Linux, Elastic Beanstalk ha utilizado el controlador de almacenamiento Device Mapper. Para mejorar el rendimiento, Elastic Beanstalk aprovisionó un volumen adicional de Amazon EBS. En las versiones de la plataforma Docker de AL2023/AL2, Elastic Beanstalk utiliza el controlador de almacenamiento OverlayFS, y logra un rendimiento aún mejor sin necesidad de un volumen independiente.</p> <p>Con la AMI de Amazon Linux, si ha utilizado la opción <code>BlockDeviceMappings</code> del espacio de nombres <code>aws:autoscaling:launchconfiguration</code> para agregar volúmenes de almacenamiento personalizados a un entorno de Docker, le aconsejamos que también agregue el volumen de Amazon EBS <code>/dev/xvdcz</code> que aprovisiona Elastic Beanstalk. Elastic Beanstalk ya no aprovisiona este volumen, por lo que debería eliminarlo de los archivos de configuración. Para obtener más información, consulte the section called “Configuración de Docker en la AMI de Amazon Linux (anterior a Amazon Linux 2)”.</p> </td> </tr> <tr> <td data-bbox="321 1722 490 1900">Autenticación de repositorio</td> <td data-bbox="490 1722 1507 1900"> <p>Cuando proporciona un archivo de autenticación generado por Docker para conectarse a un repositorio privado, ya no es necesario convertirlo al formato anterior que requerían las versiones de la</p> </td> </tr> </tbody> </table>		Área	Cambios e información	Almacenamiento	<p>Elastic Beanstalk configura Docker de tal forma que use controladores de almacenamiento para almacenar los datos de contenidos y las imágenes de Docker. En la AMI de Amazon Linux, Elastic Beanstalk ha utilizado el controlador de almacenamiento Device Mapper. Para mejorar el rendimiento, Elastic Beanstalk aprovisionó un volumen adicional de Amazon EBS. En las versiones de la plataforma Docker de AL2023/AL2, Elastic Beanstalk utiliza el controlador de almacenamiento OverlayFS, y logra un rendimiento aún mejor sin necesidad de un volumen independiente.</p> <p>Con la AMI de Amazon Linux, si ha utilizado la opción <code>BlockDeviceMappings</code> del espacio de nombres <code>aws:autoscaling:launchconfiguration</code> para agregar volúmenes de almacenamiento personalizados a un entorno de Docker, le aconsejamos que también agregue el volumen de Amazon EBS <code>/dev/xvdcz</code> que aprovisiona Elastic Beanstalk. Elastic Beanstalk ya no aprovisiona este volumen, por lo que debería eliminarlo de los archivos de configuración. Para obtener más información, consulte the section called “Configuración de Docker en la AMI de Amazon Linux (anterior a Amazon Linux 2)”.</p>	Autenticación de repositorio	<p>Cuando proporciona un archivo de autenticación generado por Docker para conectarse a un repositorio privado, ya no es necesario convertirlo al formato anterior que requerían las versiones de la</p>
Área	Cambios e información							
Almacenamiento	<p>Elastic Beanstalk configura Docker de tal forma que use controladores de almacenamiento para almacenar los datos de contenidos y las imágenes de Docker. En la AMI de Amazon Linux, Elastic Beanstalk ha utilizado el controlador de almacenamiento Device Mapper. Para mejorar el rendimiento, Elastic Beanstalk aprovisionó un volumen adicional de Amazon EBS. En las versiones de la plataforma Docker de AL2023/AL2, Elastic Beanstalk utiliza el controlador de almacenamiento OverlayFS, y logra un rendimiento aún mejor sin necesidad de un volumen independiente.</p> <p>Con la AMI de Amazon Linux, si ha utilizado la opción <code>BlockDeviceMappings</code> del espacio de nombres <code>aws:autoscaling:launchconfiguration</code> para agregar volúmenes de almacenamiento personalizados a un entorno de Docker, le aconsejamos que también agregue el volumen de Amazon EBS <code>/dev/xvdcz</code> que aprovisiona Elastic Beanstalk. Elastic Beanstalk ya no aprovisiona este volumen, por lo que debería eliminarlo de los archivos de configuración. Para obtener más información, consulte the section called “Configuración de Docker en la AMI de Amazon Linux (anterior a Amazon Linux 2)”.</p>							
Autenticación de repositorio	<p>Cuando proporciona un archivo de autenticación generado por Docker para conectarse a un repositorio privado, ya no es necesario convertirlo al formato anterior que requerían las versiones de la</p>							

Ramificación de la plataforma a AL1	Ruta de migración a AL2023/AL2	
	Área	Cambios e información
	io privado	<p>plataforma Docker de la AMI de Amazon Linux. Las versiones de la plataforma Docker de AL2023/AL2 admiten el nuevo formato. Para obtener más información, consulte the section called “Uso de imágenes de un repositorio privado”.</p>
	Servidor proxy	<p>Las versiones de la plataforma Docker de AL2023/AL2 no admiten contenedores independientes que no se ejecuten detrás de un servidor proxy. En las versiones de la plataforma Docker de la AMI de Amazon Linux, esto solía ser posible a través del valor none de la opción ProxyServer en el espacio de nombres <code>aws:elasticbeanstalk:environment:proxy</code>.</p>

Go

La siguiente tabla muestra información sobre la migración de las versiones de la plataforma AL2023/AL2 en la [plataforma Go](#).

Área	Cambios e información
Transferencia de puertos	<p>En las plataformas AL2023/AL2, Elastic Beanstalk no pasa ningún valor de puerto al proceso de la aplicación a través de la variable de entorno PORT. Si desea simular este comportamiento, puede configurar una propiedad de entorno PORT. Sin embargo, si tiene varios procesos y prevé que Elastic Beanstalk va a pasar valores de puerto incrementales a los procesos (5000, 5100, 5200, etc.), debe modificar la implementación. Para obtener más información, expanda la sección Configuración del proxy inverso de the section called “Ampliación de plataformas Linux”.</p>

Amazon Corretto

La siguiente tabla muestra información sobre la migración de las ramificaciones de la plataforma Corretto en la [plataforma Java SE](#).

Área	Cambios e información
Corretto en comparación con OpenJDK	Para implementar la plataforma Java, Standard Edition (Java SE), las ramificaciones de la plataforma AL2023/AL2 utilizan Amazon Corretto , una distribución de AWS del Open Java Development Kit (OpenJDK). Las ramificaciones anteriores de la plataforma Java SE de Elastic Beanstalk utilizan los paquetes OpenJDK que se incluyen con la AMI de Amazon Linux.
Herramientas de compilación	Las plataformas AL2023/AL2 tienen versiones más recientes de las herramientas de compilación: gradle, maven y ant.
Administración de archivos JAR	En las plataformas AL2023/AL2, si su paquete de código fuente (archivo ZIP) contiene un único archivo JAR y ningún otro archivo, Elastic Beanstalk ya no cambia el nombre del archivo JAR a <code>application.jar</code> . Este cambio de nombre solo se produce si el archivo JAR se envía solo, y no dentro de un archivo ZIP.
Transferencia de puertos	En las plataformas AL2023/AL2, Elastic Beanstalk no pasa ningún valor de puerto al proceso de la aplicación a través de la variable de entorno PORT. Si desea simular este comportamiento, puede configurar una propiedad de entorno PORT. Sin embargo, si tiene varios procesos y prevé que Elastic Beanstalk va a pasar valores de puerto incrementales a los procesos (5000, 5100, 5200, etc.), debe modificar la implementación. Para obtener más información, expanda la sección Configuración del proxy inverso de the section called “Ampliación de plataformas Linux” .
Java 7	Elastic Beanstalk no admite una ramificación de la plataforma Java 7 de AL2023/AL2. Si tiene una aplicación Java 7, mírela a Corretto 8 o a Corretto 11.

Tomcat

La siguiente tabla muestra información sobre la migración de las versiones de la plataforma AL2023/AL2 en la [plataforma Tomcat](#).

Área	Cambios e información						
Opciones de configuración	<p>En las versiones de la plataforma AL2023/AL2, Elastic Beanstalk solo admite un subconjunto de las opciones de configuración y los valores de opción en el espacio de nombres <code>aws:elasticbeanstalk:environment:proxy</code>. A continuación se muestra la información de migración para cada opción.</p> <table border="1"> <thead> <tr> <th>Opción</th> <th>Información de migración</th> </tr> </thead> <tbody> <tr> <td><code>GzipCompression</code></td> <td>No se admite en las versiones de la plataforma AL2023/AL2.</td> </tr> <tr> <td><code>ProxyServer</code></td> <td> <p>Las versiones de la plataforma Tomcat de AL2023/AL2 admiten los servidores proxy nginx y Apache HTTPD versión 2.4. Sin embargo, la versión 2.2 de Apache no es compatible.</p> <p>En las versiones de la plataforma de la AMI de Amazon Linux, el proxy predeterminado era Apache 2.4. Si ha utilizado la configuración de proxy predeterminada y ha agregado archivos de configuración del proxy personalizados, la configuración del proxy debería seguir funcionando en AL2023/AL2. Sin embargo, si utilizó el valor de opción <code>apache/2.2</code>, ahora tiene que migrar su configuración de proxy a la versión 2.4 de Apache.</p> </td> </tr> </tbody> </table> <p>La opción <code>XX:MaxPermSize</code> del espacio de nombres <code>aws:elasticbeanstalk:container:tomcat:jvmoptions</code> no se admite en las versiones de la plataforma AL2023/AL2. La configuración de JVM para modificar el tamaño de la generación permanente se aplica solo a Java 7 y versiones anteriores, y por lo tanto no es aplicable a las versiones de la plataforma AL2023/AL2.</p>	Opción	Información de migración	<code>GzipCompression</code>	No se admite en las versiones de la plataforma AL2023/AL2.	<code>ProxyServer</code>	<p>Las versiones de la plataforma Tomcat de AL2023/AL2 admiten los servidores proxy nginx y Apache HTTPD versión 2.4. Sin embargo, la versión 2.2 de Apache no es compatible.</p> <p>En las versiones de la plataforma de la AMI de Amazon Linux, el proxy predeterminado era Apache 2.4. Si ha utilizado la configuración de proxy predeterminada y ha agregado archivos de configuración del proxy personalizados, la configuración del proxy debería seguir funcionando en AL2023/AL2. Sin embargo, si utilizó el valor de opción <code>apache/2.2</code>, ahora tiene que migrar su configuración de proxy a la versión 2.4 de Apache.</p>
Opción	Información de migración						
<code>GzipCompression</code>	No se admite en las versiones de la plataforma AL2023/AL2.						
<code>ProxyServer</code>	<p>Las versiones de la plataforma Tomcat de AL2023/AL2 admiten los servidores proxy nginx y Apache HTTPD versión 2.4. Sin embargo, la versión 2.2 de Apache no es compatible.</p> <p>En las versiones de la plataforma de la AMI de Amazon Linux, el proxy predeterminado era Apache 2.4. Si ha utilizado la configuración de proxy predeterminada y ha agregado archivos de configuración del proxy personalizados, la configuración del proxy debería seguir funcionando en AL2023/AL2. Sin embargo, si utilizó el valor de opción <code>apache/2.2</code>, ahora tiene que migrar su configuración de proxy a la versión 2.4 de Apache.</p>						

Área	Cambios e información
Ruta de la aplicación	En las plataformas AL2023/AL2, la ruta del directorio de la aplicación en las instancias de Amazon EC2 de su entorno es <code>/var/app/current</code> . Estaba <code>/var/lib/tomcat8/webapps</code> en plataformas de la AMI de Amazon Linux.

Node.js

La siguiente tabla muestra información sobre la migración de las versiones de la plataforma AL2023/AL2 en la [plataforma Node.js](#).

Área	Cambios e información
Versiones de Node.js instaladas	<p>En las plataformas AL2023/AL2, Elastic Beanstalk mantiene varias ramificaciones de la plataforma Node.js y solo instala la versión más reciente de la versión principal de Node.js correspondiente a la ramificación de la plataforma de cada versión de la plataforma. Por ejemplo, cada versión de plataforma de la ramificación de la plataforma Node.js 12 solo tiene la versión Node.js 12.x.y instalada de forma predeterminada. En las versiones de la plataforma de la AMI de Amazon Linux, hemos instalado las múltiples versiones de varias versiones de Node.js en cada versión de la plataforma y hemos mantenido únicamente una ramificación de la plataforma.</p> <p>Seleccione la ramificación de la plataforma Node.js que se corresponda con la versión principal de Node.js que necesita su aplicación.</p>
Nombres de archivos de registro HTTPD de Apache	<p>En las plataformas AL2023/AL2, si utiliza el servidor proxy Apache HTTPD, los nombres de archivo de registro de HTTPD son <code>access_log</code> y <code>error_log</code> , lo cual es coherente con todas las demás plataformas compatibles con Apache HTTPD. En las versiones de la plataforma de la AMI de Amazon Linux, estos archivos de registro se denominaban <code>access.log</code> y <code>error.log</code> , respectivamente.</p> <p>Para obtener más información sobre los nombres de archivos de registro y las ubicaciones de todas las plataformas, consulte the section called “Cómo Elastic Beanstalk configura CloudWatch Logs”.</p>

Área	Cambios e información
Opciones de configuración	<p>En las plataformas AL2023/AL2, Elastic Beanstalk no admite las opciones de configuración en el espacio de nombres <code>aws:elasticbeanstalk:container:nodejs</code>. Algunas de las opciones tienen alternativas. A continuación se muestra la información de migración para cada opción.</p>
Opción	Información de migración
NodeCommand	<p>Utilice un <code>Procfile</code> o la palabra clave <code>scripts</code> en un archivo <code>package.json</code> para especificar el script de inicio.</p>
NodeVersion	<p>utilice la palabra clave <code>engines</code> en un archivo <code>package.json</code> para especificar la versión de Node.js. Tenga en cuenta que solo puede especificar una versión de Node.js que corresponde con su ramificación de la plataforma. Por ejemplo, si está utilizando la ramificación de la plataforma para Node.js 12, puede especificar solo una versión de Node.js 12.x.y. Para obtener más información, consulte the section called “Especificación de dependencias de Node.js con un archivo package.json”.</p>
GzipCompression	<p>No se admite en las versiones de la plataforma AL2023/AL2.</p>
ProxyServer	<p>En las versiones de la plataforma Node.js de AL2023/AL2, esta opción se trasladó al espacio de nombres <code>aws:elasticbeanstalk:environment:proxy</code>. Puede elegir entre <code>nginx</code> (el valor predeterminado) y <code>apache</code>.</p> <p>Las versiones de la plataforma Node.js de AL2023/AL2 no admiten aplicaciones independientes que no se ejecuten detrás de un servidor proxy. En las versiones de la plataforma Node.js de la AMI de Amazon Linux, esto solía ser posible a través del valor <code>none</code> de la opción <code>ProxyServer</code> en el espacio de nombres <code>aws:elasticbeanstalk:container:nodejs</code>. Si su entorno ejecuta una aplicación independiente, actualice el código para escuchar al puerto al que el servidor proxy (<code>nginx</code> o <code>Apache</code>) reenvía tráfico.</p>

Área	Cambios e información	
	Opción	Información de migración
		<pre>var port = process.env.PORT 5000; app.listen(port, function() { console.log('Server running at http://127.0.0.1:%s', port); });</pre>

PHP

La siguiente tabla muestra información sobre la migración de las versiones de la plataforma AL2023/AL2 en la [plataforma PHP](#).

Área	Cambios e información
Procesamiento de archivos PHP	En las plataformas AL2023/AL2, los archivos PHP se procesan mediante PHP-FPM (un administrador de procesos CGI). En las plataformas de la AMI de Amazon Linux utilizábamos mod_php (un módulo Apache).
Servidor proxy	<p>Las versiones de la plataforma PHP de AL2023/AL2 admiten los servidores proxy nginx y Apache HTTPD. El valor predeterminado es nginx.</p> <p>Las versiones de la plataforma PHP de la AMI de Amazon Linux solo admiten Apache HTTPD. Si agregó archivos de configuración personalizados de Apache, puede establecer la opción <code>ProxyServer</code> en el espacio de nombres <code>aws:elasticbeanstalk:environment:proxy</code> en <code>apache</code>.</p>

Python

La siguiente tabla muestra información sobre la migración de las versiones de la plataforma AL2023/AL2 en la [plataforma Python](#).

Área	Cambios e información
Servidor WSGI	<p>En las plataformas AL2023/AL2, Gunicorn es el servidor WSGI predeterminado. De forma predeterminada, Gunicorn escucha en el puerto 8000. El puerto puede ser distinto al que usó su aplicación en la plataforma de la AMI de Amazon Linux. Si está configurando la opción <code>WSGIPath</code> del espacio de nombres <code>aws:elasticbeanstalk:container:python</code>, reemplace el valor con la sintaxis de Gunicorn. Para obtener más información, consulte the section called “Espacios de nombres de configuración de Python”.</p> <p>Alternativamente, puede utilizar un <code>Procfile</code> para especificar y configurar el servidor WSGI. Para obtener más información, consulte the section called “Procfile”.</p>
Ruta de la aplicación	<p>En las plataformas AL2023/AL2, la ruta del directorio de la aplicación en las instancias de Amazon EC2 de su entorno es <code>/var/app/current</code>. Estaba <code>/opt/python/current/app</code> en plataformas de la AMI de Amazon Linux.</p>
Servidor proxy	<p>Las versiones de la plataforma Python de AL2023/AL2 admiten los servidores proxy <code>nginx</code> y <code>Apache HTTPD</code>. El valor predeterminado es <code>nginx</code>.</p> <p>Las versiones de la plataforma Python de la AMI de Amazon Linux solo admiten <code>Apache HTTPD</code>. Si agregó archivos de configuración personalizados de <code>Apache</code>, puede establecer la opción <code>ProxyServer</code> en el espacio de nombres <code>aws:elasticbeanstalk:environment:proxy</code> en <code>apache</code>.</p>

Ruby

La siguiente tabla muestra información sobre la migración de las versiones de la plataforma AL2023/AL2 en la [plataforma Ruby](#).

Área	Cambios e información
Versiones de Ruby instaladas	<p>En las plataformas AL2023/AL2, Elastic Beanstalk solo instala la versión más reciente de una sola versión de Ruby, correspondiente a la ramificación de la plataforma, en cada versión de la plataforma. Por ejemplo, cada versión de la plataforma en la ramificación de la plataforma Ruby 2.6 solo tiene Ruby 2.6.x instalado. En las versiones de la plataforma de la AMI de Amazon Linux, instalamos</p>

Área	Cambios e información
	<p>las versiones más recientes de varias versiones de Ruby, por ejemplo, 2.4.x, 2.5.x y 2.6.x.</p> <p>Si su aplicación utiliza una versión de Ruby que no se corresponde con la ramificación de la plataforma que está utilizando, le recomendamos que cambie a una ramificación de la plataforma que tenga la versión correcta de Ruby para su aplicación.</p>
Servidor de aplicaciones	<p>En las plataformas AL2023/AL2, Elastic Beanstalk solo instala el servidor de aplicaciones Puma en todas las versiones de la plataforma Ruby. Puede usar un <code>Procfile</code> para iniciar un servidor de aplicaciones diferente y un <code>Gemfile</code> para instalarlo.</p> <p>En la plataforma de la AMI de Amazon Linux, admitíamos dos tipos de ramificaciones de la plataforma para cada versión de Ruby: una con el servidor de aplicaciones Puma y la otra con el servidor de aplicaciones Passenger. Si su aplicación utiliza Passenger, puede configurar su entorno de Ruby para instalar y usar Passenger.</p> <p>Para obtener más información y ejemplos, consulte the section called “La plataforma a Ruby”.</p>

Preguntas frecuentes sobre la retirada de plataformas

Note

Elastic Beanstalk retiró todas las ramificaciones de la plataforma basadas en la AMI de Amazon Linux (AL1) el 18 de julio de 2022.

Las respuestas de estas preguntas frecuentes hacen referencia a los siguientes temas:

- [Política de compatibilidad con plataformas Elastic Beanstalk](#)
- [Historial de la ramificación de la plataforma retirada](#)
- [Plataformas compatibles con Elastic Beanstalk](#) en Plataformas de AWS Elastic Beanstalk

- [Migración de su aplicación de Linux de Elastic Beanstalk a Amazon Linux 2023 o Amazon Linux 2](#)
- [Preguntas frecuentes sobre Amazon Linux 2.](#)

1. ¿Qué significa la retirada de una ramificación de plataforma?

Tras la fecha anunciada de retirada de una ramificación de la plataforma, ya no podrá crear un nuevo entorno basado en la ramificación de la plataforma retirada, a menos que ya tenga un entorno activo basado en esa ramificación de la plataforma. Para obtener más información, consulte la [Pregunta frecuente n.º 11](#). Elastic Beanstalk dejará de proporcionar nuevas actualizaciones de mantenimiento para estas ramificaciones de la plataforma. No se recomienda utilizar una ramificación retirada de la plataforma en entornos de producción. Para obtener más información, consulte la [Pregunta frecuente n.º 5](#).

2. ¿Por qué AWS se han retirado las sucursales de plataformas basadas en AL1?

Elastic Beanstalk retira las ramificaciones de la plataforma cuando los componentes de la plataforma quedan obsoletos o son retirados por sus proveedores. En este caso, la AMI de Amazon Linux (AL1) ha finalizado el soporte estándar a partir del [31 de diciembre de 2020](#). Aunque Elastic Beanstalk continuó ofreciendo las plataformas basadas en AL1 durante 2022, desde entonces hemos lanzado muchas plataformas basadas en AL2 y AL2023 que incluyen las características más recientes. Para que los clientes sigan beneficiándose de la seguridad y las características más recientes en lo sucesivo, es fundamental que migren a nuestras plataformas basadas en AL2 o AL2023.

3. ¿Qué ramificaciones de la plataforma se han retirado?

Para obtener una lista de los componentes de la plataforma y las ramificaciones de plataforma que se han retirado, consulte [Historial de la ramificación de la plataforma retirada](#).

4. ¿Qué plataformas son compatibles actualmente?

Consulte las [plataformas compatibles con Elastic Beanstalk](#) en Plataformas de AWS Elastic Beanstalk .

5. ¿Elastic Beanstalk eliminará o dará de baja algún componente de mi entorno luego de la retirada?

Nuestra política para las sucursales de plataformas retiradas no elimina el acceso a los entornos ni elimina los recursos. Sin embargo, un entorno basado en una ramificación de la plataforma retirada puede acabar en una situación imprevisible, ya que Elastic Beanstalk no puede proporcionar

actualizaciones de seguridad, soporte técnico o revisiones para las ramificaciones de la plataforma retirada debido a que el proveedor marca su componente como Fin de vida útil (EOL). Por ejemplo, puede surgir una vulnerabilidad de seguridad perjudicial y crítica en un entorno que se ejecuta en una ramificación de la plataforma retirada. O una acción de la API de EB puede dejar de funcionar para el entorno si con el tiempo se vuelve incompatible con el servicio de Elastic Beanstalk. La posibilidad de este tipo de riesgos aumenta cuanto más tiempo permanezca activo un entorno basado en una ramificación de la plataforma retirada.

Si tu aplicación tiene problemas mientras se ejecuta en una rama de plataforma retirada y no puedes migrarla a una plataforma compatible, tendrás que considerar otras alternativas. Las soluciones alternativas incluyen encapsular la aplicación en una imagen de Docker para ejecutarla como contenedor de Docker. Esto permitiría a un cliente utilizar cualquiera de nuestras soluciones de Docker, como nuestras plataformas Docker de Elastic Beanstalk AL2023/AL2, u otros servicios basados en Docker, como Amazon ECS o Amazon EKS. Las alternativas que no son de Docker incluyen nuestro AWS CodeDeploy servicio, que permite una personalización completa de los tiempos de ejecución que desee.

6. ¿Puedo enviar una solicitud para ampliar la fecha de retirada?

No. Una vez transcurrida la fecha de retirada, los entornos existentes seguirán funcionando. Sin embargo, Elastic Beanstalk ya no proporcionará actualizaciones de seguridad ni de mantenimiento de la plataforma. Por lo tanto, es fundamental que migre a AL2 o AL2023 si aún ejecuta aplicaciones en una plataforma basada en AL1. Para obtener más información sobre los riesgos y las soluciones alternativas, consulte la [Pregunta frecuente n.º 5](#).

7. ¿Qué soluciones alternativas existen si no puedo completar la migración a AL2 o AL2023 a tiempo?

Los clientes pueden seguir ejecutando el entorno, aunque les recomendamos encarecidamente que planifiquen la migración de todos sus entornos de Elastic Beanstalk a una versión de plataforma compatible. De este modo, se minimizará el riesgo y se seguirán aprovechando las importantes mejoras de seguridad, rendimiento y funcionalidad que se ofrecen en las versiones más recientes. Para obtener más información sobre los riesgos y las soluciones alternativas, consulte la [Pregunta frecuente n.º 5](#).

8. ¿Cuál es el proceso recomendado para migrar a las plataformas AL2 o AL2023?

Para obtener instrucciones completas sobre la migración de AL1 a AL2/AL2023, consulte [Migración de su aplicación de Linux de Elastic Beanstalk a Amazon Linux 2023 o Amazon Linux 2](#). En este

tema, se explica que Elastic Beanstalk necesita una implementación azul/verde para realizar la actualización.

9. Si tengo un entorno que se ejecuta en una plataforma retirada, ¿cuál sería el impacto?

Un entorno basado en una ramificación de la plataforma retirada puede acabar en una situación imprevisible, ya que Elastic Beanstalk no puede proporcionar actualizaciones de seguridad, soporte técnico o revisiones para las ramificaciones de la plataforma retirada debido a que el proveedor marca su componente como Fin de vida útil (EOL). Por ejemplo, puede surgir una vulnerabilidad de seguridad perjudicial y crítica en un entorno que se ejecuta en una ramificación de la plataforma retirada. O una acción de la API de EB puede dejar de funcionar para el entorno si con el tiempo se vuelve incompatible con el servicio de Elastic Beanstalk. La posibilidad de que se produzcan este tipo de riesgos aumenta cuanto más tiempo permanezca activo un entorno en una ramificación de la plataforma retirada. Para obtener más información, consulte la [Pregunta frecuente n.º 5](#).

10. ¿Qué ocurre 90 días después de la fecha de retirada?

Nuestra política para las sucursales de plataformas retiradas no elimina el acceso a los entornos ni elimina los recursos. Sin embargo, tenga en cuenta que un entorno basado en una ramificación de la plataforma retirada puede acabar en una situación imprevisible, ya que Elastic Beanstalk no puede proporcionar actualizaciones de seguridad, soporte técnico o revisiones para las ramificaciones de la plataforma retirada debido a que el proveedor marca su componente como Fin de vida útil (EOL). Por ejemplo, puede surgir una vulnerabilidad de seguridad perjudicial y crítica en un entorno que se ejecuta en una ramificación de la plataforma retirada. O una acción de la API de EB puede dejar de funcionar para el entorno si con el tiempo se vuelve incompatible con el servicio de Elastic Beanstalk. La posibilidad de que se produzcan este tipo de riesgos aumenta cuanto más tiempo permanezca activo un entorno en una ramificación de la plataforma retirada. Para obtener más información, consulte la [Pregunta frecuente n.º 5](#).

11. ¿Puedo crear un nuevo entorno basado en una plataforma retirada?

Puede crear un nuevo entorno basado en una ramificación de la plataforma retirada, si ya ha utilizado esa ramificación de plataforma para crear un entorno existente utilizando la misma cuenta y en la misma región. La rama de plataforma retirada no estará disponible en la consola de Elastic Beanstalk. Sin embargo, para los clientes que tengan entornos existentes basados en una ramificación de la plataforma retirada, estará disponible a través de la CLI de EB, la API de EB y la AWS CLI. Además, los clientes existentes pueden utilizar las consolas para [Clonar un entorno](#) y

[Volver a crear un entorno](#). Sin embargo, tenga en cuenta que un entorno basado en una ramificación de la plataforma retirada puede acabar en una situación imprevisible. Para obtener más información, consulte la [Pregunta frecuente n.º 5](#).

12. Si tengo un entorno existente que se ejecuta en una rama de plataforma retirada, ¿hasta cuándo puedo crear un entorno nuevo basado en la rama de plataforma retirada? ¿Puedo hacerlo mediante la consola, la CLI o la API?

Puede crear el entorno después de la fecha de retirada. Sin embargo, hay que tener en cuenta que una ramificación de la plataforma retirada puede acabar en una situación imprevisible. Cuanto más lejos en el tiempo se cree o se encuentre activo un entorno de este tipo, mayor será el riesgo de que el entorno encuentre problemas inesperados. Para obtener más información sobre cómo crear un nuevo entorno, consulte la [Pregunta frecuente n.º 11](#).

13. ¿Puedo clonar o volver a crear mi entorno basado en una plataforma retirada?

Sí. Puede hacerlo mediante las consolas para [Clonar un entorno](#) y [Volver a crear un entorno](#). También puede utilizar la CLI de EB, la API de EB y AWS CLI. Para obtener más información sobre cómo crear un nuevo entorno, consulte la [Pregunta frecuente n.º 11](#).

Sin embargo, le recomendamos encarecidamente que planifique la migración de todos sus entornos de Elastic Beanstalk a una versión de plataforma compatible. De este modo, se minimizará el riesgo y se seguirán aprovechando las importantes mejoras de seguridad, rendimiento y funcionalidad que se ofrecen en las versiones más recientes. Para obtener más información sobre los riesgos y las soluciones alternativas, consulte la [Pregunta frecuente n.º 5](#).

14. Tras la fecha de retirada, ¿qué pasaría con los AWS recursos de mi entorno de Elastic Beanstalk que está basado en una rama de plataforma retirada? Por ejemplo, si la instancia de EC2 que se está ejecutando se termina, ¿podría Elastic Beanstalk lanzar una nueva instancia de EC2 basada en AL1 para mantener la capacidad?

Los recursos del entorno permanecerían activos y seguirían funcionando. Y, sí, Elastic Beanstalk escalará automáticamente para las instancias de EC2 de AL1 en el entorno. Sin embargo, Elastic Beanstalk dejará de proporcionar nuevas actualizaciones de mantenimiento de la plataforma al entorno, lo que puede hacer que el entorno acabe en una situación imprevisible con el tiempo. Para obtener más información, consulte la [Pregunta frecuente n.º 5](#).

15. ¿Cuáles son las principales diferencias entre los sistemas operativos de AL2/AL2023 y la AMI de Amazon Linux (AL1)? ¿Cómo se ven afectadas las ramificaciones de la plataforma AL2023/AL2 de Elastic Beanstalk?

Aunque la AMI de Amazon Linux y AL2023/AL2 comparten el mismo kernel de Linux, difieren en el sistema de inicialización, las versiones de `libc`, la cadena de herramientas del compilador y diversos paquetes. Para obtener más información, consulte las [Preguntas frecuentes de Amazon Linux 2](#).

El servicio Elastic Beanstalk también ha actualizado versiones específicas de la plataforma del tiempo de ejecución, las herramientas de compilación y otras dependencias. No se garantiza que las ramificaciones de la plataforma basadas en AL2023/AL2 sean retrocompatibles con su aplicación existente. Además, aunque el código de la aplicación se implemente correctamente en la nueva versión de la plataforma, es posible que se comporte o funcione de forma diferente debido a las diferencias del sistema operativo y del entorno de ejecución. Para obtener una lista y una descripción de las configuraciones y personalizaciones que tendrá que revisar y probar, consulte [Migración de su aplicación de Linux de Elastic Beanstalk a Amazon Linux 2023 o Amazon Linux 2](#).

Cancelación de actualizaciones de configuración del entorno e implementaciones de aplicaciones

Puede cancelar las actualizaciones en curso que se activan por cambios de configuración del entorno. También puede cancelar la implementación de una nueva versión de la aplicación en curso. Es posible, por ejemplo, que desee cancelar una actualización porque prefiera seguir utilizando la configuración del entorno existente en lugar de aplicar nuevas opciones de configuración. O tal vez se haya dado cuenta de que la nueva versión de la aplicación que está implementando tiene problemas que impedirán que se inicie o se ejecute correctamente. Si cancela una actualización del entorno o de una versión de la aplicación, no tendrá que esperar hasta que la actualización o el proceso de implementación termine para intentarlo de nuevo.

Note

Durante la fase de limpieza en la que se eliminan los recursos antiguos que ya no son necesarios, una vez que se ha actualizado el último lote de instancias, ya no es posible cancelar la actualización.


Elastic Beanstalk efectúa la restauración del mismo modo que realizó la última actualización correcta. Por ejemplo, si las actualizaciones continuas basadas en tiempo están habilitadas en el entorno, Elastic Beanstalk esperará a que llegue el tiempo de pausa establecido entre los cambios de restauración de un lote de instancias para restaurar los cambios en el siguiente lote. O, si hace poco que habilitó las actualizaciones continuas y la última vez que actualizó correctamente la configuración del entorno fue sin utilizar actualizaciones continuas, Elastic Beanstalk realizará la restauración de todas las instancias simultáneamente.

No se puede detener un proceso de Elastic Beanstalk para restaurar una configuración del entorno anterior una vez que ha empezado a cancelarse la actualización. El proceso de restauración continúa hasta que todas las instancias del entorno tengan la configuración del entorno anterior o hasta que el proceso de restauración produce un error. En el caso de las implementaciones de versiones de la aplicación, cuando se cancelan, simplemente se detiene la implementación, por lo que algunas instancias tendrán la nueva versión de la aplicación, mientras que otras seguirán ejecutando la versión de la aplicación existente. Posteriormente, podrá implementar la misma versión de la aplicación u otra diferente.

Para obtener más información sobre las actualizaciones continuas, consulte [Actualizaciones de configuración del entorno continuas de Elastic Beanstalk](#). Para obtener más información sobre las implementaciones por lotes de la versión de la aplicación, consulte [Políticas y ajustes de implementación](#).

Para cancelar una actualización

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

 Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En la página de información general del entorno, elija Actions (Acciones) y, a continuación, elija Abort current operation (Anular operación actual).

Volver a crear entornos de Elastic Beanstalk

El entorno de AWS Elastic Beanstalk puede resultar inservible si no utiliza la funcionalidad de Elastic Beanstalk para modificar o terminar los recursos de AWS subyacentes del entorno. Si esto ocurre, puede volver a crear el entorno para intentar restaurarlo a un estado operativo. Cuando se vuelve a crear un entorno, se terminan todos sus recursos y se reemplazan por nuevos recursos con la misma configuración.

También puede volver a crear entornos terminados en un plazo de seis semanas (42 días) a partir de su terminación. Cuando vuelve a crear un entorno, Elastic Beanstalk intenta crearlo con el mismo nombre, ID y configuración.

Volver a crear un entorno en ejecución

Puede volver a crear un entorno mediante la consola de Elastic Beanstalk o usando la API `RebuildEnvironment`.

Para volver a crear un entorno en ejecución (consola)

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. Elija Actions y, a continuación, elija Rebuild environment.
4. Elija Rebuild.

Cuando se vuelve a crear un entorno en ejecución, se crean nuevos recursos con la misma configuración que los recursos antiguos; sin embargo, los identificadores de recurso son diferentes y no se restauran los datos de los recursos antiguos. Por ejemplo, al volver a crear un entorno con una instancia de base de datos de Amazon RDS, se crea una nueva base de datos con la misma configuración, pero no se aplica una instantánea a la nueva base de datos.

Para volver a crear un entorno en ejecución con la API de Elastic Beanstalk, utilice la acción [RebuildEnvironment](#) con la AWS CLI o el SDK de AWS.

```
$ aws elasticbeanstalk rebuild-environment --environment-id e-vdnftxubwq
```

Volver a crear un entorno terminado

Puede volver a crear y restaurar un entorno terminado mediante la consola de Elastic Beanstalk, la CLI de EB o la API `RebuildEnvironment`.

Note

A menos que utilice su propio nombre de dominio personalizado con su entorno terminado, el entorno usará un subdominio de `elasticbeanstalk.com`. Estos subdominios se comparten dentro de una región de Elastic Beanstalk. Por lo tanto, se pueden utilizar en cualquier entorno creado por cualquier cliente en la misma región. Aunque su entorno se haya terminado, otro entorno podría utilizar su subdominio. En este caso, la reconstrucción del entorno producirá un error.

Puede evitar este problema usando un dominio personalizado. Para obtener más información, consulte [Nombre de dominio de su entorno de Elastic Beanstalk](#).

Los entornos recién terminados aparecen en la información de la aplicación al cabo de una hora como máximo. Durante este tiempo, puede ver los eventos del entorno en su [panel](#) y usar la acción [Restore environment](#) para volver a crearlo.

Para volver a crear un entorno que ya no está visible, utilice la opción `Restore terminated environment` de la página de la aplicación.

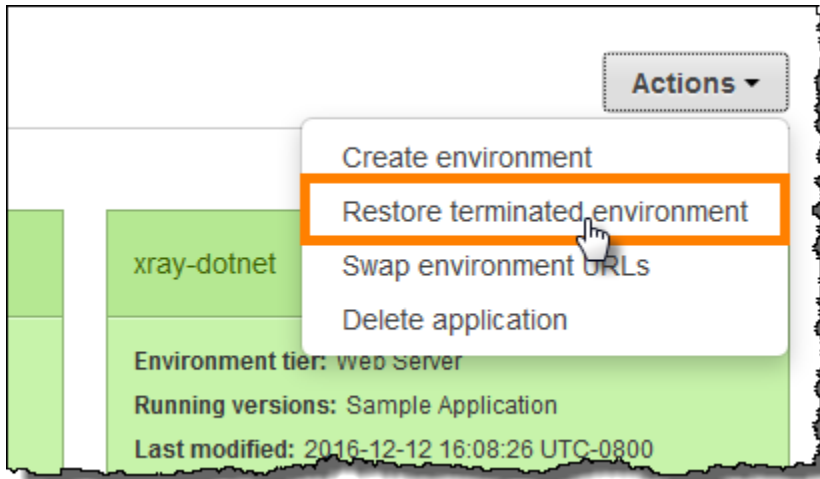
Para volver a crear un entorno terminado (consola)

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista `Regions (Regiones)`, seleccione su Región de AWS.
2. En el panel de navegación, elija `Applications (Aplicaciones)` y, a continuación, elija el nombre de la aplicación de la lista.

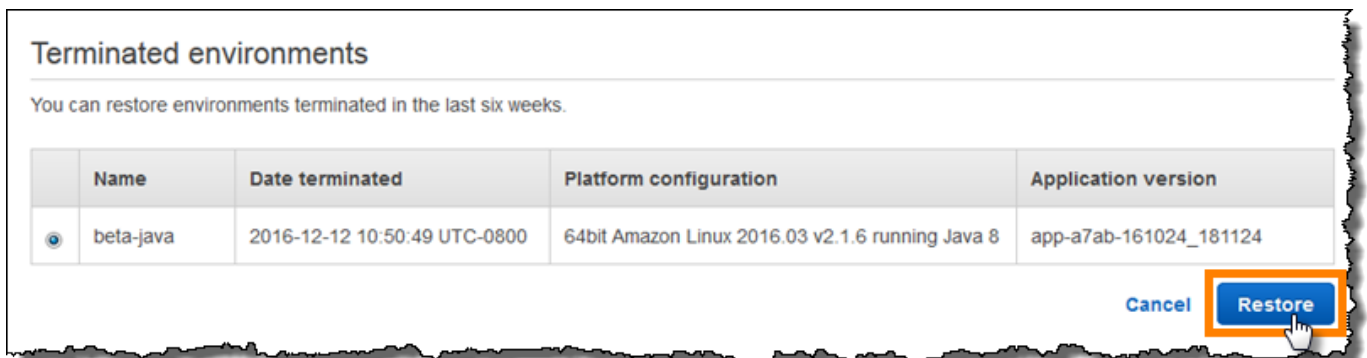
Note

Si tiene muchas aplicaciones, utilice la barra de búsqueda para filtrar la lista de aplicaciones.

3. Elija Actions y, a continuación, elija Restore terminated environment.



4. Elija un entorno terminado.
5. Elija Restore (Restaurar).



Elastic Beanstalk intenta crear un nuevo entorno con el mismo nombre, ID y configuración. Si existe un entorno con el mismo nombre o dirección URL cuando intenta reconstruirlo, la reconstrucción produce un error. La eliminación de la versión de la aplicación que se implementó en el entorno también producirá un error durante la reconstrucción del entorno.

Si utiliza la CLI de EB para administrar su entorno, utilice el comando `eb restore` para volver a crear un entorno terminado.


```
$ eb restore e-vdnftxubwq
```

Para obtener más información, consulte [eb restore](#).

Para volver a crear un entorno terminado con la API de Elastic Beanstalk, utilice la acción [RebuildEnvironment](#) con la AWS CLI o el SDK de AWS.

```
$ aws elasticbeanstalk rebuild-environment --environment-id e-vdnftxubwq
```

Tipos de entornos

En AWS Elastic Beanstalk puede crear un entorno escalable con equilibrio de carga o un entorno de una sola instancia. El tipo de entorno necesario depende de la aplicación que implemente. Por ejemplo, puede desarrollar y probar una aplicación en un entorno de una sola instancia para ahorrar costos y después actualizar ese entorno a un entorno escalable con balanceo de carga cuando la aplicación esté lista para producción.

Note

Una capa de entorno de trabajo para una aplicación web que procese tareas en segundo plano no incluye un balanceador de carga. Sin embargo, un entorno de trabajo se amplía de manera eficaz añadiendo instancias al grupo de Auto Scaling para procesar los datos de la cola de Amazon SQS cuando la carga los necesite.

Entorno escalable con balanceo de carga

Un entorno escalable y con balanceo de carga utiliza los servicios de Elastic Load Balancing y Auto Scaling de Amazon EC2 para aprovisionar las instancias de Amazon EC2 necesarias para la aplicación implementada. Auto Scaling de Amazon EC2 inicia automáticamente más instancias para acomodar el aumento de la carga que registra la aplicación. Si la carga de la aplicación se reduce, Auto Scaling de Amazon EC2 detiene instancias, pero siempre deja en ejecución el número mínimo especificado de instancias. Si la aplicación requiere escalabilidad con la opción de que se ejecute en varias zonas de disponibilidad, utilice un entorno escalable con balanceo de carga. Si no está seguro de qué tipo de entorno debe seleccionar, puede elegir uno y, si es necesario, cambiarlo más adelante.

Entorno de una sola instancia

Un entorno de una sola instancia contiene una única instancia de Amazon EC2 con una dirección IP elástica. Un entorno de una sola instancia no tiene un balanceador de carga, lo que puede ayudarle a reducir costos en comparación con un entorno escalable con balanceo de carga. Aunque un entorno de una sola instancia usa el servicio de Auto Scaling de Amazon EC2, los valores de número mínimo de instancias, número máximo de instancias y capacidad deseada están establecidos en 1. Por tanto, no se inician nuevas instancias para acomodar el aumento de la carga de su aplicación.

Utilice un entorno de una sola instancia si prevé que su aplicación en producción tendrá poco tráfico o si realiza un desarrollo remoto. Si no está seguro de qué tipo de entorno debe seleccionar, puede elegir uno y, si es necesario, puede cambiarlo más adelante. Para obtener más información, consulte [Cambiar el tipo de entorno](#).

Cambiar el tipo de entorno

Puede cambiar el tipo de entorno a un entorno de una sola instancia o a un entorno escalable con balanceo de carga editando la configuración del entorno. En algunos casos, puede ser conveniente cambiar el tipo de entorno de un tipo a otro. Por ejemplo, suponga que ha desarrollado y probado una aplicación en un entorno de una sola instancia para ahorrar costos. Cuando la aplicación esté lista para producción, puede cambiar el tipo de entorno a un entorno escalable con balanceo de carga para que pueda ampliarse con el fin de satisfacer las demandas de sus clientes.

Para cambiar un tipo de entorno

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría Capacity (Capacidad), elija Edit (Editar).
5. En la lista Environment type (Tipo de entorno), seleccione el tipo de entorno que desee.

Elastic Beanstalk > Environments > GettingStartedApp-env > Configuration

Modify capacity

Configure the compute capacity of your environment and Auto Scaling settings to optimize the number of instances used.

Auto Scaling Group

Environment type
Load balanced

Instances
Min 1
Max 2

Fleet composition
Choose a mix of On-Demand and Spot Instances with multiple instance types. Spot Instances are automatically launched at the lowest available price. [Learn more](#)

On-Demand instances
 Combine purchase options and instances

Maximum spot price
The maximum price per instance-hour, in USD, that you're willing to pay for a Spot Instance. Setting a custom price limits your chances to fulfill your target capacity using [On-Demand instances](#).

6. Seleccione Save.

Pueden transcurrir varios minutos hasta que se actualice el entorno mientras Elastic Beanstalk aprovisiona los recursos de AWS.

Si su entorno se encuentra en una VPC, seleccione las subredes en las que desea colocar las instancias de Elastic Load Balancing y Amazon EC2. Cada zona de disponibilidad en la que se ejecute la aplicación debe tener ambas. Para obtener más información, consulte [Uso de Elastic Beanstalk con Amazon VPC](#).

Entornos de trabajo de Elastic Beanstalk

Si su aplicación de AWS Elastic Beanstalk realiza operaciones o flujos de trabajo que tardan mucho tiempo en completarse, puede trasladar las tareas a un entorno de trabajo dedicado. Desacoplar el front-end de su aplicación web de un proceso que realiza operaciones de bloqueo es una forma habitual de garantizar que su aplicación siga respondiendo cuando está sometida a una carga intensa.

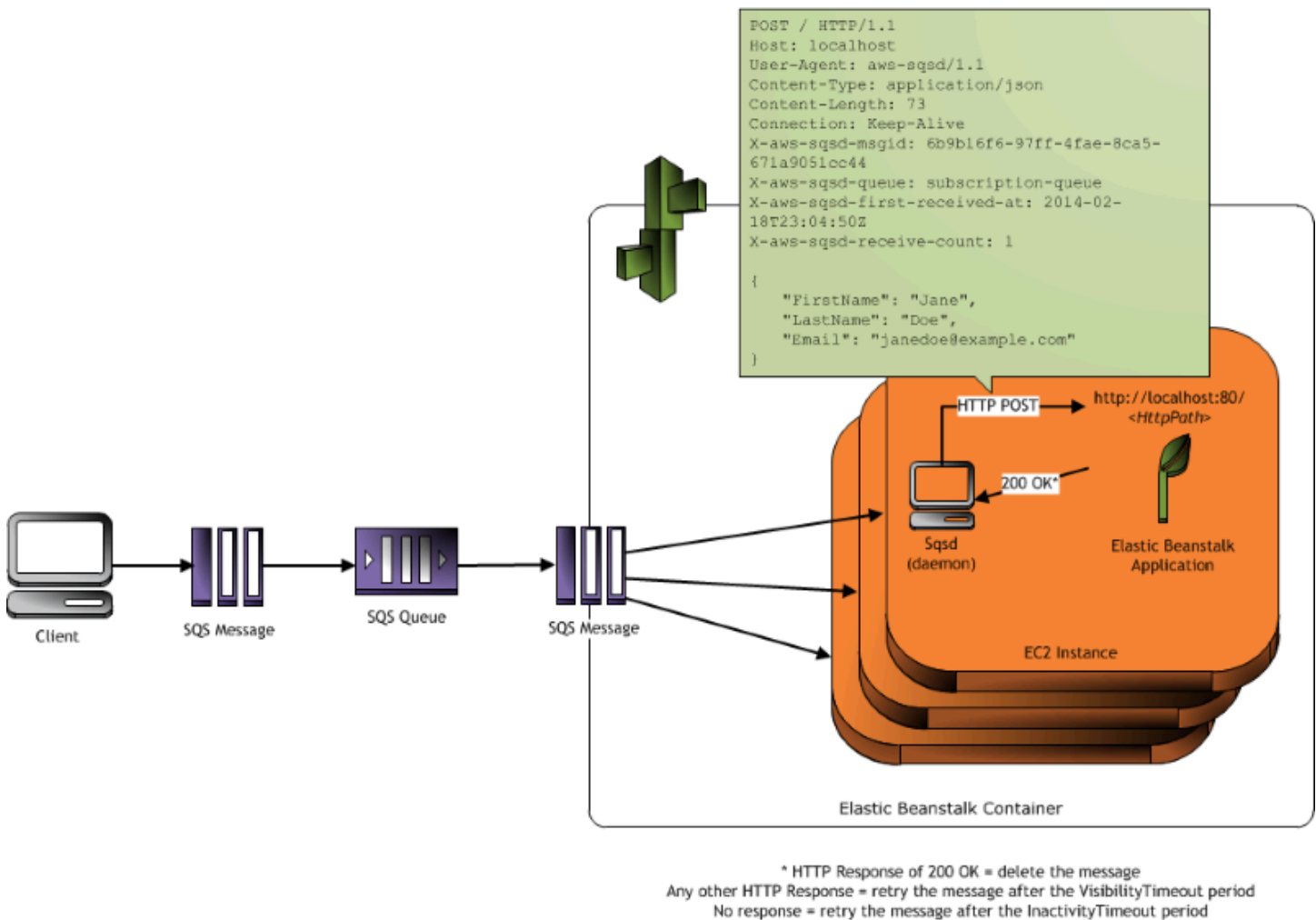
Una tarea de larga ejecución aumenta significativamente el tiempo que se tarda en completar una solicitud, como el procesamiento de imágenes o vídeos, el envío de correo electrónico o la

generación de un archivo ZIP. Estas operaciones pueden tardar solamente uno o dos segundos en completarse, pero un retraso de unos segundos es mucho para una solicitud web que de otro modo se completaría en menos de 500 ms.

Una opción consiste en generar un proceso de trabajo localmente y procesar la tarea de forma asíncrona. Esto funcionará si la instancia pueda hacerse cargo de todas las tareas que se le envían. Sin embargo, cuando el nivel de carga es alto, la instancia podría saturarse con tareas en segundo plano y dejar de responder a solicitudes de mayor prioridad. Si los distintos usuarios pueden generar varias tareas, el aumento de la carga podrían no corresponderse a un aumento de los usuarios, por lo que resultará difícil escalar la capa de servidor web de forma eficaz.

Para evitar que se ejecuten tareas de ejecución prolongada localmente, puede utilizar el SDK de AWS para su lenguaje de programación con el fin de enviarlas a una cola de Amazon Simple Queue Service (Amazon SQS) y ejecutar el proceso correspondiente a cada una de ellas en un conjunto separado de instancias. A continuación, diseñe estas instancias de trabajo para obtener los elementos de la cola solo cuando tengan la capacidad de ejecutarlos, lo que evitará que se sobrecarguen.

Los entornos de trabajo de Elastic Beanstalk simplifican este proceso mediante la administración de la cola de Amazon SQS y la ejecución de un [proceso daemon](#) en cada instancia que lea de la cola automáticamente. Cuando el daemon obtiene un elemento de la cola, envía una solicitud HTTP POST localmente a `http://localhost/` en el puerto 80 con el contenido de la cola en el cuerpo del mensaje. Todo lo que su aplicación tiene que hacer es ejecutar la tarea de larga ejecución en respuesta a la solicitud POST. Puede [configurar el daemon](#) para publicar en una ruta diferente, usar un tipo MIME distinto de una aplicación/JSON, conectarse a una cola existente o personalizar las conexiones (solicitudes simultáneas máximas), los tiempos de espera y los reintentos.



Con las [tareas periódicas](#), también puede configurar el daemon de trabajo para que ponga en cola los mensajes de acuerdo con una programación cron. Cada tarea periódica puede realizar una solicitud POST en una ruta diferente. Habilite las tareas periódicas incluyendo un archivo YAML en el código fuente que defina la programación y la ruta de cada tarea.

Note

La plataforma [.NET en Windows Server](#) no admite entornos de trabajo.

Secciones

- [El daemon de SQS del entorno de trabajo](#)
- [Colas de mensajes fallidos](#)
- [Tareas periódicas](#)

- [Uso de Amazon CloudWatch para el escalado automático en capas del entorno de trabajo](#)
- [Configuración de entornos de trabajo](#)

El daemon de SQS del entorno de trabajo

Los entornos de trabajo ejecutan un proceso daemon proporcionado por Elastic Beanstalk. Este daemon se actualiza periódicamente para agregar características y corregir errores. Para obtener la versión más reciente del daemon, actualice a la última [versión de la plataforma](#).

Cuando la aplicación en el entorno de trabajo devuelve una respuesta 200 OK para confirmar que ha recibido y procesado correctamente la solicitud, el daemon envía una llamada DeleteMessage a la cola de Amazon SQS para que el mensaje se elimine de la cola. Si la aplicación devuelve una respuesta distinta a 200 OK, Elastic Beanstalk espera el periodo especificado en `ErrorVisibilityTimeout` para volver a poner el mensaje en la cola. Si no hay respuesta, Elastic Beanstalk espera el periodo especificado en `InactivityTimeout` para volver a poner el mensaje en la cola, de forma que el mensaje esté disponible para otro intento durante el procesamiento.

Note

Las propiedades de las colas de Amazon SQS (orden de los mensajes, entrega al menos una vez y muestreo de mensajes) pueden afectar al modo en que diseña una aplicación web para un entorno de trabajo. Para obtener más información, consulte [Propiedades de colas distribuidas](#) en la [Guía para desarrolladores de Amazon Simple Queue Service](#).

Amazon SQS elimina automáticamente los mensajes que llevan en la cola más tiempo que el valor especificado en `RetentionPeriod`.

El daemon establece los siguientes encabezados HTTP.

Encabezados HTTP

Nombre	Valor
User-Agent	aws-sqsd
	aws-sqsd/1.1 1

Encabezados HTTP

<code>X-Aws-Sqs-Message-Id</code>	ID de mensaje de SQS, que se utiliza para detectar tormentas de mensajes (un número inusualmente elevado de mensajes nuevos).
<code>X-Aws-Sqs-Queue</code>	Nombre de la cola de SQS.
<code>X-Aws-Sqs-First-Received-At</code>	Hora UTC, en formato ISO 8601 , a la que se recibió por primera vez el mensaje.
<code>X-Aws-Sqs-Receive-Count</code>	Número de recepción de mensajes de SQS.
<code>X-Aws-Sqs-Attr-<i>message-attribute-name</i></code>	Atributos de mensajes personalizados asignados al mensaje que se está procesando. <code>message-attribute-name</code> es el nombre del atributo del mensaje real. Todos los atributos de mensajes de cadena y número se añaden al encabezado. Los atributos binarios se descartan y no se incluyen en el encabezado.
<code>Content-Type</code>	Configuración de tipo MIME; <code>application/json</code> , de forma predeterminada.

Colas de mensajes fallidos

Los entornos de trabajo de Elastic Beanstalk admiten las colas de mensajes fallidos de Amazon Simple Queue Service (Amazon SQS). Una cola de mensajes fallidos es una cola en la que otras colas pueden enviar mensajes que por algún motivo no se pudieron procesar correctamente. Una de las ventajas principales de usar una cola de mensajes fallidos es la capacidad de identificar y aislar los mensajes procesados sin éxito. A continuación, puede analizar los mensajes enviados a la cola de mensajes fallidos para intentar determinar por qué no se han procesado correctamente.

Si especifica una cola de Amazon SQS generada automáticamente en el momento de crear la capa del entorno de trabajo, se habilita de forma predeterminada una cola de mensajes fallidos para un entorno de trabajo. Si elige una cola de SQS existente para su entorno de trabajo, debe utilizar SQS para configurar una cola de mensajes fallidos de manera independiente. Para obtener información

sobre cómo utilizar SQS para configurar una cola de mensajes fallidos, consulte [Colas de mensajes fallidos de Amazon SQS](#).

No puede deshabilitar las colas de mensajes fallidos. Los mensajes que no se puedan entregar siempre se envían a una cola de mensajes fallidos. No obstante, puede desactivar esta funcionalidad de forma eficaz estableciendo la opción `MaxRetries` en el valor válido máximo de 100.

Si no se ha configurado una cola de mensajes fallidos para la cola de Amazon SQS de su entorno de trabajo, Amazon SQS mantiene los mensajes en la cola hasta que expira el período de retención. Para obtener información detallada sobre la configuración del período de retención, consulte [the section called "Configuración de entornos de trabajo"](#).

Note

La opción `MaxRetries` de Elastic Beanstalk es equivalente a la opción `MaxReceiveCount` de SQS. Si el entorno de trabajo no utiliza una cola de SQS generada automáticamente, use la opción `MaxReceiveCount` de SQS para desactivar de forma eficaz su cola de mensajes fallidos. Para obtener más información, consulte [Uso de colas de mensajes fallidos de Amazon SQS](#).

Para obtener más información sobre el ciclo de vida de un mensaje de SQS, consulte [Message Lifecycle \(Ciclo de vida del mensaje\)](#).

Tareas periódicas

Puede definir tareas periódicas en un archivo denominado `cron.yaml` en su paquete de código fuente para añadir automáticamente trabajos a la cola del entorno de trabajo a intervalos regulares.

Por ejemplo, el siguiente archivo `cron.yaml` crea dos tareas periódicas. La primera se ejecuta cada 12 horas y la segunda a las 11 p. m. UTC todos los días.

Example cron.yaml

```
version: 1
cron:
  - name: "backup-job"
    url: "/backup"
    schedule: "0 */12 * * *"
```



```
- name: "audit"
  url: "/audit"
  schedule: "0 23 * * *"
```

El **name** debe ser único para cada tarea. La dirección URL es la ruta a la que se envía la solicitud POST para desencadenar el trabajo. La programación es una [expresión CRON](#) que determina cuándo se ejecuta la tarea.

Cuando se ejecuta una tarea, el daemon publica un mensaje en la cola de SQS del entorno con un encabezado que indica el trabajo que debe realizarse. Cualquier instancia del entorno puede obtener el mensaje y procesar el trabajo.

Note

Si configura su entorno de trabajo con una cola de SQS existente y elige una [cola FIFO de Amazon SQS](#), las tareas periódicas no serán compatibles.

Elastic Beanstalk usa la funcionalidad de elección de líder para determinar qué instancia de su entorno de trabajo pone en cola la tarea periódica. Cada instancia intenta convertirse en líder escribiendo en una tabla de Amazon DynamoDB. La primera instancia que lo logra es el líder y debe seguir escribir en la tabla para mantener el estado de líder. Si el líder se queda fuera de servicio, otra instancia ocupa su lugar rápidamente.

Para las tareas periódicas, el daemon de trabajo establece los siguientes encabezados adicionales.

Encabezados HTTP

Nombre	Valor
X-Aws-Sqs-Taskname	Para las tareas periódicas, el nombre de la tarea que se va a realizar.
X-Aws-Sqs-Scheduled-At	Momento en el que programó la tarea periódica
X-Aws-Sqs-Sender-Id	Número de cuenta de AWS del remitente del mensaje

Uso de Amazon CloudWatch para el escalado automático en capas del entorno de trabajo

Juntos, Auto Scaling de Amazon EC2 y CloudWatch monitorizan el uso de la CPU de las instancias en ejecución en el entorno de trabajo. El modo en que configura el límite de escalado automático para la capacidad de la CPU determina el número de instancias que el grupo de Auto Scaling ejecuta para administrar correctamente el desempeño de los mensajes en la cola de Amazon SQS. Cada instancia EC2 publica sus métricas de utilización de la CPU en CloudWatch. Auto Scaling de Amazon EC2 recupera de CloudWatch el uso medio de la CPU de todas las instancias del entorno de trabajo. Debe configurar el umbral superior e inferior, así como el número de instancias que se añaden o terminan de acuerdo con la capacidad de la CPU. Cuando Auto Scaling de Amazon EC2 detecta que se ha alcanzado el umbral superior especificado para la capacidad de la CPU, Elastic Beanstalk crea nuevas instancias en el entorno de trabajo. Las instancias se eliminan cuando la carga de la CPU cae por debajo del umbral.

Note

Los mensajes que no se procesaron cuando se terminó la instancia se devuelven a la cola, donde pueden ser procesados por otro daemon de la instancia que siga ejecutándose.

También puede configurar otras alarmas de CloudWatch, según sea necesario, mediante la consola de Elastic Beanstalk, la CLI o el archivo de opciones. Para obtener más información, consulte [Uso de Elastic Beanstalk con Amazon CloudWatch](#) y [Crear un grupo de Auto Scaling con políticas de escalado por pasos](#).

Configuración de entornos de trabajo

Puede administrar la configuración de un entorno de trabajo editando la categoría Worker (Entorno de trabajo) en la página Configuration (Configuración) de la [consola de administración del entorno](#).

[Elastic Beanstalk](#) > [Environments](#) > [GettingStartedApp-env](#) > Configuration

Modify worker

You can create a new Amazon SQS queue for your worker application or pull work items from an existing queue. The worker daemon on the instances in your environment pulls an item from the queue and relays it in the body of a POST request to a local HTTP path relative to localhost.

Queue

Worker queue



SQS queue from which to read work items.

Messages

HTTP path

The daemon pulls items from the Amazon SQS queue and posts them locally to this path.

MIME type

Change the MIME type of the POST requests that the worker daemon sends to your application.

HTTP connections

Maximum number of concurrent connections to the application.

Visibility timeout

seconds

The amount of time to lock an incoming message for processing before returning it to the queue.

Error visibility timeout

seconds

The amount of time to wait before resending a message after an error response from the application.

▼ Advanced options

The following settings control advanced behavior of the worker tier daemon. [Learn more](#)

Max retries

Maximum number of retries after which the message is discarded.

Connection timeout

Inactivity timeout

Note

Puede configurar la ruta de URL para enviar los mensajes de cola de proceso de trabajo, pero no puede configurar el puerto IP. Elastic Beanstalk siempre publica mensajes de cola de entorno de trabajo en el puerto 80. La aplicación de entorno de proceso de trabajo o su proxy deben escuchar en el puerto 80.

Para configurar el daemon del entorno de trabajo

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración del Worker (Entorno de trabajo) elija Edit (Editar).

La página Modify worker (Modificar configuración del entorno de trabajo) tiene las siguientes opciones.

En la sección Queue (Cola):

- Worker queue (Cola de entorno de trabajo): especifique la cola de Amazon SQS de la que lee el daemon. Si dispone de una, puede elegir una cola existente. Si elige Autogenerated queue (Cola generada automáticamente), Elastic Beanstalk crea una nueva cola de Amazon SQS y una URL de cola de entorno de trabajo correspondiente.

Note

Al elegir Autogenerated queue (Cola generada automáticamente), la cola que crea Elastic Beanstalk es una cola de Amazon SQS [estándar](#). Cuando elija una cola existente, puede

proporcionar una cola estándar o una cola [FIFO](#) de Amazon SQS. Tenga en cuenta que, si proporciona una cola FIFO, las [tareas periódicas](#) no serán compatibles.

- Worker queue URL (URL de cola de entorno de trabajo): si elige una cola de entorno de trabajo existente, esta opción muestra la URL asociada a esa cola de Amazon SQS.

En la sección Messages (Mensajes) :

- HTTP path (Ruta HTTP): especifique la ruta relativa a la aplicación que recibirá los datos de la cola de Amazon SQS. Los datos se insertan en el cuerpo de un mensaje HTTP POST. El valor predeterminado es `/`.
- MIME type (Tipo MIME): indique el tipo MIME que usa el mensaje HTTP POST. El valor predeterminado es `application/json`. Sin embargo, cualquier valor es válido, ya que puede crear y después especificar su propio tipo MIME.
- HTTP connections (Conexiones HTTP): especifique el número máximo de conexiones simultáneas que el daemon puede realizar en cualquier aplicación dentro de una instancia de Amazon EC2. El valor predeterminado es **50**. Puede especificar de **1** a **100**.
- Visibility timeout (Tiempo de espera de visibilidad): indique la cantidad de tiempo, en segundos, durante la que se bloquea el procesamiento de un mensaje entrante procedente de la cola de Amazon SQS. Una vez transcurrido el período de tiempo configurado, el mensaje vuelve a hacerse visible en la cola para que cualquier otro daemon pueda leerlo. Elija un valor superior al que necesita su aplicación para procesar mensajes, hasta **43200** segundos.
- Error visibility timeout (Tiempo de espera de visibilidad del error): indique la cantidad de tiempo, en segundos, que transcurre antes de que Elastic Beanstalk devuelva un mensaje a la cola de Amazon SQS cuando se produce un error explícito en un intento de procesamiento. Puede especificar de **0** a **43200** segundos.

En la sección Advanced options (Opciones avanzadas):

- Max retries (Número máximo de reintentos): especifique el número máximo de veces que Elastic Beanstalk intenta enviar el mensaje a la cola de Amazon SQS antes de mover el mensaje a la [cola de mensajes fallidos](#). El valor predeterminado es **10**. Puede especificar de **1** a **100**.

Note

La opción Número máximo de reintentos solo se aplica a las colas de Amazon SQS configuradas con una cola de mensajes fallidos. Para las colas de Amazon SQS que no estén configuradas con una cola de mensajes fallidos, Amazon SQS retiene los mensajes en la cola y los procesa hasta que vence el período especificado por la opción Período de retención.

- **Connection timeout** (Tiempo de espera de la conexión): indique la cantidad de tiempo, en segundos, que se va a esperar a que las conexiones con una aplicación se realicen correctamente. El valor predeterminado es **5**. Puede especificar de **1** a **60** segundos.
- **Inactivity timeout** (Tiempo de espera de inactividad): indique la cantidad de tiempo, en segundos, que se debe a esperar para que llegue la respuesta de una conexión existente con una aplicación. El valor predeterminado es **180**. Puede especificar de **1** a **36000** segundos.
- **Retention period** (Período de retención): indique la cantidad de tiempo, en segundos, durante la que un mensaje es válido y se procesa de forma activa. El valor predeterminado es **345600**. Puede especificar de **60** a **1209600** segundos.

Si utiliza una cola de Amazon SQS existente, los valores que configura al crear un entorno de trabajo pueden entrar en conflicto con los valores establecidos directamente en Amazon SQS. Por ejemplo, si configura un entorno de trabajo con un valor de `RetentionPeriod` mayor que el valor de `MessageRetentionPeriod` definido en Amazon SQS, Amazon SQS eliminará el mensaje cuando supere el valor de `MessageRetentionPeriod`.

Asimismo, si el valor de `RetentionPeriod` que establece en la configuración del entorno de trabajo es menor que el valor de `MessageRetentionPeriod` definido en Amazon SQS, el daemon eliminará el mensaje antes que Amazon SQS. En `VisibilityTimeout`, el valor que establece para el daemon en la configuración del entorno de trabajo anula el valor de `VisibilityTimeout` de Amazon SQS. Asegúrese de que los mensajes se eliminan adecuadamente comparando la configuración de Elastic Beanstalk con la configuración de Amazon SQS.

Creación de enlaces entre entornos de Elastic Beanstalk

A medida que la aplicación crece en tamaño y complejidad, es posible que quiera dividirla en componentes con diferentes ciclos de vida de desarrollo y operaciones. Con la ejecución de servicios pequeños que interactúen entre sí a través de una interfaz bien definida, los equipos pueden

trabajar de forma independiente y las implementaciones pueden suponer menos riesgo. AWS Elastic Beanstalk le permite vincular sus entornos para compartir información entre los componentes que dependen unos de otros.

Note

Actualmente, Elastic Beanstalk admite enlaces de entorno para todas las plataformas excepto Multicontainer Docker.

Con los enlaces de entorno, puede definir las conexiones entre los entornos de los componentes de la aplicación como referencias con nombre. Cuando crea un entorno que define un enlace, Elastic Beanstalk establece una variable de entorno con el mismo nombre que el enlace. El valor de la variable es el punto de enlace que puede utilizar para conectarse al otro componente, que puede ser un servidor web o un entorno de trabajo.

Por ejemplo, si la aplicación se compone de un frontend que recopila direcciones de correo electrónico y un proceso de trabajo que envía un email de bienvenida a las direcciones de correo electrónico obtenidas por el frontend, puede crear un enlace con el proceso de trabajo del frontend y hacer que el frontend detecte automáticamente el punto de enlace (URL de la cola) del proceso de trabajo.

Defina los enlaces con otros entornos en un [manifiesto de entorno](#), un archivo con formato YAML llamado `env.yaml` situado en la raíz del código fuente de la aplicación. En el siguiente manifiesto, se define un enlace con un entorno llamado "worker":

`~/workspace/my-app/frontend/env.yaml`

```
AWSConfigurationTemplateVersion: 1.1.0.0
EnvironmentLinks:
  "WORKERQUEUE": "worker"
```

Si se crea un entorno con una versión de la aplicación que incluye el manifiesto de entorno anterior, Elastic Beanstalk buscará un entorno llamado `worker` que pertenezca a la misma aplicación. Si ese entorno existe, Elastic Beanstalk crea una propiedad de entorno llamada `WORKERQUEUE`. El valor de `WORKERQUEUE` es la URL de la cola de Amazon SQS. La aplicación frontend puede leer esta propiedad igual que una variable de entorno. Para obtener más información, consulte [Manifiesto del entorno \(env.yaml\)](#).

Para usar enlaces de entorno, añada un manifiesto de entorno a la fuente de la aplicación y cárguelo con la CLI de EB AWS CLI o un SDK. Si utiliza el SDK AWS CLI o un SDK, active la `process` marca cuando llame a `CreateApplicationVersion`:

```
$ aws elasticbeanstalk create-application-version --process --application-name  
my-app --version-label frontend-v1 --source-bundle S3Bucket="DOC-EXAMPLE-  
BUCKET",S3Key="front-v1.zip"
```

Esta opción le indica a Elastic Beanstalk que valide el manifiesto del entorno y los archivos de configuración del paquete de código fuente cuando se cree la versión de la aplicación. La CLI de EB establece esta marca automáticamente cuando hay un manifiesto de entorno en el directorio del proyecto.

Cree sus entornos con normalidad utilizando cualquier cliente. Cuando necesita terminar entornos, utilice primero el enlace. Si un entorno está vinculado a otro entorno, Elastic Beanstalk impedirá que el entorno vinculado se termine. Para anular esta medida de protección, utilice la marca `ForceTerminate`. Este parámetro está disponible en la AWS CLI como `--force-terminate`:

```
$ aws elasticbeanstalk terminate-environment --force-terminate --environment-name  
worker
```


Configuración de entornos de Elastic Beanstalk

AWS Elastic Beanstalk ofrece una amplia gama de opciones para personalizar los recursos de su entorno y la configuración de plataforma y comportamiento de Elastic Beanstalk. Cuando crea un entorno de servidor web, Elastic Beanstalk crea varios recursos para admitir la operación de la aplicación.

- **EC2 Instance (Instancia de EC2):** una máquina virtual de Amazon Elastic Compute Cloud (Amazon EC2) configurada para ejecutar aplicaciones web en la plataforma que elija.

Cada plataforma ejecuta un conjunto específico de software, archivos de configuración y scripts compatibles con una determinada versión de lenguaje, marco de trabajo y contenedor web (o una combinación de estos). La mayoría de las plataformas utilizan Apache o nginx como un proxy inverso que se sitúa delante de la aplicación web, reenvía las solicitudes a esta, administra los recursos estáticos y genera registros de acceso y errores.

- **Instance security group (Grupo de seguridad de la instancia):** grupo de seguridad de Amazon EC2 configurado para permitir el tráfico entrante en el puerto 80. Este recurso permite que el tráfico HTTP procedente del balanceador de carga llegue a la instancia EC2 en la que se ejecuta la aplicación web. De forma predeterminada, el tráfico no está permitido en otros puertos.
- **Balanceador de carga:** balanceador de carga de Elastic Load Balancing configurado para distribuir solicitudes a las instancias que se ejecutan en la aplicación. Los balanceadores de carga también permiten que las instancias no estén expuestas directamente a Internet.
- **Grupo de seguridad del balanceador de carga:** grupo de seguridad de Amazon EC2 configurado para permitir el tráfico entrante en el puerto 80. Este recurso permite que el tráfico HTTP procedente de Internet llegue al balanceador de carga. De forma predeterminada, el tráfico no está permitido en otros puertos.
- **Grupo de Auto Scaling:** grupo de Auto Scaling configurado para reemplazar una instancia si termina o deja de estar disponible.
- **Bucket de Amazon S3:** ubicación de almacenamiento para el código fuente, los registros y otros artefactos que se crean al utilizar Elastic Beanstalk.
- **CloudWatch Alarmas de Amazon:** dos CloudWatch alarmas que monitorean la carga de las instancias de su entorno y que se activan si la carga es demasiado alta o demasiado baja. Cuando se activa una alarma, en respuesta, el grupo de Auto Scaling aumenta o reduce los recursos.

- AWS CloudFormation pila: Elastic AWS CloudFormation Beanstalk se utiliza para lanzar los recursos de su entorno y propagar los cambios de configuración. Los recursos se definen en una plantilla que puede verse en la [consola deAWS CloudFormation](#).
- Nombre de dominio: nombre de dominio que direcciona el tráfico a la aplicación web con el formato *subdominio.región*.elasticbeanstalk.com.

Note

Para aumentar la seguridad de las aplicaciones de Elastic Beanstalk, el dominio elasticbeanstalk.com está registrado en la [lista de sufijos públicos \(PSL\)](#). Para mayor seguridad, se recomienda que utilice cookies con un prefijo __Host- en caso de que necesite configurar cookies confidenciales en el nombre de dominio predeterminado de sus aplicaciones de Elastic Beanstalk. Esta práctica le ayudará a defender su dominio contra intentos de falsificación de solicitudes entre sitios (CSRF). Para obtener más información, consulte la página de [configuración de cookies](#) en la red de desarrolladores de Mozilla.

Este tema se centra en las opciones de configuración de recursos disponibles en la consola de Elastic Beanstalk. Los siguientes temas muestran cómo configurar el entorno en la consola. También describen los espacios de nombres subyacentes que se corresponden a las opciones de la consola que se deben utilizar con los archivos de configuración o las opciones de configuración de la API. Para obtener información sobre los métodos de configuración avanzados, consulte [Configuración de entornos \(procedimiento avanzado\)](#).

Temas

- [Configuración del entorno usando la consola de Elastic Beanstalk](#)
- [Instancias de Amazon EC2 para el entorno de Elastic Beanstalk](#)
- [Grupo de Auto Scaling para su entorno de Elastic Beanstalk](#)
- [Balanceador de carga del entorno de Elastic Beanstalk](#)
- [Añadir una base de datos al entorno de Elastic Beanstalk](#)
- [La seguridad de su AWS Elastic Beanstalk entorno](#)
- [Etiquetado de recursos en sus entornos de Elastic Beanstalk](#)
- [Propiedades del entorno y otras opciones del software](#)
- [Notificaciones de entorno de Elastic Beanstalk con Amazon SNS](#)

- [Configuración de Amazon Virtual Private Cloud \(Amazon VPC\) con Elastic Beanstalk](#)
- [Nombre de dominio de su entorno de Elastic Beanstalk](#)

Configuración del entorno usando la consola de Elastic Beanstalk

Puede utilizar la consola de Elastic Beanstalk para ver y modificar numerosas [opciones de configuración](#) de su entorno y sus recursos. Puede personalizar el comportamiento de su entorno durante las implementaciones, habilitar características adicionales y modificar el tipo de instancia y otros ajustes que eligió durante la creación del entorno.

Para ver un resumen de la configuración del entorno.

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).

Página de configuración

La página Configuration overview (Información general de la configuración), muestra un grupo de categorías de configuración. Cada categoría de configuración resume el estado actual de un grupo de opciones relacionadas.

Elastic Beanstalk > Environments > Gettingstarteda-env > Configuration

Configuration Info

[Cancel](#) [Review changes](#) [Apply changes](#)

Service access Info

Configure the service role and EC2 instance profile that Elastic Beanstalk uses to manage your environment. Choose an EC2 key pair to securely log in to your EC2 instances. [Edit](#)

Service role
arn:aws:iam::164656829171:role/aws-elasticbeanstalk-service-role

Instance traffic and scaling Info

Customize the capacity and scaling for your environment's instances. Select security groups to control instance traffic. Configure the software that runs on your environment's instances by setting platform-specific options. [Edit](#)

Instances
IMDSv1
Deactivated

Capacity

Environment type	Fleet composition	On-demand base
Load balanced	On-Demand instances	0
On-demand above base	Processor type	Instance types
70	x86_64	t2.micro,t2.small

Load balancer
Load balancer type
application

Networking, database, and tags Info

Configure VPC settings, and subnets for your environment's EC2 instances and load balancer. Set up an Amazon RDS database that's integrated with your environment. [Edit](#)

Network

Load balancer visibility	Load balancer subnets
public	—

Database
Has coupled database
false

Updates, monitoring, and logging Info

Define when and how Elastic Beanstalk deploys changes to your environment. Manage your application's monitoring and logging settings, instances, and other environment resources. [Edit](#)

Updates

Managed updates	Update batch size	Deployment batch size
Deactivated	1	100
Deployment batch size type	Command timeout	Deployment policy
Percentage	600	AllAtOnce
Health threshold	Ignore health check	Instance replacement
Ok	false	false
Minimum capacity	Notifications email	
0	—	

Seleccione Edit (Editar) en una categoría de configuración para dirigirse a una página de configuración relacionada, en la que podrá ver los valores de opción completos y realizar cambios. Cuando haya terminado de ver y modificar opciones, puede elegir una de las siguientes acciones:

- **Cancel (Cancelar):** vuelva al panel del entorno sin aplicar los cambios de configuración. Si elige **Cancel (Cancelar)**, la consola pierde los cambios pendientes que haya realizado en cualquier categoría de configuración.

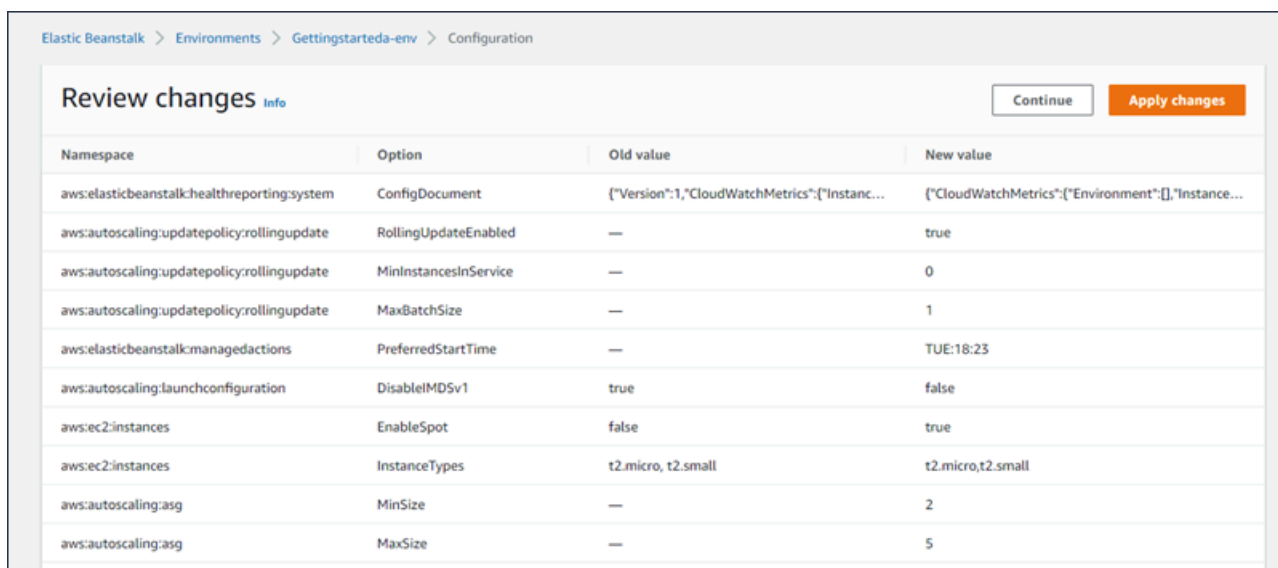
También puede cancelar los cambios en la configuración si elige otra página de la consola, como **Events (Eventos)** o **Logs (Registros)**. En este caso, si hay cambios de configuración pendientes, la consola le pedirá que confirme que acepta perderlos.

- **Review changes (Revisar cambios):** obtenga un resumen de todos los cambios pendientes que haya realizado en cualquiera de las categorías de configuración. Para obtener más información, consulte [Página Review Changes \(Revisar cambios\)](#).
- **Apply changes (Aplicar cambios):** aplique los cambios que haya realizado en cualquiera de las categorías de configuración a su entorno. En algunos casos se le pedirá que confirme una consecuencia de una de sus decisiones de configuración.

Página Review Changes (Revisar cambios)

La página **Review changes (Revisar cambios)** muestra una tabla con todos los cambios de opciones pendientes que haya realizado en cualquiera de las categorías de configuración y todavía no se hayan aplicado a su entorno.

Las tablas enumeran cada opción como una combinación de **Namespace (Espacio de nombres)** y **Option (Opción)** con la que Elastic Beanstalk lo identifica. Para obtener más información, consulte [Opciones de configuración](#).



Namespace	Option	Old value	New value
aws:elasticbeanstalk:healthreporting:system	ConfigDocument	{"Version":1,"CloudWatchMetrics":{"Instanc...	{"CloudWatchMetrics":{"Environment":[],"Instance...
aws:autoscaling:updatepolicy:rollingupdate	RollingUpdateEnabled	—	true
aws:autoscaling:updatepolicy:rollingupdate	MinInstancesInService	—	0
aws:autoscaling:updatepolicy:rollingupdate	MaxBatchSize	—	1
aws:elasticbeanstalk:managedactions	PreferredStartTime	—	TUE:18:23
aws:autoscaling:launchconfiguration	DisableIMDSv1	true	false
aws:ec2:instances	EnableSpot	false	true
aws:ec2:instances	InstanceTypes	t2.micro, t2.small	t2.micro,t2.small
aws:autoscaling:asg	MinSize	—	2
aws:autoscaling:asg	MaxSize	—	5

Cuando haya terminado de revisar los cambios, puede elegir una de las siguientes acciones:

- **Continue (Continuar):** vuelva a la página *Configuration overview* (Información general de configuración). A continuación, puede seguir realizando cambios o aplicando los pendientes.
- **Apply changes (Aplicar cambios):** aplique los cambios que haya realizado en cualquiera de las categorías de configuración a su entorno. En algunos casos se le pedirá que confirme una consecuencia de una de sus decisiones de configuración.

Instancias de Amazon EC2 para el entorno de Elastic Beanstalk

Cuando crea un entorno de servidor web, AWS Elastic Beanstalk crea una o más máquinas virtuales Amazon Elastic Compute Cloud (Amazon EC2), conocidas como instancias.

Las instancias del entorno están configuradas para ejecutar aplicaciones web en la plataforma que elija. Puede realizar cambios en diferentes propiedades y comportamientos de las instancias del entorno cuando cree el entorno o una vez que se esté ejecutando. O bien, ya puede realizar estos cambios modificando el código fuente que implementa en el entorno. Para obtener más información, consulte [the section called “Opciones de configuración”](#).

Note

El [grupo de Auto Scaling](#) del entorno administra las instancias de Amazon EC2 en las que se ejecuta la aplicación. Cuando se realizan cambios de configuración que se describen en esta página, la configuración de lanzamiento también cambia. La configuración de lanzamiento puede consistir en una plantilla de lanzamiento de Amazon EC2 o en un recurso de configuración de lanzamiento del grupo de Auto Scaling. Este cambio requiere el [reemplazo de todas las instancias](#). También desencadena una [actualización continua](#) o una [actualización inmutable](#), según la que se haya configurado.

Elastic Beanstalk admite varias [opciones de compra de instancias](#) de Amazon EC2: instancias bajo demanda, instancias reservadas e instancias de spot. Una instancia bajo demanda es un pay-as-you-go recurso; no se requiere ningún compromiso a largo plazo para utilizarla. Una instancia reservada es un descuento de facturación adquirido previamente que se aplica de forma automática a las instancias bajo demanda correspondientes del entorno. Una instancia de spot es una instancia de Amazon EC2 sin utilizar que está disponible por un precio inferior al precio bajo demanda. Puede habilitar las instancias de spot en su entorno a través de una única opción. Puede configurar el uso de instancias de spot, incluida la combinación de instancias bajo demanda y de spot, utilizando otras opciones. Para obtener más información, consulte [Grupo de Auto Scaling](#).

Secciones

- [Tipos de instancias de Amazon EC2](#)
- [Configuración de instancias de Amazon EC2 para su entorno](#)
- [Configuración de las instancias AWS EC2 para su entorno mediante el AWS CLI](#)
- [Recomendaciones para los entornos de primera oleada Graviton arm64](#)
- [El espacio de nombres aws:autoscaling:launchconfiguration](#)
- [Configuración del servicio de metadatos de instancia](#)

Tipos de instancias de Amazon EC2

Cuando se crea un nuevo entorno, Elastic Beanstalk aprovisiona instancias de Amazon EC2 que se basan en los tipos de instancia de Amazon EC2 que se elijan. Los tipos de instancia que elija determinan el hardware del host que ejecuta sus instancias. Los tipos de instancias de EC2 se pueden clasificar en función de la arquitectura del procesador en la que se basa cada una. Elastic Beanstalk admite tipos de instancias basados en las siguientes AWS arquitecturas de procesador: arquitectura Arm Graviton de 64 bits (arm64), arquitectura de 64 bits (x86) y arquitectura de 32 bits (i386). Elastic Beanstalk selecciona la arquitectura del procesador x86 de forma predeterminada al crear un entorno nuevo.

Note

La arquitectura i386 de 32 bits ya no es compatible con la mayoría de las plataformas de Elastic Beanstalk. En su lugar, se recomienda que elija los tipos de arquitectura x86 o arm64. Elastic Beanstalk proporciona [opciones de configuración](#) para los tipos de instancias del procesador i386 en el espacio de nombres [aws:ec2:instances](#).

En la configuración de un determinado entorno de Elastic Beanstalk, todos los tipos de instancia deben contar con el mismo tipo de arquitectura del procesador. Supongamos que se agrega un nuevo tipo de instancia a un entorno existente en el que ya hay un tipo de instancia t2.medium, basado en la arquitectura x86. Solo se puede agregar otro tipo de instancia de la misma arquitectura, como t2.small. Si desea reemplazar los tipos de instancias existentes por otros de una arquitectura diferente, puede hacerlo. Pero asegúrate de que todos los tipos de instancias del comando estén basados en el mismo tipo de arquitectura.

Elastic Beanstalk agrega a menudo soporte para nuevos tipos de instancias compatibles después de que Amazon EC2 los presente. Para obtener información sobre los tipos de instancias disponibles, consulte [Tipos de instancias](#) en la Guía del usuario de Amazon EC2 o [Tipos de instancias](#) en la Guía del usuario de Amazon EC2.

Note

Elastic Beanstalk ahora ofrece soporte para Graviton en todas las plataformas Amazon Linux 2 más recientes en todas las regiones compatibles con Graviton. AWS Para obtener más información sobre cómo crear un entorno de Elastic Beanstalk con tipos de instancias basados en arm64, consulte [Configuración de instancias de Amazon EC2 para su entorno](#). Cree nuevos entornos que ejecuten instancias de Amazon EC2 en una arquitectura arm64 y migre sus aplicaciones actuales a ellos con las [opciones de implementación](#) que ofrece Elastic Beanstalk.

Para obtener más información sobre los procesadores Graviton basados en arm64, consulte estos recursos: AWS

- Ventajas: el procesador [AWS Graviton](#)
- Introducción y otros temas, como consideraciones específicas del idioma: artículo sobre [cómo empezar a](#) usar Graviton AWS GitHub

Configuración de instancias de Amazon EC2 para su entorno

Puede crear o modificar la configuración de la instancia de Amazon EC2 de su entorno de Elastic Beanstalk en la consola de Elastic Beanstalk.

Note

Si bien la consola de Elastic Beanstalk no ofrece la opción de cambiar la arquitectura del procesador de un entorno existente, puede hacerlo con AWS CLI. Para ver ejemplos de comandos, consulte [Configuración de las instancias AWS EC2 para su entorno mediante el AWS CLI](#)

Para configurar las instancias de Amazon EC2 en la consola de Elastic Beanstalk durante la creación del entorno

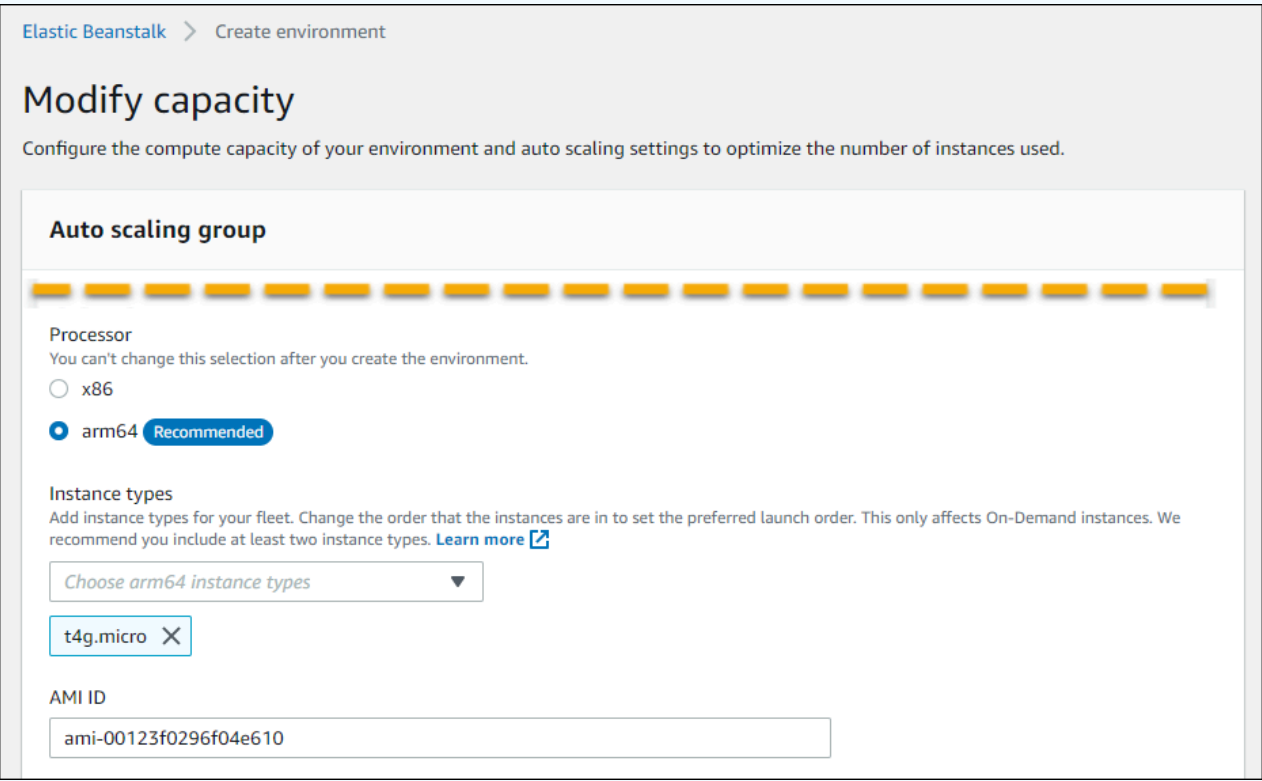
1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Entornos.
3. Elija [Create a new environment \(Crear un nuevo entorno\)](#) para empezar a crear el entorno.
4. En la página principal del asistente, antes de elegir Create environment (Crear entorno), elija Configure more options (Configurar más opciones).
5. En la categoría de configuración Instances (Instancias), elija Edit (Editar). Modifique la configuración de esta categoría y elija Apply (Aplicar). Para ver descripciones de las opciones de configuración, consulte la sección [the section called “Configuración de categorías de instancias”](#) de esta página.
6. En la categoría de configuración Capacity (Capacidad), elija Edit (Editar). Modifique la configuración de esta categoría y elija Continue (Continuar). Para ver descripciones de las opciones de configuración, consulte la sección [the section called “Configuración de la categoría de capacidad”](#) de esta página.

Selección de la arquitectura del procesador

Vaya a Processor (Procesador) y seleccione una arquitectura del procesador para sus instancias de EC2. La consola enumera las arquitecturas de los procesadores que son compatibles con la plataforma que eligió previamente en el panel Create environment (Crear entorno).

Si no aparece la arquitectura del procesador que necesita, regrese a la lista de categorías de configuración y seleccione una plataforma que sea compatible con ella. Desde el panel Modify Capacity (Modificar capacidad), elija Cancel (Cancelar).

A continuación, elija Change platform version (Cambiar versión de la plataforma) para elegir una nueva configuración de plataforma. Seguidamente, en la categoría de configuración Capacity (Capacidad), elija Edit (Editar) para visualizar de nuevo las opciones de la arquitectura del procesador.



Elastic Beanstalk > Create environment

Modify capacity

Configure the compute capacity of your environment and auto scaling settings to optimize the number of instances used.

Auto scaling group

Processor
You can't change this selection after you create the environment.

x86

arm64 **Recommended**

Instance types
Add instance types for your fleet. Change the order that the instances are in to set the preferred launch order. This only affects On-Demand instances. We recommend you include at least two instance types. [Learn more](#)

Choose arm64 instance types ▼

t4g.micro X

AMI ID

ami-00123f0296f04e610

7. Elija Save (Guardar) y haga los demás cambios de configuración que requiera su entorno.
8. Seleccione Crear entorno.

Para configurar las instancias de Amazon EC2 de un entorno en ejecución en la consola de Elastic Beanstalk

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Instances (Instancias), elija Edit (Editar). Modifique la configuración de esta categoría y elija Apply (Aplicar). Para ver descripciones de las opciones de

configuración, consulte la sección [the section called “Configuración de categorías de instancias”](#) de esta página.

5. En la categoría de configuración Capacity (Capacidad), elija Edit (Editar). Modifique la configuración de esta categoría y elija Continue (Continuar). Para ver descripciones de las opciones de configuración, consulte la sección [the section called “Configuración de la categoría de capacidad”](#) de esta página.

Configuración de categorías de instancias

Los siguientes valores relacionados con las instancias de Amazon EC2 están disponibles en la categoría de configuración Instances (Instancias).

Opciones

- [Intervalo de monitorización](#)
- [Volumen raíz \(dispositivo de arranque\)](#)
- [Servicio de metadatos de instancia](#)
- [Grupos de seguridad](#)

Modify instances

Amazon CloudWatch monitoring

The time interval between when metrics are reported from the EC2 instances.

Monitoring interval

5 minute ▼

Root volume (boot device)

Root volume type

(Container default) ▼

Size

The number of gigabytes of the root volume attached to each instance.

GB

IOPS

Input/output operations per second for a provisioned IOPS (SSD) volume.

100 IOPS

Throughput

The desired throughput to provision for the Amazon EBS root volume attached to your environment's EC2 instance

MiB/s

Instance metadata service (IMDS)

Your environment's platform supports both IMDSv1 and IMDSv2. To enforce IMDSv2, disable IMDSv1. [Learn more](#)

Disable IMDSv1

With the current setting, the environment enables both IMDSv1 and IMDSv2.

Disabled

EC2 security groups

	Group name	Group ID	Name
<input type="checkbox"/>	awseb-e-awppgphwta-stack-AWSEBLoadBalancerSecurityGroup-LUAOUHKL3SNI	sg-027aafe45182f171f	WinTest-dev
<input type="checkbox"/>	awseb-e-awppgphwta-stack-AWSEBSecurityGroup-10905QSLX6UCC	sg-020e30e60b3e80c5b	WinTest-dev
<input type="checkbox"/>	awseb-e-m5yhre5nuj-stack-AWSEBLoadBalancerSecurityGroup-PIICIFO0QHGG	sg-03879e31c4ebe98ea	Gettingstarted-env
<input checked="" type="checkbox"/>	awseb-e-m5yhre5nuj-stack-AWSEBSecurityGroup-12122MOSKFTC4	sg-05b1982101cf211ef	Gettingstarted-env
<input type="checkbox"/>	default	sg-3527cd14	

Cancel

Continue

Apply

Intervalo de monitorización

De forma predeterminada, las instancias de su entorno publican [métricas de estado básicas](#) CloudWatch en Amazon en intervalos de cinco minutos sin coste adicional.

Para obtener informes más detallados, puede establecer el intervalo de monitoreo en 1 minuto para aumentar la frecuencia con la que los recursos de su entorno publican [las métricas de estado básicas](#). CloudWatch CloudWatch se aplican cargos de servicio a las métricas de intervalos de un minuto. Para obtener más información, consulta [Amazon CloudWatch](#).

Volumen raíz (dispositivo de arranque)

Cada instancia del entorno está configurada con un volumen raíz. El volumen raíz es el dispositivo de bloques de Amazon EBS adjunto a la instancia en el que se almacena el sistema operativo, las bibliotecas, los scripts y el código fuente de la aplicación. De forma predeterminada, todas las plataformas utilizan dispositivos de bloques SSD de uso general para almacenamiento.

Si es necesario, puede modificar el Root volume type (Tipo de volumen raíz) para utilizar un volumen de almacenamiento magnético o un volumen SSD de IOPS provisionadas y, en caso necesario, aumentar el tamaño. En el caso de los volúmenes de IOPS provisionadas, también deberá seleccionar el número de IOPS que se van a aprovisionar. Throughput (Rendimiento) solo es aplicable a los tipos de volumen de SSD gp3. Puede ingresar el rendimiento deseado para aprovisionar. Puede oscilar entre 125 y 1000 mebibytes por segundo (Mib/s). Seleccione un tipo de volumen que se ajuste a sus necesidades de desempeño y precio.

Para obtener más información, consulte los [tipos de volumen de Amazon EBS](#) en la Guía del usuario de Amazon EC2 y los detalles del producto de [Amazon EBS](#).

Servicio de metadatos de instancia

El servicio de metadatos de instancia (IMDS) es un componente de la instancia que utiliza el código de instancia para acceder de forma segura a los metadatos de la instancia. El código permite acceder a los metadatos de una instancia de ejecución mediante uno de los dos métodos siguientes. Se trata de la versión 1 o 2 del Servicio de metadatos de instancia (IMDSv1 o IMDSv2). IMDSv2 es más seguro. Deshabilite IMDSv1 para aplicar IMDSv2. Para obtener más información, consulte [the section called “IMDS”](#).

Note

La sección IMDS de esta página de configuración aparece solo para versiones de la plataforma compatibles con IMDSv2.

Grupos de seguridad

Los grupos de seguridad que están adjuntos a las instancias determinan qué tráfico puede llegar a las instancias. También determinan qué tráfico puede abandonarlas. Elastic Beanstalk crea un grupo de seguridad que permite el tráfico procedente del balanceador de carga en los puertos estándar para HTTP (80) y HTTPS (443).

Puede especificar grupos de seguridad adicionales que haya creado para permitir el tráfico en otros puertos o que proceda de otros orígenes. Por ejemplo, puede crear un grupo de seguridad para el acceso de SSH que permita el tráfico entrante en el puerto 22 de un intervalo restringido de direcciones IP. De lo contrario, si desea aumentar la seguridad, cree uno que permita el tráfico procedente de un host bastión al que solo usted tenga acceso.

Note

Para permitir el tráfico entre las instancias del entorno A y las del entorno B, puede agregar una regla al grupo de seguridad que Elastic Beanstalk ha adjuntado al entorno B. Luego, puede especificar el grupo de seguridad que Elastic Beanstalk ha adjuntado al entorno A. Esto permite la entrada desde las instancias del entorno A o la salida hacia ellas. Sin embargo, esto crea una dependencia entre los dos grupos de seguridad. Si posteriormente intenta terminar el entorno A, Elastic Beanstalk no puede eliminar el grupo de seguridad del entorno, ya que el grupo de seguridad del entorno B depende de él. Por lo tanto, se recomienda crear primero un grupo de seguridad independiente. A continuación, adjúntelo al entorno A y especifíquelo en una regla del grupo de seguridad del entorno B.

Para obtener más información sobre los grupos de seguridad de Amazon EC2, consulte los grupos de seguridad de [Amazon EC2 en la Guía](#) del usuario de Amazon EC2.

Configuración de la categoría de capacidad

Las siguientes opciones relacionadas con las instancias de Amazon EC2 están disponibles en la categoría de configuración Capacity (Capacidad).

Opciones

- [Tipos de instancias](#)
- [ID de AMI](#)

Elastic Beanstalk > Environments > Gettingstartedz-env > Configuration

Modify capacity

Configure the compute capacity of your environment and auto scaling settings to optimize the number of instances used.

Auto scaling group

Instance types
Add instance types for your fleet. Change the order that the instances are in to set the preferred launch order. This only affects On-Demand instances. We recommend you include at least two instance types. [Learn more](#)

-- Choose instance types --

t4g.medium X t4g.2xlarge X

AMI ID
ami-00123f0296f04e610

Tipos de instancias

La configuración Instance type (Tipo de instancias) determina el tipo de instancia de Amazon EC2 que se lanza para que se ejecute la aplicación. En esta página de configuración se muestra una lista de Tipos de instancias. Puede seleccionar uno o varios tipos de instancias. En la consola de Elastic Beanstalk solo se muestran los tipos de instancias según la arquitectura del procesador configurada para su entorno. Por lo tanto, solo puede agregar tipos de instancias de la misma arquitectura del procesador.

Note

Si bien la consola de Elastic Beanstalk no ofrece la opción de cambiar la arquitectura del procesador de un entorno existente, puede hacerlo con AWS CLI. Para ver ejemplos de comandos, consulte [Configuración de las instancias AWS EC2 para su entorno mediante el AWS CLI](#)

Elija una instancia que tenga la capacidad suficiente como para ejecutar la aplicación con carga de trabajo, pero no una capacidad excesiva como para que esté inactiva la mayor parte del tiempo. En escenarios de desarrollo, la familia de instancias t2 ofrece una capacidad moderada, que puede ampliarse por ráfagas durante cortos períodos de tiempo. En el caso de las aplicaciones de alta disponibilidad y gran escala, utilice un grupo de instancias para garantizar que la capacidad no se ve muy afectada si una instancia deja de funcionar. Comience con un tipo de instancia que le permita ejecutar cinco instancias bajo cargas moderadas durante el horario normal. Si alguna instancia falla, las demás instancias podrán absorber el resto del tráfico. Esta reserva de capacidad también proporciona tiempo para que el entorno pueda escalarse cuando el tráfico comience a aumentar durante las horas de mayor actividad.

Para obtener más información sobre las familias y los tipos de instancias de Amazon EC2, consulte Tipos de [instancias](#) en la Guía del usuario de Amazon EC2 o [Tipos de instancias](#) en la Guía del usuario de Amazon EC2. Para determinar qué tipos de instancias cumplen sus requisitos y las regiones compatibles, consulte [Tipos de instancias disponibles](#) en la Guía del usuario de Amazon EC2 o [Tipos de instancias disponibles](#) en la Guía del usuario de Amazon EC2.

ID de AMI

Amazon Machine Image (AMI) es la imagen de máquina de Amazon Linux o Windows Server que Elastic Beanstalk utiliza para lanzar instancias de Amazon EC2 en el entorno. Elastic Beanstalk cuenta con imágenes de máquina que contienen las herramientas y recursos necesarios para ejecutar la aplicación.

Elastic Beanstalk selecciona una AMI predeterminada para su entorno en función de la región, la versión de la plataforma y la arquitectura del procesador que elija. Si ha creado una [AMI personalizada](#), sustituya el ID de la AMI predeterminada por el suyo propio.

Configuración de las instancias AWS EC2 para su entorno mediante el AWS CLI

Utilice la interfaz de línea de AWS comandos (AWS CLI) para crear y configurar entornos de Elastic Beanstalk mediante los comandos del shell de línea de comandos. En esta sección se proporcionan ejemplos de los comandos [create-environment](#) y [update-environment](#).

En los dos primeros ejemplos se crea un nuevo entorno. El comando especifica un tipo de instancias de Amazon EC2, t4g.small, que se basa en la arquitectura del procesador arm64. Elastic Beanstalk establece el ID de imagen (AMI) de forma predeterminada para las instancias de EC2 en función de la región, la versión de la plataforma y el tipo de instancia. El tipo de instancia corresponde a

una arquitectura del procesador. El parámetro `solution-stack-name` se aplica a la versión de la plataforma.

Example 1 — crear un nuevo entorno basado en arm64 (opciones de espacio de nombres en línea)

```
aws elasticbeanstalk create-environment \  
--region us-east-1 \  
--application-name my-app \  
--environment-name my-env \  
--solution-stack-name "64bit Amazon Linux 2 v3.4.7 running Docker" \  
--option-settings \  
Namespace=aws:autoscaling:launchconfiguration,OptionName=IamInstanceProfile,Value=aws-elasticbeanstalk-ec2-role \  
Namespace=aws:ec2:instances,OptionName=InstanceTypes,Value=t4g.small
```

Como alternativa, utilice un archivo `options.json` para especificar las opciones del espacio de nombres en lugar de incluirlas en línea.

Example 2 — crear un nuevo entorno basado en arm64 (opciones de espacio de nombres en el archivo **options.json**)

```
aws elasticbeanstalk create-environment \  
--region us-east-1 \  
--application-name my-app \  
--environment-name my-env \  
--solution-stack-name "64bit Amazon Linux 2 v3.4.7 running Docker" \  
--option-settings file://options.json
```

Example

```
### example options.json ###  
[  
  {  
    "Namespace": "aws:autoscaling:launchconfiguration",  
    "OptionName": "IamInstanceProfile",  
    "Value": "aws-elasticbeanstalk-ec2-role"  
  },  
  {  
    "Namespace": "aws:ec2:instances",  
    "OptionName": "InstanceTypes",  
    "Value": "t4g.small"  
  }  
]
```

```
}  
]
```

En los siguientes dos ejemplos se actualiza la configuración de un entorno existente con el comando [update-environment](#). En este ejemplo se agrega otro tipo de instancia que también se basa en la arquitectura del procesador arm64. En el caso de los entornos existentes, todos los tipos de instancia que se agreguen deben contar con la misma arquitectura del procesador. Si desea reemplazar los tipos de instancias existentes por otros de una arquitectura diferente, puede hacerlo. Pero asegúrate de que todos los tipos de instancias del comando tengan el mismo tipo de arquitectura.

Example 3 — actualizar un entorno basado en arm64 existente (opciones de espacio de nombres en línea)

```
aws elasticbeanstalk update-environment \  
--region us-east-1 \  
--application-name my-app \  
--environment-name my-env \  
--solution-stack-name "64bit Amazon Linux 2 v3.4.7 running Docker" \  
--option-settings \  
Namespace=aws:autoscaling:launchconfiguration,OptionName=IamInstanceProfile,Value=aws-elasticbeanstalk-ec2-role \  
Namespace=aws:ec2:instances,OptionName=InstanceTypes,Value=t4g.small,t4g.micro
```

Como alternativa, utilice un archivo `options.json` para especificar las opciones del espacio de nombres en lugar de incluirlas en línea.

Example 4 — actualizar un entorno basado en arm64 existente (opciones de espacio de nombres en el archivo `options.json`)

```
aws elasticbeanstalk update-environment \  
--region us-east-1 \  
--application-name my-app \  
--environment-name my-env \  
--solution-stack-name "64bit Amazon Linux 2 v3.4.7 running Docker" \  
--option-settings file://options.json
```

Example

```
### example options.json ###
```

```
[
  {
    "Namespace": "aws:autoscaling:launchconfiguration",
    "OptionName": "IamInstanceProfile",
    "Value": "aws-elasticbeanstalk-ec2-role"
  },
  {
    "Namespace": "aws:ec2:instances",
    "OptionName": "InstanceTypes",
    "Value": "t4g.small, t4g.micro"
  }
]
```

En los siguientes dos ejemplos se muestran más comandos [create-environment](#). En estos ejemplos no se proporcionan valores para InstanceTypes. Cuando no se especifican los valores InstanceTypes, Elastic Beanstalk utiliza la arquitectura del procesador basada en x86 de forma predeterminada. El ID de imagen (AMI) para las instancias de EC2 del entorno vendrá predeterminado en función de la región, la versión de la plataforma y el tipo de instancia predeterminado. El tipo de instancia corresponde a una arquitectura del procesador.

Example 5 — crear un nuevo entorno basado en x86 (opciones de espacio de nombres en línea)

```
aws elasticbeanstalk create-environment \
--region us-east-1 \
--application-name my-app \
--environment-name my-env \
--solution-stack-name "64bit Amazon Linux 2 v3.4.7 running Docker" \
--option-settings \
Namespace=aws:autoscaling:launchconfiguration,OptionName=IamInstanceProfile,Value=aws-elasticbeanstalk-ec2-role
```

Como alternativa, utilice un archivo `options.json` para especificar las opciones del espacio de nombres en lugar de incluirlas en línea.

Example 6 — crear un nuevo entorno basado en x86 (opciones de espacio de nombres en el archivo `options.json`)

```
aws elasticbeanstalk create-environment \
```

```
--region us-east-1 \  
--application-name my-app \  
--environment-name my-env \  
--solution-stack-name "64bit Amazon Linux 2 v3.4.7 running Docker" \  
--option-settings file://options.json
```

Example

```
### example options.json ###  
[  
  {  
    "Namespace": "aws:autoscaling:launchconfiguration",  
    "OptionName": "IamInstanceProfile",  
    "Value": "aws-elasticbeanstalk-ec2-role"  
  }  
]
```

Recomendaciones para los entornos de primera oleada Graviton arm64

Note

Esta sección solo se aplica a un subconjunto de clientes. Si creó un nuevo entorno con tipos de instancia basados en Graviton arm64 antes del 24 de noviembre de 2021, es posible que la información incluida en esta sección se aplique a su caso.

Acciones recomendadas para los entornos de primera oleada de Graviton arm64

A partir de octubre y noviembre de 2021, Elastic Beanstalk comenzó a agregar oleadas de compatibilidad con los procesadores Graviton arm64 en algunas regiones y para algunas versiones de la plataforma. Esta primera oleada se anunció en las Notas de la versión de AWS Elastic Beanstalk del [13 de octubre](#), [21 de octubre](#) y [19 de noviembre](#) de 2021. Si creó entornos basados en arm64, las instrucciones le indicaron que configurara las instancias con las AMI personalizadas proporcionadas en las notas de la versión. Ahora que está disponible la compatibilidad mejorada con Graviton arm64, Elastic Beanstalk establece las AMI de forma predeterminada para los tipos de instancia arm64 en las últimas versiones de la plataforma.

Si creó entornos con las AMI personalizadas proporcionadas en las primeras versiones de la oleada, se recomienda hacer lo siguiente si desea mantener un entorno en buen estado y que funcione.

1. Elimine las AMI personalizadas del entorno.
2. Actualice el entorno con la última versión de la plataforma.
3. Configure las [actualizaciones de plataforma administradas](#) para que se actualice automáticamente a la última versión de la plataforma durante un periodo de mantenimiento programado.

Note

Elastic Beanstalk no reemplazará automáticamente las AMI personalizadas. Debe eliminar las AMI personalizadas en el paso 1, para que la próxima actualización de la plataforma que se realice en el paso 2 las actualice.

El procedimiento que se describe a continuación lo guiará a través de estos pasos. Los AWS CLI ejemplos se aplican a un entorno que se creó con la siguiente información.

```
aws elasticbeanstalk create-environment \  
--region us-east-1 \  
--application-name my-app \  
--environment-name my-env \  
--solution-stack-name "64bit Amazon Linux 2 v3.4.7 running Docker" \  
--option-settings \  
Namespace=aws:autoscaling:launchconfiguration,OptionName=IamInstanceProfile,Value=aws-elasticbeanstalk-ec2-role \  
Namespace=aws:ec2:instances,OptionName=InstanceTypes,Value=t4g.small \  
Namespace=aws:autoscaling:launchconfiguration,OptionName=ImageId,Value=ami-  
0fbdb88ce139244bf
```

Para actualizar los entornos arm64 creados durante la primera oleada de compatibilidad con Graviton arm64

1. Ejecute [update-environment](#) y elimine la configuración de las AMI personalizadas.

```
aws elasticbeanstalk update-environment \  
--region us-east-1 \  
--environment-name my-env \  
--options-to-remove \  
Namespace=aws:autoscaling:launchconfiguration,OptionName=ImageId
```

2. Actualice el entorno con la última versión de la plataforma. Elija una de las siguientes opciones.

- Opción de consola: utilice la consola de Elastic Beanstalk para actualizar la versión de la plataforma. Para obtener más información, consulte [Actualización de la versión de la plataforma de su entorno](#).
- AWS CLI Opción: ejecute el comando AWS [update-environment](#), especificando la versión de plataforma disponible más reciente.

```
aws elasticbeanstalk update-environment \  
--region us-east-1 \  
--environment-name my-env \  
--solution-stack-name "64bit Amazon Linux 2 v3.4.9 running Docker"
```

Note

El [list-available-solution-stacks](#) comando proporciona una lista de las versiones de plataforma disponibles para su cuenta en una AWS región.

```
aws elasticbeanstalk list-available-solution-stacks --region us-east-1 --  
query SolutionStacks
```

3. Utilice la consola de Elastic Beanstalk y configure las actualizaciones de plataforma administradas para su entorno. Las actualizaciones de plataforma administradas actualizan automáticamente el entorno a la última versión de la plataforma durante un periodo de mantenimiento programado. Su aplicación permanece en servicio durante el proceso de actualización. Para obtener más información, consulte [Actualizaciones de la plataforma administradas](#).

El espacio de nombres **aws:autoscaling:launchconfiguration**

Puede utilizar las [opciones de configuración](#) del espacio de nombres [aws:autoscaling:launchconfiguration](#) para configurar las instancias de su entorno, incluidas otras opciones que no están disponibles en la consola.

En el siguiente ejemplo de [archivo de configuración](#) se utilizan las opciones de configuración básicas que se encuentran en este tema. Por ejemplo, utiliza la opción `DisableIMDSv1`, que se explica en [IMDS](#). También utiliza las opciones `EC2KeyName` y `IamInstanceProfile` que se analizan en [Seguridad](#) y la opción `BlockDeviceMappings`, que no está disponible en la consola.

```
option_settings:
  aws:autoscaling:launchconfiguration:
    SecurityGroups: my-securitygroup
    MonitoringInterval: "1 minute"
    DisableIMDSv1: false
    EC2KeyName: my-keypair
    IamInstanceProfile: "aws-elasticbeanstalk-ec2-role"
    BlockDeviceMappings: "/dev/sdj=:100,/dev/sdh=snap-51eef269,/dev/sdb=ephemeral0"
```

Puede utilizar `BlockDeviceMappings` para configurar dispositivos de bloques adicionales para las instancias. Para obtener más información, consulte [Block Device Mapping](#) en la Guía del usuario de Amazon EC2.

La CLI de EB y la consola de Elastic Beanstalk aplican los valores recomendados a las opciones anteriores. Debe eliminar estos ajustes si desea usar archivos de configuración para configurarlos. Para obtener más información, consulte [Valores recomendados](#).

Configuración del servicio de metadatos de instancia

Los metadatos de instancia son datos relacionados con una instancia de Amazon Elastic Compute Cloud (Amazon EC2) que las aplicaciones pueden utilizar para configurar o administrar la instancia de ejecución. El servicio de metadatos de instancia (IMDS) es un componente de la instancia que utiliza el código de instancia para acceder de forma segura a los metadatos de la instancia. Este código puede ser el código de la plataforma Elastic Beanstalk en las instancias de su entorno AWS, el SDK que pueda estar utilizando su aplicación o incluso el código propio de la aplicación. Para obtener más información, consulte [Metadatos de instancia y datos de usuario](#) en la Guía del usuario de Amazon EC2.

El código puede acceder a metadatos de instancia a partir de una instancia en ejecución mediante uno de estos dos métodos: Servicio de metadatos de instancia versión 1 (IMDSv1) o Servicio de metadatos de instancia versión 2 (IMDSv2). IMDSv2 utiliza solicitudes orientadas a la sesión y mitiga varios tipos de vulnerabilidades que podrían utilizarse para intentar acceder al IMDS. Para obtener información sobre estos dos métodos, consulte [Configuración del servicio de metadatos de la instancia](#) en la Guía del usuario de Amazon EC2.

Secciones

- [Soporte de plataformas para IMDS](#)
- [Selección de métodos IMDS](#)
- [Configuración de IMDS mediante la consola de Elastic Beanstalk](#)

- [El espacio de nombres aws:autoscaling:launchconfiguration](#)

Soporte de plataformas para IMDS

Las versiones más antiguas de la plataforma Elastic Beanstalk admiten IMDSv1. Las versiones más recientes de la plataforma Elastic Beanstalk (todas [las versiones de la plataforma Amazon Linux 2](#)) admiten tanto IMDSv1 como IMDSv2. Puede configurar el entorno para que admita ambos métodos (el valor predeterminado) o deshabilitar IMDSv1.

Note

Desactivar el IMDSv1 requiere utilizar las plantillas de lanzamiento de Amazon EC2. Cuando configura esta característica durante la creación del entorno o las actualizaciones, Elastic Beanstalk intenta configurar el entorno para que utilice plantillas de lanzamiento de Amazon EC2 (si el entorno aún no las está utilizando). En este caso, si la política de usuario carece de los permisos necesarios, la creación del entorno o las actualizaciones podrían producir un error. Por lo tanto, le recomendamos que utilice nuestra política de usuario administrada o que agregue los permisos necesarios a sus políticas personalizadas. Para obtener información detallada acerca de los permisos necesarios, consulte [the section called “Creación de una política de usuario personalizada”](#).

Selección de métodos IMDS

Al tomar una decisión sobre los métodos IMDS que desea que admita su entorno, tenga en cuenta los siguientes casos de uso:

- **AWS SDK:** si su aplicación usa un AWS SDK, asegúrese de usar la versión más reciente del SDK. Los AWS SDK realizan llamadas al IMDS, y las versiones más recientes del SDK utilizan IMDSv2 siempre que es posible. Si alguna vez desactiva IMDSv1 o si su aplicación utiliza una versión antigua del SDK, las llamadas IMDS podrían fallar.
- **El código de su aplicación:** si su aplicación realiza llamadas al IMDS, considere la posibilidad de utilizar el AWS SDK para que pueda realizar las llamadas en lugar de realizar solicitudes HTTP directas. De esta manera, no necesita realizar cambios en el código para cambiar entre los métodos IMDS. El AWS SDK usa IMDSv2 siempre que es posible.
- **Código de plataforma de Elastic Beanstalk:** nuestro código realiza llamadas de IMDS AWS a través del SDK y, por lo tanto, utiliza IMDSv2 en todas las versiones de plataforma compatibles. Si su

código usa un up-to-date AWS SDK y realiza todas las llamadas al IMDS a través del SDK, puede deshabilitar IMDSv1 de forma segura.

Configuración de IMDS mediante la consola de Elastic Beanstalk

Puede modificar la configuración de la instancia de Amazon EC2 del entorno de Elastic Beanstalk mediante la consola de Elastic Beanstalk.

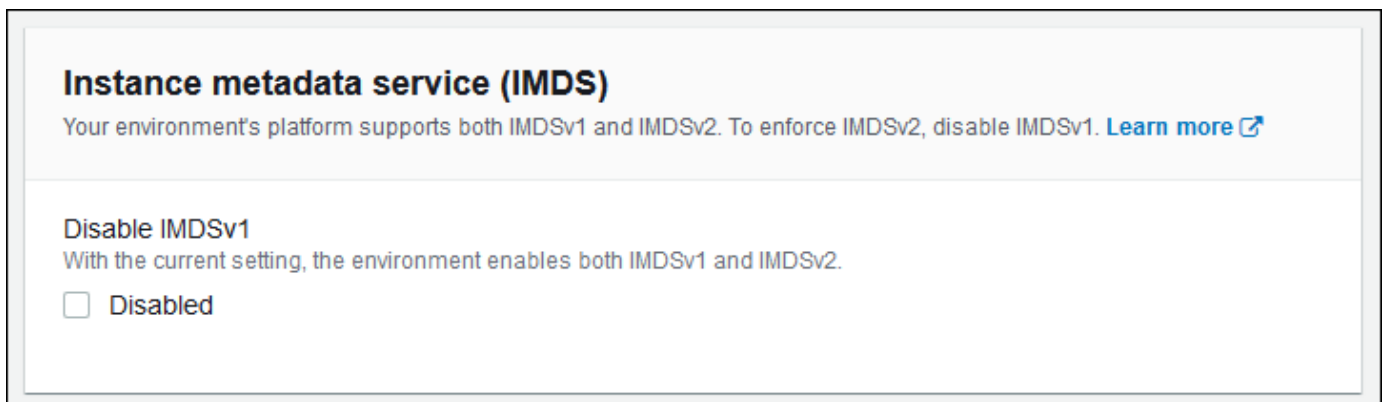
Para configurar IMDS en sus instancias de Amazon EC2 mediante la consola de Elastic Beanstalk

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Instances (Instancias), elija Edit (Editar).



5. Seleccione Disable IMDSv1 (Desactivar IMDSv1) para aplicar IMDSv2. Desactive Disable IMDSv1 (Desactivar IMDSv1) para habilitar IMDSv1 e IMDSv2.
6. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

El espacio de nombres aws:autoscaling:launchconfiguration

Puede utilizar una [opción de configuración](#) en el espacio de nombres [aws:autoscaling:launchconfiguration](#) para configurar IMDS en las instancias del entorno.

En el siguiente ejemplo del [archivo de configuración](#) se deshabilita IMDSv1 mediante la opción `DisableIMDSv1`.

```
option_settings:
  aws:autoscaling:launchconfiguration:
    DisableIMDSv1: true
```

Grupo de Auto Scaling para su entorno de Elastic Beanstalk

Su AWS Elastic Beanstalk entorno incluye un grupo de Auto Scaling que administra las [instancias de Amazon EC2](#) de su entorno. En un entorno de una sola instancia, el grupo de Auto Scaling garantiza que siempre haya una instancia en ejecución. En un entorno con balanceo de carga, se configura el grupo con una serie de instancias en ejecución y Auto Scaling agrega o elimina instancias según sea necesario y en función de la carga.

El grupo de Auto Scaling también aplica la configuración de lanzamiento de las instancias del entorno. Puede [modificar la configuración de lanzamiento](#) para cambiar el tipo de instancia, el par de claves, el almacenamiento de Amazon Elastic Block Store (Amazon EBS) y otros ajustes que solo se pueden configurar al lanzar una instancia.

El grupo Auto Scaling utiliza dos CloudWatch alarmas de Amazon para activar las operaciones de escalado. Los desencadenadores predeterminados adaptan su escala cuando el tráfico de la red saliente promedio de cada instancia es superior a 6 MiB o inferior a 2 MiB durante un periodo de cinco minutos. Para utilizar Auto Scaling de forma eficaz, [configure desencadenadores](#) adecuados para su aplicación, tipo de instancia y requisitos de servicio. Puede optar por la escala en función de varias estadísticas, como la latencia, E/S de disco, la utilización de la CPU y el recuento de solicitudes.

Para optimizar el uso de instancias de Amazon EC2 en el entorno mediante periodos predecibles de picos de tráfico, [configure el grupo de Auto Scaling para cambiar el recuento de instancias de manera programada](#). Puede programar cambios en la configuración de su grupo que pueden producirse con una periodicidad diaria o semanal, o bien programar cambios puntuales para prepararse para eventos de marketing que dirigirán una gran cantidad de tráfico a su sitio.

De manera opcional, Elastic Beanstalk puede combinar instancias en diferido y [de spot](#) para su entorno. Puede configurar Amazon EC2 Auto Scaling para monitorear y responder de manera automática a los cambios que afectan la disponibilidad de las instancias de spot al habilitar el [reequilibrio de la capacidad](#).

Auto Scaling también monitorea el estado de cada Amazon EC2 instance que lanza. Si una instancia termina de forma inesperada, Auto Scaling lo detecta y lanza una instancia de sustitución. Para configurar el grupo para utilizar el mecanismo de comprobación de estado del balanceador de carga, consulte [Configuración de comprobaciones de estado de Auto Scaling](#).

Puede configurar Auto Scaling en su entorno mediante la [consola de Elastic Beanstalk](#), la [CLI de EB](#) o las [opciones de configuración](#).

Temas

- [Compatibilidad con instancias de spot](#)
- [Configuración del grupo de Auto Scaling mediante la consola de Elastic Beanstalk](#)
- [Configuración del grupo de Auto Scaling a través de la CLI de EB](#)
- [Opciones de configuración](#)
- [Desencadenadores de Auto Scaling](#)
- [Acciones programadas de Auto Scaling](#)
- [Configuración de comprobaciones de estado de Auto Scaling](#)

Compatibilidad con instancias de spot

Para aprovechar las [instancias de spot](#) de Amazon EC2, puede habilitar una opción de spot en el entorno. El grupo de Auto Scaling de su entorno combinará entonces las opciones de compra de Amazon EC2 y mantendrá una combinación de instancias en diferido y de spot.

En este tema se describen los siguientes métodos para habilitar las solicitudes de instancias de spot del entorno:

- Consola de Elastic Beanstalk: para obtener más información, consulte Composición de flota en [the section called “Configuración del grupo de Auto Scaling mediante la consola de Elastic Beanstalk”](#).
- CLI de EB: para obtener más información, consulte [the section called “Configuración del grupo de Auto Scaling a través de la CLI de EB”](#).
- Opciones de configuración de espacio de nombres de `aws:ec2:instances`: para obtener más información, consulte [the section called “Opciones de configuración”](#).

⚠ Important

La demanda de instancias de spot puede variar enormemente de un momento a otro y la disponibilidad de las instancias de spot también puede variar significativamente en función de cuántas instancias de Amazon EC2 no utilizadas haya disponibles. Siempre es posible que la instancia de spot se vea interrumpida.

Para ayudar a minimizar el impacto de estas interrupciones en la aplicación, puede habilitar la opción de reequilibrio de la capacidad incluida con Amazon EC2 Auto Scaling. Con esta característica habilitada, EC2 intenta reemplazar de manera automática las instancias de spot en un grupo de Auto Scaling antes de que se interrumpan. Con la finalidad de habilitar esta característica, utilice la consola de Elastic Beanstalk para [configurar el grupo de Auto Scaling](#). En su defecto, puede establecer la [opción de configuración EnableCapacityRebalancing](#) de Elastic Beanstalk en `true` el espacio de nombres [aws:autoscaling:asg](#).

Para obtener más información, consulte [Reequilibrio de capacidad](#) en la Guía del usuario de Amazon EC2 Auto Scaling e [Interrupciones de instancias puntuales](#) en la Guía del usuario de Amazon EC2.

Elastic Beanstalk dispone de varias opciones de configuración que admiten la característica Spot. Estas se tratan en las siguientes secciones relacionadas con la configuración del grupo de Auto Scaling.

Dos de estas opciones, en el espacio de nombres [aws:ec2:instances](#), merecen una atención especial:

- `SpotFleetOnDemandBase`
- `SpotFleetOnDemandAboveBasePercentage`

Estas dos opciones se correlacionan con la opción `MinSize` en el espacio de nombres [aws:autoscaling:asg](#):

- Solo `MinSize` determina la capacidad inicial del entorno: el número de instancias que debe ejecutar como mínimo.
- `SpotFleetOnDemandBase` no afecta a la capacidad inicial. Si se habilita Spot, esta opción solo determina cuántas instancias en diferido se aprovisionan antes de que se considere la posibilidad de utilizar instancias de spot.

- Tenga en cuenta cuándo `SpotFleetOnDemandBase` es menor que `MinSize`. Obtendrá exactamente `MinSize` instancias como capacidad inicial. Como mínimo `SpotFleetOnDemandBase` deben ser instancias en diferido.
- Tenga en cuenta cuándo `SpotFleetOnDemandBase` es mayor que `MinSize`. A medida que el entorno escale de forma horizontal, tendrá la seguridad de obtener al menos una cantidad de instancias adicionales igual a la diferencia entre los dos valores. En otras palabras, se garantiza que obtendrá al menos $(\text{SpotFleetOnDemandBase} - \text{MinSize})$ instancias adicionales que son en diferido antes de cumplir con el requisito de `SpotFleetOnDemandBase`.

En entornos de producción, las instancias de spot resultan particularmente útiles en un entorno escalable con equilibrio de carga. No es conveniente utilizar Spot en un entorno con una única instancia. Si no hay instancias de spot disponibles, podría perder toda la capacidad del entorno (una sola instancia). Es posible que desee utilizar una instancia de spot en un entorno con una sola instancia para realizar el desarrollo o las pruebas. Si lo hace, asegúrese de establecer `SpotFleetOnDemandBase` y `SpotFleetOnDemandAboveBasePercentage` en cero. Cualquier otra configuración dará como resultado una instancia en diferido.

Notas

- Es posible que algunas AWS cuentas antiguas proporcionen a Elastic Beanstalk tipos de instancias predeterminados que no admiten instancias puntuales (por ejemplo, t1.micro). Si habilita las solicitudes de instancias de spot y aparece el error `None of the instance types you specified supports Spot` (Ninguno de los tipos de instancia especificados admite Spot), no olvide configurar los tipos de instancias compatibles con Spot. Para seleccionar tipos de instancias de spot, utilice el [Asistente de instancias de spot](#).
- Para habilitar las solicitudes de instancias de spot debe usar plantillas de lanzamiento de Amazon EC2. Cuando configura esta característica durante la creación del entorno o las actualizaciones, Elastic Beanstalk intenta configurar el entorno para que utilice plantillas de lanzamiento de Amazon EC2 (si el entorno aún no las está utilizando). En este caso, si la política de usuario carece de los permisos necesarios, la creación del entorno o las actualizaciones podrían producir un error. Por lo tanto, le recomendamos que utilice nuestra política de usuario administrada o que agregue los permisos necesarios a sus políticas personalizadas. Para obtener información detallada acerca de los permisos necesarios, consulte [the section called “Creación de una política de usuario personalizada”](#).

En los siguientes ejemplos, se muestran diversos escenarios de configuración con distintas opciones de escala. En todos los ejemplos, se presupone que se está utilizando un entorno con balanceo de carga y que las solicitudes de instancias de spot están habilitadas.

Example 1: Instancias en diferido y de spot como parte de la capacidad inicial

Opciones de configuración

Opción	Namespace	Valor
MinSize	aws:autoscaling:asg	10
MaxSize	aws:autoscaling:asg	24
SpotFleetOnDemandBase	aws:ec2:instances	4
SpotFleetOnDemandAboveBasePercentage	aws:ec2:instances	50

En este ejemplo, el entorno comienza con diez instancias, de las cuales siete son en diferido (cuatro base y el 50 % de las seis que están por encima de la base), mientras que tres son de spot. El entorno puede escalar horizontalmente hasta un máximo de 24 instancias. A medida que se amplía, la parte en diferido de la flota que está por encima de las cuatro instancias base en diferido se mantiene en el 50 %, hasta un máximo total de 24 instancias, de las cuales 14 son en diferido (cuatro base y el 50 % de las 20 que están por encima de la base) y 10 son de spot.

Example 2: Toda la capacidad inicial la conforman instancias en diferido

Opciones de configuración

Opción	Namespace	Valor
MinSize	aws:autoscaling:asg	4
MaxSize	aws:autoscaling:asg	24
SpotFleetOnDemandBase	aws:ec2:instances	4

Opción	Namespace	Valor
SpotFleetOnDemandAboveBasePercentage	aws:ec2:instances	50

En este ejemplo, el entorno comienza con cuatro instancias, todas ellas en diferido. El entorno puede ampliarse hasta un máximo de 24 instancias. A medida que se amplía, la parte en diferido de la flota que está por encima de las cuatro instancias base en diferido se mantiene en el 50 %, hasta un máximo total de 24 instancias, de las cuales 14 son en diferido (cuatro base y el 50 % de las 20 que están por encima de la base) y 10 son de spot.

Example 3: Base en diferido adicional que supera la capacidad inicial

Opciones de configuración

Opción	Namespace	Valor
MinSize	aws:autoscaling:asg	3
MaxSize	aws:autoscaling:asg	24
SpotFleetOnDemandBase	aws:ec2:instances	4
SpotFleetOnDemandAboveBasePercentage	aws:ec2:instances	50

En este ejemplo, el entorno comienza con tres instancias, todas ellas en diferido. El entorno puede escalar horizontalmente hasta un máximo de 24 instancias. La primera instancia adicional por encima de las tres iniciales es una instancia en diferido y completa las cuatro instancias base en diferido. A medida que sigue ampliándose, la parte en diferido de la flota que está por encima de las cuatro instancias base en diferido se mantiene en el 50 %, hasta un máximo total de 24 instancias, de las cuales 14 son en diferido (cuatro base y el 50 % de las 20 que están por encima de la base), mientras que diez son instancias de spot.

Configuración del grupo de Auto Scaling mediante la consola de Elastic Beanstalk

Puede configurar el funcionamiento de Auto Scaling modificando el valor de Capacity (Capacidad) en la página Configuration (Configuración) del entorno en la [consola de Elastic Beanstalk](#).

Para configurar el grupo de Auto Scaling en la consola de Elastic Beanstalk

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Capacity (Capacidad), elija Edit (Editar).
5. En la sección Auto Scaling group (Grupo de Auto Scaling), configure los siguientes ajustes.
 - Environment type (Tipo de entorno): seleccione Load balanced (Con balanceo de carga).
 - Min instances (Número mínimo de instancias): el número mínimo de instancias EC2 que el grupo debe incluir en cualquier momento. El grupo comienza con el recuento mínimo y agrega instancias cuando se cumple la condición del desencadenador de aumento.
 - Max instances (Número máximo de instancias): el número máximo de instancias EC2 que el grupo debe incluir en cualquier momento.


Note

Si utiliza las actualizaciones continuas, asegúrese de que el número máximo de instancias es mayor que el ajuste [Minimum instances in service \(Instancias mínimas en servicio\)](#) para las actualizaciones continuas.

- Composición de flota: el valor predeterminado es instancias en diferido. Para habilitar las solicitudes de instancias de spot, seleccione Combined purchase options and instances (Opciones e instancias de compra combinadas).

Las siguientes opciones están habilitadas si selecciona habilitar solicitudes de instancia de Spot:

- Precio máximo al contado: para obtener recomendaciones sobre las opciones de precio máximo para las instancias puntuales, consulte el [historial de precios de las instancias puntuales](#) en la Guía del usuario de Amazon EC2.
- Base en diferido: el número mínimo de instancias en diferido que el grupo de Auto Scaling aprovisiona antes de considerar la posibilidad de utilizar instancias de spot cuando escala de manera horizontal el entorno.
- En diferido por encima de la base: porcentaje de instancias en diferido que forman parte de la capacidad adicional que el grupo de Auto Scaling aprovisiona después de las instancias base en diferido.

 Note

Las opciones base bajo demanda y bajo demanda por encima de la base se correlacionan con las opciones de instancias Mín. y Máx. enumeradas anteriormente. Para obtener más información sobre estas opciones y ejemplos, consulte [the section called “Compatibilidad con instancias de spot”](#).

- Habilitar el reequilibrio de la capacidad: esta opción solo es relevante cuando hay al menos una instancia de spot en el grupo de Auto Scaling. Con esta característica habilitada, EC2 intenta reemplazar de manera automática las instancias de spot en el grupo de Auto Scaling antes de que se interrumpan y así minimizar las interrupciones de las instancias de spot en las aplicaciones. Para obtener más información, consulte [Reequilibrio de la capacidad](#) en la Guía del usuario de Amazon EC2 Auto Scaling.
- Instance Type (Tipo de instancia): tipo de Amazon EC2 instance lanzada para ejecutar la aplicación. Para obtener más detalles, consulte [the section called “Tipos de instancias”](#).
- AMI ID (ID de AMI): imagen de máquina que Elastic Beanstalk utiliza para lanzar instancias de Amazon EC2 en el entorno. Para obtener más detalles, consulte [the section called “ID de AMI”](#).
- Availability Zones (Zonas de disponibilidad): elija el número de zonas de disponibilidad para distribuir las instancias en su entorno. De forma predeterminada, el grupo de Auto Scaling lanza instancias de manera uniforme en todas las zonas utilizables. Para concentrar sus instancias en menos zonas, elija el número de zonas que se debe usar. Para los entornos de

producción, utilice al menos dos zonas para asegurarse de que su aplicación esté disponible si una zona de disponibilidad se queda fuera de servicio.

- Placement (Ubicación) (opcional): elija las zonas de disponibilidad que va a utilizar. Utilice este ajuste si las instancias deben conectarse a recursos en zonas específicas o si ha comprado [instancias reservadas](#), que son específicas de cada zona. Si lanza su entorno en una VPC personalizada, no puede configurar esta opción. En una VPC personalizada, usted elige zonas de disponibilidad para las subredes que asigna a su entorno.
- Scaling cooldown (Periodo de recuperación de escalado): la cantidad de tiempo que debe esperar en segundos a que las instancias se lancen o terminen después del escalado, antes de seguir evaluando disparadores. Para obtener más información, consulte [Periodos de recuperación de escalado](#).

Elastic Beanstalk > Environments > Gettingstarted-env > Configuration

Modify capacity

Configure the compute capacity of your environment and Auto Scaling settings to optimize the number of instances used.

Auto Scaling Group

Environment type
Load balanced

Instances
Min 1
Max 4

Fleet composition
Choose a mix of On-Demand and Spot Instances with multiple instance types. Spot Instances are automatically launched at the lowest available price. [Learn more](#)

On-Demand instances

Combine purchase options and instances

Maximum spot price
The maximum price per instance-hour, in USD, that you're willing to pay for a Spot Instance. Setting a custom price limits your chances to fulfill your target capacity using Spot instances.

Default - the On-Demand price for each instance type (recommended)

Set your maximum price

On-Demand base
The minimum number of On-Demand Instances that your Auto Scaling group provisions before considering Spot Instances as your environment scales out.
0

On-Demand above base
The percentage of On-Demand Instances as part of any additional capacity that your Auto Scaling group provisions beyond the On-Demand base instances.
70 %

Enable Capacity Rebalancing
Specifies whether to enable the Capacity Rebalancing feature for Spot Instances in your Auto Scaling Group. This option is only relevant when EnableSpot is true in the aws:ec2:instances namespace, and there is at least one Spot Instance in your Auto Scaling group.

Enabled

Instance types
Add acceptable instance types for your fleet. Change their order to set the launch priority of On-Demand Instances. This order doesn't affect Spot Instances. We recommend a minimum of two instance types. [Learn more](#)

-- Choose Instance Types --

t2.micro (1vCPUs, 1GiB) × t2.small (1vCPUs, 2GiB) ×

AMI ID
ami-9999999999999999

Availability Zones
Number of Availability Zones (AZs) to use.
Any

Placement
Specify Availability Zones (AZs) to use.
-- Choose Availability Zones (AZs) --

Scaling cooldown
360 seconds

6. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

Configuración del grupo de Auto Scaling a través de la CLI de EB

Si crea un entorno mediante el comando [eb create](#), puede especificar algunas opciones relacionadas con el grupo de Auto Scaling del entorno. Estas son algunas opciones que lo ayudan a controlar la capacidad del entorno.

--single

Crea el entorno con una Amazon EC2 instance y sin balanceador de carga. Si no utiliza esta opción, un balanceador de carga se agrega al entorno que se ha creado.

--enable-spot

Habilita las solicitudes de instancias de spot para el entorno.

Las siguientes opciones para el comando [eb create](#) solo pueden utilizarse con --enable-spot.

--instance-types

Enumera los tipos de Amazon EC2 instance que desea que se utilice en el entorno.

--spot-max-price

El precio máximo unidad por hora, en dólares estadounidenses, que está dispuesto a pagar por una instancia de Spot. Para obtener recomendaciones sobre las opciones de precio máximo para las instancias puntuales, consulte el [historial de precios de las instancias puntuales](#) en la Guía del usuario de Amazon EC2.

--on-demand-base-capacity

Número mínimo de instancias en diferido que el grupo de Auto Scaling aprovisiona antes de considerar la posibilidad de utilizar instancias de spot cuando se amplía el entorno.

--on-demand-above-base-capacity

El porcentaje de instancias en diferido como parte de la capacidad adicional que el grupo de Auto Scaling aprovisiona además del número de instancias especificado mediante la opción --on-demand-base-capacity.

En el siguiente ejemplo, se crea un entorno y se configura el grupo de Auto Scaling para habilitar las solicitudes de instancias de Spot en el nuevo entorno. Para este ejemplo, se pueden utilizar tres tipos de instancias posibles.

```
$ eb create --enable-spot --instance-types "t2.micro,t3.micro,t3.small"
```

⚠ Important

Hay otra opción con nombre similar que se llama `--instance-type` (sin “s”) que la CLI de EB solo reconoce al procesar instancias en diferido. No utilice `--instance-type` (sin “s”) con la opción `--enable-spot`. Si lo hace, la CLI de EB lo ignora. En su lugar utilice `--instance-types` (con “s”) con la opción `--enable-spot`.

Opciones de configuración

Elastic Beanstalk dispone de [opciones de configuración](#) para configuraciones de Auto Scaling en dos espacios de nombres: [aws:autoscaling:asg](#) y [aws:ec2:instances](#).

El espacio de nombres aws:autoscaling:asg

El espacio de nombres [aws:autoscaling:asg](#) cuenta con dos opciones de disponibilidad y escala global.

En el siguiente [archivo de configuración](#) de ejemplo, el grupo de Auto Scaling se configura para que utilice entre dos y cuatro instancias, zonas de disponibilidad específicas y un periodo de recuperación de 12 minutos (720 segundos). Se habilita el reequilibrio de la capacidad para instancias de spot. Esta última opción solo tiene efecto si `EnableSpot` se configura `true` en el espacio de nombres [aws:ec2:instances](#), como se muestra en el ejemplo del archivo de configuración siguiente a este.

```
option_settings:
  aws:autoscaling:asg:
    Availability Zones: Any
    Cooldown: '720'
    Custom Availability Zones: 'us-west-2a,us-west-2b'
    MaxSize: '4'
    MinSize: '2'
    EnableCapacityRebalancing: true
```

El espacio de nombres aws:ec2:instances

El espacio de nombres [aws:ec2:instances](#) dispone de opciones relacionadas con las instancias del entorno, incluida la administración de instancias de spot. Complementa a [aws:autoscaling:launchconfiguration](#) y [aws:autoscaling:asg](#).

Cuando actualiza la configuración del entorno y elimina uno o más tipos de instancias de la opción `InstanceTypes`, Elastic Beanstalk finaliza cualquier Amazon EC2 instance que se esté ejecutando en cualquiera de los tipos de instancias eliminados. A continuación, el grupo de Auto Scaling de su entorno lanza nuevas instancias, según sea necesario, para completar la capacidad deseada mediante sus tipos de instancia especificados actuales.

El siguiente ejemplo de [archivo de configuración](#), configura el grupo de Auto Scaling para habilitar las solicitudes de instancias de spot en su entorno. Se pueden utilizar tres tipos de instancias posibles. Al menos una instancia en diferido se utiliza para capacidad de línea base y un 33% sostenido de instancias en diferido se utiliza para cualquier capacidad adicional.

```
option_settings:
  aws:ec2:instances:
    EnableSpot: true
    InstanceTypes: 't2.micro,t3.micro,t3.small'
    SpotFleetOnDemandBase: '1'
    SpotFleetOnDemandAboveBasePercentage: '33'
```

Para seleccionar tipos de instancias de spot, utilice el [Asistente de instancias de spot](#).

Desencadenadores de Auto Scaling

El grupo de Auto Scaling en su entorno de Elastic Beanstalk utiliza dos alarmas de Amazon CloudWatch para activar operaciones de escala. Los desencadenadores predeterminados adaptan su capacidad cuando el tráfico de la red saliente promedio de cada instancia es superior a 6 MB o inferior a 2 MB durante un periodo de cinco minutos. Para utilizar Amazon EC2 Auto Scaling de forma eficaz, configure desencadenadores adecuados para su aplicación, tipo de instancia y requisitos de servicio. Puede optar por el escala en función de varias estadísticas, como la latencia, E/S de disco, la utilización de la CPU y el recuento de solicitudes.

Para obtener más información sobre las métricas y alarmas de CloudWatch, consulte [Conceptos de Amazon CloudWatch](#) en la Guía del usuario de Amazon CloudWatch.

Configuración de desencadenadores de Auto Scaling

Puede configurar los desencadenadores que ajustan el número de instancias en el grupo de Auto Scaling de su entorno en la consola de Elastic Beanstalk.

Para configurar desencadenadores en la consola de Elastic Beanstalk

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Capacity (Capacidad), elija Edit (Editar).
5. En la sección Scaling triggers (Desencadenadores de escalado), configure los siguientes ajustes:
 - Metric (Métrica): métrica utilizada para el disparador de Auto Scaling.
 - Statistic (Estadística): cálculo de estadística que debe usar el disparador como, por ejemplo, Average.
 - Unit (Unidad): unidad para la métrica de disparador como, por ejemplo, Bytes.
 - Period (Periodo): especifica la frecuencia con la que Amazon CloudWatch calcula las métricas del disparador.
 - Breach duration (Duración de la interrupción): cantidad de tiempo, en minutos, durante el cual una métrica puede superar los umbrales superior e inferior antes de activar una operación de escalado.
 - Upper threshold (Umbral superior): si el valor de la métrica se sitúa por encima de este número durante la interrupción, se activa una operación de escalado.
 - Scale up increment (Incremento de escalado ascendente): número de instancias de Amazon EC2 que deben añadirse al realizar una actividad de escalado.
 - Lower threshold (Umbral inferior): si el valor de la métrica se sitúa por debajo de este número durante la interrupción, se activa una operación de escalado.
 - Scale down increment (Incremento de escalado descendente): número de instancias de Amazon EC2 que deben eliminarse al realizar una actividad de escalado.

Scaling triggers

Metric
Change the metric that is monitored to determine if the environment's capacity is too low or too high.

NetworkOut ▼

Statistic
Choose how the metric is interpreted.

Average ▼

Unit

Bytes ▼

Period
The period between metric evaluations.

5 Min

Breach duration
The amount of time a metric can exceed a threshold before triggering a scaling operation.

5 Min

Upper threshold

6000000 Bytes

Scale up increment

1 EC2 instances

Lower threshold

2000000 Bytes

Scale down increment

-1 EC2 instances

6. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

El espacio de nombres `aws:autoscaling:trigger`

Elastic Beanstalk dispone de [opciones de configuración](#) para Auto Scaling en el espacio de nombres [aws:autoscaling:trigger](#). La configuración en este espacio de nombres se organiza por el recurso al que se aplica.

```
option_settings:  
  AWSEBAutoScalingScaleDownPolicy.aws:autoscaling:trigger:
```



```
LowerBreachScaleIncrement: '-1'  
AWSEBAutoScalingScaleUpPolicy.aws:autoscaling:trigger:  
  UpperBreachScaleIncrement: '1'  
AWSEBCloudwatchAlarmHigh.aws:autoscaling:trigger:  
  UpperThreshold: '6000000'  
AWSEBCloudwatchAlarmLow.aws:autoscaling:trigger:  
  BreachDuration: '5'  
  EvaluationPeriods: '1'  
  LowerThreshold: '2000000'  
  MeasureName: NetworkOut  
  Period: '5'  
  Statistic: Average  
  Unit: Bytes
```

Acciones programadas de Auto Scaling

Para optimizar el uso de instancias de Amazon EC2 en el entorno mediante periodos predecibles de picos de tráfico, configure el grupo de Amazon EC2 Auto Scaling para cambiar el recuento de instancias de manera programada. Puede configurar su entorno con una acción recurrente para escalar verticalmente cada día por la mañana y reducirlo por la noche cuando hay poco tráfico. Por ejemplo, si tiene un evento de marketing que incrementará el tráfico hacia su sitio durante un periodo de tiempo limitado, puede programar un evento para escalar verticalmente cuando comienza y otro para reducir verticalmente cuando finaliza.

Puede definir hasta 120 acciones programadas activas por cada entorno. Elastic Beanstalk también retiene hasta 150 acciones programadas caducadas, que puede reutilizar actualizando su configuración.

Configuración de acciones programadas

Puede crear acciones programadas para el grupo de Auto Scaling de su entorno en la consola de Elastic Beanstalk.

Para configurar acciones programadas en la consola de Elastic Beanstalk

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Capacity (Capacidad), elija Edit (Editar).
5. En la sección Time-based Scaling (Escalado basado en el tiempo), elija Add scheduled action (Añadir acción programada).

Time-based Scaling

Use the following settings to control time-based scaling actions. [Learn more](#)

Current status
1 instance(s) in service, Min: 1, Max: 2

Time zone
 UTC
 Local

Actions ▾ **Add scheduled action**

<input type="checkbox"/>	Name	Min	Max	Desired	Next occurrence (UTC)
No scheduled actions					

6. Rellene los siguientes ajustes para la acción configuración:
 - Name (Nombre): especifique un nombre único con hasta 255 caracteres alfanuméricos sin espacios.
 - Instances (Instancias): elija el número mínimo y máximo de instancias para aplicar al grupo de Auto Scaling.
 - Desired capacity (Capacidad deseada) (opcional): establezca la capacidad deseada inicial para el grupo de Auto Scaling. Después de aplicar la acción programada, los desencadenadores ajustan la capacidad deseada en función de su configuración.
 - Occurrence (Incidencia): elija Recurring (Periódica) para repetir la acción de escalado de manera programada.
 - Start time (Hora de inicio): en el caso de acciones puntuales, elija la fecha y la hora para ejecutar la acción.

En el caso de acciones recurrentes, una hora de inicio es opcional. Especifíquelo para elegir la primera vez que se realiza la acción. Después de este tiempo, la acción se repite según la expresión Recurrence (Periodicidad).

- Recurrence (Periodicidad): use una expresión [CRON](#) para especificar la frecuencia con la que desea que se produzca la acción programada. Por ejemplo, `30 6 * * 2` ejecuta la acción cada martes a las 06:30 UTC.
- End time (Hora de finalización) (opcional): opcional para acciones periódicas. Si se especifica, la acción se repite según la expresión Recurrence (Periodicidad) y no se vuelve a realizar después de este tiempo.

Cuando finaliza una acción programada, Auto Scaling no recupera automáticamente su configuración anterior. Configure una segunda acción programada para que Auto Scaling recupere la configuración original según sea necesario.

7. Elija Add (Añadir).
8. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

Note

Las acciones programadas no se guardarán hasta que se apliquen.

El espacio de nombres `aws:autoscaling:scheduledaction`

Si necesita configurar un gran número de acciones programadas, puede utilizar [archivos de configuración](#) o [la API de Elastic Beanstalk](#) para aplicar los cambios de la opción de configuración de un archivo YAML o JSON. Estos métodos también le permiten acceder a la [opción Suspend](#) para desactivar temporalmente una acción programada recurrente.

Note

Al trabajar con las opciones de configuración de acción programada fuera de la consola, utilice el formato de hora de la norma ISO 8601 para especificar la hora de inicio y de finalización en UTC. Por ejemplo: `2015-04-28T04:07:02Z`. Para obtener más información sobre el formato de hora ISO 8601, visite [Date and Time Formats \(Formatos de fecha y hora\)](#). Las fechas deben ser únicas en todas las acciones programadas.

Elastic Beanstalk dispone de opciones de configuración para la configuración de acciones programadas en el espacio de nombres [aws:autoscaling:scheduledaction](#). Utilice el campo `resource_name` para especificar el nombre de la acción programada.

Example Scheduled-scale-up-specific-time-long.config

Este archivo de configuración indica a Elastic Beanstalk que escale horizontalmente de cinco a 10 instancias en 2015-12-12T00:00:00Z.

```
option_settings:
  - namespace: aws:autoscaling:scheduledaction
    resource_name: ScheduledScaleUpSpecificTime
    option_name: MinSize
    value: '5'
  - namespace: aws:autoscaling:scheduledaction
    resource_name: ScheduledScaleUpSpecificTime
    option_name: MaxSize
    value: '10'
  - namespace: aws:autoscaling:scheduledaction
    resource_name: ScheduledScaleUpSpecificTime
    option_name: DesiredCapacity
    value: '5'
  - namespace: aws:autoscaling:scheduledaction
    resource_name: ScheduledScaleUpSpecificTime
    option_name: StartTime
    value: '2015-12-12T00:00:00Z'
```

Example Scheduled-scale-up-specific-time.config

Para utilizar la sintaxis abreviada con la CLI de EB o con los archivos de configuración, anteponga el nombre del recurso al espacio de nombres.

```
option_settings:
  ScheduledScaleUpSpecificTime.aws:autoscaling:scheduledaction:
    MinSize: '5'
    MaxSize: '10'
    DesiredCapacity: '5'
    StartTime: '2015-12-12T00:00:00Z'
```

Example Scheduled-scale-down-specific-time.config

Este archivo de configuración indica a Elastic Beanstalk que reduzca el escala en 2015-12-12T07:00:00Z.

```
option_settings:
  ScheduledScaleDownSpecificTime.aws:autoscaling:scheduledaction:
    MinSize: '1'
    MaxSize: '1'
    DesiredCapacity: '1'
    StartTime: '2015-12-12T07:00:00Z'
```

Example Scheduled-periodic-scale-up.config

Este archivo de configuración indica a Elastic Beanstalk que aumente el escala cada día a las 9:00. La acción está programada para que empiece el 14 de mayo de 2015 y termine el 12 de enero de 2016.

```
option_settings:
  ScheduledPeriodicScaleUp.aws:autoscaling:scheduledaction:
    MinSize: '5'
    MaxSize: '10'
    DesiredCapacity: '5'
    StartTime: '2015-05-14T07:00:00Z'
    EndTime: '2016-01-12T07:00:00Z'
    Recurrence: 0 9 * * *
```

Example Scheduled-periodic-scale-down.config

Este archivo de configuración indica a Elastic Beanstalk que reduzca el escala cada viernes a las 18:00 para que no haya ninguna instancia en ejecución. Si sabe que la aplicación permanece inactiva la mayor parte del tiempo fuera del horario laboral normal, puede crear una acción programada similar. Si la aplicación debe permanecer desactivada fuera del horario laboral, cambie `MaxSize` a `0`.

```
option_settings:
  ScheduledPeriodicScaleDown.aws:autoscaling:scheduledaction:
    MinSize: '0'
    MaxSize: '1'
    DesiredCapacity: '0'
    StartTime: '2015-05-14T07:00:00Z'
```

```
EndTime: '2016-01-12T07:00:00Z'  
Recurrence: 0 18 * * *
```

Example Scheduled-weekend-scale-down.config

Este archivo de configuración indica a Elastic Beanstalk que reduzca el escala cada viernes a las 18:00. Si sabe que su aplicación no recibe tanto tráfico durante el fin de semana, puede crear una acción programada similar.

```
option_settings:  
  ScheduledWeekendScaleDown.aws:autoscaling:scheduledaction:  
    MinSize: '1'  
    MaxSize: '4'  
    DesiredCapacity: '1'  
    StartTime: '2015-12-12T07:00:00Z'  
    EndTime: '2016-01-12T07:00:00Z'  
    Recurrence: 0 18 * * 5
```

Configuración de comprobaciones de estado de Auto Scaling

Amazon EC2 Auto Scaling monitoriza el estado de cada instancia de Amazon Elastic Compute Cloud (Amazon EC2) que lanza. Si una instancia termina de forma inesperada, Auto Scaling lo detecta y lanza una instancia de sustitución. De forma predeterminada, el grupo de Auto Scaling creado para su entorno utiliza las [comprobaciones de estado de Amazon EC2](#). Si una instancia de su entorno no supera una comprobación de estado de Amazon EC2, Auto Scaling la desactiva y la sustituye.

Las comprobaciones de estado de Amazon EC2 solo se aplican al estado de la instancia, no al estado de la aplicación, del servidor o de los contenedores de Docker que se ejecuten en la instancia. Si la aplicación se bloquea, pero la instancia en la que se ejecuta sigue en buen estado, puede que la instancia se expulse del balanceador de carga, pero Auto Scaling no la sustituirá automáticamente. El comportamiento predeterminado es adecuado la solución de problemas. Si Auto Scaling sustituyera la instancia en cuanto se produjera un bloqueo en la aplicación, el problema podría pasar desapercibido, incluso aunque el bloqueo se produjera nada más arrancar la instancia.

Si desea que Auto Scaling sustituya las instancias cuya aplicación ha dejado de responder, puede utilizar un [archivo de configuración](#) para configurar el grupo de Auto Scaling para que utilice las comprobaciones de estado de Elastic Load Balancing. En el siguiente ejemplo se establece el grupo que deben usar las comprobaciones de estado del balanceador de carga, además de la comprobación de estado de Amazon EC2, para determinar el estado de una instancia.

Example .ebextensions/autoscaling.config

```
Resources:
  AWSEBAutoScalingGroup:
    Type: "AWS::AutoScaling::AutoScalingGroup"
    Properties:
      HealthCheckType: ELB
      HealthCheckGracePeriod: 300
```

Para obtener más información sobre las propiedades `HealthCheckType` y `HealthCheckGracePeriod`, consulte [AWS AutoScaling AutoScalingGroup](#) en la Guía del usuario de AWS CloudFormation y [Comprobaciones de estado para instancias de Auto Scaling](#) en la Guía del usuario de Amazon EC2 Auto Scaling.

De forma predeterminada, la comprobación de estado de Elastic Load Balancing está configurada para intentar establecer una conexión TCP con su instancia en el puerto 80. Esto confirma que el servidor web que se ejecuta en la instancia está aceptando conexiones. Sin embargo, es posible que desee [personalizar la comprobación de estado del balanceador de carga](#) para asegurarse de que la aplicación, y no solo el servidor web, se encuentra en buen estado. El valor de periodo de gracia define el número de segundos durante los que una instancia puede no superar la comprobación de estado sin que se termine y reemplace. Las instancias pueden recuperarse después de haber sido expulsadas del balanceador de carga, así que asigne a la instancia una cantidad de tiempo adecuada para su aplicación.

Balanceador de carga del entorno de Elastic Beanstalk

El balanceador de carga distribuye el tráfico entre las instancias del entorno. Cuando [habilita el balanceador de carga](#), AWS Elastic Beanstalk crea un balanceador de carga [Elastic Load Balancing](#) dedicado en su entorno. Elastic Beanstalk gestiona completamente este balanceador de carga, cuidando la configuración de seguridad y finalizando el balanceador de carga cuando finalice su entorno.

Como alternativa, puede elegir compartir un balanceador de carga en varios entornos de Elastic Beanstalk. Con un balanceador de carga compartido, ahorra en costos operativos al evitar un balanceador de carga dedicado para cada entorno. También asume más responsabilidad de administración para el balanceador de carga compartido que utilizan sus entornos.

Elastic Load Balancing tiene los siguientes tipos de balanceador de carga:

- [Balanceador de carga clásico](#): el balanceador de carga de la generación anterior. Dirige el tráfico HTTP, HTTPS o TCP a los distintos puertos de las instancias del entorno.
- [Balanceador de carga de aplicaciones](#): un balanceador de carga de capa de aplicación. Dirige el tráfico HTTP o HTTPS a los distintos puertos de las instancias del entorno en función de la ruta de la solicitud.
- [Balanceador de carga de red](#): un balanceador de carga de capa de red. Dirige el tráfico TCP a los distintos puertos de las instancias del entorno. Admite comprobaciones de estado tanto activas como pasivas.

Elastic Beanstalk admite los tres tipos de balanceadores de carga. En la tabla siguiente se muestran los tipos que puede utilizar con los dos patrones de uso:

Tipo de balanceador de carga	Dedicado	Compartido
Classic Load Balancer	✓ Sí	× No
Application Load Balancer	✓ Sí	✓ Sí
Network Load Balancer	✓ Sí	× No

Note

La opción Equilibrador de carga clásico (CLB) está deshabilitada en el asistente de la consola Crear entorno. Si ya tiene un entorno configurado con un Equilibrador de carga clásico, puede crear otro nuevo [clonando el entorno existente](#) mediante la consola de Elastic Beanstalk o la [CLI de EB](#). También tiene la opción de utilizar la [CLI de EB](#) o la [AWS CLI](#) para crear un nuevo entorno configurado con un Equilibrador de carga clásico. Estas herramientas de línea de comandos crearán un nuevo entorno con un CLB incluso si aún no existe ninguno en su cuenta.

De forma predeterminada, Elastic Beanstalk crea un balanceador de carga de aplicaciones para su entorno cuando habilita el equilibrio de carga con la consola de Elastic Beanstalk o la CLI de EB. También lo configura para escuchar el tráfico HTTP en el puerto 80 y transmitirlo a las instancias en el mismo puerto. Solo es posible elegir el tipo de balanceador de carga de un entorno en el momento

de su creación. Más adelante, puede cambiar las opciones para administrar el comportamiento del balanceador de carga de un entorno en ejecución, pero no cambiar su tipo.

Note

El entorno debe estar en una VPC con subredes en al menos dos zonas de disponibilidad para crear un balanceador de carga de aplicaciones. Todas las cuentas nuevas de AWS contienen las VPC predeterminadas que cumplen este requisito.

Consulte los siguientes temas para obtener más información sobre cada tipo de balanceador de carga compatible con Elastic Beanstalk, su funcionalidad y cómo configurarlo y administrarlo en un entorno de Elastic Beanstalk, y cómo configurar un balanceador de carga para [cargar registros de acceso](#) a Amazon S3.

Temas

- [Configuración de un Classic Load Balancer](#)
- [Configuración de un balanceador de tipo Application Load Balancer](#)
- [Configuración de un balanceador de carga de aplicaciones compartido](#)
- [Configuración de un balanceador de carga de red](#)
- [Configuración de registros de acceso](#)

Configuración de un Classic Load Balancer

Si ha habilitado el [balanceador de carga](#), el entorno de AWS Elastic Beanstalk contará con un balanceador de carga Elastic Load Balancing para distribuir el tráfico entre las instancias en el entorno. Elastic Load Balancing admite varios tipos de balanceadores de carga. Para obtener más información sobre ellos, consulte la [Guía de usuario de Elastic Load Balancing](#). Elastic Beanstalk puede crear un balanceador para usted o permitirle especificar un balanceador de carga compartido que haya creado.

En este tema se describe la configuración de un [balanceador de carga clásico](#) que Elastic Beanstalk crea y dedica a su entorno. Para obtener más información sobre cómo configurar todos los tipos de balanceadores de carga compatibles con Elastic Beanstalk, consulte [Balanceador de carga del entorno de Elastic Beanstalk](#).

Note

Solo es posible elegir el tipo de balanceador de carga de un entorno en el momento de su creación. Más adelante, puede cambiar las opciones para administrar el comportamiento del balanceador de carga de un entorno en ejecución, pero no cambiar su tipo.

Introducción

Un [balanceador de carga clásico](#) es el balanceador de carga de la generación anterior de Elastic Load Balancing. Permite el direccionamiento del tráfico HTTP, HTTPS o TCP a los distintos puertos de las instancias de entorno.

Cuando su entorno utiliza un balanceador de carga clásico, Elastic Beanstalk lo configura de forma predeterminada para [escuchar](#) el tráfico HTTP en el puerto 80 y reenviarlo a instancias del mismo puerto. Aunque no puede eliminar el listener predeterminado del puerto 80, puede deshabilitarlo, lo que logra la misma funcionalidad al bloquear el tráfico. Tenga en cuenta que puede añadir o eliminar otros oyentes. Para permitir conexiones seguras, puede configurar su balanceador de carga con un agente de escucha en el puerto 443 y un certificado TLS.

El balanceador de carga utiliza una [comprobación de estado](#) para determinar si las instancias Amazon EC2 que ejecutan una aplicación están en buen estado. La comprobación de estado realiza una solicitud a una URL especificada en los intervalos indicados. Si la URL devuelve un mensaje de error o no termina en el periodo de tiempo de espera especificado, la comprobación de estado no se ha superado.

Si su aplicación funciona mejor sirviendo varias solicitudes del mismo cliente en un único servidor, puede configurar el balanceador de carga para que utilice [sesiones sticky](#). Con las sesiones sticky, el balanceador de carga añade una cookie a las respuestas HTTP para identificar la instancia Amazon EC2 que atendió la solicitud. Cuando se recibe una solicitud posterior del mismo cliente, el balanceador de carga utiliza la cookie para enviar la solicitud a la misma instancia.

Con [cross-zone load balancing](#), cada nodo del balanceador de carga de su balanceador de carga clásico distribuye las solicitudes equitativamente entre todas las instancias registradas en todas las zonas de disponibilidad habilitadas. Si cross-zone load balancing está inhabilitado, cada nodo del balanceador de carga distribuye las solicitudes equitativamente entre todas las instancias registradas solo en su zona de disponibilidad.

Cuando se elimina una instancia del balanceador de carga porque ya no está en buen estado o porque se han reducido las instancias del entorno, el [Connection Draining \(Vaciado de conexiones\)](#) deja tiempo a la instancia para que complete las solicitudes antes de cerrar la conexión entre la instancia y el balanceador de carga. Puede cambiar la cantidad de tiempo asignada a las instancias para enviar una respuesta o deshabilitar completamente el vaciado de conexiones.

Note

El Connection draining se habilita de forma predeterminada cuando se crea un entorno con la consola de Elastic Beanstalk o la CLI de EB. Para el resto de los clientes, puede habilitarlo con [opciones de configuración](#).

Puede utilizar las opciones avanzadas del balanceador de carga para configurar agentes de escucha en puertos arbitrarios, modificar la configuración adicional de sesiones sticky y configurar el balanceador de carga para que se conecte a instancias EC2 de forma segura. Estas opciones están disponibles a través de las [opciones de configuración](#) que puede definir usando los archivos de configuración del código fuente, o directamente en un entorno a través de la API de Elastic Beanstalk. Muchos de estos ajustes también están disponibles en la consola de Elastic Beanstalk. Además, también puede configurar un balanceador de carga para [cargar registros de acceso](#) en Amazon S3.

Configuración de un balanceador de carga clásico mediante la consola de Elastic Beanstalk

Puede utilizar la consola de Elastic Beanstalk para configurar los puertos de un balanceador de carga clásico, el certificado HTTPS y otras opciones durante la creación del entorno o, posteriormente, con el entorno ya en ejecución.


Note

La opción Equilibrador de carga clásico (CLB) está deshabilitada en el asistente de la consola Crear entorno. Si ya tiene un entorno configurado con un Equilibrador de carga clásico, puede crear otro nuevo [clonando el entorno existente](#) mediante la consola de Elastic Beanstalk o la [CLI de EB](#). También tiene la opción de utilizar la [CLI de EB](#) o la [AWS CLI](#) para crear un nuevo entorno configurado con un Equilibrador de carga clásico. Estas

herramientas de línea de comandos crearán un nuevo entorno con un CLB incluso si aún no existe ninguno en su cuenta.


Para configurar el balanceador de carga clásico de un entorno en ejecución en la consola de Elastic Beanstalk

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

 Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Load balancer (Balanceador de carga), elija Edit (Editar).

 Note

Si la categoría de configuración Load balancer (Balanceador de carga) no tiene un botón Edit (Editar), el entorno no tiene un balanceador de carga. Para saber cómo configurar uno, consulte [Cambiar el tipo de entorno](#)

5. Realice los cambios de configuración del balanceador de carga clásico que requiera su entorno.
6. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

Configuración del balanceador de carga clásico

- [Oyentes](#)
- [Sesiones](#)
- [Equilibrio de carga entre zonas](#)
- [Vaciado de conexiones](#)
- [Comprobación de estado](#)

Oyentes

Utilice esta lista para especificar los agentes de escucha del balanceador de carga. Cada agente de escucha dirige a las instancias el tráfico entrante desde el cliente en un puerto y con un protocolo especificados. Inicialmente, la lista muestra el agente de escucha predeterminado, que dirige el tráfico HTTP entrante en el puerto 80 a los servidores de la instancia del entorno que están esperando tráfico HTTP en el puerto 80.

Note

Si bien no puede eliminar el agente de escucha predeterminado del puerto 80, puede deshabilitarlo, lo que logra la misma funcionalidad al bloquear el tráfico.

Classic Load Balancer

You can specify listeners for your load balancer. Each listener routes incoming client traffic on a specified port using a specified protocol to your instances. By default, we've configured your load balancer with a standard web server on port 80.

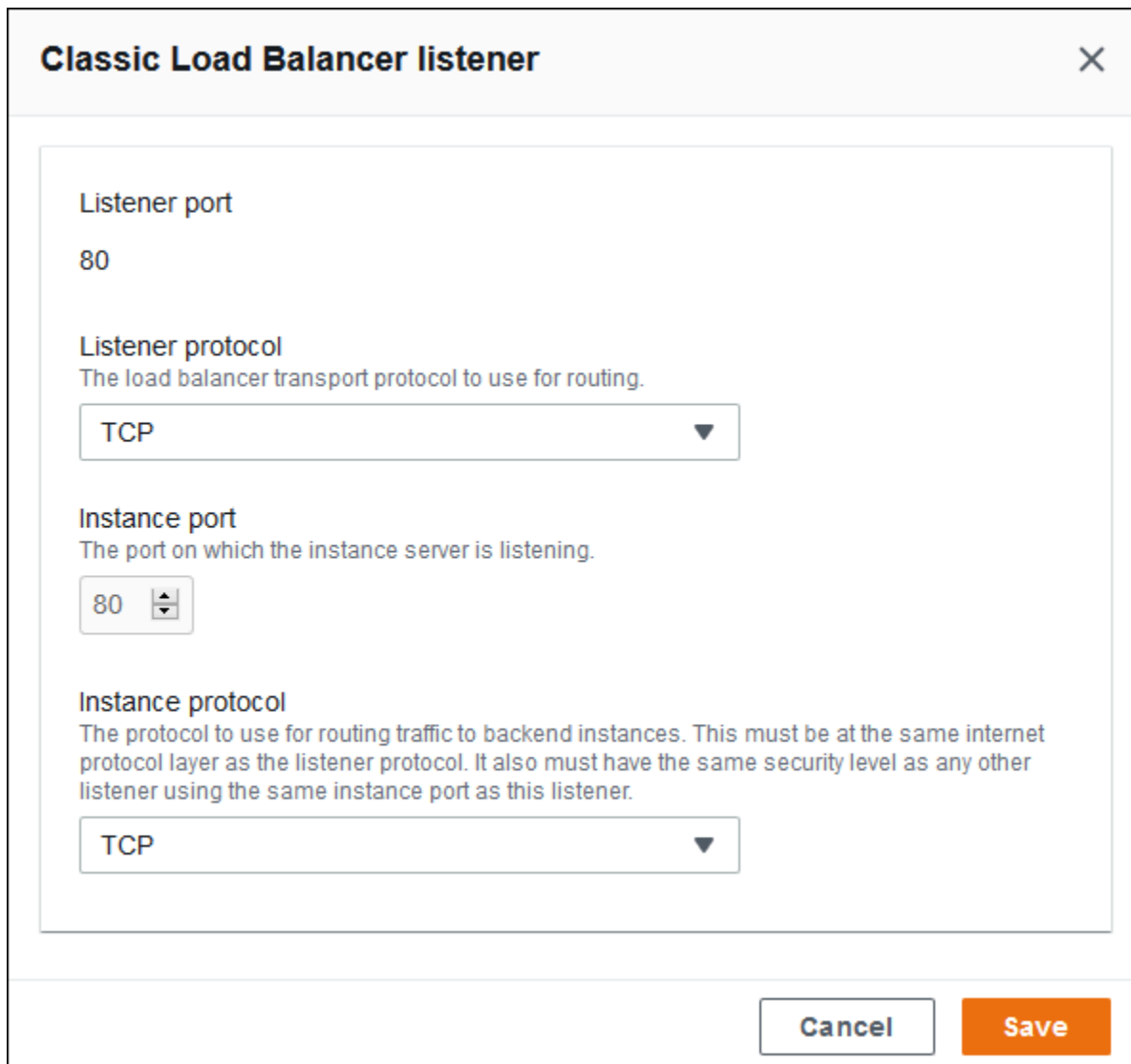
Actions ▾ Add listener

<input type="checkbox"/>	Port	Protocol	Instance port	Instance protocol	SSL certificate	Enabled
<input type="checkbox"/>	80	HTTP	80	HTTP	--	<input checked="" type="checkbox"/>

Para configurar un agente de escucha existente

1. Marque la casilla de verificación junto a la entrada correspondiente en la tabla, elija **Actions** (Acciones) y elija entonces la acción que desee.
2. Si ha elegido **Edit** (Editar), utilice el cuadro de diálogo **Classic Load Balancer listener** (Agente de escucha de Classic Load Balancer) para modificar las opciones y a continuación elija **Save** (Guardar).

Por ejemplo, puede editar el agente de escucha predeterminado y cambiar **Protocol** (Protocolo) de **HTTP** a **TCP** si desea que el balanceador de carga reenvíe las solicitudes sin modificación. Esto impide que el equilibrador de carga reescriba los encabezados (incluido **X-Forwarded-For**). La técnica no funciona con las sesiones sticky.



The image shows a dialog box titled "Classic Load Balancer listener" with a close button (X) in the top right corner. The dialog contains the following configuration options:

- Listener port:** A text input field containing the number "80".
- Listener protocol:** A dropdown menu with "TCP" selected. Below the dropdown is the text: "The load balancer transport protocol to use for routing."
- Instance port:** A spinner control with "80" selected. Below it is the text: "The port on which the instance server is listening."
- Instance protocol:** A dropdown menu with "TCP" selected. Below it is the text: "The protocol to use for routing traffic to backend instances. This must be at the same internet protocol layer as the listener protocol. It also must have the same security level as any other listener using the same instance port as this listener."

At the bottom right of the dialog, there are two buttons: "Cancel" (white with a grey border) and "Save" (orange).

Para agregar un agente de escucha

1. Elija Añadir oyente.
2. En el cuadro de diálogo Classic Load Balancer listener (Agente de escucha de Classic Load Balancer), configure las opciones que desee y, a continuación, elija Add (Añadir).

La adición de un agente de escucha seguro es un caso de uso común. En el ejemplo de la imagen siguiente se añade un agente de escucha para el tráfico HTTPS en el puerto 443. Este agente dirige el tráfico entrante a los servidores de instancia del entorno que esperan tráfico HTTPS en el puerto 443.

Para poder configurar un agente de escucha HTTPS, antes debe asegurarse de tener un certificado SSL válido. Realice una de las acciones siguientes:

- Si AWS Certificate Manager (ACM) está [disponible en su región de AWS](#), cree o importe un certificado con ACM. Para obtener más información sobre cómo solicitar un certificado de ACM, consulte [Solicitar un certificado](#) en la Guía del usuario de AWS Certificate Manager. Para obtener más información sobre la importación de certificados de terceros en ACM, consulte [Importar certificados](#) en la Guía del usuario de AWS Certificate Manager.
- Si ACM no está [disponible en su región de AWS](#), cargue el certificado y la clave existentes en IAM. Para obtener más información sobre la creación y la carga de certificados en IAM, consulte [Uso de certificados de servidor](#) en la Guía de usuario de IAM.

Para obtener más información sobre cómo configurar HTTPS y trabajar con certificados en Elastic Beanstalk, consulte [Configuración de HTTPS para su entorno de Elastic Beanstalk](#).

En SSL certificate (Certificado SSL), elija el ARN del certificado SSL. Por ejemplo,
`arn:aws:iam::123456789012:server-certificate/abc/certs/build` o
`arn:aws:acm:us-east-2:123456789012:certificate/12345678-12ab-34cd-56ef-12345678`.


Classic Load Balancer listener ✕

Listener port
443

Listener protocol
The load balancer transport protocol to use for routing.
HTTPS

Instance port
The port on which the instance server is listening.
443

Instance protocol
The protocol to use for routing traffic to backend instances. This must be at the same internet protocol layer as the listener protocol. It also must have the same security level as any other listener using the same instance port as this listener.
HTTPS

SSL certificate
arn:aws:acm:us-east-2:123456789012:certific... 

Cancel Add

Para obtener información sobre el modo de configurar HTTPS y trabajar con certificados en Elastic Beanstalk, consulte [Configuración de HTTPS para su entorno de Elastic Beanstalk](#).

Sesiones

Seleccione o desactive la casilla *Session stickiness enabled* (Sesión sticky habilitada) para habilitar o deshabilitar las sesiones sticky. Utilice *Cookie duration* (Duración de la cookie) para configurar una sesión sticky de duración, hasta **1000000** segundos. En la lista *Load balancer ports* (Puertos del balanceador de carga) seleccione los puertos de escucha a los que se aplica la política predeterminada (*AWSEB-ELB-StickinessPolicy*).

Sessions

The following settings let you control whether the load balancer routes requests for the same session to the Amazon EC2 instance with the smallest load, or consistently to the same instance.

Session stickiness enabled

Cookie duration
Lifetime of the sticky session cookie between an Amazon EC2 instance and the load balancer.

seconds

Load balancer ports
List of the listener ports that the default policy (AWSEB-ELB-StickinessPolicy) applies to.

80
443

Equilibrio de carga entre zonas

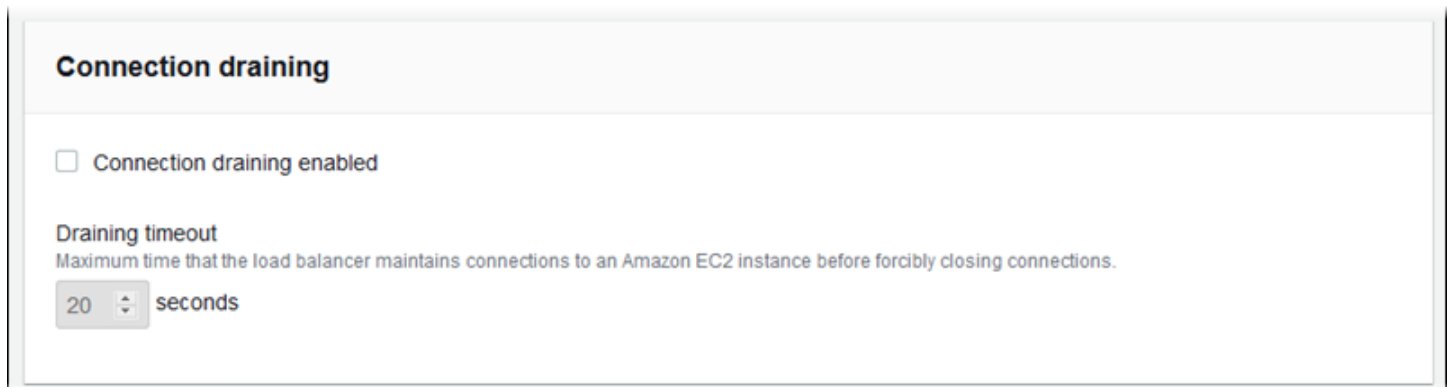
Seleccione o desactive la casilla Load balancing across multiple Availability Zones enabled (Balanceo de carga en varias zonas de disponibilidad habilitado) para habilitar o deshabilitar el balanceo de carga entre zonas.

Cross-zone load balancing

Load balancing across multiple Availability Zones enabled

Vaciado de conexiones

Seleccione o desactive la casilla Connection draining enabled (Vaciado de conexiones habilitado) para habilitar o deshabilitar el vaciado de conexiones. Establezca el Draining timeout (Tiempo de espera de vaciado), hasta **3600** segundos.



Connection draining

Connection draining enabled

Draining timeout
Maximum time that the load balancer maintains connections to an Amazon EC2 instance before forcibly closing connections.

20 seconds

Comprobación de estado

Utilice las opciones siguientes para configurar las comprobaciones de estado del balanceador de carga:

- **Health check path (Ruta de comprobación de estado):** Ruta a la que el balanceador de carga envía las solicitudes de comprobación de estado. Si no establece la ruta, el balanceador de carga intenta establecer una conexión TCP en el puerto 80 para comprobar el estado.
- **Timeout (Tiempo de espera):** Intervalo de tiempo en segundos que debe esperarse una respuesta de comprobación de estado.
- **Interval (Intervalo):** Intervalo de tiempo en segundos entre las comprobaciones de estado de una instancia individual. El intervalo debe ser mayor que el tiempo de espera.
- **Unhealthy threshold (Umbral incorrecto), Healthy threshold (Umbral correcto):** número de comprobaciones de estado que deben fallar o pasar, respectivamente, para que Elastic Load Balancing cambie el estado de una instancia.

Health check

Health check path
Path to which ELB sends an HTTP GET request to verify instance health.

Timeout
Amount of time to wait for a health check response.
5 seconds

Interval
Amount of time between health checks of an individual instance. The interval must be greater than the timeout.
10 seconds

Unhealthy threshold
The number of consecutive health check failures required to designate the instance as unhealthy.
5 requests

Healthy threshold
The number of consecutive successful health checks required to designate the instance as healthy.
3 requests

Note

La comprobación de estado de Elastic Load Balancing no afecta al comportamiento de comprobación de estado del grupo de Auto Scaling de un entorno. Las instancias que fallan en una comprobación de estado de Elastic Load Balancing no se reemplazan automáticamente por Amazon EC2 Auto Scaling, a menos que configure manualmente Amazon EC2 Auto Scaling para hacerlo. Para obtener más información, consulte [Configuración de comprobaciones de estado de Auto Scaling](#).

Para obtener más información sobre las comprobaciones de estado y la forma en que influyen en el estado general del entorno, consulte [Informes de estado básicos](#).

Configuración de un balanceador de carga clásico mediante la CLI de EB

La CLI de EB le pide que elija el tipo de balanceador de carga cuando se ejecuta [eb create](#).

```
$ eb create
Enter Environment Name
(default is my-app): test-env
Enter DNS CNAME prefix
(default is my-app): test-env-DLW24ED23SF

Select a load balancer type
1) classic
2) application
3) network
(default is 1):
```

Pulse Enter (Intro) para seleccionar `classic`.

También puede especificar un tipo de balanceador de carga mediante la opción `--elb-type`.

```
$ eb create test-env --elb-type classic
```

Espacios de nombres de la configuración del balanceador de carga clásico

Puede encontrar la configuración relacionada con balanceadores de carga clásicos en los siguientes espacios de nombres:

- [aws:elb:healthcheck](#): configure los umbrales, el intervalo de comprobación y el tiempo de espera de las comprobaciones de estado del balanceador de carga.
- [aws:elasticbeanstalk:application](#): configure la URL de comprobación de estado.
- [aws:elb:loadbalancer](#): habilite el balanceo de carga entre zonas. Asigne grupos de seguridad al balanceador de carga e invalide el grupo de seguridad predeterminado que crea Elastic Beanstalk. Este espacio de nombres incluye también opciones obsoletas para configurar los agentes de escucha estándar y seguro, que se han sustituido por opciones del espacio de nombres `aws:elb:listener`.
- [aws:elb:listener](#): configure el agente de escucha predeterminado en el puerto 80, un agente de escucha seguro en el puerto 443 o agentes de escucha adicionales para cualquier protocolo en cualquier puerto. Si especifica `aws:elb:listener` como el espacio de nombres, la configuración se aplica al agente de escucha predeterminado en el puerto 80. Si especifica un puerto (por ejemplo, `aws:elb:listener:443`), se configura un agente de escucha en dicho puerto.
- [aws:elb:policies](#): configure opciones adicionales para el balanceador de carga. Utilice las opciones de este espacio de nombres para configurar agentes de escucha en puertos arbitrarios,

modificar la configuración adicional de sesiones sticky y configurar el balanceador de carga para que se conecte a instancias de Amazon EC2 de forma segura.

La CLI de EB y la consola de Elastic Beanstalk aplican los valores recomendados a las opciones anteriores. Debe eliminar estos ajustes si desea usar archivos de configuración para configurarlos. Para obtener más información, consulte [Valores recomendados](#).

Example `.ebextensions/loadbalancer-terminatehttps.config`

El siguiente archivo de configuración de ejemplo crea un agente de escucha HTTPS en el puerto 443, asigna un certificado que el balanceador de carga utiliza para terminar la conexión segura y desactiva el agente de escucha predeterminado en el puerto 80. El balanceador de carga reenvía las solicitudes descifradas a las instancias EC2 de su entorno en el puerto HTTP 80.

```
option_settings:
  aws:elb:listener:443:
    ListenerProtocol: HTTPS
    SSLCertificateId: arn:aws:acm:us-east-2:123456789012:certificate/12345678-12ab-34cd-56ef-12345678
    InstancePort: 80
    InstanceProtocol: HTTP
  aws:elb:listener:
    ListenerEnabled: false
```

Configuración de un balanceador de tipo Application Load Balancer

Cuando [habilita el equilibrio de carga](#), su AWS Elastic Beanstalk entorno está equipado con un balanceador de carga de Elastic Load Balancing para distribuir el tráfico entre las instancias de su entorno. Elastic Load Balancing admite varios tipos de balanceadores de carga. Para obtener más información sobre ellos, consulte la [Guía de usuario de Elastic Load Balancing](#). Elastic Beanstalk puede crear un balanceador para usted o permitirle especificar un balanceador de carga compartido que haya creado.

En este tema se describe la configuración de un [balanceador de carga de aplicaciones](#) que Elastic Beanstalk crea y dedica a su entorno. Véase también [the section called “Balanceador de carga de aplicaciones compartido.”](#). Para obtener más información sobre cómo configurar todos los tipos de balanceadores de carga compatibles con Elastic Beanstalk, consulte [the section called “Balanceador de carga”](#).

Note

Solo es posible elegir el tipo de balanceador de carga de un entorno en el momento de su creación. Puede cambiar las opciones para administrar el comportamiento del balanceador de carga de un entorno en ejecución, pero no cambiar su tipo. Tampoco puede cambiar de un balanceador de carga dedicado a un balanceador de carga compartido o viceversa.

Introducción

Un balanceador de carga de aplicaciones inspecciona el tráfico en la capa de aplicación del protocolo de red para identificar la ruta de la solicitud, lo que le permite dirigir las solicitudes con rutas distintas a destinos diferentes.

Cuando su entorno utiliza un balanceador de carga de aplicaciones, Elastic Beanstalk lo configura de forma predeterminada para realizar la misma función que un balanceador de carga clásico. El agente de escucha predeterminado acepta las solicitudes HTTP en el puerto 80 y las distribuye a las instancias del entorno. Puede agregar un agente de escucha seguro en el puerto 443 con un certificado para descifrar el tráfico HTTPS, configurar el comportamiento de las comprobaciones de estado e insertar registros de acceso del balanceador de carga en un bucket de Amazon Simple Storage Service (Amazon S3).

Note

A diferencia de un balanceador de carga clásico o un balanceador de carga de red, un balanceador de carga de aplicaciones no puede tener agentes de escucha TCP o SSL/TLS de capa de transporte (capa 4). Solo admite agentes de escucha HTTP y HTTPS. Además, no puede utilizar la autenticación backend para autenticar las conexiones HTTPS entre el balanceador de carga y las instancias de backend.

En un entorno de Elastic Beanstalk, puede utilizar un balanceador de carga de aplicaciones para dirigir el tráfico de determinadas rutas a un proceso diferente en las instancias del servidor web. Con un balanceador de carga clásico, todo el tráfico que llega al agente de escucha se direcciona a un único puerto de las instancias del backend. Con un balanceador de carga de aplicaciones puede configurar varias reglas en el agente de escucha para enrutar las solicitudes con determinadas rutas a diferentes procesos de backend. Puede configurar cada proceso con el puerto en el que escucha el proceso.

Por ejemplo, podría ejecutar un proceso de inicio de sesión separándolo de la aplicación principal. Aunque la aplicación principal de las instancias del entorno aceptaría la mayoría de las solicitudes y las recibiría en el puerto 80, el proceso de inicio de sesión utiliza el puerto 5000 y acepta las solicitudes dirigidas a la ruta `/login`. Todas las solicitudes entrantes de los clientes llegan al puerto 80. Con un balanceador de carga de aplicaciones puede configurar un único agente de escucha para el tráfico entrante en el puerto 80 con dos reglas que dirijan el tráfico a dos procesos independientes, en función de la ruta de la solicitud. Puede agregar una regla personalizada que enruta tráfico a `/login` al proceso de inicio de sesión escuchando en el puerto 5000. La regla predeterminada dirige el resto del tráfico al proceso de la aplicación principal en el puerto 80.

Una regla del balanceador de carga de aplicaciones asigna una solicitud a un grupo de destino. En Elastic Beanstalk, un grupo objetivo está representado por un proceso. Puede configurar un proceso con las opciones de protocolo, puerto y comprobación de estado. Dicho proceso representa al proceso que se ejecuta en las instancias del entorno. El proceso predeterminado es un agente de escucha del puerto 80 del proxy inverso (nginx o Apache) que se ejecuta delante de la aplicación.

Note

Fuera de Elastic Beanstalk, un grupo objetivo se asigna a un grupo de instancias. Un agente de escucha puede utilizar reglas y grupos de destino para enrutar el tráfico a diferentes instancias en función de la ruta. Dentro de Elastic Beanstalk, todas las instancias del entorno son idénticas, por lo que la distinción se hace entre los procesos que utilizan diferentes puertos de escucha.

Un balanceador de carga clásico utiliza una única ruta de comprobación de estado para todo el entorno. Con un balanceador de carga de aplicaciones, cada proceso tiene una ruta de comprobación de estado independiente supervisada por el balanceador de carga y la supervisión del estado mejorada de Elastic Beanstalk.

Para utilizar un balanceador de carga de aplicaciones, el entorno debe estar en una VPC predeterminada o personalizada y debe tener un rol de servicio con el conjunto estándar de permisos. Si tiene un rol de servicio antiguo, es posible que tenga que [actualizar los permisos](#) del rol para incluir `elasticloadbalancing:DescribeTargetHealth` y `elasticloadbalancing:DescribeLoadBalancers`. Para obtener más información sobre los balanceadores de carga de aplicaciones, consulte [What is an Application Load Balancer? \(¿Qué es un Application Load Balancer?\)](#).

Note

La comprobación de estado del balanceador de carga de aplicaciones no utiliza la ruta de comprobación de estado de Elastic Beanstalk. En su lugar, usa la ruta específica configurada para cada proceso por separado.

Configuración de un balanceador de carga de aplicaciones mediante la consola de Elastic Beanstalk

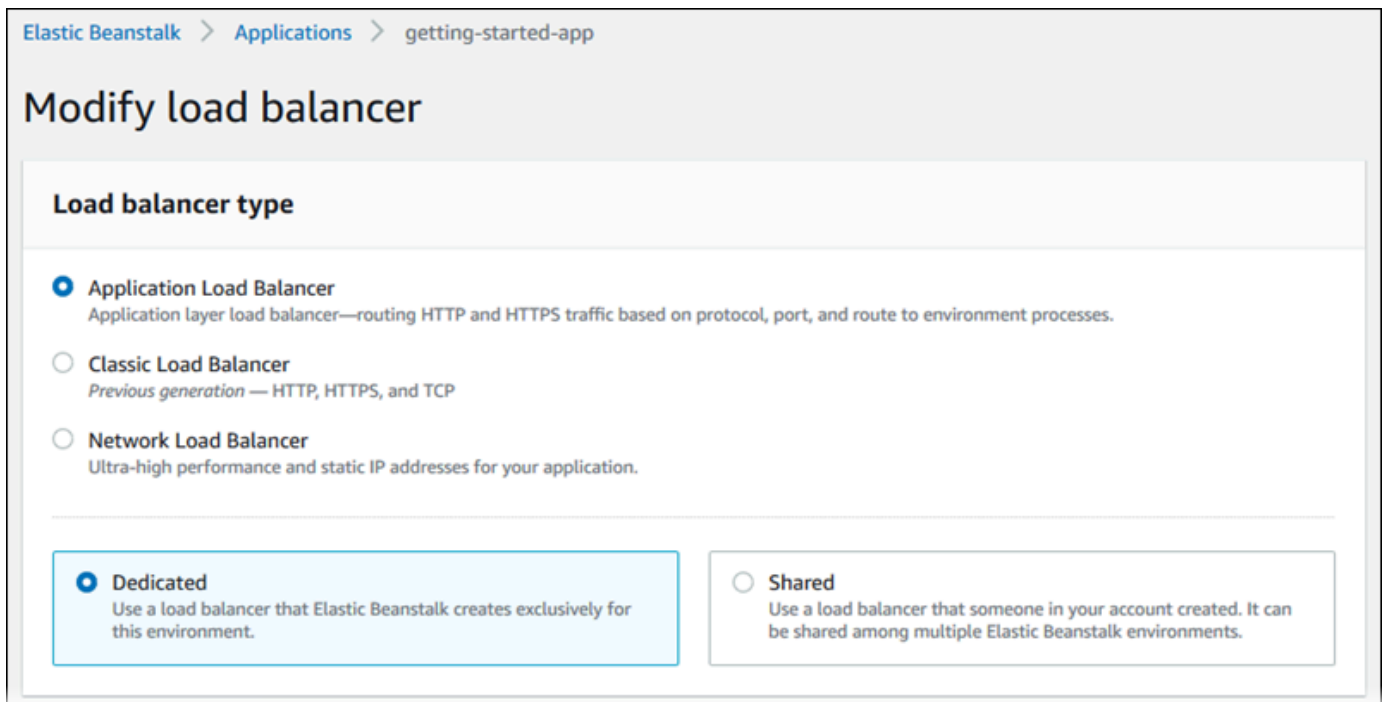
Puede utilizar la consola de Elastic Beanstalk para configurar los agentes de escucha, procesos y reglas de un balanceador de carga de aplicaciones durante la creación del entorno o posterior cuando se está ejecutando el entorno.

Para configurar un balanceador de carga de aplicaciones en la consola de Elastic Beanstalk durante la creación del entorno

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Entornos.
3. Elija [Create a new environment \(Crear un nuevo entorno\)](#) para empezar a crear el entorno.
4. En la página principal del asistente, antes de elegir Create environment (Crear entorno), elija Configure more options (Configurar más opciones).
5. Elija la configuración predefinida High availability (Alta disponibilidad).

Como alternativa, en la categoría de configuración Capacity (Capacidad), configure como tipo de entorno Load balanced (Balanceo de carga). Para obtener más detalles, consulte [Capacidad](#).

6. En la categoría de configuración Load balancer (Balanceador de carga), elija Edit (Editar).
7. Seleccione las opciones Application Load Balancer (Balanceador de carga de aplicaciones) y Dedicated (Dedicado) si aún no están seleccionadas.



8. Realice los cambios en la configuración del balanceador de carga de aplicaciones que requiera su entorno.
9. Elija Save (Guardar) y haga los demás cambios de configuración que requiera su entorno.
10. Seleccione Crear entorno.

Para configurar el balanceador de carga de aplicaciones de un entorno en ejecución en la consola de Elastic Beanstalk

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Load balancer (Balanceador de carga), elija Edit (Editar).

Note

Si la categoría de configuración Load balancer (Balanceador de carga) no tiene un botón Edit (Editar), el entorno no tiene un balanceador de carga. Para saber cómo configurar uno, consulte [Cambiar el tipo de entorno](#)

5. Realice los cambios en la configuración del balanceador de carga de aplicaciones que su entorno requiera.
6. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

Configuración de balanceadores de carga de aplicaciones

- [Oyentes](#)
- [Processes](#)
- [Reglas](#)
- [Captura de registro de acceso](#)

Oyentes

Utilice esta lista para especificar los agentes de escucha del balanceador de carga. Cada agente de escucha dirige a uno o más procesos de las instancias el tráfico entrante desde el cliente en un puerto y con un protocolo especificados. Inicialmente, la lista muestra el agente de escucha predeterminado, que enruta el tráfico HTTP entrante en el puerto 80 a un proceso llamado Predeterminado.

<input type="checkbox"/>	Port	Protocol	SSL certificate	Default process	Enabled
<input type="checkbox"/>	80	HTTP	--	default	<input checked="" type="checkbox"/>

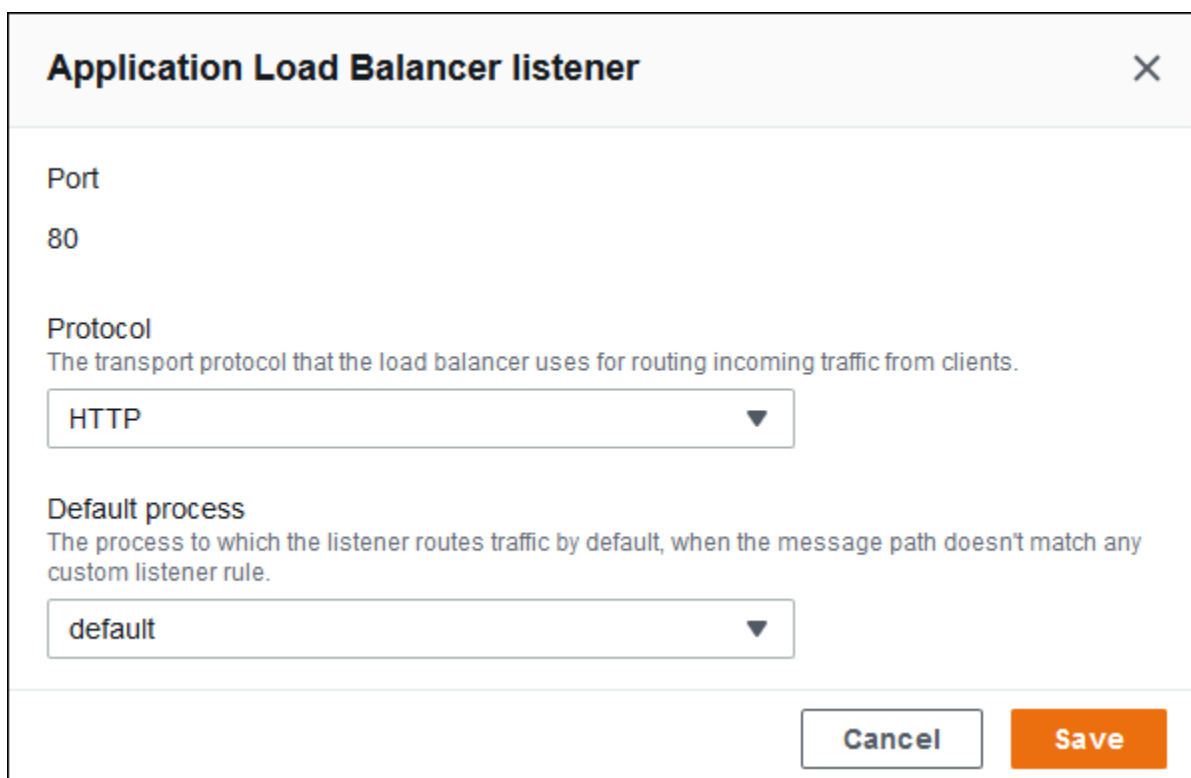
Para configurar un agente de escucha existente

1. Seleccione la casilla de verificación junto a la entrada correspondiente en la tabla y, a continuación, elija **Actions** (Acciones), **Edit** (Editar).
2. Utilice el cuadro de diálogo **Application Load Balancer listener** (Agente de escucha de Application Load Balancer) para modificar las opciones y a continuación elija **Save** (Guardar).

Para agregar un agente de escucha

1. Elija **Añadir oyente**.
2. En el cuadro de diálogo **Agente de escucha del balanceador de carga de la aplicación**, configure las opciones que desee y, a continuación, elija **Añadir**.

Utilice la configuración del cuadro de diálogo del **Agente de escucha del balanceador de carga de la aplicación** para elegir el puerto y el protocolo en los que el agente de escucha escucha el tráfico y el proceso al que se va a enrutar el tráfico. Si elige el protocolo HTTPS, configure las opciones de SSL.



Application Load Balancer listener [X]

Port
80

Protocol
The transport protocol that the load balancer uses for routing incoming traffic from clients.
HTTP

Default process
The process to which the listener routes traffic by default, when the message path doesn't match any custom listener rule.
default

Cancel Save

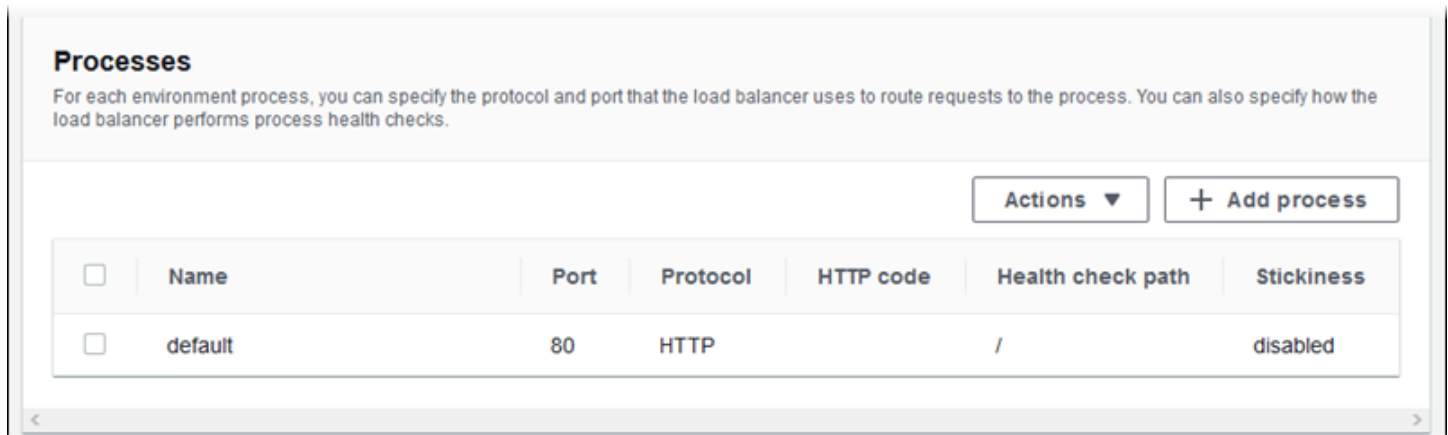
Para poder configurar un agente de escucha HTTPS, antes debe asegurarse de tener un certificado SSL válido. Realice una de las siguientes acciones siguientes:

- Si AWS Certificate Manager (ACM) está [disponible en su AWS región](#), cree o importe un certificado mediante ACM. Para obtener más información sobre cómo solicitar un certificado de ACM, consulte [Solicitar un certificado](#) en la Guía del usuario de AWS Certificate Manager . Para obtener más información sobre la importación de certificados de terceros en ACM, consulte [Importar certificados](#) en la Guía del usuario de AWS Certificate Manager .
- Si el ACM no está [disponible en su AWS región](#), cargue el certificado y la clave existentes en IAM. Para obtener más información sobre la creación y la carga de certificados en IAM, consulte [Uso de certificados de servidor](#) en la Guía de usuario de IAM.

Para obtener más información sobre cómo configurar HTTPS y trabajar con certificados en Elastic Beanstalk, consulte [Configuración de HTTPS para su entorno de Elastic Beanstalk](#).

Processes

Utilice esta lista para especificar los procesos del balanceador de carga. Un proceso es un destino al que los agentes de escucha dirigen el tráfico. Cada agente de escucha dirige a uno o más procesos de las instancias el tráfico entrante desde el cliente en un puerto y con un protocolo especificados. Inicialmente, la lista muestra el proceso predeterminado, que escucha el tráfico HTTP entrante en el puerto 80.



Processes
For each environment process, you can specify the protocol and port that the load balancer uses to route requests to the process. You can also specify how the load balancer performs process health checks.

<input type="checkbox"/>	Name	Port	Protocol	HTTP code	Health check path	Stickiness
<input type="checkbox"/>	default	80	HTTP		/	disabled

Actions ▾ + Add process

Puede editar la configuración de un proceso existente o agregar un nuevo proceso. Para empezar a editar un proceso de la lista o añadirle otro, utilice los pasos que se indican para la [lista de agentes de escucha](#). Se abre el cuadro de diálogo Environment process (Proceso del entorno).

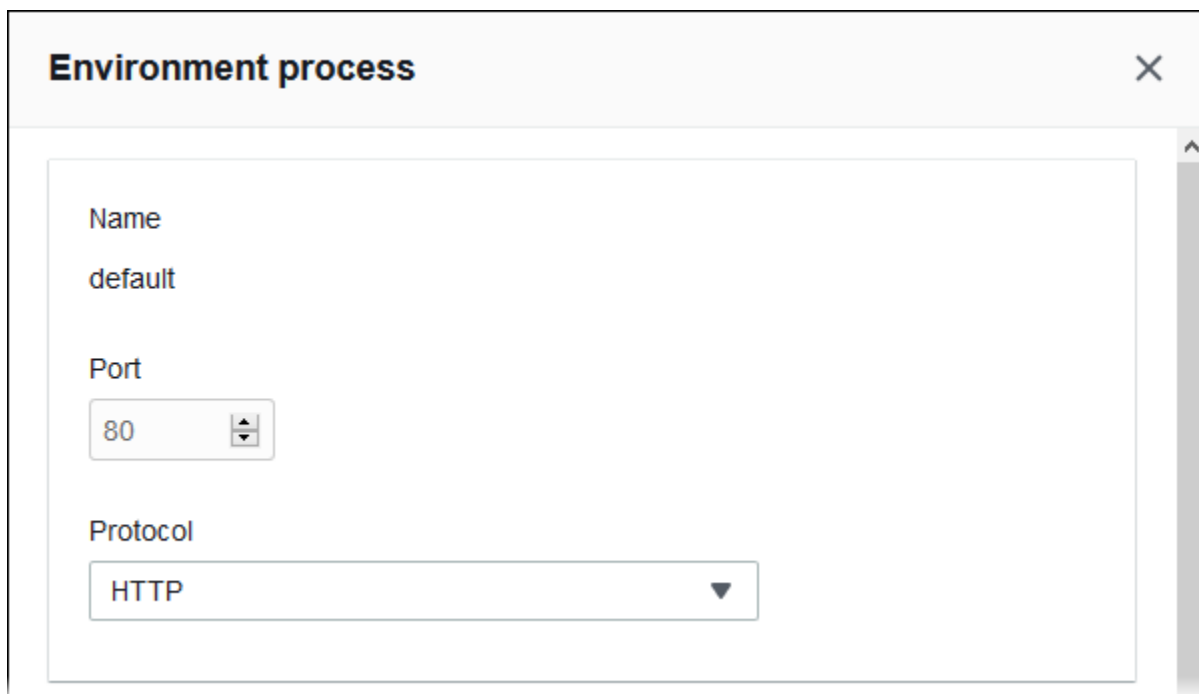
Configuración del cuadro de diálogo del proceso de entorno del balanceador de carga de aplicaciones

- [Definición](#)

- [Comprobación de estado](#)
- [Sesiones](#)

Definición

Utilice estas opciones para definir el proceso: Name (Nombre), Port (Puerto) y Protocol (Protocolo) en los que espera las solicitudes.



The screenshot shows a configuration window titled "Environment process". It contains three input fields:

- Name:** A text input field containing the value "default".
- Port:** A spinner box containing the value "80".
- Protocol:** A dropdown menu with "HTTP" selected.

Comprobación de estado

Utilice las opciones siguientes para configurar las comprobaciones de estado del proceso:

- **HTTP code (Código HTTP):** el código de estado HTTP que designa a un proceso en buen estado.
- **Path (Ruta) :** ruta de la solicitud de comprobación de estado del proceso.
- **Timeout (Tiempo de espera):** Intervalo de tiempo en segundos que debe esperarse una respuesta de comprobación de estado.
- **Interval (Intervalo):** Intervalo de tiempo en segundos entre las comprobaciones de estado de una instancia individual. El intervalo debe ser mayor que el tiempo de espera.
- **Unhealthy threshold (Umbral incorrecto), Healthy threshold (Umbral correcto):** número de comprobaciones de estado que deben fallar o pasar, respectivamente, para que Elastic Load Balancing cambie el estado de una instancia.

- **Deregistration delay (Retraso de anulación del registro):** intervalo de tiempo, en segundos, que debe esperarse a que se completen las solicitudes activas antes de anular el registro de una instancia.

Health check

HTTP code

HTTP status code of a healthy instance in your environment.

Path

Path to which the load balancer sends HTTP health check requests.

Timeout

Amount of time to wait for a health check response.

 seconds

Interval

Amount of time between health checks of an individual instance. The interval must be greater than the timeout.

 seconds

Unhealthy threshold

The number of consecutive health check failures required to designate the instance as unhealthy.

 requests

Healthy threshold

The number of consecutive successful health checks required to designate the instance as healthy.

 requests

Deregistration delay

Amount of time to wait for active requests to complete before deregistering.

 seconds

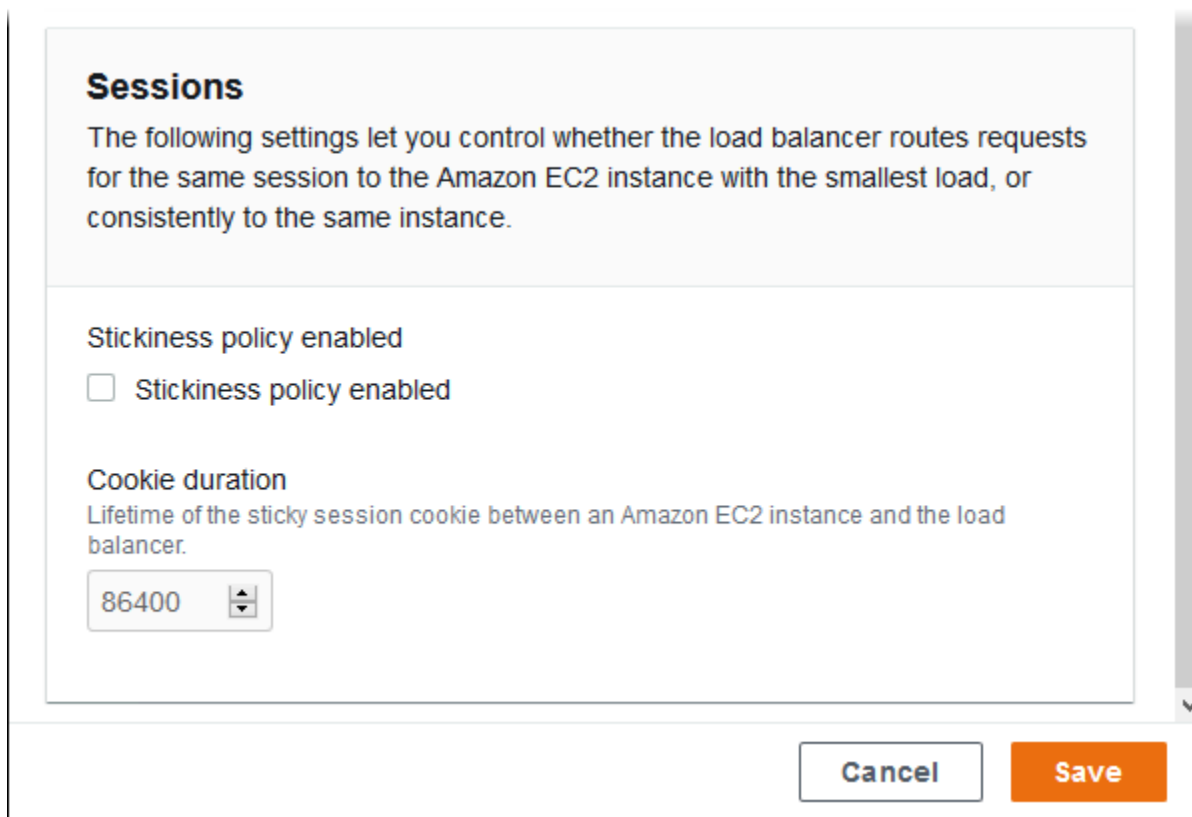
Note

La comprobación de estado de Elastic Load Balancing no afecta al comportamiento de comprobación de estado del grupo de Auto Scaling de un entorno. Las instancias que fallan en una comprobación de estado de Elastic Load Balancing no se reemplazan automáticamente por Amazon EC2 Auto Scaling, a menos que configure manualmente Amazon EC2 Auto Scaling para hacerlo. Para obtener más información, consulte [Configuración de comprobaciones de estado de Auto Scaling](#).

Para obtener más información sobre las comprobaciones de estado y la forma en que influyen en el estado general del entorno, consulte [Informes de estado básicos](#).

Sesiones

Seleccione o desactive la casilla **Stickiness policy enabled** (Política de persistencia habilitada) para habilitar o deshabilitar las sesiones sticky. Utilice **Cookie duration** (Duración de la cookie) para configurar una sesión sticky de duración, hasta **604800** segundos.



Sessions

The following settings let you control whether the load balancer routes requests for the same session to the Amazon EC2 instance with the smallest load, or consistently to the same instance.

Stickiness policy enabled

Stickiness policy enabled

Cookie duration

Lifetime of the sticky session cookie between an Amazon EC2 instance and the load balancer.

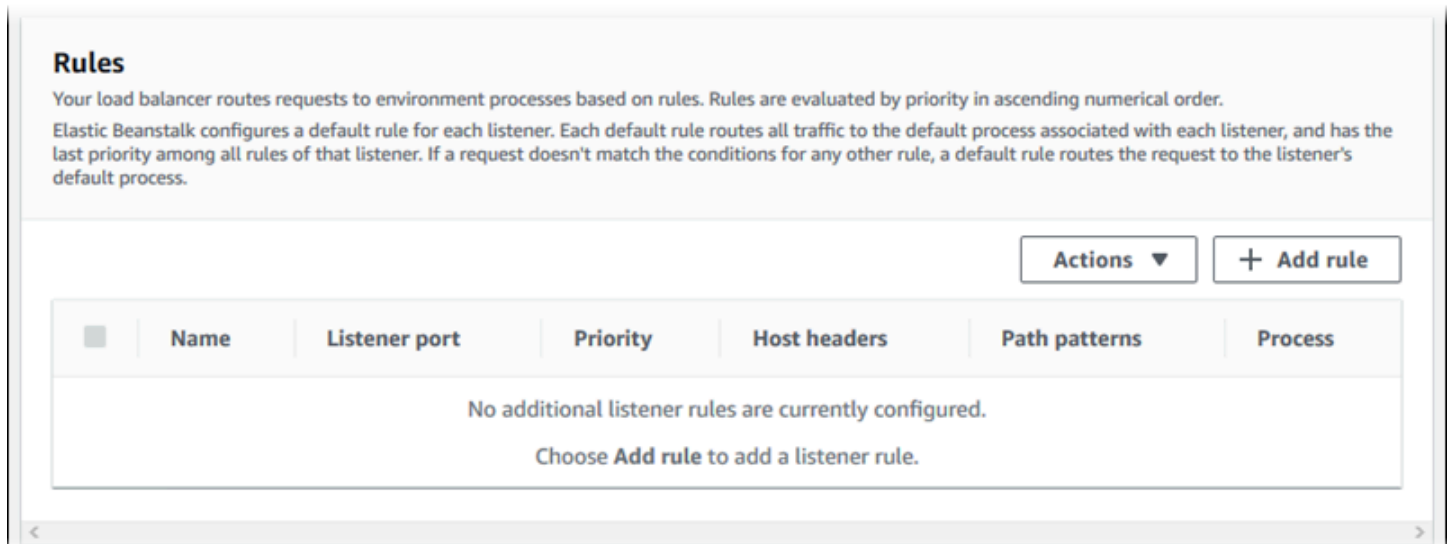
86400

Cancel Save

Reglas

Utilice esta lista para especificar las reglas del agente de escucha personalizado del balanceador de carga. Una regla asigna un destino a las solicitudes que reciba el agente de escucha con un patrón de ruta específico. Cada agente de escucha puede tener varias reglas, dirigiendo así las solicitudes con rutas distintas a procesos distintos de las instancias.

Las reglas tienen prioridades numéricas que determinan la prioridad con que se aplican a las solicitudes entrantes. Para cada nuevo agente de escucha que se agrega, Elastic Beanstalk añade una regla predeterminada que dirige todo su tráfico al proceso predeterminado. La prioridad de la regla predeterminada es la más baja; solo se aplica si ninguna otra regla del mismo agente de escucha es aplicable a la solicitud entrante. Inicialmente, si no ha agregado reglas personalizadas, la lista está vacía. No se muestran las reglas predeterminadas de todos los agentes de escucha.



Rules

Your load balancer routes requests to environment processes based on rules. Rules are evaluated by priority in ascending numerical order. Elastic Beanstalk configures a default rule for each listener. Each default rule routes all traffic to the default process associated with each listener, and has the last priority among all rules of that listener. If a request doesn't match the conditions for any other rule, a default rule routes the request to the listener's default process.

Actions ▾ + Add rule

	Name	Listener port	Priority	Host headers	Path patterns	Process
No additional listener rules are currently configured. Choose Add rule to add a listener rule.						

Puede editar las opciones de una regla existente o añadir una regla nueva. Para empezar a editar una regla de la lista o añadirle otra, siga los pasos que se indican para la [lista de agentes de escucha](#). Se abre el cuadro de diálogo Listener rule (Regla de agente de escucha) con las opciones siguientes:

- Name (Nombre): el nombre de la regla.
- Listener port (Puerto del agente de escucha): el puerto del agente de escucha al que se aplica la regla.
- Priority (Prioridad): la prioridad de la regla. Cuanto menor sea el número mayor es la prioridad. Las prioridades de las reglas de un agente de escucha deben ser únicas.

- **Match conditions (Condiciones de coincidencia):** una lista de condiciones de URL de solicitud a las que se aplica la regla. Existen dos tipos de condiciones: **HostHeader**(la parte del dominio de la URL) y **PathPattern**(la parte de la ruta de la URL). Puede agregar hasta cinco condiciones. Cada valor de condición tiene hasta 128 caracteres y puede incluir caracteres comodín.
- **Process (Proceso):** el proceso al que el balanceador de carga dirige las solicitudes a las que se aplica la regla.

Al editar cualquier regla existente, no puede cambiar el valor de **Name (Nombre)** ni el de **Listener port (Puerto del agente de escucha)**.

Listener rule ✕

Name

Listener port

Priority
Evaluated in ascending numerical order. Must be unique across all rules.

Match conditions
A listener rule can have up to five match conditions.

Type	Value	
<input type="text" value="PathPattern"/>	<input type="text" value="/images/*"/>	<input type="button" value="Remove"/>

Process

Captura de registro de acceso

Utilice estas opciones para configurar Elastic Load Balancing para capturar registros con información detallada sobre las solicitudes enviadas a su balanceador de carga de aplicaciones. La captura de registro de acceso está desactivada de forma predeterminada. Cuando los registros de almacenamiento están habilitados, Elastic Load Balancing almacena los registros en el bucket de S3 que ha configurado. La configuración Prefix (Prefijo) especifica una carpeta de nivel superior en el bucket para los registros. Elastic Load Balancing coloca los registros en una carpeta denominada AWSLogs en el prefijo. Si no especifica un prefijo, Elastic Load Balancing coloca su carpeta en el nivel raíz del bucket.

Note

Si el depósito de Amazon S3 que configuró para la captura de registros de acceso no es el que Elastic Beanstalk creó para su cuenta, asegúrese de añadir una política de usuario con los permisos AWS Identity and Access Management adecuados para sus usuarios (de IAM). Las [políticas de usuario administradas](#) que Elastic Beanstalk proporciona solo incluyen permisos para los recursos administrados por Elastic Beanstalk.

Para obtener más información acerca de los registros de acceso, incluidos los permisos y otros requisitos, [Registros de acceso para el balanceador de carga de aplicaciones](#).

Access log files

Configure Elastic Load Balancing to capture logs with detailed information about requests sent to your Load Balancer. Logs are stored in Amazon S3. [Learn more](#)

Store logs
(Standard Amazon S3 charges apply.)
 Enabled

S3 bucket
(You must first configure bucket permissions. [Learn more](#))
-- Choose an Amazon S3 bucket --
Choose a bucket.

Prefix
Logical hierarchy in the bucket. If you don't specify a prefix, Elastic Load Balancing stores access logs at the bucket's root.

Ejemplo: Balanceador de carga de aplicaciones con un agente de escucha seguro y dos procesos

En este ejemplo, la aplicación requiere el cifrado end-to-end del tráfico y un proceso independiente para gestionar las solicitudes administrativas.

Para configurar el balanceador de carga de aplicaciones de su entorno de modo que satisfaga estos requisitos, debe eliminar el agente de escucha predeterminado, agregar un agente de escucha HTTPS, indicar que el proceso predeterminado escucha en el puerto 443 para HTTPS y agregar un proceso y una regla de agente de escucha para dirigir el tráfico administrativo con una ruta diferente.

Para configurar el balanceador de carga para este ejemplo

1. Adición de un agente de escucha seguro. En Puerto, escriba **443**. En Protocolo, seleccione **HTTPS**. En Certificado SSL, seleccione el ARN del certificado SSL. Por ejemplo, **arn:aws:iam::123456789012:server-certificate/abc/certs/build** o **arn:aws:acm:us-east-2:123456789012:certificate/12345678-12ab-34cd-56ef-12345678**.

En Proceso predeterminado, mantenga seleccionado **default**.

Application Load Balancer listener ✕

Port
443

Protocol
The transport protocol that the load balancer uses for routing incoming traffic from clients.
HTTPS

SSL certificate
arn:aws:acm:us-east-2:123456789012:certific... ↻

SSL policy
The Secure Sockets Layer (SSL) negotiation configuration, known as a security policy, that this load balancer uses to negotiate SSL connections with clients.
ELBSecurityPolicy-2016-08

Default process
The process to which the listener routes traffic by default, when the message path doesn't match any custom listener rule.
default

Cancel Add

Ahora puede ver el agente de escucha adicional en la lista.

<input type="checkbox"/>	Port	Protocol	SSL certificate	Default process	Enabled
<input type="checkbox"/>	80	HTTP	--	default	<input checked="" type="checkbox"/>
<input type="checkbox"/>	443	HTTPS	arn:aws:acm:us-east-2:123456789012:certificate/12345678-12ab-34cd-56ef-12345678	default	<input checked="" type="checkbox"/>

2. Deshabilitar el agente de escucha predeterminado para HTTP en el puerto 80. Para el agente de escucha predeterminado, desactive la opción Enabled (Habilitado).

<input type="checkbox"/>	Port	Protocol	SSL certificate	Default process	Enabled
<input type="checkbox"/>	80	HTTP	--	default	<input type="checkbox"/>
<input type="checkbox"/>	443	HTTPS	arn:aws:acm:us-east-2:123456789012:certificate/12345678-12ab-34cd-56ef-12345678	default	<input checked="" type="checkbox"/>

3. Configuración del proceso predeterminado con HTTPS. Seleccione el proceso predeterminado y, a continuación, en Actions (Acciones), elija Edit (Editar). En Puerto, escriba **443**. En Protocolo, seleccione **HTTPS**.

Environment process

Name
default

Port
443

Protocol
HTTPS

4. Adición de un proceso administrativo. En Name (Nombre), escriba **admin**. En Puerto, escriba **443**. En Protocolo, seleccione **HTTPS**. En Comprobación de estado, en Ruta escriba **/admin**.

Environment process ✕

Name

admin

Port

443

Protocol

HTTPS

Health check

HTTP code
HTTP status code of a healthy instance in your environment.

200

Path
Path to which the load balancer sends HTTP health check requests.

/admin

5. Adición de una regla para el tráfico administrativo. En Name (Nombre), escriba **admin**. En Puerto de agente de escucha, escriba **443**. Para las condiciones de coincidencia, añada a PathPattern con el valor **/admin/***. En Procesar, seleccione **admin**.

Listener rule ✕

Name

Listener port

Priority
Evaluated in ascending numerical order. Must be unique across all rules.

Match conditions
A listener rule can have up to five match conditions.

Type	Value	
<input type="text" value="PathPattern"/>	<input type="text" value="/admin/*"/>	<input type="button" value="Remove"/>

Process

Configuración de un balanceador de carga de aplicaciones mediante la CLI de EB

La CLI de EB le pide que elija el tipo de balanceador de carga cuando se ejecuta [eb create](#).

```
$ eb create
Enter Environment Name
(default is my-app): test-env
Enter DNS CNAME prefix
(default is my-app): test-env-DLW24ED23SF

Select a load balancer type
```



```
1) classic
2) application
3) network
(default is 2):
```

También puede especificar un tipo de balanceador de carga con la opción `--elb-type`.

```
$ eb create test-env --elb-type application
```

Espacios de nombres de Application Load Balancer

Puede encontrar la configuración relacionada con balanceadores de carga de aplicaciones en los siguientes espacios de nombres:

- [aws:elasticbeanstalk:environment](#): elija el tipo de balanceador de carga para el entorno. El valor de un balanceador de carga de aplicaciones es `application`.

No puede establecer esta opción en los archivos de configuración ([.Ebextensions](#)).

- [aws:elbv2:loadbalancer](#): configure los registros de acceso y otras opciones que se aplican al balanceador de carga de aplicaciones como un todo.
- [aws:elbv2:listener](#): configure los agentes de escucha en el balanceador de carga de aplicaciones. Estos valores se corresponden con los de `aws:elb:listener` en los balanceadores de carga clásicos.
- [aws:elbv2:listenerrule](#): configure reglas que redirijan el tráfico a diferentes procesos, en función de la ruta de la solicitud. Las reglas son exclusivas de los balanceadores de carga de aplicaciones.
- [aws:elasticbeanstalk:environment:process](#): configure las comprobaciones de estado y especifique el puerto y el protocolo de los procesos que se ejecutan en las instancias del entorno. Los valores del puerto y del protocolo se asignan a los valores del puerto y del protocolo de la instancia especificados en `aws:elb:listener` para un agente de escucha de un balanceador de tipo Classic Load Balancer. Los valores de la comprobación de estado se corresponden con los valores de los espacios de nombres `aws:elb:healthcheck` y `aws:elasticbeanstalk:application`.

Example `.ebextensions/ .config alb-access-logs`

El siguiente archivo de configuración permite obtener acceso a las cargas de registros de un entorno con un balanceador de carga de aplicaciones.

```
option_settings:
  aws:elbv2:loadbalancer:
    AccessLogsS3Bucket: DOC-EXAMPLE-BUCKET
    AccessLogsS3Enabled: 'true'
    AccessLogsS3Prefix: beanstalk-alb
```

Example .ebextensions/ alb-default-process .config

El siguiente archivo de configuración modifica los ajustes de comprobación de estado y persistencia del proceso predeterminado.

```
option_settings:
  aws:elasticbeanstalk:environment:process:default:
    DeregistrationDelay: '20'
    HealthCheckInterval: '15'
    HealthCheckPath: /
    HealthCheckTimeout: '5'
    HealthyThresholdCount: '3'
    UnhealthyThresholdCount: '5'
    Port: '80'
    Protocol: HTTP
    StickinessEnabled: 'true'
    StickinessLBCookieDuration: '43200'
```

Example .ebextensions/ alb-secure-listener .config

El siguiente archivo de configuración agrega un agente de escucha seguro y un proceso coincidente al puerto 443.

```
option_settings:
  aws:elbv2:listener:443:
    DefaultProcess: https
    ListenerEnabled: 'true'
    Protocol: HTTPS
    SSLCertificateArns: arn:aws:acm:us-
east-2:123456789012:certificate/21324896-0fa4-412b-bf6f-f362d6eb6dd7
  aws:elasticbeanstalk:environment:process:https:
    Port: '443'
    Protocol: HTTPS
```

Example .ebextensions/ alb-admin-rule .config

El siguiente archivo de configuración agrega un agente de escucha seguro con una regla que direcciona el tráfico con la ruta de solicitud /admin a un proceso llamado admin que utiliza el puerto 4443.

```
option_settings:
  aws:elbv2:listener:443:
    DefaultProcess: https
    ListenerEnabled: 'true'
    Protocol: HTTPS
    Rules: admin
    SSLCertificateArns: arn:aws:acm:us-east-2:123456789012:certificate/21324896-0fa4-412b-bf6f-f362d6eb6dd7
  aws:elasticbeanstalk:environment:process:https:
    Port: '443'
    Protocol: HTTPS
  aws:elasticbeanstalk:environment:process:admin:
    HealthCheckPath: /admin
    Port: '4443'
    Protocol: HTTPS
  aws:elbv2:listenerrule:admin:
    PathPatterns: /admin/*
    Priority: 1
    Process: admin
```

Configuración de un balanceador de carga de aplicaciones compartido

Si ha [habilitado el balanceador de carga](#), el entorno de AWS Elastic Beanstalk contará con un balanceador de carga Elastic Load Balancing para distribuir el tráfico entre las instancias en el entorno. Elastic Load Balancing admite varios tipos de balanceadores de carga. Para obtener más información sobre ellos, consulte la [Guía de usuario de Elastic Load Balancing](#). Elastic Beanstalk puede crear un balanceador para usted o permitirle especificar un balanceador de carga compartido que haya creado.

En este tema se describe la configuración de un [balanceador de carga de aplicaciones](#) compartido que se crea y se asocia con el entorno. Véase también [the section called “Equilibrador de carga de aplicación”](#). Para obtener más información sobre cómo configurar todos los tipos de balanceadores de carga compatibles con Elastic Beanstalk, consulte [Balanceador de carga del entorno de Elastic Beanstalk](#).

Note

Solo es posible elegir el tipo de balanceador de carga de un entorno en el momento de su creación. Puede cambiar las opciones para administrar el comportamiento del balanceador de carga de un entorno en ejecución, pero no cambiar su tipo. Tampoco puede cambiar de un balanceador de carga dedicado a un balanceador de carga compartido o viceversa.

Introducción

Un balanceador de carga compartido es un balanceador de carga que crea y administra usted mismo mediante el servicio Amazon Elastic Compute Cloud (Amazon EC2) y que puede utilizar en varios entornos de Elastic Beanstalk.

Cuando crea un entorno de escalado balanceado de carga y elige utilizar un balanceador de carga de aplicaciones, Elastic Beanstalk crea un balanceador de carga dedicado al entorno de forma predeterminada. Para saber qué es un balanceador de carga de aplicaciones y cómo funciona en un entorno de Elastic Beanstalk, consulte la [introducción](#) a la configuración de balanceadores de carga de aplicaciones para Elastic Beanstalk.

En algunas situaciones puede que desee ahorrarse el costo de tener varios balanceadores de carga dedicados. Esto puede ser útil cuando multiplique varios entornos, por ejemplo, si su aplicación es un conjunto de microservicios en vez de un servicio monolítico. En tales casos, puede optar por utilizar un balanceador de carga compartido.

Para utilizar un balanceador de carga compartido, créelo primero en Amazon EC2 y agregue uno o más agentes de escucha. Durante la creación de un entorno de Elastic Beanstalk, proporcione el balanceador de carga y elija un puerto de escucha. Elastic Beanstalk asocia el agente de escucha con el proceso predeterminado en su entorno. Puede agregar reglas de agente de escucha personalizado para enrutar el tráfico desde encabezados y rutas de host específicos a otros procesos del entorno.

Elastic Beanstalk agrega una etiqueta al balanceador de carga compartido. El nombre de la etiqueta es `elasticbeanstalk:shared-elb-environment-count` y su valor es el número de entornos que comparten este balanceador de carga.

El uso de un balanceador de carga compartido es distinto de usar uno dedicado de varias maneras.

Regarding	Balanceador de carga de aplicaciones específico	Balanceador de carga de aplicaciones compartido.
Administración	Elastic Beanstalk crea y administra el balanceador de carga, los oyentes, las reglas de escucha y los procesos (grupos de destino). Elastic Beanstalk también los elimina cuando finaliza su entorno. Elastic Beanstalk puede configurar la captura del registro de acceso del balanceador de carga, si elige esa opción.	Puede crear y administrar el balanceador de carga y los oyentes fuera de Elastic Beanstalk. Elastic Beanstalk crea y administra una y un proceso por defecto y usted puede agregar reglas y procesos. Elastic Beanstalk elimina las reglas y procesos de escucha que se agregaron durante la creación del entorno.
Reglas del agente de escucha	Elastic Beanstalk crea una regla predeterminada para cada agente de escucha, para enrutar todo el tráfico al proceso predeterminado del agente de escucha.	<p>Elastic Beanstalk asocia una regla predeterminada solo con un agente de escucha del puerto 80, si existe. Si elige un puerto de agente de escucha predeterminado diferente, tiene que asociar la regla predeterminada con él (la consola de Elastic Beanstalk y la CLI de EB lo hacen automáticamente).</p> <p>Para resolver conflictos de condición de regla de agente de escucha entre entornos que comparten el balanceador de carga, Elastic Beanstalk agrega el CNAME del entorno a la regla del agente de escucha como condición de encabezado del host.</p> <p>Elastic Beanstalk trata la configuración de prioridad de regla como relativa entre entornos que comparten el balanceador de carga y los asigna a prioridades absolutas durante la creación.</p>

Regarding	Balanceador de carga de aplicaciones específico	Balanceador de carga de aplicaciones compartido.
Grupos de seguridad	Elastic Beanstalk crea un grupo de seguridad predeterminado y lo asocia al balanceador de carga.	Puede configurar uno o más grupos de seguridad para utilizarlos en el balanceador de carga. Si no lo hace, Elastic Beanstalk comprueba si un grupo de seguridad existente que gestiona Elastic Beanstalk ya está asociado al balanceador de carga. De lo contrario, Elastic Beanstalk crea un grupo de seguridad y lo adjunta al balanceador de carga. Elastic Beanstalk elimina este grupo de seguridad cuando finaliza el último entorno que comparte el balanceador de carga.
Actualizaciones	Puede actualizar el balanceador de carga de aplicaciones después de la creación del entorno. Puede editar agentes de escucha, reglas de escucha y procesos. Puede configurar la captura del registro de acceso del balanceador de carga.	No puede usar Elastic Beanstalk para configurar la captura del registro de acceso en el Balanceador de carga de aplicaciones y no puede actualizar listeners ni reglas de escucha después de la creación del entorno. Solo puede actualizar procesos (grupos de destino). Para configurar la captura de registro de acceso y actualizar los agentes de escucha y las reglas de agentes de escucha, utilice Amazon EC2.

Configuración de un balanceador de carga de aplicaciones compartido mediante la consola de Elastic Beanstalk

Puede utilizar la consola de Elastic Beanstalk para configurar un balanceador de carga de aplicaciones compartido durante la creación del entorno. Puede seleccionar uno de los balanceadores de carga compartibles de su cuenta para utilizarlo en el entorno, seleccionar el puerto del agente de escucha predeterminado y configure los procesos y reglas de agente de escucha adicionales.

No puede editar la configuración compartida del Balanceador de carga de aplicaciones en la consola del balanceador de carga de aplicaciones después de crear el entorno. Para configurar agentes

de escucha, reglas de agente de escucha, procesos (grupos de destino) y captura de registro de acceso, utilice Amazon EC2.

Para configurar un balanceador de carga de aplicaciones en la consola de Elastic Beanstalk durante la creación del entorno

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos).
3. Elija [Create a new environment \(Crear un nuevo entorno\)](#) para empezar a crear el entorno.
4. En la página principal del asistente, antes de elegir Create environment (Crear entorno), elija Configure more options (Configurar más opciones).
5. Elija la configuración predefinida High availability (Alta disponibilidad).

Como alternativa, en la categoría de configuración Capacity (Capacidad), configure como tipo de entorno Load balanced (Balanceo de carga). Para obtener más información, consulte [Capacidad](#).

6. En la categoría de configuración Load balancer (Balanceador de carga), elija Edit (Editar).
7. Seleccione la opción balanceador de carga de aplicaciones si aún no está seleccionada y, a continuación, seleccione la opción Compartido .

Elastic Beanstalk > Applications > getting-started-app

Modify load balancer

Load balancer type

- Application Load Balancer**
Application layer load balancer—routing HTTP and HTTPS traffic based on protocol, port, and route to environment processes.
- Classic Load Balancer**
Previous generation — HTTP, HTTPS, and TCP
- Network Load Balancer**
Ultra-high performance and static IP addresses for your application.

Dedicated
Use a load balancer that Elastic Beanstalk creates exclusively for this environment.

Shared
Use a load balancer that someone in your account created. It can be shared among multiple Elastic Beanstalk environments.

8. Realice los cambios compartidos en la configuración del balanceador de carga de aplicaciones que requiera su entorno.
9. Elija Save (Guardar) y haga los demás cambios de configuración que requiera su entorno.
10. Seleccione Create environment (Crear entorno).

Configuración del balanceador de carga de aplicaciones compartidas

- [Balanceador de carga de aplicaciones compartido.](#)
- [Processes](#)
- [Reglas](#)

Balanceador de carga de aplicaciones compartido.

Utilice esta sección para elegir un balanceador de carga de aplicaciones compartido para su entorno y configurar el enrutamiento de tráfico predeterminado.

Antes de configurar aquí un balanceador de carga de aplicaciones compartido, utilice Amazon EC2 para definir al menos un balanceador de carga de aplicaciones para compartirlo, con al menos un agente de escucha, en su cuenta. Si aún no lo ha hecho, puede elegir Administrar balanceadores de carga. Elastic Beanstalk abre la consola de Amazon EC2 en una nueva pestaña del navegador.

Cuando haya terminado de configurar balanceadores de carga compartidos fuera de Elastic Beanstalk, configure las siguientes opciones en esta sección de consola:

- Load balancer ARN (ARN de balanceador de carga): el balanceador de carga compartido que se va a utilizar en este entorno. Seleccione de una lista de balanceadores de carga o introduzca un nombre de recurso de Amazon (ARN) de balanceador de carga.
- Default listener port (Puerto de agente de escucha predeterminado): un puerto de agente de escucha en el que escucha el balanceador de carga compartido. Seleccione de una lista de puertos de agente de escucha existentes. El tráfico de este agente de escucha con el CNAME del entorno en el encabezado del host se redirige a un proceso predeterminado en este entorno.

Shared Application Load Balancer

Select a shared load balancer and default listener for your environment. To manage load balancers and listeners, choose **Manage load balancers**.

Manage load balancers

Load balancer ARN

Must be an active Application Load Balancer in vpc-5732152e

Default listener

The default process and rule are associated with this listener.

Processes

Utilice esta lista para especificar procesos para el balanceador de carga compartido. Un proceso es un destino al que los agentes de escucha dirigen el tráfico. Inicialmente, la lista muestra el proceso predeterminado, que recibe tráfico del agente de escucha predeterminado.

Processes

For each environment process, you can specify the protocol and port that the load balancer uses to route requests to the process. You can also specify how the load balancer performs process health checks.

Actions ▼
+ Add process

	Name	Port	Protocol	HTTP code	Health check path	Stickiness
<input type="checkbox"/>	default	80	HTTP		/	disabled

Para configurar un proceso existente

1. Seleccione la casilla de verificación junto a la entrada correspondiente en la tabla y, a continuación, elija **Actions** (Acciones), **Edit** (Editar).
2. Utilice el cuadro de diálogo **Proceso** del entorno para editar la configuración y, a continuación, elija **Guardar**.

Para agregar un proceso

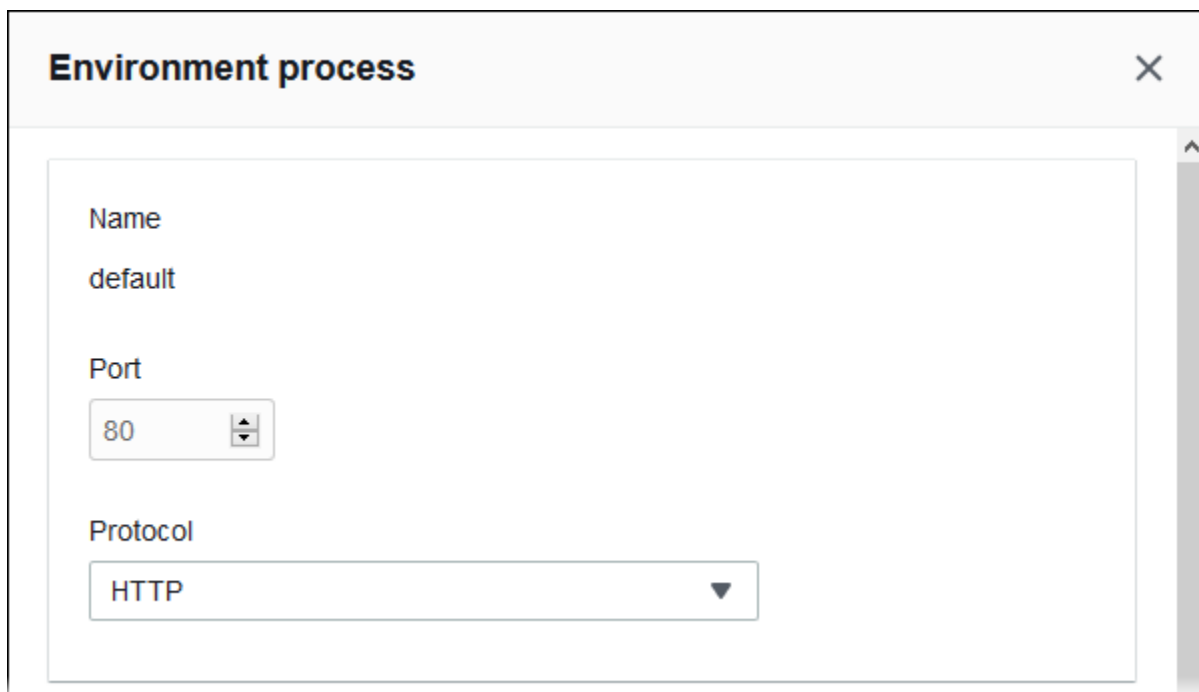
1. Elija Agregar un proceso.
2. En el cuadro de diálogo Proceso del entorno configure la configuración que desee y, a continuación, elija Añadir.

Configuración del cuadro de diálogo del proceso de entorno del balanceador de carga de aplicaciones

- [Definición](#)
- [Comprobación de estado](#)
- [Sesiones](#)

Definición

Utilice estas opciones para definir el proceso: Name (Nombre), Port (Puerto) y Protocol (Protocolo) en los que espera las solicitudes.



The screenshot shows a dialog box titled "Environment process" with a close button (X) in the top right corner. The dialog contains three configuration fields:

- Name:** A text input field containing the value "default".
- Port:** A spinner control showing the value "80".
- Protocol:** A dropdown menu with "HTTP" selected.

Comprobación de estado

Utilice las opciones siguientes para configurar las comprobaciones de estado del proceso:

- HTTP code (Código HTTP): el código de estado HTTP que designa a un proceso en buen estado.

- Path (Ruta) : ruta de la solicitud de comprobación de estado del proceso.
- Timeout (Tiempo de espera): Intervalo de tiempo en segundos que debe esperarse una respuesta de comprobación de estado.
- Interval (Intervalo): Intervalo de tiempo en segundos entre las comprobaciones de estado de una instancia individual. El intervalo debe ser mayor que el tiempo de espera.
- Unhealthy threshold (Umbral incorrecto), Healthy threshold (Umbral correcto): número de comprobaciones de estado que deben fallar o pasar, respectivamente, para que Elastic Load Balancing cambie el estado de una instancia.
- Deregistration delay (Retraso de anulación del registro): intervalo de tiempo, en segundos, que debe esperarse a que se completen las solicitudes activas antes de anular el registro de una instancia.

Health check

HTTP code

HTTP status code of a healthy instance in your environment.

Path

Path to which the load balancer sends HTTP health check requests.

Timeout

Amount of time to wait for a health check response.

 seconds

Interval

Amount of time between health checks of an individual instance. The interval must be greater than the timeout.

 seconds

Unhealthy threshold

The number of consecutive health check failures required to designate the instance as unhealthy.

 requests

Healthy threshold

The number of consecutive successful health checks required to designate the instance as healthy.

 requests

Deregistration delay

Amount of time to wait for active requests to complete before deregistering.

 seconds

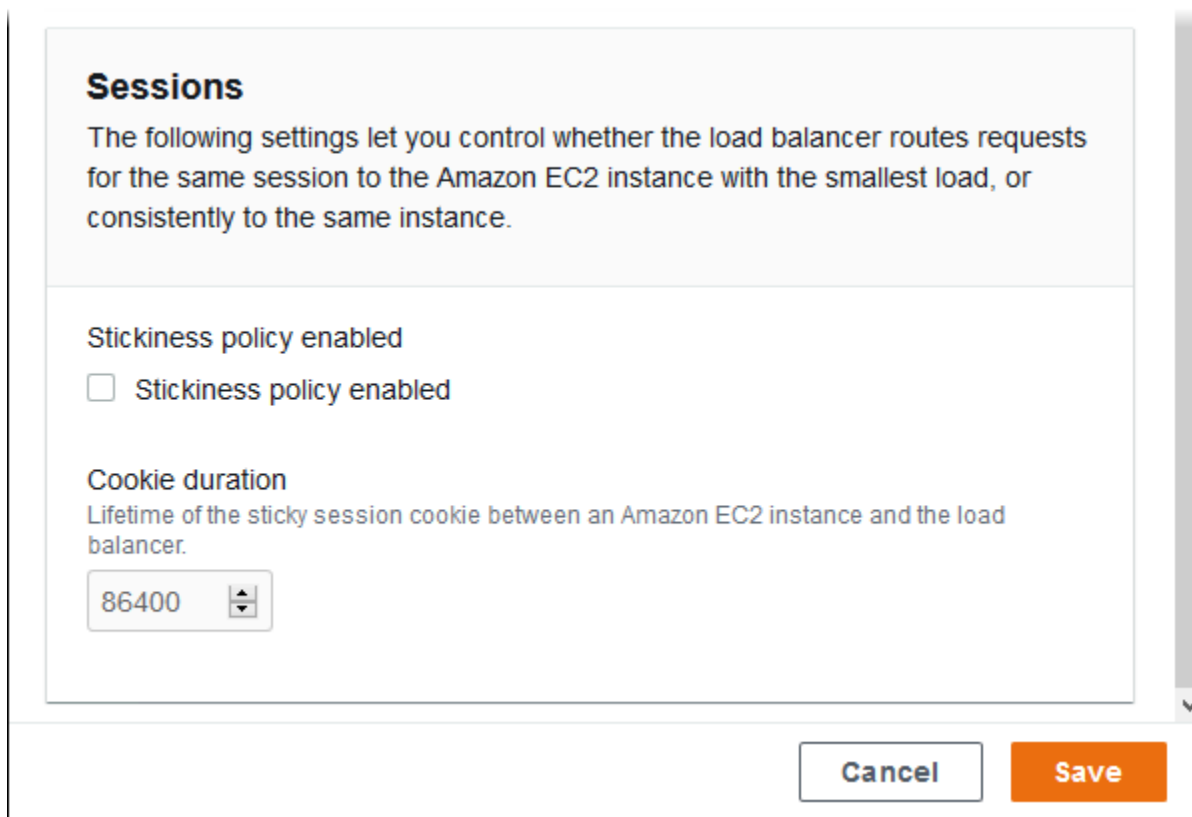
Note

La comprobación de estado de Elastic Load Balancing no afecta al comportamiento de comprobación de estado del grupo de Auto Scaling de un entorno. Las instancias que fallan en una comprobación de estado de Elastic Load Balancing no se reemplazan automáticamente por Amazon EC2 Auto Scaling, a menos que configure manualmente Amazon EC2 Auto Scaling para hacerlo. Para obtener más información, consulte [Configuración de comprobaciones de estado de Auto Scaling](#).

Para obtener más información sobre las comprobaciones de estado y la forma en que influyen en el estado general del entorno, consulte [Informes de estado básicos](#).

Sesiones

Seleccione o desactive la casilla **Stickiness policy enabled** (Política de persistencia habilitada) para habilitar o deshabilitar las sesiones sticky. Utilice **Cookie duration** (Duración de la cookie) para configurar una sesión sticky de duración, hasta **604800** segundos.



The screenshot shows the 'Sessions' configuration panel in the AWS Management Console. The panel has a title 'Sessions' and a descriptive paragraph: 'The following settings let you control whether the load balancer routes requests for the same session to the Amazon EC2 instance with the smallest load, or consistently to the same instance.' Below this, there are two settings: 'Stickiness policy enabled' with an unchecked checkbox, and 'Cookie duration' with a text input field containing '86400' and a spinner control. At the bottom right, there are 'Cancel' and 'Save' buttons.

Reglas

Utilice esta lista para especificar las reglas del agente de escucha personalizadas para el balanceador de carga compartido. Una regla asigna un destino a las solicitudes que reciba el agente de escucha con un patrón de ruta específico. Cada agente de escucha puede tener varias reglas, redireccionando solicitudes en distintas rutas a distintos procesos en instancias de los diferentes entornos que comparten el agente de escucha.

Las reglas tienen prioridades numéricas que determinan la prioridad con que se aplican a las solicitudes entrantes. Elastic Beanstalk agrega una regla predeterminada que enruta todo el tráfico del listener predeterminado al proceso predeterminado de su nuevo entorno. La prioridad de la regla predeterminada es la más baja; solo se aplica si ninguna otra regla del mismo agente de escucha es aplicable a la solicitud entrante. Inicialmente, si no ha agregado reglas personalizadas, la lista está vacía. No se muestra la regla predeterminada.

Rules

Your load balancer routes requests to environment processes based on rules. Rules are evaluated by priority in ascending numerical order. If the shared load balancer has existing rules configured, this environment's rules are adjusted to have lower priority than existing rules. You can manage rules across environments in the EC2 console.

Elastic Beanstalk configures a default rule for this environment. This rule routes all traffic from the default listener on port **80** to the default process, and has the last priority among all rules of this environment. If a request doesn't match the conditions for any other rule, the default rule routes the request to the default process.

Shared load balancer environment rules

After environment creation, you can't add or edit rules for this environment using Elastic Beanstalk. When you terminate the environment, listener rules created outside of Elastic Beanstalk aren't automatically removed by Elastic Beanstalk.

Actions ▾ + Add rule

	Name	Listener port	Priority	Host headers	Path patterns	Process
No additional listener rules are currently configured. Choose Add rule to add a listener rule.						

Cancel Save

Puede editar las opciones de una regla existente o añadir una regla nueva. Para comenzar a editar una regla en la lista o agregarle otra, siga los mismos pasos que se indican para la [lista de procesos](#). Se abre el cuadro de diálogo Listener rule (Regla de agente de escucha) con las opciones siguientes:

- **Name (Nombre):** el nombre de la regla.
- **Listener port (Puerto del agente de escucha):** el puerto del agente de escucha al que se aplica la regla.
- **Priority (Prioridad):** la prioridad de la regla. Cuanto menor sea el número mayor es la prioridad. Las prioridades de las reglas de un agente de escucha deben ser únicas. Elastic Beanstalk trata las prioridades de regla como relativas en todos los entornos de uso compartido y las asigna a prioridades absolutas durante la creación.
- **Match conditions (Condiciones de coincidencia):** una lista de condiciones de URL de solicitud a las que se aplica la regla. Hay dos tipos de condiciones: **HostHeader** (la parte del dominio de la URL) y **PathPattern** (la parte de la ruta de la URL). Una condición está reservada para el subdominio del entorno y puede agregar hasta cuatro condiciones. El valor de cada condición tiene una longitud de hasta 128 caracteres y puede incluir caracteres comodín.
- **Process (Proceso):** el proceso al que el balanceador de carga dirige las solicitudes a las que se aplica la regla.

Listener rule ✕

Name
images

Listener port
80 ▼

Priority
Evaluated in ascending numerical order. Must be unique across all rules.
1 ▲▼

Match conditions
A listener rule can have up to five match conditions.

Type	Value	
PathPattern ▼	/images/*	Remove

Add condition

Process
images ▼

Cancel Add

Ejemplo: utilizar un balanceador de carga de aplicaciones compartido para una aplicación segura basada en microservicios

En este ejemplo, la aplicación consta de varios microservicios, cada uno implementado como un entorno de Elastic Beanstalk. Además, necesita cifrado de tráfico de extremo a extremo. Demostraremos uno de los entornos de microservicios, que tiene un proceso principal para las solicitudes de los usuarios y un proceso independiente para la gestión de solicitudes administrativas.

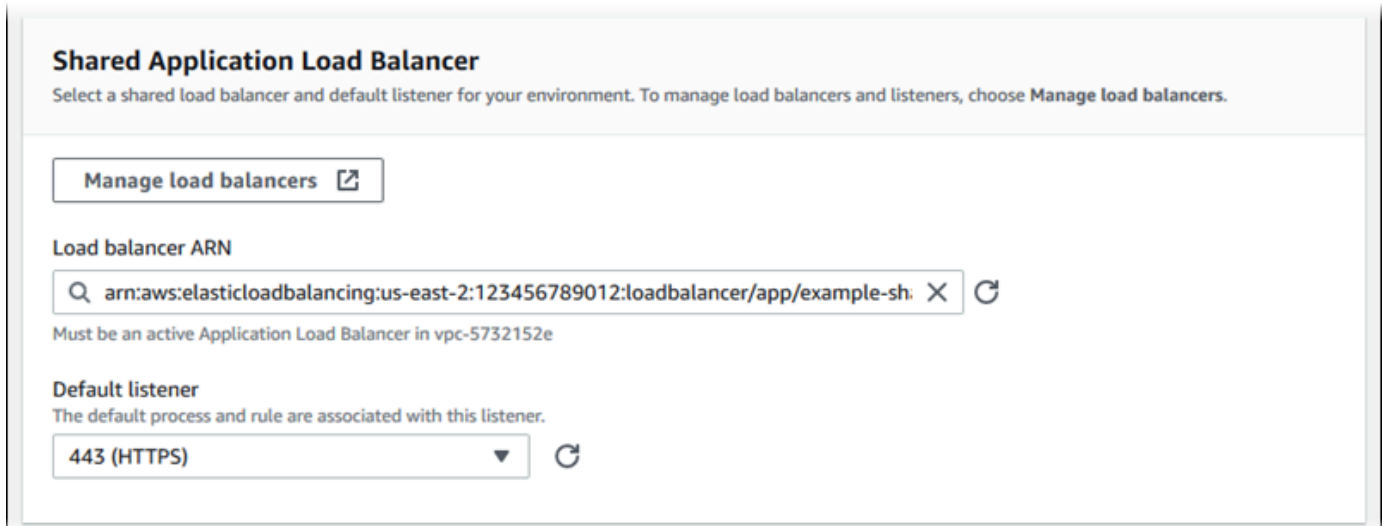
Para cumplir estos requisitos, utilice Amazon EC2 para crear un balanceador de carga de aplicaciones que compartirá entre sus microservicios. Agregue un agente de escucha seguro en el puerto 443 y el protocolo HTTPS. A continuación, agregue varios certificados SSL al agente de

escucha: uno por dominio de microservicio. Para obtener más información sobre cómo crear el balanceador de carga de aplicaciones y el agente de escucha seguro, consulte [Crear un balanceador de carga de aplicaciones](#) y [Crear un agente de escucha HTTPS para el balanceador de carga de aplicaciones](#) en la Guía del usuario de balanceadores de carga de aplicaciones.

En Elastic Beanstalk, configure cada entorno de microservicios para utilizar el balanceador de carga de aplicaciones compartido y establezca el puerto de escucha predeterminado en 443. En el caso del entorno particular que estamos demostrando aquí, indique que el proceso predeterminado escucha el puerto 443 en HTTPS y agregue un proceso y una regla de agente de escucha para el tráfico de administración en una ruta diferente.

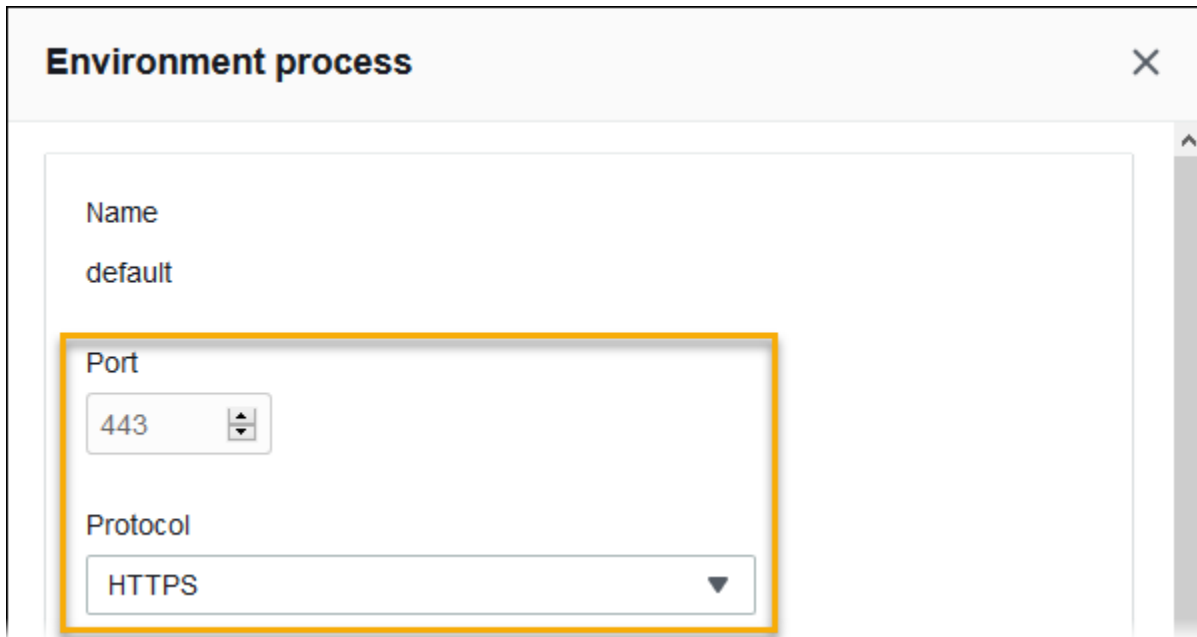
Para configurar el balanceador de carga compartido para este ejemplo

1. En la sección Balanceador de carga de aplicaciones compartidas seleccione el balanceador de carga y, a continuación, en Puerto de escucha predeterminado, seleccione **443**. El puerto del agente de escucha ya debe estar seleccionado si es el único agente de escucha que tiene el balanceador de carga.



The screenshot shows the configuration for a Shared Application Load Balancer. At the top, it says "Shared Application Load Balancer" and provides instructions to "Select a shared load balancer and default listener for your environment. To manage load balancers and listeners, choose Manage load balancers." Below this is a button labeled "Manage load balancers" with an external link icon. The "Load balancer ARN" field contains the value "arn:aws:elasticloadbalancing:us-east-2:123456789012:loadbalancer/app/example-sh" and has a search icon, a clear icon, and a refresh icon. Below the ARN field is a note: "Must be an active Application Load Balancer in vpc-5732152e". The "Default listener" section has a note: "The default process and rule are associated with this listener." Below this is a dropdown menu showing "443 (HTTPS)" with a clear icon.

2. Configuración del proceso predeterminado con HTTPS. Seleccione el proceso predeterminado y, a continuación, en Actions (Acciones), elija Edit (Editar). En Puerto, escriba **443**. En Protocolo, seleccione **HTTPS**.



Environment process [X]

Name
default

Port
443

Protocol
HTTPS

3. Adición de un proceso administrativo. En Name (Nombre), ingrese **admin**. En Puerto, escriba **443**. En Protocolo, seleccione **HTTPS**. En Comprobación de estado, en Ruta escriba **/admin**.

Environment process

Name
admin

Port
443

Protocol
HTTPS

Health check

HTTP code
HTTP status code of a healthy instance in your environment.
200

Path
Path to which the load balancer sends HTTP health check requests.
/admin

4. Adición de una regla para el tráfico administrativo. En Name (Nombre), ingrese **admin**. En Puerto de agente de escucha, escriba **443**. En Condiciones de coincidencia, agregue un PathPattern con el valor **/admin/***. En Procesar, seleccione **admin**.

Listener rule ✕

Name
admin

Listener port
443 ▼

Priority
Evaluated in ascending numerical order. Must be unique across all rules.
1 ▲▼

Match conditions
A listener rule can have up to five match conditions.

Type	Value	
PathPattern ▼	/admin/*	Remove

Add condition

Process
admin ▼

Cancel Add

Configuración de un balanceador de carga de aplicaciones compartido mediante la CLI de EB

La CLI de EB le pide que elija el tipo de balanceador de carga cuando se ejecuta [eb create](#). Si elige `application` (el valor predeterminado) y si su cuenta tiene al menos un balanceador de carga de aplicaciones compatible, la CLI de EB también le pregunta si desea utilizar un balanceador de carga de aplicaciones compartido. Si responde `y`, también se le pedirá que seleccione el balanceador de carga y el puerto predeterminado.

```
$ eb create
```

```

Enter Environment Name
(default is my-app): test-env
Enter DNS CNAME prefix
(default is my-app): test-env-DLW24ED23SF

Select a load balancer type
1) classic
2) application
3) network
(default is 2):

Your account has one or more sharable load balancers. Would you like your new
environment to use a shared load balancer?(y/N) y

Select a shared load balancer
1)MySharedALB1 - arn:aws:elasticloadbalancing:us-east-2:123456789012:loadbalancer/app/
MySharedALB1/6d69caa75b15d46e
2)MySharedALB2 - arn:aws:elasticloadbalancing:us-east-2:123456789012:loadbalancer/app/
MySharedALB2/e574ea4c37ad2ec8
(default is 1): 2

Select a listener port for your shared load balancer
1) 80
2) 100
3) 443
(default is 1): 3

```

También puede especificar un balanceador de carga compartido mediante las opciones de comando.

```
$ eb create test-env --elb-type application --shared-lb MySharedALB2 --shared-lb-
port 443
```

Espacios de nombres de balanceador de carga de aplicaciones compartidos.

Puede encontrar la configuración relacionada con los balanceadores de carga de aplicaciones compartidos en los siguientes espacios de nombres:

- [aws:elasticbeanstalk:environment](#): elija el tipo de balanceador de carga para el entorno e indique a Elastic Beanstalk que utilizará un balanceador de carga compartido.

No puede establecer estas dos opciones en los archivos de configuración ([.Ebextensions](#)).

- [aws:elbv2:loadbalancer](#) : configure el ARN compartido del balanceador de carga de aplicaciones y los grupos de seguridad.
- [aws:elbv2:listener](#) : asocie agentes de escucha del balanceador de carga de aplicaciones compartido con procesos de entorno mediante la lista de reglas de escucha.
- [aws:elbv2:listenerrule](#): configure reglas de agente de escucha que redirijan el tráfico a diferentes procesos, en función de la ruta de la solicitud. Las reglas son exclusivas de los balanceadores de carga de aplicaciones, tanto los específicos como los compartidos.
- [aws:elasticbeanstalk:environment:process](#): configure las comprobaciones de estado y especifique el puerto y el protocolo de los procesos que se ejecutan en las instancias del entorno.

Example `.ebextensions/application-load-balancer-shared.config`

Para empezar a utilizar un balanceador de carga de aplicaciones compartido, utilice la consola Elastic Beanstalk, la CLI de EB o la API para establecer el tipo de balanceador de carga en `application` y elegir utilizar un balanceador de carga compartido. Utilice un [archivo de configuración](#) para configurar el balanceador de carga compartido.

```
option_settings:
  aws:elbv2:loadbalancer:
    SharedLoadBalancer: arn:aws:elasticloadbalancing:us-
east-2:123456789012:loadbalancer/app/MySharedALB2/e574ea4c37ad2ec8
```

Note

Puede configurar esta opción solo durante la creación del entorno.

Example `.ebextensions/alb-shared-secure-listener.config`

El siguiente archivo de configuración selecciona un agente de escucha seguro predeterminado en el puerto 443 para el balanceador de carga compartido y establece el proceso predeterminado para escuchar el puerto 443.

```
option_settings:
  aws:elbv2:loadbalancer:
    SharedLoadBalancer: arn:aws:elasticloadbalancing:us-
east-2:123456789012:loadbalancer/app/MySharedALB2/e574ea4c37ad2ec8
```

```
aws:elbv2:listener:443:
  rules: default
aws:elasticbeanstalk:environment:process:default:
  Port: '443'
  Protocol: HTTPS
```

Example .ebextensions/alb-shared-admin-rule.config

El siguiente archivo de configuración se basa en el ejemplo anterior y agrega una regla que redirige el tráfico con una ruta de solicitud de /admin a un proceso denominado admin que escucha en el puerto 4443.

```
option_settings:
  aws:elbv2:loadbalancer:
    SharedLoadBalancer: arn:aws:elasticloadbalancing:us-
east-2:123456789012:loadbalancer/app/MySharedALB2/e574ea4c37ad2ec8
  aws:elbv2:listener:443:
    rules: default,admin
  aws:elasticbeanstalk:environment:process:default:
    Port: '443'
    Protocol: HTTPS
  aws:elasticbeanstalk:environment:process:admin:
    HealthCheckPath: /admin
    Port: '4443'
    Protocol: HTTPS
  aws:elbv2:listenerrule:admin:
    PathPatterns: /admin/*
    Priority: 1
    Process: admin
```

Configuración de un balanceador de carga de red

Si ha [habilitado el balanceador de carga](#), el entorno de AWS Elastic Beanstalk contará con un balanceador de carga Elastic Load Balancing para distribuir el tráfico entre las instancias en el entorno. Elastic Load Balancing admite varios tipos de balanceadores de carga. Para obtener más información sobre ellos, consulte la [Guía de usuario de Elastic Load Balancing](#). Elastic Beanstalk puede crear un balanceador para usted o permitirle especificar un balanceador de carga compartido que haya creado.

En este tema se describe la configuración de un [balanceador de carga de red](#) que Elastic Beanstalk crea y dedica a su entorno. Para obtener más información sobre cómo configurar todos los tipos

de balanceadores de carga compatibles con Elastic Beanstalk, consulte [Balanceador de carga del entorno de Elastic Beanstalk](#).

Note

Solo es posible elegir el tipo de balanceador de carga de un entorno en el momento de su creación. Puede cambiar las opciones para administrar el comportamiento del balanceador de carga de un entorno en ejecución, pero no cambiar su tipo.

Introducción

Con un balanceador de carga de red, el agente de escucha predeterminado acepta las solicitudes TCP en el puerto 80 y las distribuye a las instancias del entorno. Puede configurar el comportamiento de las comprobaciones de estado, configurar el puerto de agente de escucha o agregar un agente de escucha a otro puerto.

Note

A diferencia de un balanceador de carga clásico o un balanceador de carga de aplicaciones, un balanceador de carga de red no puede tener oyentes HTTP o HTTPS de capa de aplicación (capa 7) HTTP o HTTPS. Solo admite la capa de transporte (capa 4) de los agentes de escucha TCP. El tráfico HTTP y HTTPS se pueden dirigir a su entorno a través de TCP. Para establecer las conexiones HTTPS seguras entre clientes web y su entorno, instale un [certificado autofirmado](#) en las instancias del entorno y configure las instancias para escuchar en el puerto correspondiente (por lo general, 443) y terminar las conexiones HTTPS. La configuración varía en función de la plataforma. Para obtener instrucciones, consulte [Configurar su aplicación para terminar las conexiones HTTPS en la instancia](#). A continuación, configure su balanceador de carga de red para añadir un agente de escucha que se asigne a un proceso que escuche en el puerto correspondiente.

Un balanceador de carga de red admite comprobaciones de estado activas. Estas comprobaciones se basan en mensajes a la ruta raíz (/). Además, un balanceador de carga de red admite comprobaciones de estado pasivas. Detecta automáticamente instancias backend defectuosas y direcciona el tráfico solo a las instancias con estado correcto.

Configuración de un balanceador de carga de red mediante la consola de Elastic Beanstalk

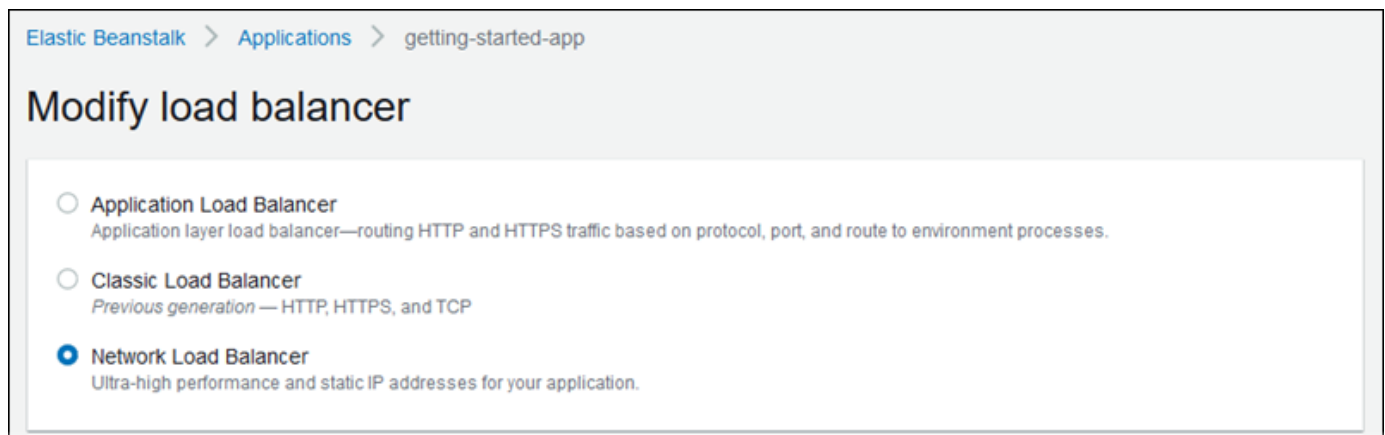
Puede utilizar la consola de Elastic Beanstalk para configurar los oyentes y los procesos de un balanceador de carga de red durante la creación del entorno o posteriormente, cuando el entorno se está ejecutando.

Para configurar un balanceador de carga de red en la consola de Elastic Beanstalk durante la creación del entorno

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos).
3. Elija [Create a new environment \(Crear un nuevo entorno\)](#) para empezar a crear el entorno.
4. En la página principal del asistente, antes de elegir Create environment (Crear entorno), elija Configure more options (Configurar más opciones).
5. Elija la configuración predefinida High availability (Alta disponibilidad).

Como alternativa, en la categoría de configuración Capacity (Capacidad), configure como tipo de entorno Load balanced (Balanceo de carga). Para obtener más información, consulte [Capacidad](#).

6. En la categoría de configuración Load balancer (Balanceador de carga), elija Edit (Editar).
7. Seleccione la opción balanceador de carga de red si aún no está seleccionada.



8. Realice los cambios en la configuración del balanceador de carga de red que requiera su entorno.
9. Elija Save (Guardar) y haga los demás cambios de configuración que requiera su entorno.

10. Seleccione Create environment (Crear entorno).

Para configurar el balanceador de carga de red de un entorno en ejecución en la consola de Elastic Beanstalk

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Load balancer (Balanceador de carga), elija Edit (Editar).

Note

Si la categoría de configuración Load balancer (Balanceador de carga) no tiene un botón Edit (Editar), el entorno no tiene un balanceador de carga. Para saber cómo configurar uno, consulte [Cambiar el tipo de entorno](#)

5. Realice los cambios en la configuración del balanceador de carga de red que su entorno requiera.
6. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

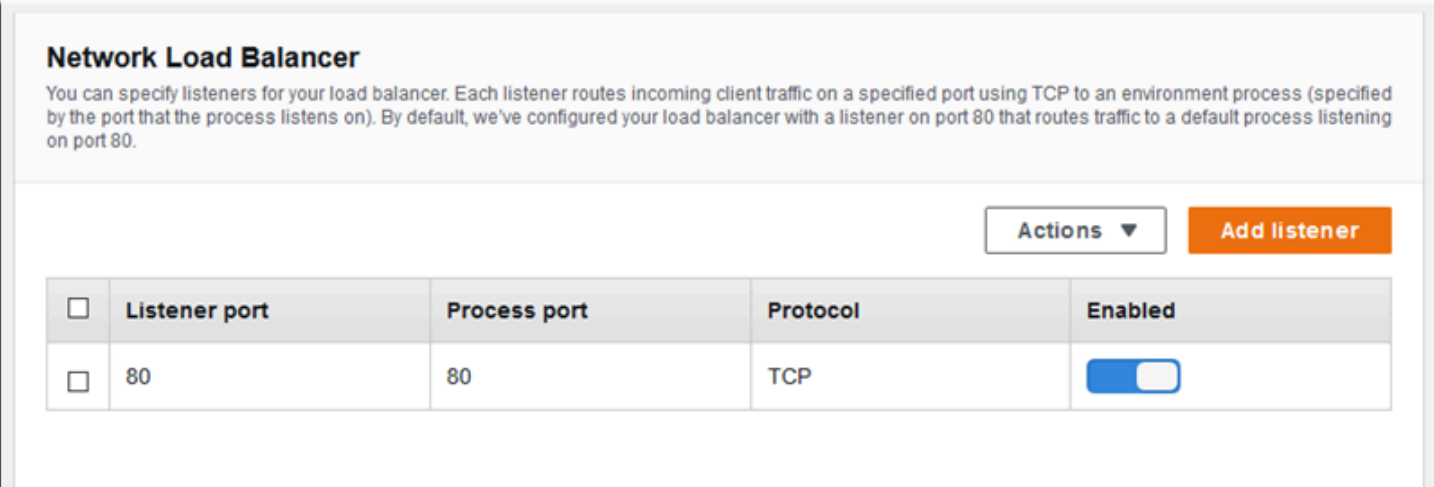
Configuración del balanceador de carga de red

- [Agentes de escucha](#)
- [Processes](#)

Agentes de escucha

Utilice esta lista para especificar los agentes de escucha del balanceador de carga. Cada agente de escucha dirige a las instancias el tráfico entrante desde el cliente en un puerto a un proceso en sus instancias. Inicialmente, la lista muestra el agente de escucha predeterminado, que dirige el tráfico

entrante en el puerto 80 a un proceso llamado default (predeterminado), que escucha en el puerto 80.



Network Load Balancer

You can specify listeners for your load balancer. Each listener routes incoming client traffic on a specified port using TCP to an environment process (specified by the port that the process listens on). By default, we've configured your load balancer with a listener on port 80 that routes traffic to a default process listening on port 80.

Actions ▾ Add listener

<input type="checkbox"/>	Listener port	Process port	Protocol	Enabled
<input type="checkbox"/>	80	80	TCP	<input checked="" type="checkbox"/>

Para configurar un agente de escucha existente

1. Seleccione la casilla de verificación junto a la entrada correspondiente en la tabla y, a continuación, elija Actions (Acciones), Edit (Editar).
2. Utilice el cuadro de diálogo Network Load Balancer listener (Agente de escucha de balanceador de carga de red) para modificar las opciones y a continuación elija Save (Guardar).

Para agregar un agente de escucha

1. Elija Add listener (Añadir agente de escucha).
2. En el cuadro de diálogo Network Load Balancer listener (Agente de escucha de balanceador de carga de red), configure las opciones requeridas y, a continuación, elija Add (Añadir).

Utilice el cuadro de diálogo Network Load Balancer listener (Agente de escucha del balanceador de carga de red) para configurar el puerto en el que el agente escucha el tráfico, y elija el proceso al que desea dirigir el tráfico (especificado por el puerto que el proceso escucha).

Network Load Balancer listener ✕

Listener port
80

Protocol
The transport protocol that the load balancer uses for routing incoming traffic from clients.
TCP ▼

Process port
The port to which this listener routes traffic. It determines the environment process that receives traffic from the listener.
80 ▼

Cancel **Save**

Processes

Utilice esta lista para especificar los procesos del balanceador de carga. Un proceso es un destino al que los agentes de escucha dirigen el tráfico. Cada agente de escucha dirige a las instancias el tráfico entrante desde el cliente en un puerto a un proceso en sus instancias. Inicialmente, la lista muestra el proceso predeterminado, que escucha el tráfico entrante en el puerto 80.

Processes

For each environment process, you can specify the port that the load balancer uses to route requests to the process. You can also specify how the load balancer performs process health checks.

Actions ▾ Add process

<input type="checkbox"/>	Process name	Process port	Interval	Healthy threshold	Unhealthy threshold
<input type="checkbox"/>	default	80	10	5	5

Cancel Save

Puede editar la configuración de un proceso existente o agregar un nuevo proceso. Para empezar a editar un proceso de la lista o añadirle otro, utilice los pasos que se indican para la [lista de agentes de escucha](#). Se abre el cuadro de diálogo Environment process (Proceso del entorno).

Cuadro de diálogo de configuración del proceso de entorno del balanceador de carga de red

- [Definición](#)
- [Comprobación de estado](#)

Definición

Utilice estas opciones para definir el proceso: Name (Nombre) y el Process port (Puerto del proceso) en los que espera las solicitudes.

Environment process ✕

Name
default

Process port
80 ▾

Comprobación de estado

Utilice las opciones siguientes para configurar las comprobaciones de estado del proceso:

- **Interval (Intervalo):** Intervalo de tiempo en segundos entre las comprobaciones de estado de una instancia individual.
- **Healthy threshold (Umbral de correcto):** Número de comprobaciones de estado que deben pasar para que Elastic Load Balancing cambie el estado de una instancia. (Para el balanceador de carga de red, **Unhealthy threshold (Umbral de estado incorrecto)** es una configuración de solo lectura que siempre es igual al valor de umbral en buen estado).
- **Deregistration delay (Retraso de anulación del registro):** intervalo de tiempo, en segundos, que debe esperarse a que se completen las solicitudes activas antes de anular el registro de una instancia.

Health check

Interval
Amount of time between health checks of an individual instance.

seconds

Healthy threshold
The number of consecutive successful health checks required to designate the instance as healthy.

 requests

Unhealthy threshold
The number of consecutive health check failures required to designate the instance as unhealthy.

 requests

Deregistration delay
Amount of time to wait for active requests to complete before deregistering.

 seconds

Note

La comprobación de estado de Elastic Load Balancing no afecta al comportamiento de comprobación de estado del grupo de Auto Scaling de un entorno. Las instancias que fallan en una comprobación de estado de Elastic Load Balancing no serán reemplazadas automáticamente por Amazon EC2 Auto Scaling, a menos que configure manualmente Amazon EC2 Auto Scaling para hacerlo. Para obtener más información, consulte [Configuración de comprobaciones de estado de Auto Scaling](#).

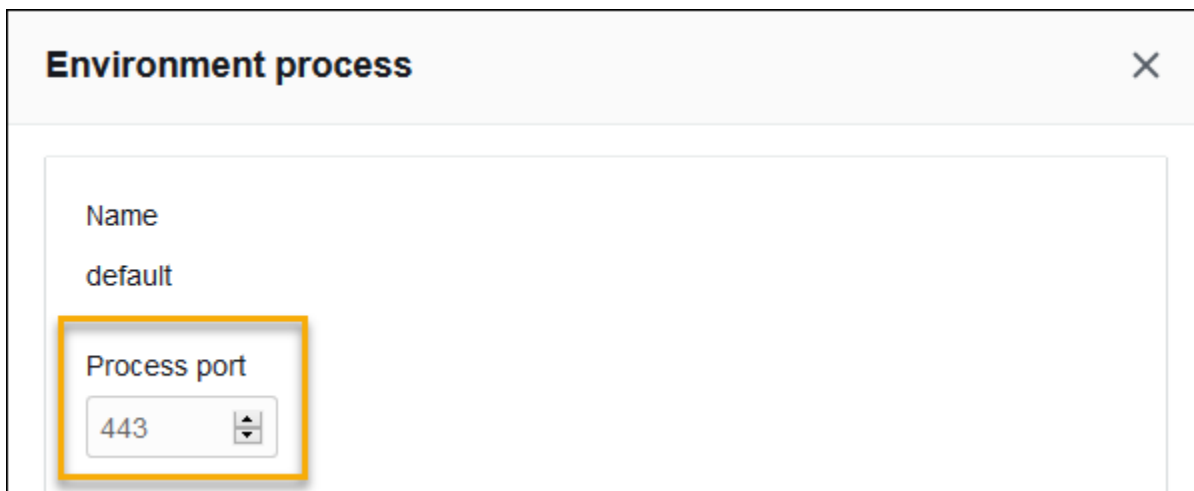
Para obtener más información sobre las comprobaciones de estado y la forma en que influyen en el estado general del entorno, consulte [Informes de estado básicos](#).

Ejemplo: balanceador de carga de red para un entorno con cifrado de extremo a extremo

En este ejemplo, la aplicación requiere el cifrado del tráfico integral. Para configurar el balanceador de carga de red de su entorno para satisfacer estos requisitos, configure el proceso predeterminado para escuchar el puerto 443, agregue un agente de escucha al puerto 443 que redirija el tráfico al proceso predeterminado y deshabilite el agente de escucha predeterminado.

Para configurar el balanceador de carga para este ejemplo

1. Configuración del proceso predeterminado. Seleccione el proceso predeterminado y, a continuación, en Actions (Acciones), elija Edit (Editar). En Process port (Puerto del proceso), escriba 443.



2. Adición de un agente de escucha de puerto 443. Añada un nuevo agente de escucha. En Listener port (Puerto de agente de escucha), escriba 443. Para Process port (Puerto del proceso), asegúrese de que 443 está seleccionado.

Network Load Balancer listener ✕

Listener port

443

Protocol
The transport protocol that the load balancer uses for routing incoming traffic from clients.

TCP

Process port
The port to which this listener routes traffic. It determines the environment process that receives traffic from the listener.

443

Ahora puede ver el agente de escucha adicional en la lista.

<input type="checkbox"/>	Listener port	Process port	Protocol	Enabled
<input type="checkbox"/>	80	443	TCP	<input checked="" type="checkbox"/>
<input type="checkbox"/>	443	443	TCP	<input checked="" type="checkbox"/>

3. Deshabilitar el agente de escucha predeterminado en el puerto 80. Para el agente de escucha predeterminado, desactive la opción Enabled (Habilitado).

<input type="checkbox"/>	Listener port	Process port	Protocol	Enabled
<input type="checkbox"/>	80	443	TCP	<input type="checkbox"/>
<input type="checkbox"/>	443	443	TCP	<input checked="" type="checkbox"/>

Configuración de un balanceador de carga de red mediante la CLI de EB

La CLI de EB le pide que elija el tipo de balanceador de carga cuando se ejecuta [eb create](#).

```
$ eb create
Enter Environment Name
(default is my-app): test-env
Enter DNS CNAME prefix
(default is my-app): test-env-DLW24ED23SF

Select a load balancer type
1) classic
2) application
3) network
(default is 1): 3
```

También puede especificar un tipo de balanceador de carga con la opción `--elb-type`.

```
$ eb create test-env --elb-type network
```

Espacios de nombres del balanceador de carga de red

Puede encontrar la configuración relacionada con balanceadores de carga de red en los siguientes espacios de nombres:

- [aws:elasticbeanstalk:environment](#): elija el tipo de balanceador de carga para el entorno. El valor de un balanceador de carga de red es `network`.
- [aws:elbv2:listener](#) — Configure los oyentes en el balanceador de carga de red. Estos valores se corresponden con los de `aws:elb:listener` en los balanceadores de carga clásicos.
- [aws:elasticbeanstalk:environment:process](#): configure las comprobaciones de estado y especifique el puerto y el protocolo de los procesos que se ejecutan en las instancias del entorno. Los valores del puerto y del protocolo se asignan a los valores del puerto y del protocolo de la instancia especificados en `aws:elb:listener` para un agente de escucha de un balanceador de tipo Classic Load Balancer. Los valores de la comprobación de estado se corresponden con los valores de los espacios de nombres `aws:elb:healthcheck` y `aws:elasticbeanstalk:application`.

Example .ebextensions/network-load-balancer.config

Para comenzar a utilizar un balanceador de carga de red, utilice un [archivo de configuración](#) para establecer el tipo de balanceador de carga en network.

```
option_settings:
  aws:elasticbeanstalk:environment:
    LoadBalancerType: network
```

Note

Solo puede establecer el tipo de balanceador de carga durante la creación del entorno.

Example .ebextensions/nlb-default-process.config

El siguiente archivo de configuración modifica los ajustes de comprobación de estado del proceso predeterminado.

```
option_settings:
  aws:elasticbeanstalk:environment:process:default:
    DeregistrationDelay: '20'
    HealthCheckInterval: '10'
    HealthyThresholdCount: '5'
    UnhealthyThresholdCount: '5'
    Port: '80'
    Protocol: TCP
```

Example .ebextensions/nlb-secure-listener.config

El siguiente archivo de configuración agrega un agente de escucha para el tráfico seguro en el puerto 443 y un proceso de destino coincidente que escucha en el puerto 443.

```
option_settings:
  aws:elbv2:listener:443:
    DefaultProcess: https
    ListenerEnabled: 'true'
  aws:elasticbeanstalk:environment:process:https:
    Port: '443'
```

La opción `DefaultProcess` se denomina de esta forma debido a los balanceadores de carga de aplicaciones, que pueden tener agentes de escucha no predeterminados en el mismo puerto correspondiente al tráfico a rutas específicas (consulte [Equilibrador de carga de aplicación](#) para obtener más información). Para un Network Load Balancer, la opción especifica el único proceso de destino de este agente de escucha.

En este ejemplo, hemos denominado al proceso `https` porque escucha el tráfico seguro (HTTPS). El agente de escucha envía el tráfico al proceso en el puerto designado mediante el protocolo TCP, porque un Network Load Balancer solo funciona con TCP. Esto es correcto, porque el tráfico de red HTTP y HTTPS está implementado sobre TCP.

Configuración de registros de acceso

Puede utilizar [archivos de configuración](#) para configurar el balanceador de carga de su entorno de forma que cargue los logs de acceso en un bucket de Amazon S3. Consulte los siguientes archivos de configuración de ejemplo en GitHub para obtener instrucciones:

- [loadbalancer-accesslogs-existingbucket.config](#): configurar el balanceador de carga para cargar logs de acceso en un bucket de Amazon S3 existente.
- [loadbalancer-accesslogs-newbucket.config](#): configurar el balanceador de carga para cargar registros de acceso en un nuevo bucket.

Añadir una base de datos al entorno de Elastic Beanstalk

Elastic Beanstalk provee integración con [Amazon Relational Database Service \(Amazon RDS\)](#). Puede utilizar Elastic Beanstalk para agregar una base de datos de MySQL, PostgreSQL, Oracle o SQL Server a un entorno existente o a uno nuevo durante la creación del entorno o después de haberlo creado. Cuando agrega una instancia de base de datos, Elastic Beanstalk provee la información de conexión a la aplicación. Para ello, establece las propiedades del entorno para el nombre de host de la base de datos, el puerto, el nombre de usuario, la contraseña y el nombre de base de datos.

Si no ha usado una instancia de base de datos con su aplicación anteriormente, le recomendamos que utilice primero el proceso descrito en este tema para agregar una base de datos a un entorno de prueba utilizando el servicio Elastic Beanstalk. De este modo, puede verificar que la aplicación puede leer las propiedades del entorno, crear una cadena de conexión y conectarse a una instancia de base de datos, sin el trabajo de configuración adicional que una base de datos externa a Elastic Beanstalk necesita.

Después de comprobar que la aplicación funciona correctamente con la base de datos, puede considerar la posibilidad de avanzar hacia un entorno de producción. Aquí tiene la opción de desacoplar la base de datos de su entorno de Elastic Beanstalk para avanzar hacia una configuración que ofrezca mayor flexibilidad. La base de datos desacoplada puede permanecer en funcionamiento como una instancia de base de datos externa de Amazon RDS. El estado del entorno no se ve afectado por el desacoplamiento de la base de datos. Si necesita terminar el entorno, puede hacerlo y elegir también la opción de mantener la base de datos disponible y en funcionamiento fuera de Elastic Beanstalk.

Utilizar una base de datos externa tiene varias ventajas. Puede conectarse a la base de datos externa desde varios entornos, utilizar tipos de bases de datos que no son compatibles con bases de datos integradas y realizar implementaciones azul-verde. Como alternativa al uso de una base de datos desacoplada creada por Elastic Beanstalk, también puede crear una instancia de base de datos fuera del entorno de Elastic Beanstalk. Ambas opciones dan como resultado una instancia de base de datos que es externa a su entorno de Elastic Beanstalk y requieren un grupo de seguridad adicional y la configuración de la cadena de conexión. Para obtener más información, consulte [Uso de Elastic Beanstalk con Amazon RDS](#).

Secciones

- [Ciclo de vida de la base de datos](#)
- [Agregar una instancia de base de datos de Amazon RDS al entorno utilizando la consola](#)
- [Conexión a la base de datos](#)
- [Configuración de una instancia de base de datos de RDS integrada mediante la consola](#)
- [Configuración de una instancia de base de datos de RDS integrada utilizando archivos de configuración](#)
- [Desacoplamiento de una instancia de base de datos de RDS utilizando la consola](#)
- [Desacoplamiento de una instancia de base de datos de RDS utilizando archivos de configuración](#)

Ciclo de vida de la base de datos

Puede elegir lo que desea que le ocurra a la base de datos después de desacoplarla del entorno de Elastic Beanstalk. Las opciones entre las que puede elegir se denominan colectivamente como políticas de eliminación. Las siguientes políticas de eliminación se aplican a una base de datos después de [desacoplarla de un entorno de Elastic Beanstalk](#) o de que se termine el entorno de Elastic Beanstalk.

- **Instantánea:** antes de que Elastic Beanstalk termine la base de datos, guarda una instantánea de la misma. Puede restaurar una base de datos a partir de una instantánea cuando agregue una instancia de base de datos al entorno de Elastic Beanstalk o cuando cree una base de datos independiente. Para obtener más información acerca de la creación de una nueva instancia de base de datos independiente a partir de una instantánea, consulte [Restauración desde una instantánea de base de datos](#) en la Guía del usuario de Amazon RDS. El almacenamiento de instantáneas de bases de datos podría generar cargos. Para obtener más información, consulte la sección Almacenamiento de copias de seguridad de [Precios de Amazon RDS](#).
- **Eliminar:** Elastic Beanstalk termina la base de datos. Una vez terminada, la instancia de base de datos ya no está disponible para ninguna operación.
- **Retener:** la instancia de base de datos no se termina. Sigue disponible y en funcionamiento aunque está desacoplada de Elastic Beanstalk. A continuación, puede configurar uno o varios entornos para conectarse a la base de datos como una instancia de base de datos externa de Amazon RDS. Para obtener más información, consulte [Uso de Elastic Beanstalk con Amazon RDS](#).

Agregar una instancia de base de datos de Amazon RDS al entorno utilizando la consola

Puede añadir una instancia de base de datos al entorno mediante la consola de Elastic Beanstalk.

Para agregar una instancia de base de datos al entorno

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Database (Base de datos), seleccione Edit (Editar).
5. Seleccione un motor de base de datos y escriba el nombre de usuario y la contraseña.
6. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

Puede configurar las siguientes opciones:

- **Snapshot (Instantánea):** seleccione una instantánea de base de datos existente. Elastic Beanstalk restablece la instantánea y la añade a su entorno. El valor predeterminado es None (Ninguno). Cuando el valor predeterminado es None (Ninguno), puede configurar una base de datos nueva utilizando el resto de la configuración de esta página.
- **Engine (Motor):** seleccione un motor de base de datos.
- **Engine version (Versión del motor):** elija una versión específica del motor de base de datos.
- **Instance class (Clase de instancia):** seleccione la clase de instancia de base de datos. Para obtener más información sobre clases de instancias de base de datos, consulte <https://aws.amazon.com/rds/>.
- **Storage (Almacenamiento):** seleccione el espacio de almacenamiento que va a aprovisionar para la base de datos. Posteriormente, podrá aumentar el almacenamiento asignado, pero no reducirlo. Para obtener información acerca de la asignación de almacenamiento, consulte las [características](#).
- **Username (Nombre de usuario):** ingrese el nombre de usuario que quiera combinando solo números y letras.
- **Password (Contraseña):** introduzca la contraseña que desee. Debe incluir entre 8 y 16 caracteres ASCII imprimibles (menos /, \ y @).
- **Availability (Disponibilidad):** seleccione High (Multi-AZ) (Alta [múltiples zonas de disponibilidad]) para ejecutar una "warm backup" en una segunda zona de disponibilidad y contar con una alta disponibilidad.
- **Database deletion policy (Política de eliminación de base de datos):** la política de eliminación determina lo que ocurre con la base de datos después de que se [desacople](#) de su entorno. Se puede establecer en los siguientes valores: Create Snapshot, Retain o Delete. Estos valores se describen en [Ciclo de vida de la base de datos](#) en este mismo tema.

Note

Elastic Beanstalk crea un usuario maestro para la base de datos con el nombre de usuario y la contraseña que proporcione. Para obtener más información sobre el usuario maestro y sus privilegios, consulte [Privilegios de la cuenta de usuario maestro](#).

Se tarda alrededor de 10 minutos en agregar una instancia de base de datos. Cuando finaliza la actualización, la nueva base de datos está acoplada a su entorno. El nombre de host y otra

información de conexión de la instancia de base de datos están disponibles para su aplicación mediante las siguientes propiedades de entorno.

Nombre de la propiedad	Descripción	Valor de la propiedad
RDS_HOSTNAME	El nombre de host de la instancia de base de datos.	En la pestaña Connectivity & security (Conectividad y seguridad) de la consola de Amazon RDS: Endpoint (Punto de enlace).
RDS_PORT	El puerto en donde la instancia de base de datos acepta las conexiones. El valor predeterminado varía dependiendo del motor de base de datos.	En la pestaña Connectivity & security (Conectividad y seguridad) de la consola de Amazon RDS: Port (Puerto).
RDS_DB_NAME	El nombre de la base de dato, ebdb .	En la pestaña Configuration (Configuración) de la consola de Amazon RDS: DB Name (Nombre de la base de datos).
RDS_USERNAME	El nombre de usuario que ha configurado para la base de datos.	En la pestaña Configuration (Configuración) de la consola de Amazon RDS: Master username (Nombre de usuario maestro).
RDS_PASSWORD	La contraseña que ha configurado para la base de datos.	No está disponible como referencia en la consola de Amazon RDS.

Conexión a la base de datos

Utilice la información de conectividad para conectarse a la base de datos desde la aplicación a través de variables de entorno. Para obtener más información sobre el uso de Amazon RDS con las aplicaciones, consulte los siguientes temas.

- Java SE: [Conexión a una base de datos \(plataformas Java SE\)](#)
- Java con Tomcat: [Conexión a una base de datos \(plataformas Tomcat\)](#)
- Node.js: [Conexión a una base de datos](#)
- .NET: [Conexión a una base de datos](#)
- PHP: [Conexión a una base de datos con PDO o MySQLi](#)
- Python: [Conexión a una base de datos](#)
- Ruby: [Conexión a una base de datos](#)

Configuración de una instancia de base de datos de RDS integrada mediante la consola

Puede ver y modificar los valores de configuración de la instancia de base de datos en la sección Database (Base de datos) de la página Configuration (Configuración) del entorno en la [consola de Elastic Beanstalk](#).

Para configurar la instancia de base de datos del entorno en la consola de Elastic Beanstalk

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Database (Base de datos), seleccione Edit (Editar).

Puede modificar la configuración de Instance class (Clase de instancia), Storage (Almacenamiento), Password (Contraseña), Availability (Disponibilidad) y Database deletion policy (Política de eliminación de la base de datos) después de la creación de la base de datos. Si cambia la clase de instancia, Elastic Beanstalk vuelve a aprovisionar la instancia de base de datos.

Si ya no necesita Elastic Beanstalk para asociar la base de datos al entorno, puede elegir desacoplarla seleccionando Decouple database (Desacoplar base de datos). Es importante entender las opciones y consideraciones que implican esta operación. Para obtener más información, consulte [the section called “Desacoplamiento de una instancia de base de datos de RDS utilizando la consola”](#).

Advertencia

No modifique la configuración de la instancia de base de datos acoplada con otra funcionalidad distinta que la que proporciona Elastic Beanstalk (por ejemplo, en la consola de Amazon RDS). Si lo hace, la configuración de base de datos de Amazon RDS podría no estar sincronizada con la definición del entorno. Al actualizar o reiniciar su entorno, la configuración especificada en el entorno sustituye cualquier configuración que ha realizado fuera de Elastic Beanstalk.

Si necesita modificar la configuración que Elastic Beanstalk no admite directamente, utilice los [archivos de configuración](#) de Elastic Beanstalk.

Configuración de una instancia de base de datos de RDS integrada utilizando archivos de configuración

Puede configurar la instancia de base de datos del entorno mediante [archivos de configuración](#). Utilice las opciones del espacio de nombres `aws:rds:dbinstance`. En el siguiente ejemplo se modifica el tamaño de almacenamiento de base de datos asignado a 100 GB.

Example `.ebextensions/db-instance-options.config`

```
option_settings:
  aws:rds:dbinstance:
    DBAllocatedStorage: 100
```

Si desea configurar propiedades de la instancia de base de datos que Elastic Beanstalk no admite, puede seguir utilizando un archivo de configuración y especificar la configuración con la

clave `resources`. En el siguiente ejemplo se establecen los valores en las propiedades `Iops` y `StorageType` de Amazon RDS.

Example `.ebextensions/db-instance-properties.config`

```
Resources:
  AWSEBRDSDatabase:
    Type: AWS::RDS::DBInstance
    Properties:
      StorageType: io1
      Iops: 1000
```

Desacoplamiento de una instancia de base de datos de RDS utilizando la consola

Puede desacoplar la base de datos de un entorno de Elastic Beanstalk sin que esto afecte al estado del entorno. Tenga en cuenta los siguientes requisitos antes de desacoplar la base de datos:

- ¿Qué debería pasar con la base de datos después de desacoplarla?

Puede elegir entre crear una instantánea de la base de datos para luego terminarla, retenerla en funcionamiento como una base de datos independiente externa a Elastic Beanstalk o eliminar permanentemente la base de datos. La configuración de la Database deletion policy (Política de eliminación de base de datos) determina este resultado. Para obtener una descripción detallada de las políticas de eliminación, consulte [Ciclo de vida de la base de datos](#) en este mismo tema.


- ¿Necesita realizar algún cambio en la configuración de la base de datos antes de desacoplarla?

Si necesita realizar algún cambio de configuración en la base de datos, debe aplicarlos antes de desacoplarla. Esto incluye los cambios en la Database deletion policy (Política de eliminación de base de datos). Cualquier cambio pendiente que se envíe simultáneamente con la configuración Decouple database (Desacoplar base de datos) se ignorará, mientras que solo se aplica la configuración de desacoplamiento.

Para desacoplar una instancia de base de datos de un entorno

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.

2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

 Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Database (Base de datos), seleccione Edit (Editar).
5. Revise todos los valores de configuración de la sección Database settings (Configuración de la base de datos), especialmente la Database deletion policy (Política de eliminación de base de datos), que determina qué sucede con la base de datos una vez desacoplada.

Database settings

Choose an engine and instance type for your environment's database.

Engine
mysql

Engine version
--

Instance class
db.t2.micro

Storage
Choose a number between 5 GB and 1024 GB.
5

Username
test

Password

Availability
Low (one AZ)

Database deletion policy
This policy applies when you decouple a database or terminate the environment coupled to it.

Create snapshot
Elastic Beanstalk saves a snapshot of the database and then deletes it. You can restore a database from a snapshot when you add a DB to an Elastic Beanstalk environment or when you create a standalone database. You might incur charges for storing database snapshots.

Retain
The decoupled database will remain available and operational external to Elastic Beanstalk.

Delete
Elastic Beanstalk terminates the database. The database will no longer be available.

Cancel Continue Apply

Si todos los demás ajustes de configuración son correctos, vaya al Paso 6 para desacoplar la base de datos.

⚠ Warning

Es importante aplicar la configuración de Database deletion policy (Política de eliminación de base de datos) por separado de Decouple database (Desacoplar la base de datos). Si selecciona aplicar con la intención de guardar ambos Desacoplar la base de datos (Desacoplar base de datos) y una Database deletion policy (Política de eliminación de base de datos) seleccionada recientemente, la nueva política de eliminación que haya elegido será ignorada. Elastic Beanstalk desacoplará la base de datos siguiendo la política de eliminación establecida anteriormente. Si la política de eliminación establecida anteriormente es Delete o Create Snapshot, corre el riesgo de perder la base de datos en lugar de seguir la política pendiente prevista.

Si alguno de los ajustes de configuración requiere una actualización, haga lo siguiente:

1. Realice las modificaciones necesarias en el panel Database settings (Configuración de la base de datos).
2. Seleccione Apply (Aplicar). Tarda unos minutos en guardar los cambios de configuración de la base de datos.
3. Vuelva al Paso 3 y elija Configuration (Configuración) en el panel de navegación.
6. Vaya a la sección del panel Database connection (Conexión de la base de datos).

Database connection

Environment/database connection
Add a database to your environment or decouple an existing database from it.

Couple database
Elastic Beanstalk creates a database coupled to your environment. If you terminate an environment with a coupled database, the database lifecycle follows the deletion policy that you choose.

Decouple database
The database is decoupled from your environment. Decoupling a database doesn't affect the health of your environment. The database follows the deletion policy that you chose.

7. Elegir Decouple database (Desacoplar base de datos).
8. Elegir Apply (Aplicar) para iniciar la operación de desacoplamiento de la base de datos.

La configuración de la política de eliminación determina el resultado de la base de datos y el tiempo necesario para desacoplar la base de datos.

- Si la política de eliminación está establecida en `Delete`, la base de datos se elimina. La operación puede durar aproximadamente 10 a 20 minutos, dependiendo del tamaño de la base de datos.
- Si la política de eliminación está establecida en `Snapshot`, se crea una instantánea de la base de datos. A continuación, la base de datos se elimina. El tiempo necesario para este proceso varía de acuerdo con el tamaño de la base de datos.
- Si la política de eliminación está establecida en `Retain`, la base de datos permanece en funcionamiento de manera externa al entorno de Elastic Beanstalk. Por lo general, se tarda menos de cinco minutos en desacoplar una base de datos.

Si ha decidido retener la base de datos externa a su entorno de Elastic Beanstalk, deberá adoptar nuevas medidas para configurarla. Para obtener más información, consulte [Uso de Elastic Beanstalk con Amazon RDS](#). Si planea utilizar la base de datos que desacopla para un entorno de producción, verifique que el tipo de almacenamiento que utiliza la base de datos es el adecuado para su carga de trabajo. Para obtener más información, consulte [Almacenamiento de instancias de base de datos](#) y [Modificación de una instancia de base de datos](#) en la Guía del usuario de Amazon RDS.

Desacoplamiento de una instancia de base de datos de RDS utilizando archivos de configuración

Puede desacoplar la instancia de base de datos de un entorno de Elastic Beanstalk sin afectar el estado del entorno. La instancia de base de datos sigue la política de eliminación de base de datos que se aplicó cuando se desacopló la base de datos.

Ambas opciones necesarias para desacoplar la base de datos se encuentran en el espacio de nombres [the section called “aws:rds:dbinstance”](#). Son los siguientes:

- La opción `DBDeletionPolicy` establece la política de eliminación. Se puede establecer en los siguientes valores: `Snapshot`, `Delete` o `Retain`. Estos valores se describen en [Ciclo de vida de la base de datos](#) en este mismo tema.
- La opción `HasCoupledDatabase` determina si el entorno tiene una base de datos acoplada.
 - Si se cambia a `true`, Elastic Beanstalk crea una nueva instancia de base de datos acoplada a su entorno.

- Si se cambia a `false`, Elastic Beanstalk comienza a desacoplar la instancia de base de datos de su entorno.

Si desea cambiar la configuración de la base de datos antes de desacoplarla, primero aplique los cambios de configuración en una operación independiente. Esto incluye cambiar la configuración de `DBDeletionPolicy`. Después de aplicar los cambios, ejecute un comando independiente para configurar la opción de desacoplamiento. Si envía otros ajustes de configuración y la configuración de desacoplamiento al mismo tiempo, los otros ajustes de configuración se ignoran y solo se aplica la configuración de desacoplamiento.

Warning

Es importante que ejecute los comandos para aplicar las configuraciones `DBDeletionPolicy` y `HasCoupledDatabase` como dos operaciones independientes. Si la política de eliminación activa ya está establecida en `Delete` o `Snapshot`, corre el riesgo de perder la base de datos. La base de datos cumple con la política de eliminación que está activa actualmente, en lugar de con la política de eliminación pendiente que pretendía.

Para desacoplar una instancia de base de datos de un entorno

Siga estos pasos para desacoplar la base de datos de su entorno de Elastic Beanstalk. Puede utilizar la CLI de EB o la AWS CLI para completar los pasos. Para obtener más información, consulte [Personalización avanzada del entorno con archivos de configuración](#).

1. Si desea cambiar la política de eliminación, configure un archivo de configuración en el siguiente formato. En este ejemplo, la política de eliminación se establece para retener.

Example

```
option_settings:
  aws:rds:dbinstance:
    DBDeletionPolicy: Retain
```

2. Ejecute el comando utilizando la herramienta que prefiera para completar la actualización de la configuración.
3. Configure un archivo de configuración para definir `HasCoupledDatabase` como `false`.

Example

```
option_settings:
  aws:rds:dbinstance:
    HasCoupledDatabase: false
```

4. Ejecute el comando utilizando la herramienta que prefiera para completar la actualización de la configuración.

La configuración de la política de eliminación determina el resultado de la base de datos y el tiempo necesario para desacoplar la base de datos.

- Si la política de eliminación está establecida en `Delete`, la base de datos se elimina. La operación puede durar aproximadamente 10 a 20 minutos, dependiendo del tamaño de la base de datos.
- Si la política de eliminación está establecida en `Snapshot`, se crea una instantánea de la base de datos. A continuación, la base de datos se elimina. El tiempo necesario para este proceso varía de acuerdo con el tamaño de la base de datos.
- Si la política de eliminación está establecida en `Retain`, la base de datos permanece en funcionamiento de manera externa al entorno de Elastic Beanstalk. Por lo general, se tarda menos de cinco minutos en desacoplar una base de datos.

Si ha decidido retener la base de datos externa a su entorno de Elastic Beanstalk, deberá adoptar nuevas medidas para configurarla. Para obtener más información, consulte [Uso de Elastic Beanstalk con Amazon RDS](#). Si planea utilizar la base de datos que desacopla para un entorno de producción, verifique que el tipo de almacenamiento que utiliza la base de datos es el adecuado para su carga de trabajo. Para obtener más información, consulte [Almacenamiento de instancias de base de datos](#) y [Modificación de una instancia de base de datos](#) en la Guía del usuario de Amazon RDS.

La seguridad de su AWS Elastic Beanstalk entorno

Elastic Beanstalk proporciona varias opciones que controlan el acceso al servicio (seguridad) de su entorno y de las instancias de Amazon EC2 en este. Este tema trata la configuración de estas opciones.

Secciones

- [Configuración de la seguridad de su entorno](#)

- [Espacios de nombres de configuración de seguridad del entorno](#)

Configuración de la seguridad de su entorno

Puede modificar la configuración de seguridad del entorno de Elastic Beanstalk en la consola de Elastic Beanstalk.

Para configurar el acceso al servicio (seguridad) del entorno en la consola de Elastic Beanstalk

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Service access (Acceso al servicio), elija Edit (Editar).

Dispone de las siguientes opciones.

Configuración

- [Rol de servicio](#)
- [EC2 key pair](#)
- [Perfil de instancia IAM](#)

Elastic Beanstalk > Environments > Gettingstarted-env > Configuration

Configure service access [Info](#)

Service access

IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profiles allow Elastic Beanstalk to create and manage your environment. Both the IAM role and instance profile must be attached to IAM managed policies that contain the required permissions. [Learn more](#)

Service role

Choose an existing IAM role for Elastic Beanstalk to assume as a service role. The existing IAM role must have the required IAM managed policies.

aws-elasticbeanstalk-service-role

EC2 key pair

Select an EC2 key pair to securely log in to your EC2 instances. [Learn more](#)

Choose a key pair

EC2 instance profile

Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.

aws-elasticbeanstalk-ec2-role

Rol de servicio

Seleccione un [rol de servicio](#) para asociar con su entorno de Elastic Beanstalk. Elastic Beanstalk asume la función de servicio cuando AWS accede a otros servicios en su nombre. Para obtener más detalles, consulte [Administración de roles de servicio de Elastic Beanstalk](#).

EC2 key pair

Puede iniciar sesión de forma segura en las instancias de Amazon Elastic Compute Cloud (Amazon EC2) aprovisionadas para su aplicación de Elastic Beanstalk con un par de claves de Amazon EC2. Para obtener instrucciones sobre la creación de un par de claves, consulte [Creación de un par de claves con Amazon EC2](#) en la Guía del usuario de Amazon EC2.

Note

Al crear un par de claves, Amazon EC2 almacena una copia de la clave pública. Si ya no la necesita para conectarse a cualquier instancia de entorno, puede eliminarla de Amazon EC2. Para obtener más información, [consulte Eliminar un par de claves](#) en la Guía del usuario de Amazon EC2.

Elija un par de claves de EC2 en el menú desplegable para asignarlo a las instancias del entorno. Cuando asigna un par de claves, la clave pública se guarda en la instancia para autenticar la clave privada, que se guarda en local. La clave privada nunca se almacena en AWS.

Para obtener más información sobre la conexión a instancias de Amazon EC2, consulte [Conectarse a su instancia y Conectarse a instancias de Linux/UNIX desde Windows mediante PuTTY en la Guía del usuario de Amazon EC2](#).

Perfil de instancia IAM

Un [perfil de instancia](#) de EC2 es un rol de IAM que se aplica a las instancias lanzadas en el entorno de Elastic Beanstalk. Las instancias de Amazon EC2 asumen la función de perfil de instancia para firmar las solicitudes AWS y acceder a las API, por ejemplo, para cargar registros en Amazon S3.

La primera vez que crea un entorno en la consola de Elastic Beanstalk, Elastic Beanstalk le pide que cree un perfil de instancia con un conjunto de permisos predefinidos. Puede añadir permisos a este perfil para proporcionar a sus instancias acceso a otros AWS servicios. Para obtener más detalles, consulte [Administración de perfiles de instancia de Elastic Beanstalk](#).

Note

Anteriormente, Elastic Beanstalk creaba un `aws-elasticbeanstalk-ec2-role` perfil de instancia EC2 predeterminado denominado la primera vez AWS que una cuenta creaba un entorno. Este perfil de instancia incluía políticas administradas predeterminadas. Si su cuenta ya tiene este perfil de instancia, seguirá estando disponible para que lo asigne a sus entornos.

Sin embargo, las directrices AWS de seguridad recientes no permiten que un AWS servicio cree funciones automáticamente con políticas de confianza para otros AWS servicios, en este caso EC2. Debido a estas directrices de seguridad, Elastic Beanstalk ya no crea un perfil de instancia `aws-elasticbeanstalk-ec2-role` predeterminado.

Espacios de nombres de configuración de seguridad del entorno

Elastic Beanstalk proporciona [opciones de configuración](#) en los espacios de nombres siguientes para permitirle personalizar la seguridad de su entorno:

- [aws:elasticbeanstalk:environment](#): configure el rol de servicio del entorno utilizando la opción `ServiceRole`.

- [aws:autoscaling:launchconfiguration](#): configure permisos para las instancias de Amazon EC2 del entorno utilizando las opciones `EC2KeyName` e `IamInstanceProfile`.

La CLI de EB y la consola de Elastic Beanstalk aplican los valores recomendados a las opciones anteriores. Debe eliminar estos ajustes si desea usar archivos de configuración para configurarlos. Para obtener más información, consulte [Valores recomendados](#).

Etiquetado de recursos en sus entornos de Elastic Beanstalk

Puede aplicar etiquetas a sus AWS Elastic Beanstalk entornos. Las etiquetas son pares clave-valor asociados AWS a los recursos. Para obtener información sobre el etiquetado de recursos de Elastic Beanstalk, los casos de uso, las restricciones de las claves y los valores de las etiquetas y los tipos de recursos admitidos, consulte [Etiquetar recursos de la aplicación Elastic Beanstalk](#).

Elastic Beanstalk aplica etiquetas de entorno al propio recurso del entorno, así como a otros AWS recursos que Elastic Beanstalk cree para el entorno. También puede usar las etiquetas para administrar los permisos en el nivel de recurso específico dentro de un entorno. Para obtener más información, [consulte Etiquetado de los recursos de Amazon EC2](#) en la Guía del usuario de Amazon EC2.

De forma predeterminada, Elastic Beanstalk aplica algunas etiquetas a su entorno:

- `elasticbeanstalk:environment-name`: el nombre del entorno.
- `elasticbeanstalk:environment-id`: el ID del entorno.
- `Name`: también es el nombre del entorno. `Name` se utiliza en el panel de Amazon EC2 para identificar y clasificar recursos.

No puede editar estas etiquetas predeterminadas.

Puede especificar etiquetas cuando crea el entorno de Elastic Beanstalk. En un entorno existente, puede añadir o eliminar etiquetas, y actualizar los valores de etiquetas existentes. Un entorno puede tener hasta 50 etiquetas, incluidas las etiquetas predeterminadas.

Adición de etiquetas durante la creación del entorno

Al utilizar la consola de Elastic Beanstalk para crear un entorno, puede especificar claves y valores de etiquetas en la página de configuración `Modify tags` (Modificar etiquetas) del [asistente de creación de nuevo entorno](#).

Elastic Beanstalk > Applications > getting-started-app

Modify tags

Apply up to 50 tags to the resources in your environment in addition to the default tags.

Key	Value	
<input type="text" value="mytag1"/>	<input type="text" value="value1"/>	<input type="button" value="Remove"/>

49 remaining

Si utiliza la CLI de EB para crear un entorno, use la opción `--tags` con [eb create](#) para añadir etiquetas.

```
~/workspace/my-app$ eb create --tags mytag1=value1,mytag2=value2
```

Con este AWS CLI u otros clientes basados en API, utilice el `--tags` parámetro del comando. [create-environment](#)

```
$ aws elasticbeanstalk create-environment \
  --tags Key=mytag1,Value=value1 Key=mytag2,Value=value2 \
  --application-name my-app --environment-name my-env --cname-prefix my-app --
  version-label v1 --template-name my-saved-config
```

Las [configuraciones guardadas](#) incluyen etiquetas definidas por el usuario. Cuando aplique una configuración guardada que contiene etiquetas durante la creación del entorno, las etiquetas se aplican al nuevo entorno, siempre y cuando no especifique nuevas etiquetas. Si añade etiquetas a un entorno utilizando uno de los métodos anteriores, se descartan las etiquetas definidas en la configuración guardada.

Administración de etiquetas de un entorno existente

Puede añadir, actualizar y eliminar etiquetas en un entorno de Elastic Beanstalk existente. Elastic Beanstalk aplica los cambios a los recursos de su entorno.

Sin embargo, no es posible editar las etiquetas predeterminadas que Elastic Beanstalk aplica a cada entorno.

Para administrar las etiquetas de un entorno en la consola de Elastic Beanstalk

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, seleccione Tags.

La página de administración de etiquetas muestra la lista de etiquetas existentes actualmente en el entorno.

The screenshot shows the AWS Elastic Beanstalk console interface for managing tags on an environment named 'GettingStartedApp-env'. The breadcrumb navigation is 'Elastic Beanstalk > Environments > GettingStartedApp-env > Tags'. The main heading is 'Tags for GettingStartedApp-env'. Below the heading is a descriptive text: 'Apply up to 47 tags in addition to the default tags to the resources in your environment. You can use tags to group and filter your environments. A tag is a key-value pair. The key must be unique within the environment and is case-sensitive. [Learn more](#)'. The interface features a table with two columns: 'Key' and 'Value'. The first two rows show default tags: 'elasticbeanstalk:environment-id' with value 'e-cubmdjm6ga' and 'elasticbeanstalk:environment-name' with value 'GettingStartedApp-env'. Below these, there are two rows of user-defined tags: 'mytag1' with value 'value1' and 'mytag2' with value 'value2'. Each user-defined tag row includes a 'Remove' button. At the bottom left, there is an 'Add tag' button and a counter indicating '45 remaining' tags. At the bottom right, there are 'Cancel' and 'Apply' buttons.

Key	Value	
elasticbeanstalk:environment-id	e-cubmdjm6ga	
elasticbeanstalk:environment-name	GettingStartedApp-env	
Name	GettingStartedApp-env	
mytag1	value1	Remove
mytag2	value2	Remove

4. Añadir, actualizar o eliminar etiquetas:
 - Para añadir una etiqueta, escríbala en las casillas vacías de la parte inferior de la lista. Para añadir otra etiqueta, elija Add tag (Añadir etiqueta) y Elastic Beanstalk añadirá otro par de casillas vacías.

- Para actualizar la clave o el valor de una etiqueta, edite la casilla correspondiente en la fila de la etiqueta.
 - Para eliminar una etiqueta, elija Remove (Eliminar) junto a la casilla de valor de la etiqueta.
5. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

Si utiliza la CLI de EB para actualizar su entorno, utilice [eb tags](#) para añadir, actualizar, eliminar o enumerar etiquetas.

Por ejemplo, el siguiente comando enumera las etiquetas en el entorno predeterminado.

```
~/workspace/my-app$ eb tags --list
```

El siguiente comando actualiza la etiqueta mytag1 y elimina la etiqueta mytag2.

```
~/workspace/my-app$ eb tags --update mytag1=newvalue --delete mytag2
```

Para obtener una lista de las opciones y más ejemplos, consulte [eb tags](#).

Con este AWS CLI u otros clientes basados en API, utilice el [list-tags-for-resource](#) comando para enumerar las etiquetas de un entorno.

```
$ aws elasticbeanstalk list-tags-for-resource --resource-arn  
"arn:aws:elasticbeanstalk:us-east-2:my-account-id:environment/my-app/my-env"
```

Utilice el comando [update-tags-for-resource](#) para añadir, actualizar o eliminar etiquetas en un entorno.

```
$ aws elasticbeanstalk update-tags-for-resource \  
  --tags-to-add Key=mytag1,Value=newvalue --tags-to-remove mytag2 \  
  --resource-arn "arn:aws:elasticbeanstalk:us-east-2:my-account-id:environment/my-  
app/my-env"
```

Especifique las etiquetas que desea añadir y las que desea actualizar en el parámetro `--tags-to-add` de `update-tags-for-resource`. Se añade una etiqueta no existente y se actualiza el valor de una etiqueta existente.

Note

Para usar estos dos AWS CLI comandos con un entorno de Elastic Beanstalk, necesita el ARN del entorno. Puede recuperar el ARN mediante el siguiente comando.

```
$ aws elasticbeanstalk describe-environments
```

Propiedades del entorno y otras opciones del software

La página Configurar actualizaciones, monitoreo y registro permite configurar el software en las instancias de Amazon Elastic Compute Cloud (Amazon EC2) que ejecutan la aplicación. Puede configurar las propiedades del entorno, la depuración de AWS X-Ray, el almacenamiento y el streaming de registros de instancias y la configuración específica de la plataforma.

Temas

- [Configurar la configuración específica de la plataforma](#)
- [Configuración de las propiedades del entorno \(variables de entorno\)](#)
- [Espacios de nombres para configuración de software](#)
- [Acceso a las propiedades del entorno](#)
- [Configuración de la depuración de AWS X-Ray](#)
- [Visualización de los registros del entorno de Elastic Beanstalk](#)

Configurar la configuración específica de la plataforma


Además del conjunto estándar de opciones disponibles para todos los entornos, la mayoría de plataformas de Elastic Beanstalk le permiten especificar una configuración específica del lenguaje o de marco. Aparecen en la sección Software de la plataforma de la página Configurar actualizaciones, monitoreo y registro y pueden adoptar las siguientes formas.

- Propiedades preestablecidas del entorno: la plataforma Ruby utiliza propiedades del entorno para la configuración del marco como RACK_ENV y BUNDLE_WITHOUT.
- Propiedades comodín del entorno: la plataforma Tomcat define una propiedad del entorno llamada JDBC_CONNECTION_STRING que no se ha establecido en ningún valor. Este tipo de configuración era más habitual en las versiones anteriores de la plataforma.

- Opciones de configuración: la mayoría de las plataformas definen las [opciones de configuración](#) en espacios de nombres compartidos o específicos de plataforma como `aws:elasticbeanstalk:xray` o `aws:elasticbeanstalk:container:python`.

Para configurar los parámetros específicos de la plataforma en la consola de Elastic Beanstalk

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

 Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Actualizaciones, supervisión y registro, seleccione Editar.
5. En Software de plataforma, realice los cambios necesarios en los valores de configuración.
6. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

Para obtener más información sobre opciones específicas de la plataforma y sobre la obtención de valores de propiedad del entorno, consulte el tema de la plataforma para su lenguaje o marco:

- Docker: [the section called “Configuración del entorno”](#)
- Go: [Uso de la plataforma Elastic Beanstalk Go](#)
- Java SE: [Uso de la plataforma Java SE de Elastic Beanstalk](#)
- Tomcat: [Uso de la plataforma Tomcat de Elastic Beanstalk](#)
- .NET Core en Linux: [.NET Core en la plataforma Linux](#)
- .NET: [Uso de la plataforma .NET de Elastic Beanstalk](#)
- Node.js: [Uso de la plataforma Node.js de Elastic Beanstalk](#)
- PHP: [Uso de la plataforma PHP de Elastic Beanstalk](#)
- Python: [Uso de la plataforma Elastic Beanstalk Python](#)
- Ruby: [Uso de la plataforma Elastic Beanstalk Ruby](#)

Configuración de las propiedades del entorno (variables de entorno)

Puede utilizar las propiedades del entorno (también conocidas como variables de entorno) para pasar claves secretas, puntos de conexión, ajustes de depuración y otros datos a su aplicación. Las propiedades del entorno le ayudarán a ejecutar la aplicación en varios entornos para distintos fines, por ejemplo, desarrollo, pruebas, almacenamiento provisional y producción.

Además, al [agregar una base de datos a su entorno](#), Elastic Beanstalk establece las propiedades del entorno, por ejemplo, `RDS_HOSTNAME`, que puede leer en el código de la aplicación para construir una cadena u objeto de conexión.

Variables de entorno

En la mayoría de los casos, las propiedades del entorno se transfieren a la aplicación como variables de entorno, pero el comportamiento depende de la plataforma. Por ejemplo, [la plataforma Java SE](#) establece las variables de entorno que se recuperan con `System.getenv`, mientras que [la plataforma Tomcat](#) establece las propiedades del sistema Java que se recuperan con `System.getProperty`. En general, las propiedades no están visibles si se conecta a una instancia y ejecuta `env`.

Para configurar las propiedades del entorno en la consola de Elastic Beanstalk

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Actualizaciones, supervisión y registro, seleccione Editar.
5. Desplácese hacia abajo hasta Propiedades del entorno.
6. Seleccione Agregar propiedad de entorno.
7. Introduzca los pares Nombre y Valor de la propiedad.

8. Si necesita agregar más variables, repita Paso 6 y Paso 7.
9. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

Límites de la propiedad de entorno

- Las claves pueden contener caracteres alfanuméricos y los siguientes símbolos: `_ . : / + \ - @`

Los símbolos indicados son válidos para las clave de propiedad de entorno, pero podrían no ser válidos para nombres de variable de entorno en la plataforma del entorno. Para lograr la compatibilidad con todas las plataformas, limite las propiedades del entorno al siguiente patrón:

`[A-Z_][A-Z0-9_]*`

- Los valores pueden contener cualquier carácter alfanumérico y los siguientes símbolos: `_ . : / = + \ - @ ' "`

Note

Se debe aplicar escape a algunos caracteres de los valores de propiedades del entorno. Usa el carácter de barra invertida (`\`) para representar algunos caracteres especiales y caracteres de control. La siguiente lista incluye ejemplos para representar algunos caracteres a los que hay que aplicar escape:

- barra invertida (`\`): para representar el uso `\\`
- comillas simples (`'`): para representar el uso `\'`
- comillas dobles (`"`): para representar el uso `\"`

- Las claves y los valores distinguen entre mayúsculas y minúsculas.
- El tamaño combinado de todas las propiedades del entorno no puede superar 4096 bytes cuando se almacena como cadenas con el formato `clave=valor`.

Espacios de nombres para configuración de software

Puede usar un [archivo de configuración](#) para definir opciones de configuración y realizar otras tareas de configuración en las instancias durante las implementaciones. Las opciones de configuración se pueden definir a través del servicio de Elastic Beanstalk o la plataforma que utilice y están organizadas por espacios de nombres.

Puede utilizar [archivos de configuración](#) de Elastic Beanstalk para establecer propiedades de entorno y opciones de configuración en el código fuente. Utilice el [aws:elasticbeanstalk:application:environment espacio de nombres](#) para definir las propiedades del entorno.

Example `.ebextensions/options.config`

```
option_settings:
  aws:elasticbeanstalk:application:environment:
    API_ENDPOINT: www.example.com/api
```

Si utiliza archivos de configuración o plantillas de AWS CloudFormation para crear [recursos personalizados](#), puede usar una función de AWS CloudFormation para obtener información acerca del recurso y asignarlo a una propiedad de entorno dinámicamente durante la implementación. El siguiente ejemplo del repositorio de GitHub [elastic-beanstalk-samples](#) utiliza la [función Ref](#) para obtener el ARN de un tema de Amazon SNS que crea y se lo asigna a una propiedad de entorno denominada NOTIFICATION_TOPIC.

Notas

- Si utiliza una función de AWS CloudFormation para definir una propiedad de entorno, la consola de Elastic Beanstalk muestra el valor de la propiedad antes de evaluar la función. Puede utilizar el [script de plataforma get-config](#) para confirmar los valores de las propiedades de entorno que están disponibles para su aplicación.
- La plataforma [Multicontainer Docker](#) no utiliza AWS CloudFormation para crear recursos de contenedor. Como resultado, esta plataforma no admite la definición de las propiedades del entorno usando funciones de AWS CloudFormation.

Example `.Ebextensions/sns-topic.config`

```
Resources:
  NotificationTopic:
    Type: AWS::SNS::Topic

option_settings:
  aws:elasticbeanstalk:application:environment:
    NOTIFICATION_TOPIC: '`{"Ref" : "NotificationTopic"}``'
```

También puede utilizar esta característica para propagar información de [pseudoparámetros de AWS CloudFormation](#). Este ejemplo obtiene la región actual y la asigna a una propiedad llamada `AWS_REGION`.

Example `.Ebextensions/env-regionname.config`

```
option_settings:
  aws:elasticbeanstalk:application:environment:
    AWS_REGION: ``{"Ref" : "AWS::Region"}``
```

La mayoría de las plataformas de Elastic Beanstalk define espacios de nombres adicionales con opciones para configurar software que se ejecuta en la instancia, como el proxy inverso que transmite solicitudes a su aplicación. Para obtener más información sobre los espacios de nombres disponibles para su plataforma, consulte las siguientes secciones:

- Go: [Espacios de nombres de configuración de Go](#)
- Java SE: [Espacios de nombres de configuración de Java SE](#)
- Tomcat: [Espacios de nombres de configuración de Tomcat](#)
- .NET Core en Linux: [.NET Core en la configuración de espacio de nombres de Linux](#)
- .NET: [El espacio de nombres aws:elasticbeanstalk:container:dotnet:apppool](#)
- Node.js: [Espacio de nombres de configuración de Node.js](#)
- PHP: [El espacio de nombres aws:elasticbeanstalk:container:php:phpini](#)
- Python: [Espacios de nombres de configuración de Python](#)
- Ruby: [Espacios de nombres de la configuración de Ruby](#)

Elastic Beanstalk cuenta con numerosas opciones de configuración para personalizar el entorno. Además de los archivos de configuración, también puede definir opciones en la consola, configuraciones guardadas, la CLI de EB o la AWS CLI. Para obtener más información, consulte [Opciones de configuración](#).

Acceso a las propiedades del entorno

En la mayoría de los casos, accede a propiedades de entorno en su código de aplicación como una variable de entorno. No obstante, las propiedades del entorno suelen transferirse únicamente a la aplicación y no se pueden visualizar conectando una instancia en su entorno y ejecutando `env`.

- [Go](#): `os.Getenv`

```
endpoint := os.Getenv("API_ENDPOINT")
```

- [Java SE](#): `System.getenv`

```
String endpoint = System.getenv("API_ENDPOINT");
```

- [Tomcat](#): `System.getProperty`

```
String endpoint = System.getProperty("API_ENDPOINT");
```

- [.NET Core en Linux](#): `Environment.GetEnvironmentVariable`

```
string endpoint = Environment.GetEnvironmentVariable("API_ENDPOINT");
```

- [.NET](#): `appConfig`

```
NameValueCollection appConfig = ConfigurationManager.AppSettings;  
string endpoint = appConfig["API_ENDPOINT"];
```

- [Node.js](#): `process.env`

```
var endpoint = process.env.API_ENDPOINT
```

- [PHP](#): `$_SERVER`

```
$endpoint = $_SERVER['API_ENDPOINT'];
```

- [Python](#): `os.environ`

```
import os  
endpoint = os.environ['API_ENDPOINT']
```

- [Ruby](#): `ENV`

```
endpoint = ENV['API_ENDPOINT']
```

Fuera del código de aplicación como, por ejemplo, en un script que se ejecuta durante la implementación, puede obtener acceso a las propiedades del entorno con el [get-config script](#)

[de la plataforma](#). Consulte el repositorio de GitHub [elastic-beanstalk-samples](#) para obtener configuraciones de ejemplo que utilizan `get-config`.

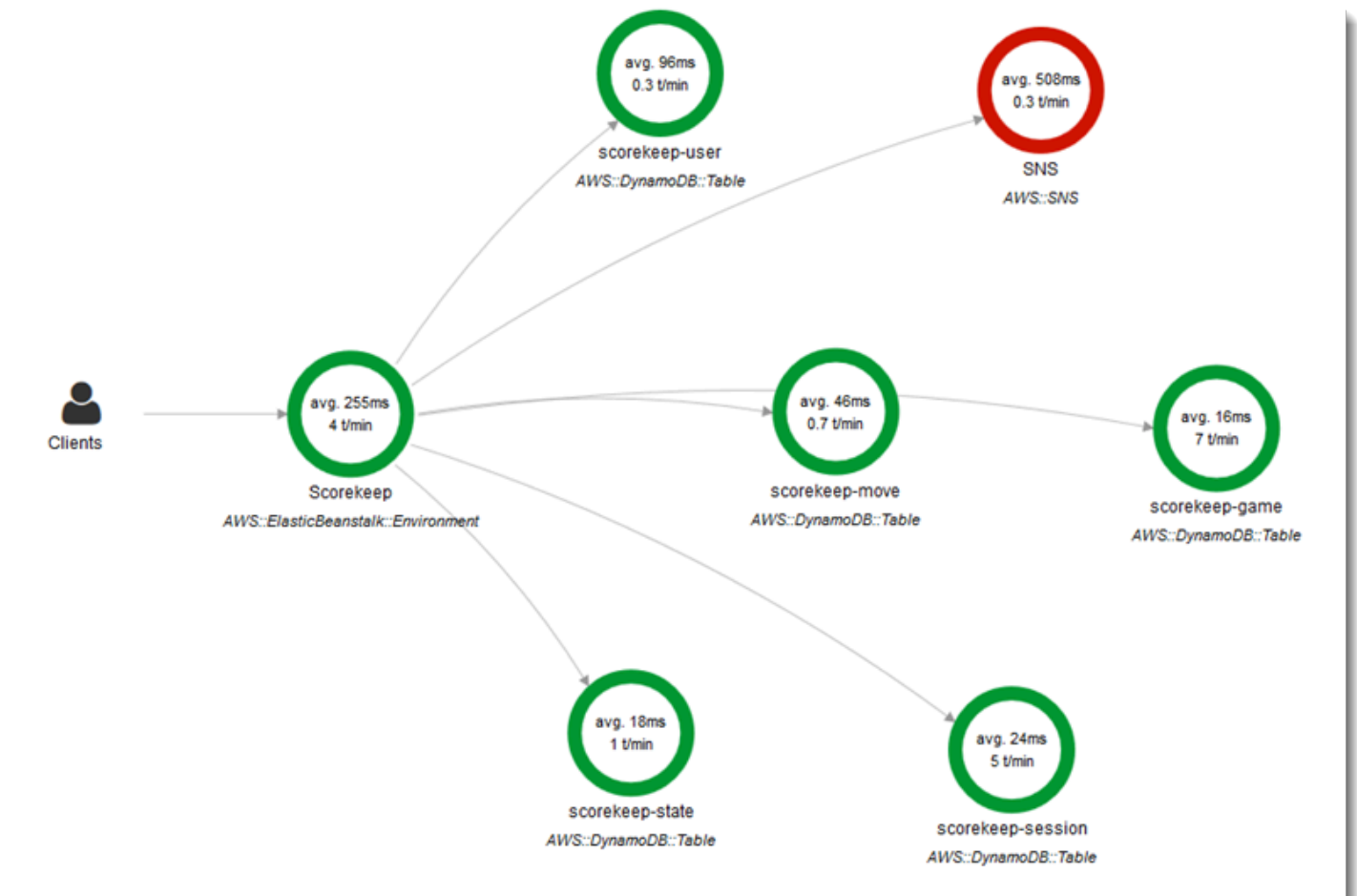
Configuración de la depuración de AWS X-Ray

Puede utilizar la consola AWS Elastic Beanstalk o un archivo de configuración para ejecutar el daemon de AWS X-Ray en las instancias del entorno. X-Ray es un servicio de AWS que recopila datos sobre las solicitudes que la aplicación atiende y los utiliza para crear un mapa de servicios que puede utilizar para identificar problemas con la aplicación y oportunidades de optimización.

Note

Algunas regiones no ofrecen X-Ray. Si crea un entorno en una de estas regiones, no puede ejecutar el daemon X-Ray en las instancias de su entorno.

Para obtener información sobre los servicios de AWS que se ofrecen en cada región, consulte la [Tabla de regiones](#).



X-Ray cuenta con un SDK que puede utilizar para configurar el código de la aplicación y con una aplicación daemon que transmite la información de depuración del SDK a la API de X-Ray.

Plataformas admitidas

Puede utilizar el SDK de X-Ray con las siguientes plataformas Elastic Beanstalk:

- Go: versión 2.9.1 y posteriores
- Java 8: versión 2.3.0 y posteriores
- Java 8 con Tomcat 8: versión 2.4.0 y posteriores
- Node.js: versión 3.2.0 y posteriores
- Windows Server: todas las versiones de plataforma que se haya publicado a partir del 18 de diciembre de 2016
- Python: versión 2.5.0 y posteriores

En las plataformas compatibles, puede utilizar una opción de configuración para ejecutar el daemon de X-Ray en las instancias del entorno. Puede habilitar el daemon a través de la [consola de Elastic Beanstalk](#) o utilizando un [archivo de configuración](#).

Para cargar los datos en X-Ray, el daemon de X-Ray necesita permisos de IAM en la política administrada AWSXrayWriteOnlyAccess. Estos permisos se incluyen en [el perfil de instancia de Elastic Beanstalk](#). Si no utiliza el perfil de instancias predeterminado, consulte [Conceder permiso al daemon con el fin de enviar datos a X-Ray](#) en la Guía para desarrolladores de AWS X-Ray.

Para realizar la depuración con X-Ray se requiere el uso del SDK de X-Ray. Consulte [Introducción a AWS X-Ray](#) en la Guía para desarrolladores de AWS X-Ray con la finalidad de obtener instrucciones y aplicaciones de muestra.

Si utiliza una versión de plataforma que no incluye el daemon, puede ejecutarlo con un script de un archivo de configuración. Con el fin de obtener más información, consulte [Descargar y ejecutar de forma manual el daemon de X-Ray \(avanzado\)](#) en la Guía para desarrolladores de AWS X-Ray.

Secciones

- [Configuración de la depuración](#)
- [El espacio de nombres aws:elasticbeanstalk:xray](#)

Configuración de la depuración

Puede habilitar el daemon de X-Ray en un entorno en ejecución en la consola de Elastic Beanstalk.

Para habilitar la depuración en la consola de Elastic Beanstalk

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Actualizaciones, supervisión y registro, seleccione Editar.

5. En la sección Amazon X-Ray, seleccione Activado.
6. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

También puede activar esta opción durante la creación del entorno. Para obtener más información, consulte [El asistente de creación de nuevo entorno](#).

El espacio de nombres aws:elasticbeanstalk:xray

Puede utilizar la opción `XRayEnabled` del espacio de nombres `aws:elasticbeanstalk:xray` para habilitar la depuración.

Para habilitar la depuración automáticamente al implementar la aplicación, defina la opción en un [archivo de configuración](#) del código fuente tal y como se indica a continuación.

Example `.ebextensions/debugging.config`

```
option_settings:
  aws:elasticbeanstalk:xray:
    XRayEnabled: true
```

Visualización de los registros del entorno de Elastic Beanstalk

AWS Elastic Beanstalk ofrece dos maneras de ver periódicamente los registros en las instancias de Amazon EC2 que ejecutan su aplicación:

- Configure su entorno de Elastic Beanstalk para cargar registros de instancias rotadas en el bucket de Amazon S3 del entorno.
- Configure el entorno para transmitir los registros de instancias a Amazon CloudWatch Logs.

Al configurar el streaming de registros de instancia en CloudWatch Logs, Elastic Beanstalk crea grupos de registro de CloudWatch Logs para registros de proxy y de implementación en las instancias de Amazon EC2, y transfiere estos archivos de registro a CloudWatch Logs en tiempo real. Para obtener más información acerca de los logs de instancia, consulte [Visualización de registros de instancias de Amazon EC2 en su entorno de Elastic Beanstalk](#).

Además de los registros de instancias, si habilita el [estado mejorado](#) de su entorno, puede configurar dicho entorno para que realice la transmisión de la información de estado a CloudWatch Logs.

Cuando cambia el estado del entorno, Elastic Beanstalk agrega un registro a un grupo de registros de estado, con el nuevo estado y una descripción de la causa del cambio. Para obtener información acerca del streaming del estado del entorno, consulte [Transmisión de información sobre el estado del entorno Elastic Beanstalk a Amazon CloudWatch Logs](#).

Configuración de la visualización de registros de instancias

Para ver los registros de instancias, puede habilitar la rotación de registros de instancias y el streaming de registros en la consola de Elastic Beanstalk.

Para configurar la rotación del registro de instancias y el streaming de registros en la consola de Elastic Beanstalk

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Actualizaciones, supervisión y registro, seleccione Editar.
5. En la sección Almacenamiento de registros de S3, seleccione Activado debajo de Rotar registros para habilitar la carga de registros rotados en Amazon S3.
6. En la sección Instance log streaming to CloudWatch Logs (Streaming de registros de instancias a CloudWatch Logs), configure los siguientes ajustes:
 - Streaming de registros: seleccione Activado para habilitar el streaming de registros.
 - Retention (Retención): especifique el número de días durante el cual se conservan registros en CloudWatch Logs.
 - Lifecycle (Ciclo de vida): establézcalo en Delete logs upon termination (Eliminar los registros al terminar) para eliminar los registros de CloudWatch Logs inmediatamente si el entorno ha finalizado, en lugar de esperar a que caduquen.
7. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

Después de habilitar el streaming de registros, puede volver a la categoría o página de configuración Software y buscar el enlace Log Groups (Grupos de registros). Haga clic en este enlace para ver los registros de instancias en la consola de CloudWatch.

Configuración de la visualización de registros de estado de entorno

Para ver los registros de estado de entorno, puede habilitar el streaming de registros de estado de entorno en la consola de Elastic Beanstalk.

Para configurar el streaming de registros de estado del entorno en la consola de Elastic Beanstalk

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Actualizaciones, supervisión y registro, seleccione Editar.
5. Desplácese hasta la sección Monitoreo.
6. En Health event streaming to CloudWatch Logs (Streaming de eventos de estado a CloudWatch Logs), configure los siguientes ajustes:
 - Streaming de registros: elija Activado para habilitar el streaming de registros.
 - Retention (Retención): especifique el número de días durante el cual se conservan registros en CloudWatch Logs.
 - Lifecycle (Ciclo de vida): establézcalo en Delete logs upon termination (Eliminar los registros al terminar) para eliminar los registros de CloudWatch Logs inmediatamente si el entorno ha finalizado, en lugar de esperar a que caduquen.
7. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

Espacios de visualización de registros

Los siguientes espacios de nombres contienen opciones para la visualización de logs:

- [aws:elasticbeanstalk:hostmanager](#): configure la actualización de registros rotados en Amazon S3.
- [aws:elasticbeanstalk:cloudwatch:logs](#) – configure el streaming de registros de instancias a CloudWatch.
- [aws:elasticbeanstalk:cloudwatch:logs:health](#): configure el streaming de estado de entorno a CloudWatch.

Notificaciones de entorno de Elastic Beanstalk con Amazon SNS

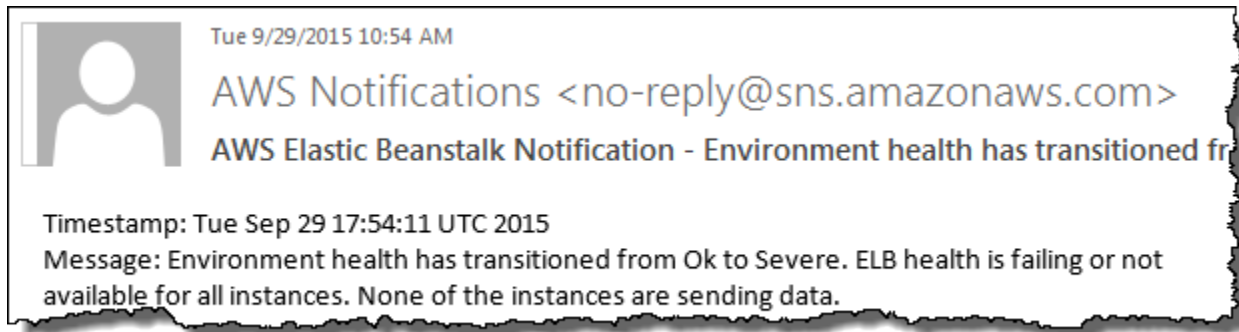
Puede configurar el entorno de AWS Elastic Beanstalk para que utilice Amazon Simple Notification Service (Amazon SNS) y lo notifique sobre eventos importantes que afecten a su aplicación. Para recibir mensajes de correo electrónico de AWS siempre que se produzca un error o cambie el estado del entorno, especifique una dirección de correo electrónico cuando cree un entorno o más adelante.

Note

Elastic Beanstalk utiliza Amazon SNS para las notificaciones. Para obtener información sobre los precios de Amazon SNS, consulte <https://aws.amazon.com/sns/pricing/>.

Cuando configura notificaciones para su entorno, Elastic Beanstalk crea un tema de Amazon SNS para su entorno en su nombre. Para enviar mensajes a un tema de Amazon SNS, Elastic Beanstalk debe tener el permiso requerido. Para obtener más información, consulte [Configuración de los permisos para enviar notificaciones](#).

Cuando se produce un [evento](#) notable, Elastic Beanstalk envía un mensaje al tema. A continuación, Amazon SNS transmite los mensajes que recibe a los suscriptores del tema. Los eventos destacables incluyen errores de creación del entorno y todos los cambios en el [estado del entorno y las instancias](#). Los eventos para las operaciones de Amazon EC2 Auto Scaling (como agregar y eliminar instancias del entorno) y otros eventos informativos no activan las notificaciones.



Puede escribir una dirección de correo electrónico en la consola de Elastic Beanstalk cuando cree un entorno o en algún momento posterior. Esto creará un tema de Amazon SNS y se suscribirá a él. Elastic Beanstalk administra el ciclo de vida del tema y lo elimina cuando se termina el entorno o cuando elimina su dirección de correo electrónico en la [consola de administración del entorno](#).

El espacio de nombres de `aws:elasticbeanstalk:sns:topics` proporciona opciones para configurar un tema de Amazon SNS con archivos de configuración, una CLI o un SDK. Mediante uno de estos métodos, puede configurar el tipo de suscriptor y el punto de enlace. Para el tipo de suscriptor, puede elegir una cola de Amazon SQS o una URL HTTP.

Solo puede activar o desactivar notificaciones de Amazon SNS. La frecuencia de las notificaciones enviadas al tema puede ser alta, en función del tamaño y la composición del entorno. Para configurar las notificaciones que se enviarán en circunstancias específicas, tiene otras opciones. Puede [configurar reglas controladas por eventos](#) con Amazon EventBridge que lo notifiquen cuando Elastic Beanstalk emita eventos que cumplan criterios específicos. O bien, puede [configurar su entorno para publicar métricas personalizadas](#) y [establecer alarmas de Amazon CloudWatch](#) a fin de notificarlo cuando esas métricas alcancen un umbral incorrecto.

Configuración de notificaciones mediante la consola de Elastic Beanstalk

Puede escribir una dirección de correo electrónico en la consola de Elastic Beanstalk a fin de crear un tema de Amazon SNS para su entorno.

Para configurar notificaciones en la consola de Elastic Beanstalk

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Actualizaciones, supervisión y registro, seleccione Editar.
5. Desplácese hasta la sección Notificaciones de correo electrónico.
6. Escriba una dirección de correo electrónico.
7. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

Cuando escribe una dirección de correo electrónico para notificaciones, Elastic Beanstalk crea un tema de Amazon SNS para su entorno y añade una suscripción. Amazon SNS envía un correo electrónico a la dirección suscrita para confirmar la suscripción. Debe hacer clic en el enlace del correo electrónico de confirmación para activar la suscripción y recibir notificaciones.

Configuración de las notificaciones con las opciones de configuración

Utilice las opciones del [espacio de nombres de aws:elasticbeanstalk:sns:topics](#) para configurar notificaciones de Amazon SNS para el entorno. Puede definir estas opciones utilizando [archivos de configuración](#), una CLI o un SDK.

- Punto de enlace de notificación: la dirección de correo electrónico, cola de Amazon SQS o URL a la que se envían las notificaciones. Si establece esta opción, se crearán una cola de SQS y una suscripción para el punto de enlace especificado. Si el punto de enlace no es una dirección de correo electrónico, también debe establecer la opción `Notification Protocol`. SNS valida el valor de `Notification Endpoint` en función del valor de `Notification Protocol`. Si define esta opción varias veces, se crean suscripciones adicionales al tema. Si quita esta opción, el tema se elimina.
- Protocolo de notificación: el protocolo que se utiliza para enviar notificaciones a `Notification Endpoint`. El valor predeterminado de esta opción es `email`. Defina esta opción para `email-json` enviar correo electrónico con formato JSON, `http` o `https` para publicar notificaciones con formato JSON a un punto de enlace HTTP, o `sqs` para enviar notificaciones a una cola de SQS.

Note

No se admiten las notificaciones de AWS Lambda.

- ARN de tema de notificación: después de establecer un punto de enlace de notificación para su entorno, consulte este valor para obtener el ARN del tema de SNS. También puede definir esta opción para que se utilice un tema de SNS para recibir notificaciones. Un tema asociado a su entorno mediante esta opción no se elimina cuando cambia esta opción o termina el entorno.

Para configurar las notificaciones de Amazon SNS, debe tener los permisos necesarios.

Si el usuario de IAM utiliza la política de usuario administrada [AdministratorAccess-AWSElasticBeanstalk](#), ya debe disponer de los permisos necesarios para configurar el tema predeterminado de Amazon SNS que Elastic Beanstalk crea para el entorno. Sin embargo, si configura un tema de Amazon SNS que Elastic Beanstalk no administra, deberá agregar la siguiente política a su rol de usuario.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sns:SetTopicAttributes",
        "sns:GetTopicAttributes",
        "sns:Subscribe",
        "sns:Unsubscribe",
        "sns:Publish"
      ],
      "Resource": [
        "arn:aws:sns:us-east-2:123456789012:sns_topic_name"
      ]
    }
  ]
}
```

- Nombre de tema de notificación: defina esta opción para personalizar el nombre del tema de Amazon SNS utilizado para las notificaciones del entorno. Si existe un tema con el mismo nombre, Elastic Beanstalk asocia ese tema al entorno.

⚠ Warning

Si asocia un tema de SNS existente a un entorno con `Notification Topic Name`, Elastic Beanstalk eliminará el tema en caso de que termine el entorno o cambie esta configuración en el futuro.

Si cambia esta opción, también se cambia `Notification Topic ARN`. Si ya hay un tema asociado al entorno, Elastic Beanstalk eliminará el tema anterior y creará un tema y suscripción nuevos.

Al utilizar un nombre de tema personalizado, también debe proporcionar un ARN de un tema personalizado creado externamente. La política de usuario administrada no detecta automáticamente un tema con un nombre personalizado, por lo que debe proporcionar permisos personalizados de Amazon SNS a los usuarios de IAM. Utilice una política similar a la que se utiliza para un ARN de tema personalizado, pero incluya las siguientes adiciones:

- Incluya dos acciones más en la lista `Actions`, concretamente: `sns:CreateTopic`, `sns>DeleteTopic`
- Si va a cambiar el `Notification Topic Name` de un nombre de tema personalizado a otro, incluya los ARN de ambos temas en la lista `Resource`. También puede incluir una expresión regular que cubra ambos temas. De esta forma, Elastic Beanstalk tiene permisos para eliminar el tema anterior y crear el nuevo.

La CLI de EB y la consola de Elastic Beanstalk aplican los valores recomendados a las opciones anteriores. Debe eliminar estos ajustes si desea usar archivos de configuración para configurarlos. Para obtener más información, consulte [Valores recomendados](#).

Configuración de los permisos para enviar notificaciones

En esta sección se describen las consideraciones de seguridad relacionadas con las notificaciones que utilizan Amazon SNS. Existen dos casos distintos:

- Utilice el tema predeterminado de Amazon SNS que Elastic Beanstalk crea para su entorno.
- Proporcione un tema externo de Amazon SNS a través de las opciones de configuración.

La política de acceso predeterminada para un tema de Amazon SNS permite únicamente al propietario del tema publicarlo o suscribirse a él. Sin embargo, mediante la configuración adecuada de la política, se puede conceder permiso a Elastic Beanstalk para publicar en un tema de Amazon SNS en cualquiera de los dos casos descritos en esta sección. En las siguientes subsecciones se proporciona más información.

Permisos de un tema predeterminado

Cuando configura notificaciones para su entorno, Elastic Beanstalk crea un tema de Amazon SNS para su entorno. Para enviar mensajes a un tema de Amazon SNS, Elastic Beanstalk debe tener el permiso requerido. Si su entorno utiliza el [rol de servicio](#) que la consola de Elastic Beanstalk o la CLI de EB han generado para él, o el [rol vinculado al servicio de monitoreo](#) de la cuenta, no es necesario hacer nada más. Estos roles administrados incluyen el permiso necesario que permite a Elastic Beanstalk enviar mensajes al tema de Amazon SNS.

Sin embargo, si ha proporcionado un rol de servicio personalizado al crear su entorno, asegúrese de que incluya la siguiente política.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sns:Publish"
      ],
      "Resource": [
        "arn:aws:sns:us-east-2:123456789012:ElasticBeanstalkNotifications*"
      ]
    }
  ]
}
```

Permisos de un tema externo

En [Configuración de las notificaciones con las opciones de configuración](#) se explica cómo puede reemplazar el tema de Amazon SNS que Elastic Beanstalk proporciona, por otro tema de Amazon SNS. Si reemplazó el tema, Elastic Beanstalk debe comprobar que tiene permiso para publicar en este tema de SNS a fin de poder asociar el tema SNS con el entorno. Debería tener `sns:Publish`. El rol de servicio utiliza el mismo permiso. Para verificar que este es el caso, Elastic Beanstalk envía

una notificación de prueba a SNS como parte de su acción a fin de crear o actualizar el entorno. Si esta prueba falla, el intento de crear o actualizar el entorno también falla. Elastic Beanstalk muestra un mensaje que explica el motivo de esta falla.

Si proporciona un rol de servicio personalizado para su entorno, asegúrese de que incluya la siguiente política a fin de permitir que Elastic Beanstalk envíe mensajes al tema de Amazon SNS. En el código siguiente, reemplace *sns_topic_name* con el nombre del tema de Amazon SNS que proporcionó en las opciones de configuración.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sns:Publish"
      ],
      "Resource": [
        "arn:aws:sns:us-east-2:123456789012:sns_topic_name"
      ]
    }
  ]
}
```

Para obtener más información sobre el control de acceso de Amazon SNS, consulte [Casos de ejemplo para el control de acceso de Amazon SNS](#) en la Guía para desarrolladores de Amazon Simple Notification Service.

Configuración de Amazon Virtual Private Cloud (Amazon VPC) con Elastic Beanstalk

[Amazon Virtual Private Cloud](#) (Amazon VPC) es el servicio de red que enruta el tráfico de forma segura a las instancias de EC2 que ejecutan la aplicación en Elastic Beanstalk. Si no configura una VPC al lanzar el entorno, Elastic Beanstalk utilizará la VPC predeterminada.

Puede lanzar el entorno en una VPC personalizada para adaptar las opciones de red y seguridad. Elastic Beanstalk le permite elegir las subredes que desea utilizar para sus recursos y cómo configurar las direcciones IP para las instancias y el balanceador de carga de entorno. Un entorno

está asociado a una VPC al crearlo, pero puede cambiar las opciones de subred y de dirección IP cuando está en ejecución.

Note

Si creó su cuenta de AWS antes del 4 de diciembre de 2013, es posible que tenga entornos que utilicen la configuración de red Amazon EC2-Classik en algunas regiones de AWS en lugar de Amazon VPC. Para obtener información sobre cómo migrar entornos de una configuración de red EC2 clásica a VPC, consulte [Migración de entornos Elastic Beanstalk de EC2 classic a una VPC](#).

Configuración de la VPC en la consola de Elastic Beanstalk

Si eligió una VPC personalizada al crear el entorno, puede modificar las opciones de VPC en la consola de Elastic Beanstalk.

Para configurar las opciones de VPC del entorno

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Network (Red), seleccione Edit (Editar).

Dispone de las siguientes opciones.

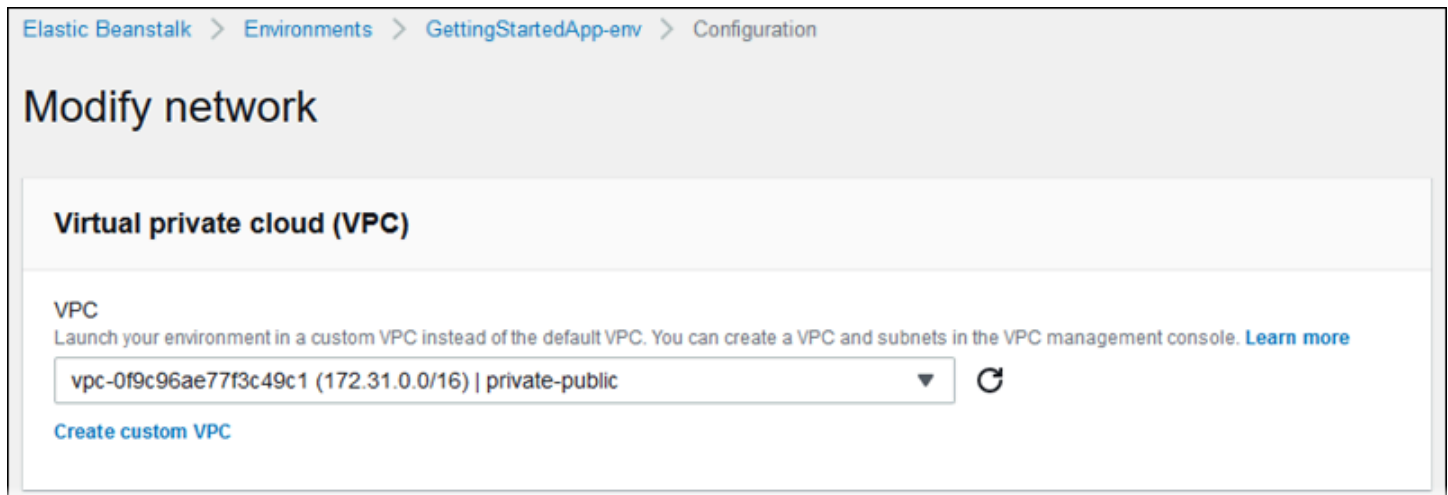
Opciones

- [VPC](#)
- [Visibilidad del balanceador de carga](#)
- [Subredes del balanceador de carga](#)

- [Dirección IP pública de la instancia](#)
- [Subredes de instancia](#)
- [Subredes de base de datos](#)

VPC

Elija una VPC para el entorno. Solo puede cambiar esta opción durante la creación del entorno.



Visibilidad del balanceador de carga

Para un entorno con balanceo de carga, elija el modo del balanceador de carga. De forma predeterminada, el balanceador de carga es público, con una dirección IP pública y un nombre de dominio. Si la aplicación solo atiende a tráfico proveniente del interior de la VPC o de una VPN conectada, desactive esta opción y elija subredes privadas para que el balanceador de carga sea interno y quede deshabilitado el acceso desde Internet.

Subredes del balanceador de carga

Para un entorno con balanceo de carga, elija las subredes que el balanceador de carga utiliza para distribuir el tráfico. Para una aplicación pública, elija subredes públicas. Utilice subredes en varias zonas de disponibilidad para lograr un mayor nivel de disponibilidad. Para una aplicación interna, elija subredes privadas y deshabilite la visibilidad del balanceador de carga.

Load balancer settings

Assign your load balancer to a subnet in each Availability Zone (AZ) in which your application runs. For a publicly accessible application, set **Visibility** to **Public** and choose public subnets.

Visibility
Make your load balancer internal if your application serves requests only from connected VPCs. Public load balancers serve requests from the Internet.

Public ▼

Load balancer subnets

<input type="checkbox"/>	Availability Zone	Subnet	CIDR	Name
<input checked="" type="checkbox"/>	us-east-2a	subnet-04a707767b8ca8023	172.31.0.0/24	public-a
<input type="checkbox"/>	us-east-2a	subnet-0c559ebeb1a89adb4	172.31.3.0/24	private-a
<input checked="" type="checkbox"/>	us-east-2b	subnet-034a813125cd2077a	172.31.2.0/24	private-b
<input type="checkbox"/>	us-east-2b	subnet-09a24e24e7f7359fa	172.31.1.0/24	public-b

Dirección IP pública de la instancia

Si elige subredes públicas para las instancias de aplicación, habilite direcciones IP públicas para poder incluirlas en rutas desde Internet.

Subredes de instancia

Elija subredes para las instancias de aplicación. Elija al menos una subred por cada zona de disponibilidad que el balanceador de carga utilice. Si elige subredes privadas para las instancias, la VPC debe tener un gateway NAT en una subred pública que las instancias puedan utilizar para el acceso a Internet.

Instance settings

Choose a subnet in each AZ for the instances that run your application. To avoid exposing your instances to the Internet, run your instances in private subnets and load balancer in public subnets. To run your load balancer and instances in the same public subnets, assign public IP addresses to the instances.

Public IP address
Assign a public IP address to the Amazon EC2 instances in your environment.

Instance subnets

	Availability Zone	Subnet	CIDR	Name
<input type="checkbox"/>	us-east-2a	subnet-04a707767b8ca8023	172.31.0.0/24	public-a
<input checked="" type="checkbox"/>	us-east-2a	subnet-0c559eeeb1a89adb4	172.31.3.0/24	private-a
<input type="checkbox"/>	us-east-2b	subnet-034a813125cd2077a	172.31.2.0/24	private-b
<input checked="" type="checkbox"/>	us-east-2b	subnet-09a24e24e7f7359fa	172.31.1.0/24	public-b

Cancel Continue Apply

Subredes de base de datos

Cuando ejecute una base de datos de Amazon RDS asociada a su entorno de Elastic Beanstalk, elija subredes para las instancias de base de datos. Para una alta disponibilidad, haga que la base de datos sea de tipo Multi-AZ y elija una subred por cada zona de disponibilidad. Para asegurarse de que la aplicación puede conectar con la base de datos, ejecute ambos en las mismas subredes.

El espacio de nombres `aws:ec2:vpc`

Puede utilizar las opciones de configuración del espacio de nombres [aws:ec2:vpc](#) para configurar las opciones de red del entorno.

Este [archivo de configuración](#) utiliza las opciones de este espacio de nombres para establecer en la VPC y las subredes del entorno una configuración pública-privada. Con el fin de establecer el ID de VPC en un archivo de configuración, el archivo debe incluirse en el paquete de código fuente de la aplicación durante la creación del entorno. Consulte [Definición de opciones de configuración durante la creación del entorno](#) para ver otros modos de configurar estas opciones durante la creación del entorno.

Example .ebextensions/vpc.config: Public-Private

```
option_settings:
  aws:ec2:vpc:
    VPCId: vpc-087a68c03b9c50c84
    AssociatePublicIpAddress: 'false'
    ELBScheme: public
    ELBSubnets: subnet-0fe6b36bcb0ffc462,subnet-032fe3068297ac5b2
    Subnets: subnet-026c6117b178a9c45,subnet-0839e902f656e8bd1
```

En este ejemplo se muestra una configuración pública-pública, donde el balanceador de carga y las instancias EC2 se ejecutan en las mismas subredes públicas.

Example .ebextensions/vpc.config: Public-Public

```
option_settings:
  aws:ec2:vpc:
    VPCId: vpc-087a68c03b9c50c84
    AssociatePublicIpAddress: 'true'
    ELBScheme: public
    ELBSubnets: subnet-0fe6b36bcb0ffc462,subnet-032fe3068297ac5b2
    Subnets: subnet-0fe6b36bcb0ffc462,subnet-032fe3068297ac5b2
```

Migración de entornos Elastic Beanstalk de EC2 classic a una VPC

En este tema se describen las diferentes opciones de cómo migrar los entornos Elastic Beanstalk desde una plataforma de red de EC2-Classic a una red de [Amazon Virtual Private Cloud](#) (Amazon VPC).

Si creó la cuenta de AWS antes del 4 de diciembre de 2013, es posible que tenga entornos que utilizan la configuración de red EC2-Classic en algunas Regiones de AWS. Todas las cuentas de AWS creadas a partir del 4 de diciembre de 2013 ya tienen una VPC en cada región de AWS. Las únicas excepciones son si Amazon EC2-Classic estaba habilitada como resultado de una solicitud de soporte.

Note

Puede ver las opciones de configuración de red de su entorno en la categoría Network configuration (Configuración de red) en la página [Información general de la configuración](#) de la [consola de Elastic Beanstalk](#).

Por qué debe migrar

La compatibilidad estándar de Amazon EC2-Classic llegará a su fin el 15 de agosto de 2022. Para evitar interrupciones en sus cargas de trabajo, recomendamos que migre de Amazon EC2-Classic a una VPC antes del 15 de agosto de 2022. También le pedimos que no lance ningún recursos de AWS en Amazon EC2-Classic de ahora en más y que utilice Amazon VPC en su lugar.

Al migrar los entornos de Elastic Beanstalk de Amazon EC2-Classic a Amazon VPC, debe crear una nueva cuenta de AWS. También debe volver a crear sus entornos EC2-Classic de AWS en su nueva cuenta de AWS. No se necesita trabajo de configuración adicional para que sus entornos utilicen la VPC predeterminada. Si la VPC predeterminada no cumple sus requisitos, cree manualmente una VPC personalizada y asíciela a sus entornos.

Como alternativa, si la cuenta de AWS existente tiene recursos que no puede migrar a una cuenta nueva de AWS, puede agregar una VPC a la cuenta actual. Luego, configure los entornos para que utilicen la VPC.

Para obtener más información, consulte la entrada del blog [EC2-Classic Networking se retira: cómo prepararse](#).

Migrar un entorno de EC2-Classic a una cuenta nueva de AWS (recomendado)


Si aún no dispone de una cuenta de AWS creada a partir del 4 de diciembre de 2013, cree una nueva. Va a migrar sus entornos a esta nueva cuenta.

1. La cuenta nueva de AWS proporciona a sus entornos una VPC predeterminada. Si no tiene que crear una VPC personalizada, vaya al paso 2.

Puede crear una VPC personalizada de una de las siguientes maneras:


- Cree una VPC rápidamente utilizando el asistente de consola de Amazon VPC con una de las opciones de configuración disponibles. Para obtener más información, consulte [Configuraciones del Asistente de consola de Amazon VPC](#).
- Cree una VPC personalizada en la consola de Amazon VPC si tiene más requisitos específicos para su VPC. Recomendamos que haga esto, por ejemplo, si su caso de uso requiere un número específico de subredes. Para obtener más información, consulte [VPC y subredes](#).
- Cree una VPC utilizando el repositorio [elastic-beanstalk-samples](#) del sitio web de GitHub si prefiere utilizar plantillas de AWS CloudFormation con sus entornos de Elastic Beanstalk..

Este repositorio incluye plantillas de AWS CloudFormation. Para obtener más información, consulte . [Uso de Elastic Beanstalk con Amazon VPC](#).

 Note

También puede crear una VPC personalizada al mismo tiempo que vuelve a crear el entorno en la cuenta nueva de AWS usando el [asistente para crear un entorno nuevo](#). Si utiliza el asistente y elige crear una VPC personalizada, el asistente lo va a redirigir a la consola de Amazon VPC.

2. En su nueva cuenta de AWS, cree un nuevo entorno. Recomendamos que el entorno nuevo incluya la misma configuración que el entorno existente en la cuenta de AWS desde la que hará la migración. Puede hacerlo mediante alguna de las siguientes opciones.


 Note

Si el nuevo entorno debe seguir utilizando el mismo CNAME después de la migración, termine el entorno original en la plataforma de EC2-Classic. Esta acción libera el CNAME para poder utilizarlo. Sin embargo, esto puede producir un tiempo de inactividad para ese entorno y existe el riesgo de que otro cliente pueda seleccionar su CNAME en el tiempo transcurrido entre la terminación del entorno de EC2-Classic y la creación del nuevo. Para obtener más información, consulte . [Terminación de un entorno de Elastic Beanstalk](#).

Para entornos que tengan su propio nombre de dominio propietario, CNAME no tiene este problema. Puede actualizar su sistema de nombres de dominio (DNS) para reenviar solicitudes a su nuevo CNAME.

- Utilice el [asistente para crear nuevo entorno](#) en la [consola de Elastic Beanstalk](#). El asistente proporciona una opción para crear una VPC personalizada. Si no elige crear una VPC personalizada, se asigna una VPC predeterminada.
- Utilice la interfaz de línea de comandos de Elastic Beanstalk (CLI de EB) para volver a crear el entorno en la cuenta nueva de AWS. Uno de los [ejemplos](#) de la descripción del comando `eb create` demuestra la creación de un entorno en una VPC personalizada. Si no proporciona un ID de VPC, el entorno utiliza la VPC predeterminada.

Utilizando este enfoque, puede utilizar un archivo de configuraciones guardado en las dos cuentas de AWS. Como resultado, no necesita ingresar manualmente toda la información de configuración. Sin embargo, debe guardar las opciones de configuración para el entorno de EC2-Classic que está migrando con el comando [eb config save](#). Copie el archivo de configuración guardado en un directorio nuevo para el nuevo entorno de cuenta.

 Note

Debe editar algunos de los datos en el archivo de configuración guardado antes de poder utilizarlos en la nueva cuenta. También debe actualizar la información relativa a su cuenta anterior con los datos correctos para la nueva cuenta. Por ejemplo, debe reemplazar el nombre de recurso de Amazon (ARN) del rol de AWS Identity and Access Management (IAM) por el ARN de rol de IAM para la cuenta nueva.

Si utiliza el comando [eb create](#) con la opción `cfg`, se crea el nuevo entorno utilizando el archivo de configuración guardado especificado. Para obtener más información, consulte [Uso de configuraciones guardadas de Elastic Beanstalk](#).

Migrar un entorno desde EC2-Classic dentro de la misma cuenta de AWS

Es posible que la cuenta existente de AWS tenga recursos que no puede migrar a una cuenta nueva de AWS. En este caso, tendrá que volver a crear sus entornos y configurar manualmente una VPC para cada entorno que cree.

Migre sus entornos a una VPC personalizada

Requisitos previos


Antes de comenzar, debe tener una VPC. Puede crear una VPC no predeterminada (personalizada) de una de las siguientes maneras:

- Cree una VPC rápidamente utilizando el asistente de consola de Amazon VPC con una de las opciones de configuración disponibles. Para obtener más información, consulte [Configuraciones del Asistente de consola de Amazon VPC](#).
- Cree una VPC personalizada en la consola de Amazon VPC si tiene más requisitos específicos para su VPC. Recomendamos que haga esto, por ejemplo, si su caso de uso requiere un número específico de subredes. Para obtener más información, consulte [VPC y subredes](#).

- Cree una VPC utilizando el repositorio [elastic-beanstalk-samples](#) del sitio web de GitHub si prefiere utilizar plantillas de AWS CloudFormation con sus entornos de Elastic Beanstalk.. Este repositorio incluye plantillas de AWS CloudFormation. Para obtener más información, consulte . [Uso de Elastic Beanstalk con Amazon VPC](#).

En los pasos siguientes, utilice el ID de VPC y los ID de subred generados al configurar la VPC en el nuevo entorno.

1. Cree un nuevo entorno que incluya la misma configuración que el entorno existente. Puede hacerlo mediante alguna de las siguientes opciones.

 Note

La función Configuraciones guardadas puede ayudarle a volver a crear sus entornos en la nueva cuenta. Esta característica puede guardar la configuración del entorno para que pueda aplicarla cuando cree o actualice otros entornos. Para obtener más información, consulte . [Uso de configuraciones guardadas de Elastic Beanstalk](#).

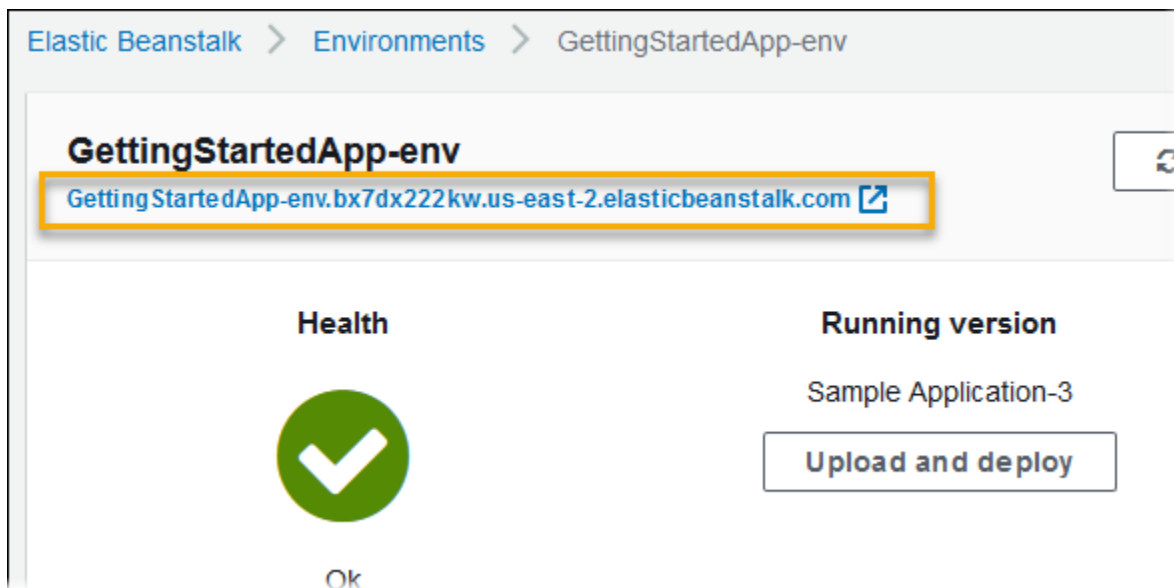
- Utilizando la [consola de Elastic Beanstalk](#), aplique una configuración guardada desde su entorno de EC2-Classic durante la configuración del nuevo entorno. Esta configuración utilizará la VPC. Para obtener más información, consulte . [Uso de configuraciones guardadas de Elastic Beanstalk](#).
 - Mediante la interfaz de línea de comandos de Elastic Beanstalk (EB CLI), ejecute el comando [eb create](#) para volver a crear el entorno. Proporcione los parámetros del entorno original y el identificador de VPC. Uno de los [ejemplos](#) de la descripción del comando `eb create` muestra cómo crear un entorno en una VPC personalizada.
 - Utilice la AWS Command Line Interface (AWS CLI) y vuelva a crear el entorno con el comando `elasticbeanstalk create-environment`. Proporcione los parámetros de su entorno original con el identificador de VPC. Para obtener instrucciones, consulte [Creación de entornos de Elastic Beanstalk con la CLI de AWS](#).
2. Intercambie el CNAME del entorno existente con el de un nuevo entorno. De este modo, se puede hacer referencia al nuevo entorno que creó con la dirección familiar. Puede utilizar la CLI de EB o la AWS CLI.

- Mediante la CLI de EB, cambie los CNAME del entorno ejecutando el comando `eb swap`. Para obtener más información, consulte [Uso de la interfaz de línea de comandos de Elastic Beanstalk \(CLI de EB\)](#).
- Use la AWS CLI, intercambie los CNAME del entorno con el comando `elasticbeanstalk swap-environment-cnames`. Para obtener más información, consulte la [Referencia de comandos de la AWS CLI](#).

Nombre de dominio de su entorno de Elastic Beanstalk

De forma predeterminada, su entorno está disponible para los usuarios en un subdominio de `elasticbeanstalk.com`. Cuando se [crea un entorno](#), puede elegir un nombre de host para su aplicación. El subdominio y el dominio se completan automáticamente `region.elasticbeanstalk.com`.

Para dirigir a los usuarios a su entorno, Elastic Beanstalk lleva un registro CNAME que apunta al balanceador de carga de su entorno. Puede ver la URL de la aplicación de su entorno con el valor actual de CNAME en la página de [información general del entorno](#) de la consola de Elastic Beanstalk.



The screenshot shows the AWS Elastic Beanstalk console interface. At the top, the breadcrumb navigation reads 'Elastic Beanstalk > Environments > GettingStartedApp-env'. Below this, the environment name 'GettingStartedApp-env' is displayed with a refresh icon to its right. A yellow box highlights the CNAME URL: 'GettingStartedApp-env.bx7dx222kw.us-east-2.elasticbeanstalk.com', which includes an external link icon. Underneath, the 'Health' section shows a green checkmark icon and the text 'Ok'. The 'Running version' section shows 'Sample Application-3' and an 'Upload and deploy' button.


Elija la dirección URL en la página de información general o elija Go to environment (Ir al entorno) en el panel de exploración para navegar a la página web de la aplicación.

Puede cambiar el registro CNAME en su entorno intercambiándolo por el registro CNAME de otro entorno. Para obtener instrucciones, consulte [Implementaciones Blue/Green \(azul/verde\) con Elastic Beanstalk](#).

Si dispone de un nombre de dominio, puede utilizar Amazon Route 53 para resolverlo en su entorno. Puede adquirir un nombre de dominio con Amazon Route 53 o utilizar uno que compre a otro proveedor.

Para comprar un nombre de dominio con Route 53, consulte [Registrar un nuevo dominio](#) en la Guía para desarrolladores de Amazon Route 53.

Para obtener más información sobre el uso de un dominio personalizado, consulte [Enrutamiento del tráfico a un entorno de AWS Elastic Beanstalk](#) en la Guía para desarrolladores de Amazon Route 53.

 Important

Si termina un entorno, también debe eliminar todas las asignaciones de CNAME que ha creado, ya que otros clientes pueden reutilizar un nombre de host disponible. Asegúrese de eliminar los registros de DNS que apuntan a su entorno terminado para evitar que se cree una entrada de DNS colgante. Una entrada de DNS colgante puede exponer el tráfico de Internet destinado a su dominio a vulnerabilidades de seguridad. También puede presentar otros riesgos.

Para más información, consulte [Protección contra registros de delegación colgantes en Route 53](#) en la Guía para desarrolladores de Amazon Route 53. También puede obtener más información sobre las entradas de DNS colgantes en [Protecciones de dominio mejoradas para solicitudes de Amazon CloudFront](#) en el Blog de seguridad de AWS.

Configuración de entornos de Elastic Beanstalk (avanzado)

Cuando crea un entorno de AWS Elastic Beanstalk, Elastic Beanstalk aprovisiona y configura todos los recursos de AWS necesarios para ejecutar y utilizar la aplicación. Además de configurar los metadatos del entorno y actualizar el comportamiento, puede personalizar estos recursos proporcionando valores para las [opciones de configuración](#). Por ejemplo, es posible que desee agregar una cola de Amazon SQS y una alarma sobre la profundidad de cola o un clúster de Amazon ElastiCache.

La mayoría de las opciones de configuración tienen valores predeterminados que se aplican automáticamente en Elastic Beanstalk. Puede cambiar estos valores predeterminados a través de los archivos de configuración, las configuraciones guardadas, las opciones de la línea de comando o llamando directamente a la API de Elastic Beanstalk. La CLI de EB y la consola de Elastic Beanstalk también aplican los valores recomendados en algunas opciones.

Puede personalizar fácilmente el entorno al mismo tiempo que implementa la versión de la aplicación incluyendo un archivo de configuración con el paquete de código fuente. Cuando personalice el software de la instancia, le resultará más útil utilizar un archivo de configuración que crear una AMI personalizada, ya que no es necesario mantener un conjunto de AMI.

Es posible que, cuando implemente las aplicaciones, quiera personalizar y configurar el software del que depende la aplicación. Estos archivos podrían ser dependencias necesarias para la aplicación (por ejemplo, paquetes adicionales del repositorio yum) o archivos de configuración (por ejemplo, un archivo sustituto de httpd.conf que anule ciertos valores específicos establecidos de forma predeterminada por AWS Elastic Beanstalk).

Temas

- [Opciones de configuración](#)
- [Personalización avanzada de entornos con archivos de configuración \(.ebextensions\)](#)
- [Uso de configuraciones guardadas de Elastic Beanstalk](#)
- [Manifiesto del entorno \(env.yaml\)](#)
- [Uso de una imagen de máquina de Amazon \(AMI\) personalizada](#)
- [Distribución de archivos estáticos](#)
- [Configuración de HTTPS para su entorno de Elastic Beanstalk](#)

Opciones de configuración

Elastic Beanstalk define un gran número de opciones de configuración que puede utilizar para configurar el comportamiento del entorno y los recursos que contiene. Las opciones de configuración se organizan en espacios de nombres como, por ejemplo, `aws:autoscaling:asg`, que define opciones para un grupo de Auto Scaling del entorno.

La consola de Elastic Beanstalk y la CLI de EB establecen las opciones de configuración al crear un entorno, incluidas las opciones que establece explícitamente y [valores recomendados](#) definidos por el cliente. También puede establecer opciones de configuración en las configuraciones guardadas y en los archivos de configuración. Si se ha establecido la misma opción en varias ubicaciones, el valor utilizado se determina según el [orden de prioridad](#).

Los valores de las opciones de configuración están en formato de texto y se guardan antes de que se cree el entorno, se aplican durante la creación del entorno mediante cualquier cliente compatible y se agregan, modifican o eliminan después de crear el entorno. Para obtener un listado detallado de todos los métodos disponibles para trabajar con opciones de configuración en cada una de estas tres fases, consulte los siguientes temas:

- [Ajuste de opciones de configuración antes de crear el entorno](#)
- [Definición de opciones de configuración durante la creación del entorno](#)
- [Ajuste de opciones de configuración después de crear el entorno](#)

Para obtener una lista completa de espacios de nombres y opciones, incluidos los valores predeterminados y admitidos para cada uno, consulte [Opciones generales para todos los entornos](#) y [Opciones específicas de la plataforma](#).

Prioridad

Durante la creación del entorno, las opciones de configuración se aplican desde diferentes orígenes con la siguiente prioridad, de mayor a menor:

- Configuración aplicada directamente al entorno: configuración especificada durante una operación de creación o actualización del entorno en la API de Elastic Beanstalk por cualquier cliente, incluida la consola de Elastic Beanstalk, la CLI de EB, la AWS CLI y los SDK. La consola de Elastic Beanstalk y la CLI de EB aplican también los [valores recomendados](#) para algunas opciones que se aplican en este nivel a menos que se invaliden.

- **Configuraciones guardadas:** los valores de las opciones que no se aplican directamente al entorno se cargan desde una configuración guardada, si se ha especificado.
- **Archivos de configuración (.ebextensions):** los valores de las opciones que no se aplican directamente al entorno y que tampoco se han especificado en una configuración guardada se cargan de los archivos de configuración de la carpeta `.ebextensions` en la raíz del paquete de código fuente de la aplicación.

Los archivos de configuración se ejecutan en orden alfabético. Por ejemplo, `.ebextensions/01run.config` se ejecuta antes que `.ebextensions/02do.config`.

- **Valores predeterminados:** si una opción de configuración tiene un valor predeterminado, este solo se aplica cuando la opción no se ha configurado en ninguno de los niveles anteriores.

Si la misma opción de configuración se define en más de una ubicación, se aplica el valor con la prioridad más alta. Cuando se aplica un valor de una configuración guardada o cuando se aplica un valor directamente al entorno, el valor se almacena como parte de la configuración del entorno. Esta configuración se puede eliminar [con la AWS CLI](#) o [con la CLI de EB](#).

Los valores de los archivos de configuración no se aplican directamente al entorno y no pueden eliminarse sin modificar los archivos de configuración e implementar una nueva versión de la aplicación. Si un valor aplicado con uno de los otros métodos se elimina, el mismo valor se carga de los archivos de configuración del paquete de código fuente.

Suponga, por ejemplo, que establece el número mínimo de instancias en su entorno en 5 durante la creación del entorno, mediante la consola de Elastic Beanstalk, una opción de la línea de comandos o una configuración guardada. El paquete de código fuente de su aplicación también incluye un archivo de configuración que establece el número mínimo de instancias en 2.

Al crear el entorno, Elastic Beanstalk establece la opción `MinSize` en el espacio de nombres `aws:autoscaling:asg` en 5. Si a continuación elimina la opción del entorno de configuración, se carga el valor del archivo de configuración y el número mínimo de instancias se establece en 2. Si después elimina el archivo de configuración del paquete de código fuente y lo vuelve a implementar, Elastic Beanstalk utiliza el valor predeterminado de 1.

Valores recomendados

La interfaz de línea de comandos de Elastic Beanstalk (CLI de EB) y la consola de Elastic Beanstalk proporcionan valores recomendados para algunas opciones de configuración. Estos valores pueden

ser diferentes de los valores predeterminados y se establecen en el nivel de API cuando se crea el entorno. Los valores recomendados permiten a Elastic Beanstalk mejorar la configuración del entorno predeterminada sin realizar cambios incompatibles con versiones anteriores en la API.

Por ejemplo, la CLI de EB y la consola de Elastic Beanstalk definen la opción de configuración del tipo de instancia EC2 (`InstanceType` en el espacio de nombres `aws:autoscaling:launchconfiguration`). Cada cliente proporciona una forma diferente de invalidar la configuración predeterminada. En la consola puede elegir un tipo de instancia diferente en un menú desplegable de la página `Configuration Details` (Detalles de configuración) del asistente `Create New Environment` (Crear un entorno nuevo). Con la CLI de EB, puede utilizar el parámetro `--instance_type` para [eb create](#).

Como los valores recomendados se establecen en el nivel de API, estos valores invalidarán los valores de las mismas opciones que defina en archivos de configuración o configuraciones guardadas. Se definen las siguientes opciones:

Consola de Elastic Beanstalk

- Espacio de nombres: `aws:autoscaling:launchconfiguration`
Nombres de opción: `IamInstanceProfile`, `EC2KeyName`, `InstanceType`
- Espacio de nombres: `aws:autoscaling:updatepolicy:rollingupdate`
Nombres de opción: `RollingUpdateType` y `RollingUpdateEnabled`
- Espacio de nombres: `aws:elasticbeanstalk:application`
Nombre de la opción: `Application Healthcheck URL`
- Espacio de nombres: `aws:elasticbeanstalk:command`
Nombre de opción: `DeploymentPolicy`, `BatchSize` y `BatchSizeType`
- Espacio de nombres: `aws:elasticbeanstalk:environment`
Nombre de la opción: `ServiceRole`
- Espacio de nombres: `aws:elasticbeanstalk:healthreporting:system`
Nombre de opción: `SystemType` y `HealthCheckSuccessThreshold`
- Espacio de nombres: `aws:elasticbeanstalk:sns:topics`
Nombre de la opción: `Notification Endpoint`

- Espacio de nombres: `aws:elasticbeanstalk:sqs`

Nombre de la opción: `HttpConnections`

- Espacio de nombres: `aws:elb:loadbalancer`

Nombre de la opción: `CrossZone`

- Espacio de nombres: `aws:elb:policies`

Nombres de opción: `ConnectionDrainingTimeout` y `ConnectionDrainingEnabled`

CLI DE EB

- Espacio de nombres: `aws:autoscaling:launchconfiguration`

Nombres de opción: `IamInstanceProfile`, `InstanceType`

- Espacio de nombres: `aws:autoscaling:updatepolicy:rollingupdate`

Nombres de opción: `RollingUpdateType` y `RollingUpdateEnabled`

- Espacio de nombres: `aws:elasticbeanstalk:command`

Nombre de opción: `BatchSize` y `BatchSizeType`

- Espacio de nombres: `aws:elasticbeanstalk:environment`

Nombre de la opción: `ServiceRole`

- Espacio de nombres: `aws:elasticbeanstalk:healthreporting:system`

Nombre de la opción: `SystemType`

- Espacio de nombres: `aws:elb:loadbalancer`

Nombre de la opción: `CrossZone`

- Espacio de nombres: `aws:elb:policies`

Nombres de opción: `ConnectionDrainingEnabled`

Ajuste de opciones de configuración antes de crear el entorno

AWS Elastic Beanstalk admite un gran número de [opciones de configuración](#) que le permiten modificar los valores que se aplican a los recursos de su entorno. Varias de estas opciones tienen

valores predeterminados que pueden invalidarse para personalizar su entorno. Otras opciones se pueden configurar para habilitar características adicionales.

Elastic Beanstalk admite dos métodos para guardar valores de opciones de configuración. Se pueden incluir archivos de configuración en formato YAML o JSON en el código fuente de su aplicación en un directorio denominado `.ebextensions` e implementarse como parte del paquete de código fuente de su aplicación. Los archivos de configuración se crean y administran localmente.

Las configuraciones guardadas son plantillas que crea desde un entorno en ejecución o archivo de opciones JSON y almacena en Elastic Beanstalk. Las configuraciones guardadas existentes también se pueden ampliar para crear una nueva configuración.

Note

Los valores definidos en los archivos de configuración y las configuraciones guardadas tienen menos prioridad que los valores configurados durante o después de la creación del entorno, incluidos los valores recomendados aplicados por la consola de Elastic Beanstalk y la [CLI de EB](#). Para obtener más información, consulte [Prioridad](#).

Las opciones se pueden especificar también en un documento JSON y proporcionarse directamente a Elastic Beanstalk al crear o actualizar un entorno con la CLI de EB o la AWS CLI. Las opciones proporcionadas directamente a Elastic Beanstalk de esta forma invalidan todos los demás métodos.

Para ver la lista completa de opciones disponibles, consulte [Opciones de configuración](#).

Métodos

- [Archivos de configuración \(.ebextensions\)](#)
- [Configuraciones guardadas](#)
- [Documento JSON](#)
- [Configuración de la CLI de EB](#)

Archivos de configuración (**.ebextensions**)

Utilice `.ebextensions` para configurar opciones que sean necesarias para que su aplicación funcione y para proporcionar valores predeterminados para otras opciones que se puedan invalidar en un nivel mayor de [prioridad](#). Las opciones especificadas en `.ebextensions` tienen el nivel de prioridad más bajo y se invalidan con los valores de cualquier otro nivel.

Para utilizar archivos de configuración, cree una carpeta con el nombre `.ebextensions` en la parte superior del código fuente de su proyecto. Agrega un archivo con la extensión `.config` y especifique las opciones de la siguiente manera:

```
option_settings:
  - namespace: namespace
    option_name: option name
    value: option value
  - namespace: namespace
    option_name: option name
    value: option value
```

Por ejemplo, el siguiente archivo de configuración establece la URL de comprobación de estado de la aplicación en `/health`:

`healthcheckurl.config`

```
option_settings:
  - namespace: aws:elasticbeanstalk:application
    option_name: Application Healthcheck URL
    value: /health
```

En JSON:

```
{
  "option_settings" :
    [
      {
        "namespace" : "aws:elasticbeanstalk:application",
        "option_name" : "Application Healthcheck URL",
        "value" : "/health"
      }
    ]
}
```

Esto configura el balanceador de carga de Elastic Load Balancing de su entorno de Elastic Beanstalk para que realice una solicitud HTTP a la ruta `/health` a cada instancia EC2 con el fin de determinar si está en buen estado o no.

Note

YAML usa la sangría uniforme. Utilice el mismo nivel de sangría cuando sustituya el contenido en el archivo de configuración de ejemplo y asegúrese de que el editor de texto utiliza espacios para la sangría, no tabuladores.

Incluya el directorio `.ebextensions` en el [paquete de código fuente de la aplicación](#) e impleméntelo en un entorno de Elastic Beanstalk nuevo o existente.

Los archivos de configuración admiten varias secciones además de `option_settings` para personalizar el software y los archivos que se ejecutan en los servidores de su entorno. Para obtener más información, consulte [.Ebextensions](#).

Configuraciones guardadas

Cree una configuración guardada para guardar los valores que ha aplicado a un entorno existente durante o después de la creación del entorno mediante la consola de Elastic Beanstalk, la CLI de EB o la AWS CLI. Las configuraciones guardadas pertenecen a una aplicación y se pueden aplicar a entornos nuevos o existentes de dicha aplicación.

Clientes

- [Consola de Elastic Beanstalk](#)
- [CLI DE EB](#)
- [AWS CLI](#)

Consola de Elastic Beanstalk

Para crear una configuración guardada (consola de Elastic Beanstalk)

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. Elija Actions (Acciones) y, a continuación, seleccione Save configuration (Guardar configuración).
4. Utilice el cuadro de diálogo en pantalla para completar la acción.

Las configuraciones guardadas se almacenan en el bucket de S3 de Elastic Beanstalk en una carpeta en la que está incluida el nombre de la aplicación. Por ejemplo, las configuraciones de una aplicación llamada my-app en la región us-west-2 para el número de cuenta 123456789012 se encuentran en `s3://elasticbeanstalk-us-west-2-123456789012/resources/templates/my-app`.

CLI DE EB

La [CLI de EB](#) proporciona también subcomandos para interactuar con las configuraciones guardadas en [eb config](#):

Para crear una configuración guardada (CLI de EB)

1. Guarde la configuración actual del entorno asociado:

```
~/project$ eb config save --cfg my-app-v1
```

La CLI de EB guarda la configuración en `~/project/.elasticbeanstalk/saved_configs/my-app-v1.cfg.yml`

2. Modifique la configuración guardada localmente en caso necesario.
3. Cargue la configuración guardada en S3:

```
~/project$ eb config put my-app-v1
```

AWS CLI

Creación de una configuración guardada desde un entorno en ejecución con `aws elasticbeanstalk create-configuration-template`

Para crear una configuración guardada (AWS CLI)

1. Identifique el ID de entorno de su entorno de Elastic Beanstalk con `describe-environments`:

```
$ aws elasticbeanstalk describe-environments --environment-name my-env
{
  "Environments": [
    {
      "ApplicationName": "my-env",
      "EnvironmentName": "my-env",
      "VersionLabel": "89df",
      "Status": "Ready",
      "Description": "Environment created from the EB CLI using \"eb create
      \",
      "EnvironmentId": "e-vcghmm2zwk",
      "EndpointURL": "awseb-e-v-AWSEBLoa-1JUM8159RA11M-43V6ZI1194.us-
      west-2.elb.amazonaws.com",
      "SolutionStackName": "64bit Amazon Linux 2015.03 v2.0.2 running Multi-
      container Docker 1.7.1 (Generic)",
      "CNAME": "my-env-nfptuqaper.elasticbeanstalk.com",
      "Health": "Green",
      "AbortableOperationInProgress": false,
      "Tier": {
        "Version": " ",
        "Type": "Standard",
        "Name": "WebServer"
      },
      "HealthStatus": "Ok",
      "DateUpdated": "2015-10-01T00:24:04.045Z",
      "DateCreated": "2015-09-30T23:27:55.768Z"
    }
  ]
}
```

2. Guarde la configuración actual del entorno con `create-configuration-template`:

```
$ aws elasticbeanstalk create-configuration-template --environment-id e-vcghmm2zwk
--application-name my-app --template-name v1
```

Elastic Beanstalk guarda la configuración en su bucket de Elastic Beanstalk en Amazon S3.

Documento JSON

Si utiliza la AWS CLI para crear y actualizar entornos, también puede proporcionar opciones de configuración en formato JSON. Una biblioteca de archivos de configuración en JSON es útil si utiliza la AWS CLI para crear y administrar entornos.

Por ejemplo, el siguiente documento JSON establece la URL de comprobación de estado de la aplicación en `/health`:

```
~/ebconfigs/healthcheckurl.json
```

```
[
  {
    "Namespace": "aws:elasticbeanstalk:application",
    "OptionName": "Application Healthcheck URL",
    "Value": "/health"
  }
]
```

Configuración de la CLI de EB

Además de admitir configuraciones guardadas y la configuración directa del entorno con comandos `eb config`, la CLI de EB tiene un archivo de configuración con una opción denominada `default_ec2_keyname` que puede utilizar para especificar un par de claves de Amazon EC2 para el acceso SSH a las instancias de su entorno. La CLI de EB utiliza esta opción para establecer la opción de configuración `EC2KeyName` en el espacio de nombres `aws:autoscaling:launchconfiguration`.

```
~/workspace/my-app/.elasticbeanstalk/config.yml
```

```
branch-defaults:
  master:
    environment: my-env
  develop:
    environment: my-env-dev
deploy:
  artifact: ROOT.war
global:
  application_name: my-app
  default_ec2_keyname: my-keypair
  default_platform: Tomcat 8 Java 8
```

```
default_region: us-west-2
profile: null
sc: git
```

Definición de opciones de configuración durante la creación del entorno

Cuando crea un entorno de AWS Elastic Beanstalk con la consola de Elastic Beanstalk, la CLI de EB, la AWS CLI, un SDK o la API de Elastic Beanstalk, puede proporcionar valores para las opciones de configuración con el fin de personalizar su entorno y los recursos de AWS que se lanzan en él.

Para cualquier otra cosa distinta de un cambio de configuración puntual, puede [almacenar los archivos de configuración](#) localmente en su paquete de código fuente o en Amazon S3.

Este tema incluye procedimientos para todos los métodos de definición de opciones de configuración durante la creación del entorno.

Clientes

- [En la consola de Elastic Beanstalk.](#)
- [Uso de la CLI de EB](#)
- [Uso de AWS CLI](#)

En la consola de Elastic Beanstalk.

Cuando crea un entorno de Elastic Beanstalk en la consola de Elastic Beanstalk, puede proporcionar las opciones de configuración mediante archivos de configuración, configuraciones guardadas y formularios del asistente Create New Environment (Crear nuevo entorno).

Métodos

- [Uso de archivos de configuración \(.ebextensions\)](#)
- [Uso de una configuración guardada](#)
- [Uso del asistente de creación de nuevo entorno](#)

Uso de archivos de configuración (**.ebextensions**)

Incluya los archivos `.config` del [paquete de código fuente de la aplicación](#) en una carpeta denominada `.ebextensions`.

Para obtener información detallada sobre los archivos de configuración, consulte [.Ebextensions](#).

```
~/workspace/my-app-v1.zip
|-- .ebextensions
|   |-- environmentvariables.config
|   `-- healthcheckurl.config
|-- index.php
`-- styles.css
```

Cargue el paquete de origen en Elastic Beanstalk normalmente, durante la [creación del entorno](#).

La consola de Elastic Beanstalk aplica los [valores recomendados](#) para algunas opciones de configuración y dispone de campos de formulario para otras. Las opciones configuradas por la consola de Elastic Beanstalk se aplican directamente al entorno e invalidan los ajustes de los archivos de configuración.

Uso de una configuración guardada

Cuando crea un nuevo entorno usando la consola de Elastic Beanstalk, uno de los primeros pasos consiste en elegir una configuración. La configuración puede ser una [predefined configuration \(configuración predeterminada\)](#) (por lo general, la última versión de una plataforma como PHP o Tomcat) o puede ser una saved configuration (configuración guardada).

Para aplicar una configuración guardada durante la creación del entorno (consola de Elastic Beanstalk)

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Applications (Aplicaciones) y, a continuación, elija el nombre de la aplicación de la lista.

Note

Si tiene muchas aplicaciones, utilice la barra de búsqueda para filtrar la lista de aplicaciones.

3. En el panel de navegación, busque el nombre de la aplicación y elija Saved configurations (Configuraciones guardadas).
4. Seleccione la configuración guardada que desea aplicar y, a continuación, elija Launch environment (Iniciar entorno).
5. Avance por el asistente para crear su entorno.

Las configuraciones guardadas son específicas de la aplicación. Consulte [Configuraciones guardadas](#) para obtener información detallada sobre cómo crear configuraciones guardadas.

Uso del asistente de creación de nuevo entorno

La mayoría de las opciones de configuración estándar se presentan en la página Configure more options (Configurar más opciones) del [Create New Environment wizard \(Asistente Crear nuevo entorno\)](#). Si crea una base de datos de Amazon RDS o configura una VPC para su entorno, hay disponibles otras opciones de configuración para esos recursos.

Para establecer opciones de configuración durante la creación del entorno (consola de Elastic Beanstalk)

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Applications (Aplicaciones).
3. Elija o [cree](#) una aplicación.
4. Elija Actions (Acciones) y, a continuación, seleccione Create environment (Crear entorno).
5. Continúe a través del asistente y elija Configure more options (Configurar más opciones).
6. Seleccione cualquiera de los configuration presets (preajustes de configuración) y, a continuación, elija Edit (Editar) en una o varias de las categorías de configuración para cambiar un grupo de opciones de configuración relacionadas.
7. Cuando haya terminado de seleccionar opciones, seleccione Create environment (Crear entorno).

Todas las opciones que define en el asistente para crear un nuevo entorno se establecen directamente en el entorno e invalidan los valores de las opciones en las configuraciones guardadas o archivos de configuración (.ebextensions) que aplique. Puede eliminar los valores después de crear el entorno usando la [CLI de EB](#) o la [AWS CLI](#) para permitir que se apliquen los valores de las configuraciones guardadas o los archivos de configuración.

Para obtener información acerca del nuevo asistente de entorno, consulte [El asistente de creación de nuevo entorno](#).

Uso de la CLI de EB

Métodos

- [Uso de archivos de configuración \(.ebextensions\)](#)
- [Uso de configuraciones guardadas](#)
- [Uso de las opciones de la línea de comandos](#)

Uso de archivos de configuración (.ebextensions)

Incluya los archivos `.config` en la carpeta del proyecto situada bajo `.ebextensions` para implementarlos con el código de la aplicación.

Para obtener información detallada sobre los archivos de configuración, consulte [.Ebextensions](#).

```
~/workspace/my-app/  
|-- .ebextensions  
|   |-- environmentvariables.config  
|   `-- healthcheckurl.config  
|-- .elasticbeanstalk  
|   `-- config.yml  
|-- index.php  
`-- styles.css
```

Cree su entorno e implemente el código fuente con `eb create`.

```
~/workspace/my-app$ eb create my-env
```

Uso de configuraciones guardadas

Para aplicar una configuración guardada al crear un entorno con `eb create`, utilice la opción `--cfg`.

```
~/workspace/my-app$ eb create --cfg savedconfig
```

Puede almacenar la configuración guardada en la carpeta del proyecto o en la ubicación de almacenamiento de Elastic Beanstalk en Amazon S3. En el ejemplo anterior, la CLI de EB primero busca un archivo de configuración guardada denominado `savedconfig.cfg.yml` en la carpeta `.elasticbeanstalk/saved_configs/`. No incluya las extensiones de nombre de archivo (`.cfg.yml`) cuando aplique una configuración guardada con `--cfg`.

```
~/workspace/my-app/  
|-- .ebextensions  
|   `-- healthcheckurl.config
```

```
|-- .elasticbeanstalk
|   |-- saved_configs
|   |   `-- savedconfig.cfg.yml
|   `-- config.yml
|-- index.php
`-- styles.css
```

Si la CLI de EB no encuentra la configuración localmente, la busca en la ubicación de almacenamiento de Elastic Beanstalk en Amazon S3. Consulte para obtener información detallada sobre cómo crear, editar y cargar configuraciones guardada [Configuraciones guardadas](#).

Uso de las opciones de la línea de comandos

El comando `eb create` de la CLI de EB tiene varias [opciones](#) que puede utilizar para definir opciones de configuración durante la creación del entorno. Puede utilizar estas opciones para agregar una base de datos de RDS al entorno, configurar una VPC o invalidar los [valores recomendados](#).

Por ejemplo, la CLI de EB utiliza el tipo de instancia `t2.micro` de forma predeterminada. Para elegir un tipo de instancia diferente, utilice la opción `--instance_type`.

```
$ eb create my-env --instance_type t2.medium
```

Para crear una instancia de base de datos de Amazon RDS y asociarla a su entorno, utilice las opciones `--database`.

```
$ eb create --database.engine postgres --database.username dbuser
```

Si omite el nombre del entorno, la contraseña de la base de datos y otros parámetros necesarios para crear su entorno, la CLI de EB le pedirá que los introduzca.

Consulte [eb create](#) para ver una lista completa de las opciones disponibles y ejemplos de uso.

Uso de AWS CLI

Cuando utiliza el comando `create-environment` para crear un entorno de Elastic Beanstalk con la AWS CLI, la AWS CLI no aplica ninguno de los [valores recomendados](#). Todas las opciones de configuración se definen en los archivos de configuración del paquete de código fuente que especifique.

Métodos

- [Uso de archivos de configuración \(.ebextensions\)](#)
- [Uso de una configuración guardada](#)
- [Uso de las opciones de la línea de comandos](#)

Uso de archivos de configuración (.ebextensions)

Para aplicar los archivos de configuración en un entorno que crea con la AWS CLI, incluya estos archivos en el paquete de código fuente de la aplicación que cargó en Amazon S3.

Para obtener información detallada sobre los archivos de configuración, consulte [.Ebextensions](#).

```
~/workspace/my-app-v1.zip
|-- .ebextensions
|   |-- environmentvariables.config
|   |-- healthcheckurl.config
|-- index.php
`-- styles.css
```

Para cargar un paquete de código fuente de la aplicación y crear un entorno con la AWS CLI

1. Si aún no tiene un bucket de Elastic Beanstalk en Amazon S3, cree uno con `create-storage-location`.

```
$ aws elasticbeanstalk create-storage-location
{
  "S3Bucket": "elasticbeanstalk-us-west-2-123456789012"
}
```

2. Cargue el paquete de código fuente de la aplicación en Amazon S3.

```
$ aws s3 cp sourcebundle.zip s3://elasticbeanstalk-us-west-2-123456789012/my-app/sourcebundle.zip
```

3. Cree la versión de la aplicación.

```
$ aws elasticbeanstalk create-application-version --application-name my-app --
version-label v1 --description MyAppv1 --source-bundle S3Bucket="elasticbeanstalk-
us-west-2-123456789012",S3Key="my-app/sourcebundle.zip" --auto-create-application
```

4. Cree el entorno.

```
$ aws elasticbeanstalk create-environment --application-name my-app --environment-name my-env --version-label v1 --solution-stack-name "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 8 Java 8"
```

Uso de una configuración guardada

Para aplicar una configuración guardada al entorno durante su creación, utilice el parámetro `--template-name`.

```
$ aws elasticbeanstalk create-environment --application-name my-app --environment-name my-env --template-name savedconfig --version-label v1
```

Cuando especifique una configuración guardada, no especifique un nombre de la pila de soluciones. Las configuraciones guardadas ya especifican una pila de soluciones y Elastic Beanstalk devolverá un error si intenta utilizar ambas opciones.

Uso de las opciones de la línea de comandos

Utilice el parámetro `--option-settings` para especificar las opciones de configuración en formato JSON.

```
$ aws elasticbeanstalk create-environment --application-name my-app --environment-name my-env --version-label v1 --template-name savedconfig --option-settings '[
  {
    "Namespace": "aws:elasticbeanstalk:application",
    "OptionName": "Application Healthcheck URL",
    "Value": "/health"
  }
]
```

Para cargar el JSON desde un archivo, utilice el prefijo `file://`.

```
$ aws elasticbeanstalk create-environment --application-name my-app --environment-name my-env --version-label v1 --template-name savedconfig --option-settings file://healthcheckurl.json
```

Elastic Beanstalk aplica los valores de las opciones que especifica con la opción `--option-settings` directamente a su entorno. Si se especifican las mismas opciones en una configuración guardada o archivo de configuración, `--option-settings` invalida esos valores.

Ajuste de opciones de configuración después de crear el entorno

Puede modificar los valores de configuración de un entorno en ejecución aplicando configuraciones guardadas, cargando un nuevo paquete de código fuente con archivos de configuración (.ebextensions) o utilizando un documento JSON. La CLI de EB y la consola de Elastic Beanstalk también tienen una funcionalidad específica para los clientes que permite definir y actualizar las opciones de configuración.

Cuando se define o se cambia una opción de configuración, en función de la importancia del cambio, se puede activar una actualización completa del entorno. Por ejemplo, los cambios que se realizan en [aws:autoscaling:launchconfiguration](#), como InstanceType, requieren que las instancias de Amazon EC2 del entorno se aprovisionen de nuevo. Esto desencadena una [actualización continua](#). Otros cambios de configuración pueden aplicarse sin interrupciones ni reaprovisionamientos.

Puede quitar valores de configuración de un entorno con los comandos de la CLI de EB o la AWS CLI. Si se elimina una opción que se estableció directamente en un entorno en el nivel de una API, se aplicarán las opciones de los archivos de configuración, que de otra forma permanecerían ocultas por los valores que se aplicaron directamente en el entorno.

Los ajustes en configuraciones guardadas y archivos de configuración se pueden anular definiendo la misma opción directamente en el entorno con uno de los otros métodos de configuración. Sin embargo, solo se pueden eliminar completamente aplicando una configuración guardada actualizada o un archivo de configuración. Cuando una opción no está establecida en una configuración guardada, en un archivo de configuración o directamente en el entorno, se aplica el valor predeterminado, si existe. Para obtener más información, consulte [Prioridad](#).

Clientes

- [La consola de Elastic Beanstalk](#)
- [CLI de EB](#)
- [El AWS CLI](#)

La consola de Elastic Beanstalk

Puede actualizar los valores de configuración de la consola de Elastic Beanstalk implementando un paquete de código fuente de la aplicación que contenga los archivos de configuración, aplicando una configuración guardada o modificando directamente el entorno con la página Configuration (Configuración) de la consola de administración del entorno.

Métodos

- [Uso de archivos de configuración \(.ebextensions\)](#)
- [Uso de una configuración guardada](#)
- [Usar la consola de Elastic Beanstalk](#)

Uso de archivos de configuración (**.ebextensions**)

Actualice los archivos de configuración del directorio de código fuente, cree un nuevo paquete de código fuente e implemente la nueva versión en el entorno de Elastic Beanstalk para aplicar los cambios.

Para obtener información detallada sobre los archivos de configuración, consulte [.Ebextensions](#).

Para implementar un paquete de código fuente

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En la página de información general del entorno, elija Upload and deploy (Cargar e implementar).
4. Utilice el cuadro de diálogo en pantalla para cargar el paquete de código fuente.
5. Elija Deploy (Implementar).
6. Cuando se complete la implementación, seleccione la URL del para abrir el sitio web en una nueva pestaña.

Los cambios realizados en los archivos de configuración no sustituyen a las opciones de las configuraciones guardadas ni a los valores que se aplican directamente en el entorno en el nivel de la API. Consulte [Prioridad](#) para obtener más información.

Uso de una configuración guardada

Utilice una configuración guardada en un entorno en ejecución para aplicar los valores definidos en ella.

Para aplicar una configuración guardada a un entorno en ejecución (consola de Elastic Beanstalk)

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Applications (Aplicaciones) y, a continuación, elija el nombre de la aplicación de la lista.

Note

Si tiene muchas aplicaciones, utilice la barra de búsqueda para filtrar la lista de aplicaciones.

3. En el panel de navegación, busque el nombre de la aplicación y elija Saved configurations (Configuraciones guardadas).
4. Seleccione la configuración guardada que desea aplicar y, a continuación, elija Load (Cargar).
5. Seleccione un entorno y, a continuación, elija Load (Cargar).

Los valores definidos en una configuración anulan a los de los archivos de configuración, aunque no tienen prioridad frente a los valores configurados a través de la consola de administración del entorno.

Consulte [Configuraciones guardadas](#) para obtener información detallada sobre cómo crear configuraciones guardadas.

Usar la consola de Elastic Beanstalk

La consola de Elastic Beanstalk dispone de varias opciones de configuración en la página Configuration (Configuración) de cada entorno.

Para cambiar las opciones de configuración en un entorno en ejecución (consola de Elastic Beanstalk)

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.

- En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

- En el panel de navegación, elija Configuration (Configuración).
- Busque la página de configuración que desea editar:
 - Si aparece la opción en la que está interesado o sabe de qué categoría de configuración se trata, elija Edit (Editar) en su categoría de configuración.
 - Para buscar una opción, active Table view (Vista de tabla) y escriba términos de búsqueda en el campo de búsqueda. A medida que escribe, la lista se reduce y muestra solo las opciones que coinciden con los términos de búsqueda.

Cuando vea la opción que está buscando, elija Edit (Editar) en la categoría de configuración que contiene.



- Cambie la configuración y, a continuación, elija Save (Guardar).
- Repita los dos pasos anteriores en las categorías de configuración adicionales según sea necesario.
- Seleccione Apply (Aplicar).

Los cambios de las opciones de configuración que se realizan a través de la consola de administración del entorno se aplican directamente al entorno. Estos cambios invalidan los valores de

las mismas opciones en los archivos de configuración o en las configuraciones guardadas. Consulte [Prioridad](#) para obtener más información.

Para obtener más información sobre cómo cambiar las opciones de configuración en un entorno de ejecución a través de la consola de Elastic Beanstalk, consulte los temas incluidos en [Configuración de entornos de Elastic Beanstalk](#).

CLI de EB

Puede actualizar los valores de configuración con la CLI de EB implementando un código fuente que contenga los archivos de configuración, aplicando los valores de una configuración guardada o modificando directamente la configuración del entorno con el comando `eb config`.

Métodos

- [Uso de archivos de configuración \(.ebextensions\)](#)
- [Uso de una configuración guardada](#)
- [Uso eb config](#)
- [Uso eb setenv](#)

Uso de archivos de configuración (**.ebextensions**)

Incluya los archivos `.config` en la carpeta del proyecto situada bajo `.ebextensions` para implementarlos con el código de la aplicación.

Para obtener información detallada sobre los archivos de configuración, consulte [.Ebextensions](#).

```
~/workspace/my-app/  
|-- .ebextensions  
|   |-- environmentvariables.config  
|   `-- healthcheckurl.config  
|-- .elasticbeanstalk  
|   `-- config.yml  
|-- index.php  
`-- styles.css
```

Implemente el código fuente con `eb deploy`.

```
~/workspace/my-app$ eb deploy
```

Uso de una configuración guardada

Puede utilizar el comando `eb config` para aplicar una configuración guardada en un entorno en ejecución. Utilice la opción `--cfg` con el nombre de la configuración guardada para aplicar la configuración al entorno.

```
$ eb config --cfg v1
```

En este ejemplo, `v1` es el nombre de un [archivo de configuración que se creó y guardó anteriormente](#).

Los valores que se aplican a un entorno con este comando anulan los valores que se aplicaron durante la creación del entorno y los valores definidos en los archivos de configuración del paquete de código fuente de la aplicación.

Uso `eb config`

El comando `eb config` de la CLI de EB le permite establecer y eliminar valores de configuración directamente en el entorno a través de un editor de texto.

Cuando ejecuta `eb config`, la CLI de EB muestra la configuración que se aplica al entorno desde todos los orígenes, incluidos los archivos de configuración, las configuraciones guardadas, los valores recomendados, las opciones definidas directamente en el entorno y los valores predeterminados de la API.

Note

`eb config` no muestra las propiedades del entorno. Para configurar las propiedades del entorno que se pueden consultar desde la aplicación, utilice [`eb setenv`](#).

En el siguiente ejemplo se muestra la configuración aplicada en el espacio de nombres `aws:autoscaling:launchconfiguration`. Esta configuración incluye:

- Dos valores recomendados, para `IamInstanceProfile` y `InstanceType`, aplicadas por CLI de EB durante la creación del entorno.
- La opción `EC2KeyName`, definida directamente en el entorno durante la creación en función de la configuración del repositorio.
- Valores predeterminados del API para las demás opciones.

```
ApplicationName: tomcat
DateUpdated: 2015-09-30 22:51:07+00:00
EnvironmentName: tomcat
SolutionStackName: 64bit Amazon Linux 2015.03 v2.0.1 running Tomcat 8 Java 8
settings:
...
aws:autoscaling:launchconfiguration:
  BlockDeviceMappings: null
  EC2KeyName: my-key
  IamInstanceProfile: aws-elasticbeanstalk-ec2-role
  ImageId: ami-1f316660
  InstanceType: t2.micro
...
```

Para configurar o cambiar las opciones de configuración con `eb config`

1. Ejecute `eb config` para ver la configuración del entorno.

```
~/workspace/my-app/$ eb config
```

2. Cambie los valores de configuración utilizando el editor de texto predeterminado.

```
aws:autoscaling:launchconfiguration:
  BlockDeviceMappings: null
  EC2KeyName: my-key
  IamInstanceProfile: aws-elasticbeanstalk-ec2-role
  ImageId: ami-1f316660
  InstanceType: t2.medium
```

3. Guarde el archivo de configuración temporal y salga.
4. La CLI de EB actualiza la configuración del entorno.

Si los valores de configuración se establecen con `eb config`, se anulan los valores de todos los demás orígenes.

También puede eliminar opciones del entorno con `eb config`.

Para eliminar las opciones de configuración (CLI de EB)

1. Ejecute `eb config` para ver la configuración del entorno.

```
~/workspace/my-app/$ eb config
```

2. Sustituya los valores mostrados por la cadena `null`. También puede eliminar toda la línea que contiene la opción que desea eliminar.

```
aws:autoscaling:launchconfiguration:
  BlockDeviceMappings: null
  EC2KeyName: my-key
  IamInstanceProfile: aws-elasticbeanstalk-ec2-role
  ImageId: ami-1f316660
  InstanceType: null
```

3. Guarde el archivo de configuración temporal y salga.
4. La CLI de EB actualiza la configuración del entorno.

Si elimina las opciones del entorno con `eb config`, se activan las mismas opciones de los archivos de configuración del paquete de código fuente de la aplicación. Consulte [Precedence](#) para obtener más información.

Uso `eb setenv`

Para establecer las propiedades de entorno con la CLI de EB, utilice `eb setenv`.

```
~/workspace/my-app/$ eb setenv ENVVAR=TEST
INFO: Environment update is starting.
INFO: Updating environment my-env's configuration settings.
INFO: Environment health has transitioned from Ok to Info. Command is executing on all
instances.
INFO: Successfully deployed new configuration to environment.
```

Este comando establece las propiedades de entorno en el espacio de nombres [aws:elasticbeanstalk:application:environment](#). Las propiedades de entorno establecidas con `eb setenv` están disponibles en la aplicación tras un breve periodo de tiempo.

Consulte las propiedades configuradas en el entorno con `eb printenv`.

```
~/workspace/my-app/$ eb printenv
Environment Variables:
  ENVVAR = TEST
```


El AWS CLI

Puede actualizar los valores de configuración con la AWS CLI implementando un paquete de código fuente que contenga los archivos de configuración, aplicando una configuración guardada en remoto o modificando el entorno directamente con el comando `aws elasticbeanstalk update-environment`.

Métodos

- [Uso de archivos de configuración \(.ebextensions\)](#)
- [Uso de una configuración guardada](#)
- [Uso de las opciones de la línea de comandos](#)

Uso de archivos de configuración (.ebextensions)

Para aplicar los archivos de configuración a un entorno en ejecución con la AWS CLI, inclúyalos en el paquete de código fuente de la aplicación que cargó en Amazon S3.

Para obtener información detallada sobre los archivos de configuración, consulte [.Ebextensions](#).

```
~/workspace/my-app-v1.zip
|-- .ebextensions
|   |-- environmentvariables.config
|   `-- healthcheckurl.config
|-- index.php
`-- styles.css
```

Para cargar un paquete de código fuente de la aplicación y aplicarlo a un entorno en ejecución (AWS CLI)

1. Si aún no tiene un bucket de Elastic Beanstalk en Amazon S3, cree uno con `create-storage-location`:

```
$ aws elasticbeanstalk create-storage-location
{
  "S3Bucket": "elasticbeanstalk-us-west-2-123456789012"
}
```

2. Cargue el paquete de código fuente de la aplicación en Amazon S3.

```
$ aws s3 cp sourcebundlev2.zip s3://elasticbeanstalk-us-west-2-123456789012/my-app/sourcebundlev2.zip
```

3. Cree la versión de la aplicación.

```
$ aws elasticbeanstalk create-application-version --application-name my-app --version-label v2 --description MyAppv2 --source-bundle S3Bucket="elasticbeanstalk-us-west-2-123456789012",S3Key="my-app/sourcebundlev2.zip"
```

4. Actualice el entorno.

```
$ aws elasticbeanstalk update-environment --environment-name my-env --version-label v2
```

Uso de una configuración guardada

Puede aplicar una configuración guardada en un entorno en ejecución con la opción `--template-name` del comando `aws elasticbeanstalk update-environment`.

La configuración guardada debe estar en el bucket de Elastic Beanstalk, en una ruta con el mismo nombre que la aplicación que se encuentra en `resources/templates`. Por ejemplo, la plantilla `v1` de la aplicación `my-app` para la región EE. UU. Oeste (Oregón) (`us-west-2`) de la cuenta `123456789012` se encuentra en `s3://elasticbeanstalk-us-west-2-123456789012/resources/templates/my-app/v1`

Para aplicar una configuración guardada en un entorno en ejecución (AWS CLI)

- Especifique la configuración guardada en una llamada a `update-environment` con la opción `--template-name`.

```
$ aws elasticbeanstalk update-environment --environment-name my-env --template-name v1
```

Elastic Beanstalk sitúa las configuraciones guardadas en esta ubicación cuando se crean con `aws elasticbeanstalk create-configuration-template`. También puede modificar las configuraciones guardadas localmente y guardarlas usted mismo en esta ubicación.

Uso de las opciones de la línea de comandos

Para cambiar las opciones de configuración con un documento JSON (AWS CLI)

1. Defina los valores de configuración con formato JSON en un archivo local.
2. Ejecute `update-environment` con la opción `--option-settings`.

```
$ aws elasticbeanstalk update-environment --environment-name my-env --option-  
settings file://~/ebconfigs/as-zero.json
```

En este ejemplo, `as-zero.json` define las opciones que configuran el entorno con el valor cero para el número mínimo y máximo de instancias. De este modo, las instancias del entorno pueden detenerse sin necesidad de terminar el entorno.

~/ebconfigs/as-zero.json

```
[  
  {  
    "Namespace": "aws:autoscaling:asg",  
    "OptionName": "MinSize",  
    "Value": "0"  
  },  
  {  
    "Namespace": "aws:autoscaling:asg",  
    "OptionName": "MaxSize",  
    "Value": "0"  
  },  
  {  
    "Namespace": "aws:autoscaling:updatepolicy:rollingupdate",  
    "OptionName": "RollingUpdateEnabled",  
    "Value": "false"  
  }  
]
```

Note

Si los valores de configuración se establecen con `update-environment`, se anulan los valores de todos los demás orígenes.

También puede eliminar opciones del entorno con `update-environment`.

Para eliminar las opciones de configuración (AWS CLI)

- Ejecute el comando `update-environment` con la opción `--options-to-remove`.

```
$ aws elasticbeanstalk update-environment --environment-name my-env --options-to-remove Namespace=aws:autoscaling:launchconfiguration,OptionName=InstanceType
```

Si elimina las opciones del entorno con `update-environment`, se activan las mismas opciones de los archivos de configuración del paquete de código fuente de la aplicación. Si hay una opción que no se ha configurado con ninguno de estos métodos, se aplicará el valor predeterminado de la API, si existe. Consulte [Prioridad](#) para obtener más información.

Opciones generales para todos los entornos

Espacios de nombres

- [aws:autoscaling:asg](#)
- [aws:autoscaling:launchconfiguration](#)
- [aws:autoscaling:scheduledaction](#)
- [aws:autoscaling:trigger](#)
- [aws:autoscaling:updatepolicy:rollingupdate](#)
- [aws:ec2:instances](#)
- [aws:ec2:vpc](#)
- [aws:elasticbeanstalk:application](#)
- [aws:elasticbeanstalk:application:environment](#)
- [aws:elasticbeanstalk:cloudwatch:logs](#)
- [aws:elasticbeanstalk:cloudwatch:logs:health](#)
- [aws:elasticbeanstalk:command](#)
- [aws:elasticbeanstalk:environment](#)
- [aws:elasticbeanstalk:environment:process:default](#)
- [aws:elasticbeanstalk:environment:process:process_name](#)
- [aws:elasticbeanstalk:environment:proxy:staticfiles](#)
- [aws:elasticbeanstalk:healthreporting:system](#)

- [aws:elasticbeanstalk:hostmanager](#)
- [aws:elasticbeanstalk:managedactions](#)
- [aws:elasticbeanstalk:managedactions:platformupdate](#)
- [aws:elasticbeanstalk:monitoring](#)
- [aws:elasticbeanstalk:sns:topics](#)
- [aws:elasticbeanstalk:sqs](#)
- [aws:elasticbeanstalk:trafficsplitting](#)
- [aws:elasticbeanstalk:xray](#)
- [aws:elb:healthcheck](#)
- [aws:elb:loadbalancer](#)
- [aws:elb:listener](#)
- [aws:elb:listener:listener_port](#)
- [aws:elb:policies](#)
- [aws:elb:policies:policy_name](#)
- [aws:elbv2:listener:default](#)
- [aws:elbv2:listener:listener_port](#)
- [aws:elbv2:listenerrule:rule_name](#)
- [aws:elbv2:loadbalancer](#)
- [aws:rds:dbinstance](#)

aws:autoscaling:asg

Configure el grupo de Auto Scaling del entorno. Para obtener más información, consulte [the section called “Grupo de Auto Scaling”](#).

Espacio de nombres: **aws:autoscaling:asg**

Nombre	Descripción	Predeterminado	Valores válidos
Availability Zones	Las zonas de disponibilidad (AZ) son ubicaciones distintas dentro de una AWS región que están diseñadas para aislarse de los errores en otras AZ. Proporcionan conectividad de red	Any	Any Any 1 Any 2

Nombre	Descripción	Predeterminado	Valores válidos
	económica de baja latencia con otras zonas de disponibilidad de la misma región. Elija el número de zonas de disponibilidad de las instancias.		Any 3
Cooldown	Los periodos de recuperación (cooldown) ayudan a evitar que Amazon EC2 Auto Scaling inicie nuevas actividades de escala antes de que los efectos de las actividades anteriores sean visibles. El periodo de recuperación (cooldown) es la cantidad de tiempo, en segundos, que una actividad de escala tarda en completarse antes de que pueda comenzar la siguiente.	360	De 0 a 10000
Custom Availability Zones	Defina las zonas de disponibilidad de las instancias.	Ninguno	us-east-1a us-east-1b us-east-1c us-east-1d us-east-1e eu-central-1-1

Nombre	Descripción	Predeterminado	Valores válidos
EnableCapacityRebalancing	<p>Especifica si se habilita la característica de reequilibrio de la capacidad para instancias de spot en el grupo de Auto Scaling. Para obtener más información, consulte Reequilibrio de la capacidad en la Guía del usuario de Amazon EC2 Auto Scaling.</p> <p>Esta opción solo es relevante cuando <code>EnableSpot</code> se configura <code>true</code> en el espacio de nombres <code>aws:ec2:instances</code> y hay al menos una instancia de spot en el grupo de Auto Scaling.</p>	false	true false
MinSize	El número mínimo de instancias que desea en el grupo de Auto Scaling.	1	1 De a 10000
MaxSize	El número máximo de instancias que desea en el grupo de Auto Scaling.	4	De 1 a 10000


aws:autoscaling:launchconfiguration


Configure las instancias de Amazon Elastic Compute Cloud (Amazon EC2) para su entorno.

Las instancias que se usan para su entorno se crean a través de una plantilla de lanzamiento de Amazon EC2 o un recurso de configuración de lanzamiento de un grupo de Auto Scaling. Las siguientes opciones funcionan con los dos tipos de recursos.


Para obtener más información, consulte [the section called “Instancias de Amazon EC2”](#). También puede consultar más información sobre Amazon Elastic Block Store (EBS) en el capítulo [Amazon EBS de la Guía](#) del usuario de Amazon EC2.


Espacio de nombres: **aws:autoscaling:launchconfiguration**

Nombre	Descripción	Predeterminado	Valores válidos
DisableIMDSv1	<p>Establezca esta opción como <code>true</code> para deshabilitar el Servicio de metadatos de instancia versión 1 (IMDSv1).</p> <p>Las instancias del entorno se configuran de forma predeterminada de la siguiente manera, según el sistema operativo de la plataforma:</p> <ul style="list-style-type: none"> Windows server, AL2 y las versiones anteriores: habilita ambas IMDSv1 e IMDSv2 AL2023: habilita solo IMDSv2 <p>Para obtener más información, consulte Configurar el servicio de metadatos de instancia (Amazon Linux) o Configuración del servicio de metadatos de instancia (Windows server).</p>	<p><code>false</code>: plataformas basadas en Windows server, Amazon Linux 2 y versiones anteriores</p> <p><code>true</code>: plataformas basadas en Amazon Linux 2023</p>	<p><code>true</code></p> <p><code>false</code></p>
EC2KeyName	<p>Puede utilizar un par de claves para iniciar sesión de forma segura en su instancia EC2.</p> <div data-bbox="326 1549 889 1873" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>Si utiliza la consola de Elastic Beanstalk para crear un entorno, no podrá establecer esta opción en un archivo de configuración. La consola</p> </div>	Ninguna	


Nombre	Descripción	Predeterminado	Valores válidos
	<p>invalida esta opción con un valor recomendado.</p>		
iamInstanceProfile	<p>Un perfil de instancia permite a los usuarios y AWS servicios AWS Identity and Access Management (IAM) acceder a las credenciales de seguridad temporales para realizar llamadas a la API. AWS Especifique el nombre del perfil de instancia o su ARN.</p> <p>Ejemplos:</p> <ul style="list-style-type: none"> aws-elasticbeanstalk-ec2-role arn:aws:iam::123456789012:instance-profile/aws-elasticbeanstalk-ec2-role <div data-bbox="326 1226 889 1726" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 20px;"> <p> Note</p> <p>Si utiliza la consola de Elastic Beanstalk o la CLI de EB para crear un entorno, no podrá establecer esta opción en un archivo de configuración. La consola y la CLI de EB invalidan esta opción con un valor recomendado.</p> </div>	Ninguno	Nombre de perfil de instancia o ARN.

Nombre	Descripción	Predeterminado	Valores válidos
ImageId	<p>Puede cambiar la imagen de Amazon Machine Image (AMI) predeterminada y especificar su propio ID de AMI personalizado.</p> <p>Ejemplo: ami-1f316660</p>	Ninguna	

Nombre	Descripción	Predeterminado	Valores válidos
InstanceType	<p>El tipo de instancias que se utiliza para ejecutar la aplicación en un entorno de Elastic Beanstalk.</p> <div data-bbox="326 401 889 1753" style="border: 1px solid #f08080; padding: 10px; margin-top: 10px;"> <p> Important</p> <p>La opción InstanceType está obsoleta. Se ha sustituido por la nueva opción InstanceTypes del espacio de nombres aws:ec2:instances, que es más eficaz. Puede utilizar esta nueva opción le permite especificar una lista con uno o varios tipos de instancias para el entorno. El primer valor de esa lista es equivalente al valor de la opción InstanceType que está incluida en el espacio de nombres <code>aws:autoscaling:launchconfiguration</code> que se describe aquí. Recomendamos que especifique los tipos de instancias utilizando la nueva opción. Si se especifica esta nueva opción, tendrá prioridad sobre la anterior. Para obtener más información, consulte the section called “El espacio de nombres aws:ec2:instances”.</p> </div>	Varía según la cuenta y la región.	<p>Un tipo de instancia EC2.</p> <p>Varía según la cuenta, la región y la zona de disponibilidad. Puede obtener una lista de tipos de instancia de Amazon EC2 filtrados por estos valores. Para obtener más información, consulte Tipos de instancias disponibles en la Guía del usuario de Amazon EC2 o Tipos de instancias disponibles en la Guía del usuario de Amazon EC2.</p>

Nombre	Descripción	Predeterminado	Valores válidos
	<p>Los tipos de instancias que están disponibles dependen de las zonas de disponibilidad y la región utilizadas. Si elige una subred, la zona de disponibilidad que contiene esa subred determina el tipo de instancias disponible.</p> <ul style="list-style-type: none"> • Elastic Beanstalk no es compatible con los tipos de instancias de Amazon EC2 Mac. • Para obtener más información sobre las familias y los tipos de instancias de Amazon EC2, consulte Tipos de instancias en la Guía del usuario de Amazon EC2 o Tipos de instancias en la Guía del usuario de Amazon EC2. • Para obtener más información sobre los tipos de instancias disponibles en todas las regiones, consulte Tipos de instancias disponibles en la Guía del usuario de Amazon EC2 o Tipos de instancias disponibles en la Guía del usuario de Amazon EC2. <div data-bbox="326 1465 889 1837" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 20px;"> <p> Note</p> <p>Si utiliza la consola de Elastic Beanstalk o la CLI de EB para crear un entorno, no podrá establecer esta opción en un archivo de configuración. La consola y la CLI de EB</p> </div>		

Nombre	Descripción	Predeterminado	Valores válidos
	invalidan esta opción con un valor recomendado .		

Nombre	Descripción	Predeterminado	Valores válidos
LaunchTemplateTagPropagationEnabled	<p>Utilice <code>true</code> para permitir la propagación de etiquetas de entorno a las plantillas de lanzamiento de recursos específicos aprovisionados en el entorno.</p> <p>Elastic Beanstalk solo puede propagar etiquetas en plantillas de lanzamiento para los siguientes recursos:</p> <ul style="list-style-type: none"> • Volúmenes de EBS • instancias de EC2 • Interfaces de red de EC2 • AWS CloudFormation lance plantillas que definan un recurso <p>Esta restricción existe porque CloudFormation solo se permiten etiquetas en la creación de plantillas para recursos específicos. Para obtener más información, consulte TagSpecification la Guía del AWS CloudFormation usuario.</p> <div style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Important</p> <ul style="list-style-type: none"> • Cambiar el valor de esta opción de <code>false</code> a <code>true</code> para un entorno existente puede suponer un cambio sustancial para las etiquetas que ya existían anteriormente. </div>	<code>false</code>	<code>true</code> <code>false</code>

Nombre	Descripción	Predeterminado	Valores válidos
	<ul style="list-style-type: none"> • Cuando se habilita esta característica, la propagación de las etiquetas requerirá la sustitución de EC2, lo que puede provocar un tiempo de inactividad. Puede habilitar actualizaciones escalonadas para aplicar los cambios de configuración por lotes y evitar el tiempo de inactividad durante el proceso de actualización. Para obtener más información, consulte Cambios de configuración. <p>Para obtener más información sobre las plantillas de lanzamiento, consulte lo siguiente:</p> <ul style="list-style-type: none"> • Launch templates en la Guía del usuario de Amazon EC2 Auto Scaling • Trabajo con plantillas en la Guía del usuario de AWS CloudFormation • Fragmentos de código de plantillas de Elastic Beanstalk en la Guía del usuario de AWS CloudFormation <p>Para obtener más información acerca de esta opción, consulte Propagación de etiquetas para plantillas de lanzamiento.</p>		


Nombre	Descripción	Predeterminado	Valores válidos
MonitoringInterval	El intervalo (en minutos) en el que quieres que se devuelvan CloudWatch las métricas de Amazon.	5 minute	1 minute 5 minute
SecurityGroups	<p>Muestra los grupos de seguridad de Amazon EC2 que se van a asignar a las instancias EC2 del grupo de Auto Scaling para definir las reglas de firewall de las instancias.</p> <p>Puede proporcionar una sola cadena de valores separados por comas que contenga el nombre de los grupos de seguridad de Amazon EC2 existentes o referencias AWS::EC2::Security Group a los recursos creados en la plantilla. En los nombres de los grupos de seguridad se distingue entre mayúsculas y minúsculas.</p> <p>Si utiliza Amazon Virtual Private Cloud (Amazon VPC) con Elastic Beanstalk para que las instancias se lancen dentro de una nube privada virtual (VPC), especifique los identificadores de grupo de seguridad en lugar de los nombres de los grupos de seguridad.</p>	elasticbeanstalk-default	

Nombre	Descripción	Predeterminado	Valores válidos
SSHSrcRestriction	<p>Se utiliza para bloquear el acceso de SSH a un entorno. Por ejemplo, puede bloquear el acceso de SSH a las instancias EC2, de forma que solo los host bastión puedan acceder a las instancias de la subred privada.</p> <p>Esta cadena tiene el siguiente formato:</p> <p><i>protocol, fromPort, toPort, source_restriction</i></p> <p><i>protocolo</i></p> <p>Protocolo de la regla de entrada.</p> <p><i>fromPort</i></p> <p>Número de puerto inicial.</p> <p><i>toPort</i></p> <p>Número de puerto final.</p> <p><i>source_restriction</i></p> <p>El rango de CIDR o el nombre de un grupo de seguridad por el que debe enrutar el tráfico. Si va a especificar un grupo de seguridad desde otra cuenta (solo EC2-Classic; debe estar en la misma región), incluya el ID de la cuenta antes que el nombre de grupo de seguridad. Use el siguiente formato: <i>other_account_id /security_group_name</i> . Si utiliza Amazon Virtual Private Cloud (Amazon VPC)</p>	Ninguno	

Nombre	Descripción	Predeterminado	Valores válidos
	<p>con Elastic Beanstalk para que las instancias se lancen en una nube privada virtual (VPC), deberá especificar un ID de grupo de seguridad en lugar de un nombre de grupo de seguridad.</p> <p>Ejemplo: tcp, 22, 22, 54.240.196.185/32</p> <p>Ejemplo: tcp, 22, 22, my-security-group</p> <p>Ejemplo (EC2 clásico): tcp, 22, 22, 123456789012/their-security-group</p> <p>Ejemplo (VPC): tcp, 22, 22, sg-903004f8</p>		

Nombre	Descripción	Predeterminado	Valores válidos
BlockDeviceMappings	<p>Puede adjuntar más volúmenes de Amazon EBS o volúmenes de almacén de instancias en todas las instancias del grupo de Auto Scaling.</p> <p>Cuando mapea volúmenes de almacén de instancias, solo necesita mapear el nombre del dispositivo a un nombre de volumen. Sin embargo, recomendamos que, al mapear volúmenes de Amazon EBS, especifique además algunos o todos los campos a continuación (cada campo debe separarse por dos puntos):</p> <ul style="list-style-type: none"> • ID de instantánea • tamaño, en GB • eliminar al terminar (<code>true</code> o <code>false</code>) • tipo de almacenamiento (solo para <code>gp3</code>, <code>gp2</code>, <code>standard</code>, <code>st1</code>, <code>sc1</code>, o <code>io1</code>) • IOPS (solo para <code>gp3</code> o <code>io1</code>) • rendimiento (solo para <code>gp3</code>) <p>En el ejemplo siguiente, se adjuntan tres volúmenes de Amazon EBS: un volumen <code>gp2</code> vacío de 100 GB y una instantánea, un volumen <code>io1</code> vacío de 20 GB con 2000 IOPS provisionadas y un volumen de almacén de instancias <code>ephemeral0</code>. Si la instancia lo admite, pueden adjuntarse varios volúmenes de almacén de instancias.</p>	Ninguna	<ul style="list-style-type: none"> • tamaño: debe estar entre 500 y 16384 GiB • rendimiento: debe estar entre 125 y 1000 mebibytes por segundo (Mib/s)

Nombre	Descripción	Predeterminado	Valores válidos
	<code>/dev/sdj=:100:true:gp2,/dev/sdh=snap-51eef269,/dev/sdi=:20:true:io1:2000,/dev/sdb=ephemeral0</code>		
RootVolumeType	Tipo de volumen (magnético, SSD de uso general o SSD de IOPS provisionadas) que se va a utilizar con el volumen raíz de Amazon EBS asociado a las instancias EC2 del entorno.	Varía según la plataforma.	<p>standard para almacenamiento magnético.</p> <p>gp2 o gp3 para SSD de uso general.</p> <p>io1 para SSD de IOPS provisionadas.</p>
RootVolumeSize	<p>La capacidad de almacenamiento del volumen raíz de Amazon EBS en GB totales.</p> <p>Obligatorio si RootVolumeType se establece en SSD de IOPS provisionadas.</p> <p>Por ejemplo, "64".</p>	<p>Varía en función de la plataforma para almacenamiento magnético y SSD de uso general.</p> <p>Ninguno para SSD de IOPS provisionadas.</p>	<p>De 10 a 16384 GB para SSD de uso general y SSD de IOPS provisionadas.</p> <p>De 8 a 1024 GB para almacenamiento magnético.</p>

Nombre	Descripción	Predeterminado	Valores válidos
RootVolumeIOPS	<p>Las operaciones de entrada/salida por segundo (IOPS) deseadas para un volumen raíz SSD de IOPS aprovisionadas o para un volumen de raíz SSD gp3 de uso general.</p> <p>La proporción máxima entre las IOPS y el tamaño del volumen es de 500 a 1. Por ejemplo, un volumen con 3000 IOPS debe tener al menos 6 GiB.</p>	Ninguna	<p>100 a 20000 para los volúmenes raíz de SSD de IOPS aprovisionadas io1.</p> <p>3000 a 16000 para los volúmenes raíz de SSD de gp3 de uso general.</p>
RootVolumeThroughput	<p>El rendimiento deseado de mebibytes por segundo (MiB/s) para aprovisionar el volumen raíz de Amazon EBS asociado a la instancia EC2 del entorno.</p> <div data-bbox="326 1087 889 1354" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>Esta opción solo es aplicable a los tipos de almacenamiento gp3.</p> </div>	Ninguno	De 125 a 1000

aws:autoscaling:scheduledaction

Configure las [acciones programadas](#) para el grupo de Auto Scaling de su entorno. Para cada acción, especifique un `resource_name`, además del nombre de opción, espacio de nombres y valor para cada ajuste. Para ver ejemplos, consulte [El espacio de nombres aws:autoscaling:scheduledaction](#).

Espacio de nombres: **aws:autoscaling:scheduledaction**

Nombre	Descripción	Predeterminado	Valores válidos
StartTime	En el caso de las acciones puntuales, elija la fecha y la hora para ejecutar la acción. En el caso de acciones recurrentes, elija cuándo activar la acción.	Ninguno	Una marca de tiempo ISO-8601 única para todas las acciones de escalado programadas.
EndTime	Fecha y hora futuras (en la zona horaria UTC/GMT) en las que desea que la acción de escala programada deje de repetirse. Si no especifica ninguno EndTime, la acción se repite según la expresión. Recurrence Ejemplo: 2015-04-28T04:07:2Z Cuando una acción programada finaliza, Amazon EC2 Auto Scaling no regresa automáticamente a su configuración anterior. Configure una segunda acción programada para volver a la configuración original según sea necesario.	Ninguno	Una marca de tiempo ISO-8601 única para todas las acciones de escalado programadas.
MaxSize	El recuento de instancia máxima que se debe aplicar cuando se ejecuta la acción.	Ninguno	De 0 a 10000
MinSize	El recuento de instancia mínima que se debe aplicar cuando se ejecuta la acción.	Ninguno	De 0 a 10000
DesiredCapacity	Establezca la capacidad deseada inicial para el grupo de Auto Scaling.	Ninguno	De 0 a 10000

Nombre	Descripción	Predeterminado	Valores válidos
	Después de aplicar la acción programada, los desencadenadores ajustan la capacidad deseada en función de su configuración.		
Recurrence	La frecuencia con la que desea que se produzca la acción programada. Si no se especifica la recurrencia, la acción de escalado se ejecuta una sola vez en el momento especificado en <code>StartTime</code> .	Ninguno	Una expresión cron .
Suspend	Establecer en <code>true</code> para desactivar una acción programada recurrente de forma temporal.	<code>false</code>	<code>true</code> <code>false</code>

aws:autoscaling:trigger

Configure los desencadenadores de escala para el grupo de Auto Scaling de su entorno.


Note

Hay tres opciones en este espacio de nombres que determinan el tiempo que la métrica de un desencadenador puede permanecer más allá de sus límites definidos antes de que el desencadenador se inicie. Estas opciones están relacionadas de la siguiente manera:

$$\text{BreachDuration} = \text{Period} * \text{EvaluationPeriods}$$

Los valores predeterminados de estas opciones (5, 5 y 1, respectivamente) cumplen esta ecuación. Si especifica valores incoherentes, Elastic Beanstalk podría modificar uno de los valores para que la ecuación se siga cumpliendo.

Espacio de nombres: **aws:autoscaling:trigger**

Nombre	Descripción	Predeterminado	Valores válidos
BreachDuration	El periodo de tiempo, en minutos, que una métrica puede sobrepasar el tiempo definido (especificado en <code>UpperThreshold</code> y <code>LowerThreshold</code>) antes de que el disparador se active.	5	De 1 a 600
LowerBreachScaleIncrement	Número de instancias de Amazon EC2 que se van a eliminar cuando se efectúe una actividad de escala.	-1	
LowerThreshold	Si el valor de la medición se sitúa por debajo de este número durante la interrupción, se activa un desencadenador.	2000000	De 0 a 20000000
MeasureName	La métrica que es utilizada para el desencadenador de Auto Scaling. <div data-bbox="464 1186 1010 1850" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p> Note</p> <p><code>HealthyHostCount</code> , <code>UnhealthyHostCount</code> y <code>TargetResponseTime</code> solo son aplicables a entornos con un balanceador de carga dedicado. Estos valores no son valores de métrica válidos para entornos configurados con un balanceador de carga compartido. Para obtener más información acerca de los tipos de balanceador de</p> </div>	NetworkOut	CPUUtilization NetworkIn NetworkOut DiskWriteOps DiskReadBytes DiskReadOps DiskWriteBytes Latency RequestCount

Nombre	Descripción	Predeterminado	Valores válidos
	<p>carga, consulte Balanceador de carga del entorno de Elastic Beanstalk.</p>		<p>HealthyHostCount</p> <p>UnhealthyHostCount</p> <p>TargetResponseTime</p>
Period	<p>Especifica la frecuencia con la que Amazon CloudWatch mide las métricas de tu activador. El valor es el número de minutos entre dos periodos consecutivos.</p>	5	De 1 a 600
EvaluationPeriods	<p>El número de periodos de evaluación consecutivos que se utiliza para determinar si se está produciendo una interrupción.</p>	1	De 1 a 600
Statistic	<p>La estadística que el desencadenador utiliza; por ejemplo, Average.</p>	Average	<p>Minimum</p> <p>Maximum</p> <p>Sum</p> <p>Average</p>


Nombre	Descripción	Predeterminado	Valores válidos
Unit	La unidad de medición del disparador; por ejemplo, Bytes.	Bytes	Seconds Percent Bytes Bits Count Bytes/Second Bits/Second Count/Second None
UpperBreachScaleIncrement	Especifica el número de instancias de Amazon EC2 que se van a agregar cuando se efectúe una actividad de escala.	1	
UpperThreshold	Si el valor de la medición se sitúa por encima de este número durante la interrupción, se activa un desencadenador.	6000000	De 0 a 20000000

aws:autoscaling:updatepolicy:rollingupdate


Configure las actualizaciones continuas del grupo de Auto Scaling del entorno.

Espacio de nombres: **aws:autoscaling:updatepolicy:rollingupdate**

Nombre	Descripción	Predeterminado	Valores válidos
MaxBatchSize	Número de instancias incluidas en cada lote de la actualización continua.	Un tercio del tamaño mínimo del grupo de Auto Scaling, redondeado al número entero superior más próximo.	De 1 a 10000
MinInstancesInService	Número mínimo de instancias que deben estar en servicio en el grupo de Auto Scaling mientras otras instancias terminan.	El tamaño mínimo del grupo de Auto Scaling o un tamaño inferior al tamaño máximo del grupo de Auto Scaling, el que sea más bajo.	De 0 a 9999
RollingUpdateEnabled	Si el valor es <code>true</code> , pueden realizarse actualizaciones continuas en un entorno. Las actualizaciones continuas resultan útiles cuando es necesario realizar actualizaciones pequeñas y frecuentes en la aplicación de software de Elastic Beanstalk y desea evitarse que se produzcan períodos	<code>false</code>	<code>true</code> <code>false</code>

Nombre	Descripción	Predeterminado	Valores válidos
	<p>de inactividad de la aplicación.</p> <p>Si este valor se establece en <code>true</code>, las opciones <code>MaxBatchSize</code>, <code>MinInstancesInService</code> y <code>PauseTime</code> se habilitan automáticamente. Al establecer cualquiera de estas opciones, el valor de la opción <code>RollingUpdateEnabled</code> también se establece automáticamente en <code>true</code>. Si esta opción se establece en <code>false</code>, se deshabilitan las actualizaciones continuas.</p> <div data-bbox="558 1276 878 1843" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> Note</p><p>Si utiliza la consola de Elastic Beanstalk o la CLI de EB para crear un entorno, no podrá establecer esta opción</p></div>		

Nombre	Descripción	Predeterminado	Valores válidos
	<p>en un archivo de configuración. La consola y la CLI de EB invalidan esta opción con un valor recomendado.</p>		

Nombre	Descripción	Predeterminado	Valores válidos
RollingUpdateType	<p>Esto incluye tres tipos: actualizaciones sucesivas basadas en tiempo, actualizaciones sucesivas basadas en el estado y actualizaciones inmutables.</p> <p>Las actualizaciones sucesivas basadas en el tiempo se aplican <code>PauseTime</code> entre lotes. Las actualizaciones continuas basadas en estado esperan a que las nuevas instancias superen las comprobaciones de estado antes de pasar al siguiente lote. Las actualizaciones inmutables inician un conjunto completo de instancias en un nuevo grupo de Auto Scaling.</p> <div data-bbox="560 1575 876 1850" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>Si utiliza la consola de Elastic Beanstalk o</p> </div>	Time	<p>Time</p> <p>Health</p> <p>Immutable</p>

Nombre	Descripción	Predeterminado	Valores válidos
	<p>la CLI de EB para crear un entorno, no podrá establecer esta opción en un archivo de configuración. La consola y la CLI de EB invalidan esta opción con un valor recomendado.</p>		
PauseTime	<p>El periodo de tiempo (en segundos, minutos u horas) que el servicio de Elastic Beanstalk espera una vez que se han completado las actualizaciones de un lote de instancias y antes de pasar al siguiente lote.</p>	<p>Se calcula automáticamente en función del tipo de instancia y del contenedor.</p>	<p>De PT0S* (0 segundos) a PT1H (1 hora)</p>

Nombre	Descripción	Predeterminado	Valores válidos
Timeout	El periodo de tiempo máximo (en minutos u horas) que se va esperar para que todas las instancias de un lote superen las comprobaciones de estado antes de cancelar la actualización.	PT30M (30 minutos)	De PT5M* (5 minutos) a PT1H (1 hora) *Formato de duración de ISO8601 : PT#H#M#S, donde cada # es el número de horas, minutos y segundos, respectivamente.

aws:ec2:instances


Configure las instancias del entorno, incluidas las opciones de spot. Este espacio de nombres complementa a [aws:autoscaling:launchconfiguration](#) y [aws:autoscaling:asg](#).

Para obtener más información, consulte [the section called “Grupo de Auto Scaling”](#).

Espacio de nombres: **aws:ec2:instances**


Nombre	Descripción	Predeterminado	Valores válidos
EnableSpot	Habilite las solicitudes de instancias de spot para su entorno. Cuando su valor es false, algunas opciones de este espacio de nombres no surten efecto.	false	true false
InstanceTypes	Una lista separada por comas de los tipos de instancias que desea utilizar en el entorno (por ejemplo t2.micro, t3.micro).	Lista con dos tipos	Entre uno y diez tipos de instancias de EC2 Se recomiendan como mínimo dos

Nombre	Descripción	Predet inado	Valores válidos
	<p>Cuando las instancias de spot no se encuentran activadas (<code>EnableSpot</code> es <code>false</code>), solo se utiliza el primer tipo de instancia de la lista.</p> <p>El primer tipo de instancia de la lista de esta opción es equivalente al valor de la opción <code>InstanceType</code> del espacio de nombres aws:autoscaling:launchconfiguration. No recomendamos usar esta última opción porque está obsoleta. Si especifica los dos, se utilizará el primer tipo de instancia de la lista de la opción <code>InstanceTypes</code> y <code>InstanceType</code> se omitirá.</p> <p>Los tipos de instancias que están disponibles dependen de las zonas de disponibilidad y la región utilizadas. Si elige una subred, la zona de disponibilidad que contiene esa subred determina el tipo de instancias disponible.</p> <ul style="list-style-type: none"> • Elastic Beanstalk no es compatible con los tipos de instancias de Amazon EC2 Mac. • Para obtener más información sobre las familias y los tipos de instancias de Amazon EC2, consulte Tipos de instancias en la Guía del usuario de Amazon EC2 o Tipos de instancias 	<p>de instancias.</p> <p>Varía según la cuenta y la región.</p>	<p>Varía según la cuenta, la región y la zona de disponibilidad. Puede obtener una lista de tipos de instancia de Amazon EC2 filtrados por estos valores. Para obtener más información, consulte Tipos de instancias disponibles en la Guía del usuario de Amazon EC2 o Tipos de instancias disponibles en la Guía del usuario de Amazon EC2.</p> <p>Todos los tipos de instancias deben formar parte de la misma arquitectura (<code>arm64</code>, <code>x86_64</code>, <code>i386</code>).</p> <p><code>Supported Architectures</code> también forma parte de este espacio de nombres. Si proporciona algún valor para <code>Supported Architectures</code>, el valor o los valores que ingrese para <code>InstanceTypes</code> debe pertenecer a una, y solo</p>

Nombre	Descripción	Predet inado	Valores válidos
	<p>en la Guía del usuario de Amazon EC2.</p> <ul style="list-style-type: none"> Para obtener más información sobre los tipos de instancias disponibles en todas las regiones, consulte Tipos de instancias disponibles en la Guía del usuario de Amazon EC2 o Tipos de instancias disponibles en la Guía del usuario de Amazon EC2. <div data-bbox="446 756 998 1734" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>Es posible que algunas AWS cuentas antiguas proporcionen a Elastic Beanstalk tipos de instancias predeterminados que no admiten instancias puntuales (por ejemplo, t1.micro). Si activa las solicitudes de instancias de spot y aparece un error sobre un tipo de instancia que no admite la característica spot, asegúrese de configurar tipos de instancias que sean compatibles con la característica spot. Para seleccionar tipos de instancias de spot, utilice el Asistente de instancias de spot.</p> </div>		una, de las arquitecturas que proporciona para Supported Architectures .

Nombre	Descripción	Predet inado	Valores válidos
	<p>Cuando actualiza la configuración del entorno y elimina uno o más tipos de instancias de la opción <code>InstanceTypes</code>, Elastic Beanstalk finaliza cualquier Amazon EC2 instance que se esté ejecutando en cualquiera de los tipos de instancias eliminados. A continuación, el grupo de Auto Scaling de su entorno lanza nuevas instancias, según sea necesario, para completar la capacidad deseada mediante sus tipos de instancia especificados actuales.</p>		
<p><code>SpotFleetOnDemandBase</code></p>	<p>Número mínimo de instancias en diferido que el grupo de Auto Scaling aprovisiona antes de considerar la posibilidad de utilizar instancias de spot cuando se amplía el entorno.</p> <p>Esta opción solo es pertinente cuando el valor de <code>EnableSpot</code> es <code>true</code>.</p>	<p>0</p>	<p>Entre 0 y el valor de la opción <code>MaxSize</code> en el espacio de nombres aws:autoscaling:asg</p>

Nombre	Descripción	Predeterminado	Valores válidos
SpotFleetOnDemandAvailabilityPercentage	<p>Porcentaje de instancias bajo demanda que forman parte de la capacidad adicional que el grupo de Auto Scaling aprovisiona después de las instancias SpotOnDemandBase .</p> <p>Esta opción solo es pertinente cuando el valor de EnableSpot es true.</p>	<p>0 en los entornos con una única instancia.</p> <p>70 en los entornos con balanceo de carga.</p>	De 0 a 100
SpotMaxPrice	<p>Precio máximo unidad por hora en USD que está dispuesto a pagar por una instancia de spot. Para obtener recomendaciones sobre las opciones de precio máximo para las instancias puntuales, consulte el historial de precios de las instancias puntuales en la Guía del usuario de Amazon EC2.</p> <p>Esta opción solo es pertinente cuando el valor de EnableSpot es true.</p>	<p>Precio en diferido para cada tipo de instancia.</p> <p>. En este caso, el valor de la opción es null.</p>	De 0.001 a 20.0 null

Nombre	Descripción	Predeterminado	Valores válidos
SupportedArchitectures	<p>Una lista separada por comas de los tipos de arquitectura de instancia de EC2 que utilizará para su entorno.</p> <p>Elastic Beanstalk es compatible con los tipos de instancia basados en las siguientes arquitecturas de procesador:</p> <ul style="list-style-type: none"> • AWS Arquitectura Arm Graviton de 64 bits (arm64) • Arquitectura de 64 bits (x86_64) • Arquitectura de 32 bits (i386) <p>Para obtener más información sobre la arquitectura de procesador y los tipos de instancias de Amazon EC2, consulte the section called “Tipos de instancias de Amazon EC2”.</p>	Ninguno	<p>arm64</p> <p>x86_64</p> <p>i386</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>La arquitectura de 32 bits i386 no es compatible con la mayoría de las plataformas de Elastic Beanstalk. En su lugar, se recomienda que elija los tipos de arquitectura x86_64 o arm64.</p> </div>

aws:ec2:vpc

Configure su entorno para lanzar recursos en una [Amazon Virtual Private Cloud](#) (Amazon VPC) personalizada. Si no configura ninguna opción de este espacio de nombres, Elastic Beanstalk inicia los recursos en la VPC predeterminada.

Espacio de nombres: **aws:ec2:vpc**

Nombre	Descripción	Predeterminado	Valores válidos
VPCId	ID de la Amazon VPC.	Ninguno	

Nombre	Descripción	Predeterminado	Valores válidos
Subnets	Identificadores de la subred o subredes del grupo de Auto Scaling. Si tiene varias subredes, especifique el valor en una única cadena separada por comas con los identificadores de subred (por ejemplo, "subnet-11111111, subnet-22222222").	Ninguna	
ELBSubnets	Identificadores de la subred o subredes de Elastic Load Balancing. Si tiene varias subredes, especifique el valor en una única cadena separada por comas con los identificadores de subred (por ejemplo, "subnet-11111111, subnet-22222222").	Ninguno	
ELBScheme	Especifique <code>internal</code> si desea crear un balanceador de carga interno en la Amazon VPC para que no se pueda obtener acceso a la aplicación de Elastic Beanstalk desde fuera de la Amazon VPC. Si especifica un valor distinto a <code>public</code> o <code>internal</code> , Elastic Beanstalk ignora el valor.	<code>public</code>	<code>public</code> <code>internal</code>
DBSubnets	Contiene los identificadores de las subredes de la base de datos. Solo se utiliza cuando se desea agregar una instancia de base de datos de Amazon RDS como parte de la aplicación. Si tiene varias subredes, especifique el valor en una única cadena separada por comas con los identificadores de subred (por ejemplo, "subnet-11111111, subnet-22222222").	Ninguno	


Nombre	Descripción	Predeterminado	Valores válidos
AssociatePublicIpAddress	<p>Especifica si se van a lanzar instancias con direcciones IP públicas en la Amazon VPC. Las instancias con direcciones IP públicas no necesitan un dispositivo NAT para comunicarse con Internet. Establezca el valor en <code>true</code> si quiere incluir el balanceador de carga y las instancias en una única subred pública.</p> <p>Esta opción no tiene efecto en un entorno de instancia única, que siempre tiene una sola Amazon EC2 instance con una dirección IP elástica. La opción es pertinente para entornos escalables con balanceo de carga.</p>	Ninguno	<code>true</code> <code>false</code>

aws:elasticbeanstalk:application

Configure una ruta de comprobación de estado para la aplicación. Para obtener más información, consulte [Informes de estado básicos](#).

Espacio de nombres: **aws:elasticbeanstalk:application**

Nombre	Descripción	Predeterminado	Valores válidos
Application Healthcheck URL	Ruta de acceso a la que se envían las solicitudes de comprobación de estado. Si no se establece esta ruta, el balanceador de carga intenta establecer una conexión TCP en el puerto 80 para verificar el estado de la aplicación. Utilice como valor una ruta que empiece por <code>/</code> para enviar una solicitud HTTP GET a esa ruta. También puede incluir un protocolo (HTTP, HTTPS, TCP o SSL) y un puerto delante de la ruta para	Ninguno	Los valores válidos son: <code>/</code> (HTTP GET para la ruta raíz) <code>/health</code> <code>HTTPS:443/</code> <code>HTTPS:443/ health</code>

Nombre	Descripción	Predeterminado	Valores válidos
	<p>comprobar la conectividad HTTPS o utilizar otro puerto diferente al predeterminado.</p> <div style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> Note</p> <p>Si utiliza la consola de Elastic Beanstalk para crear un entorno, no podrá establecer esta opción en un archivo de configuración. La consola invalida esta opción con un valor recomendado.</p> </div>		

La CLI de EB y la consola de Elastic Beanstalk aplican los valores recomendados a las opciones anteriores. Debe eliminar estos ajustes si desea usar archivos de configuración para configurarlos. Para obtener más información, consulte [Valores recomendados](#).

aws:elasticbeanstalk:application:environment

Configure las propiedades del entorno de la aplicación.

Espacio de nombres: **aws:elasticbeanstalk:application:environment**

Nombre	Descripción	Predeterminado	Valores válidos
Cualquier nombre de variable de entorno.	Debe pasarse en pares de clave-valor.	Ninguno	Cualquier valor de variable de entorno.

Para obtener más información, consulte [Propiedades del entorno y otras opciones del software](#).

aws:elasticbeanstalk:cloudwatch:logs

Configure el streaming de logs de instancias en la aplicación.

Espacio de nombres: **aws:elasticbeanstalk:cloudwatch:logs**

Nombre	Descripción	Predeterminado	Valores válidos
StreamLogs	Especifica si se deben crear grupos en los CloudWatch registros para los registros de proxy y de despliegue, y si se deben transmitir los registros de cada instancia de su entorno.	false	true false
DeleteOnTerminate	Especifica si se van a eliminar los grupos de logs cuando termine el entorno. Si el valor es false, los registros se guardan RetentionInDays días.	false	true false
RetentionInDays	Número de días que se van a conservar los eventos de registro antes de que expiren.	7	1, 3, 5, 7, 14, 30, 60, 90, 120, 150, 180, 365, 400, 545, 731, 1827, 3653

aws:elasticbeanstalk:cloudwatch:logs:health

Configura el streaming de logs de estado de entorno de la aplicación.

Espacio de nombres: **aws:elasticbeanstalk:cloudwatch:logs:health**

Nombre	Descripción	Predeterminado	Valores válidos
HealthStreamingEnabled	Para los entornos con los informes de estado mejorados habilitados, especifica si se debe crear un grupo en CloudWatch Logs para el estado del entorno	false	true false

Nombre	Descripción	Predeterminado	Valores válidos
	y archivar los datos de estado del entorno de Elastic Beanstalk. Para obtener información acerca de cómo habilitar el estado mejorado, consulte aws:elasticbeanstalk:healthreporting:system .		
DeleteOnTerminate	Si se va a eliminar el grupo de logs cuando termine el entorno. Si se usa <code>false</code> , los datos de estado se mantienen <code>RetentionInDays</code> días.	<code>false</code>	<code>true</code> <code>false</code>
RetentionInDays	El número de días que se van a conservar los datos de estado archivados antes de que venzan.	7	1, 3, 5, 7, 14, 30, 60, 90, 120, 150, 180, 365, 400, 545, 731, 1827, 3653


aws:elasticbeanstalk:command

Configure la política de implementación para el código de aplicación. Para obtener más información, consulte [the section called “Opciones de implementación”](#).

Espacio de nombres: **aws:elasticbeanstalk:command**

Nombre	Descripción	Predeterminado	Valores válidos
DeploymentPolicy	Seleccione una política de implementación para las implementaciones de las versiones de la aplicación.	AllAtOnce	AllAtOnce

Nombre	Descripción	Predeterminado	Valores válidos
	<p>Note</p> <p>Si utiliza la consola de Elastic Beanstalk para crear un entorno, no podrá establecer esta opción en un archivo de configuración. La consola invalida esta opción con un valor recomendado.</p>		<p>Rolling</p> <p>RollingWithAdditionalBatch</p> <p>Immutable</p> <p>TrafficSplitting</p>
Timeout	<p>La cantidad de tiempo, en segundos, que se espera a que una instancia complete la ejecución de comandos.</p> <p>Elastic Beanstalk agrega internamente 240 segundos (cuatro minutos) al valor Timeout. Por ejemplo, el tiempo de espera efectivo de forma predeterminada es 840 segundos (600+240) o 14 minutos.</p>	600	De 1 a 3600
BatchSizeType	<p>El tipo de número que se especifica en BatchSize</p> <p>Note</p> <p>Si utiliza la consola de Elastic Beanstalk o la CLI de EB para crear un entorno, no podrá establecer esta opción en un archivo de configuración. La consola y la CLI de EB invalidan esta opción con un valor recomendado.</p>	Percentage	<p>Percentage</p> <p>Fixed</p>


Nombre	Descripción	Predeterminado	Valores válidos
BatchSize	<p>El porcentaje o número fijo de instancias de Amazon EC2 en el grupo de Auto Scaling en las que las implementaciones se van a realizar simultáneamente. Los valores válidos varían en función de la BatchSizeType configuración utilizada.</p> <div style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>Si utiliza la consola de Elastic Beanstalk o la CLI de EB para crear un entorno, no podrá establecer esta opción en un archivo de configuración. La consola y la CLI de EB invalidan esta opción con un valor recomendado.</p> </div>	100	De 1 a 100 (Porcentaje). 1a aws:autoscaling:asg::() MaxSize Fixed
IgnoreHealthCheck	No cancele la implementación si no supera las comprobaciones de estado.	false	true false

aws:elasticbeanstalk:environment

Configure el rol de servicio y la arquitectura del entorno.

Espacio de nombres: **aws:elasticbeanstalk:environment**

Nombre	Descripción	Predeterminado	Valores válidos
EnvironmentType	Establézcalo en SingleInstance para lanzar una instancia EC2 sin ningún balanceador de carga.	LoadBalanced	SingleInstance

Nombre	Descripción	Predeterminado	Valores válidos
			LoadBalanced
ServiceRole	<p>Nombre del rol de IAM que Elastic Beanstalk utiliza para administrar los recursos del entorno. Especifique un nombre de rol (opcionalmente prefijado una ruta personalizada) o su ARN.</p> <p>Ejemplos:</p> <ul style="list-style-type: none"> aws-elasticbeanstalk-service-role <i>custom-path /custom-role</i> arn:aws:iam::123456789012:role/aws-elasticbeanstalk-service-role <div data-bbox="391 1104 1032 1562" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>Si utiliza la consola de Elastic Beanstalk o la CLI de EB para crear un entorno, no podrá establecer esta opción en un archivo de configuración. La consola y la CLI de EB invalidan esta opción con un valor recomendado.</p> </div>	Ninguno	Nombre de rol de IAM, ruta/nombre o ARN
LoadBalancerType	El tipo de balanceador de carga de su entorno. Para obtener más información, consulte the section called “Balanceador de carga” .	classic	classic application network

Nombre	Descripción	Predeterminado	Valores válidos
LoadBalancerIsShared	<p>Especifica si el balanceador de carga del entorno está dedicado o compartido. Esta opción solo se puede aplicar a un Application Load Balancer. No se puede cambiar después de que se creó el entorno.</p> <p>Cuando <code>false</code>, el entorno tiene su propio balanceador de carga específico, creado y administrado por Elastic Beanstalk. Cuando <code>true</code>, el entorno utiliza un balanceador de carga compartido, creado por usted y especificado en la opción <code>SharedLoadBalancer</code> del espacio de nombres aws:elbv2:loadbalancer.</p>	<code>false</code>	<code>true</code> <code>false</code>

aws:elasticbeanstalk:environment:process:default

Configure el proceso predeterminado del entorno.

Espacio de nombres: **aws:elasticbeanstalk:environment:process:default**

Nombre	Descripción	Predeterminado	Valores válidos
DeregistrationDelay	La cantidad de tiempo, en segundos, que debe esperar a que se completen las solicitudes activas antes de anular el registro.	20	De 0 a 3600
HealthCheckInterval	El intervalo de tiempo, en segundos, que Elastic Load Balancing comprueba	Con un balanceador de carga clásico o de aplicaciones: 15	Con un balanceador de carga clásico o de aplicaciones: 5 a 300

Nombre	Descripción	Predeterminado	Valores válidos
	el estado de las instancias de Amazon EC2 de la aplicación.	Con un balanceador de carga de red: 30	Con un balanceador de carga de red: 10, 30
HealthCheckPath	La ruta de acceso a la que se envían las solicitudes HTTP para comprobaciones de estado.	/	Una ruta de direccionamiento.
HealthCheckTimeout	La cantidad de tiempo, en segundos, que se va a esperar para obtener una respuesta durante una comprobación de estado. Esta opción solo se aplica a los entornos con un balanceador de carga de aplicaciones.	5	De 1 a 60
HealthyThresholdCount	El número de solicitudes consecutivas correctas antes de que Elastic Load Balancing cambie el estado de mantenimiento de la instancia.	Con un balanceador de carga clásico o de aplicaciones: 3 Con un balanceador de carga de red: 5	De 2 a 10

Nombre	Descripción	Predeterminado	Valores válidos
MatcherHTTPCode	<p>Lista separada por comas de códigos HTTP que indican que una instancia tiene un estado correcto.</p> <p>Esta opción solo se aplica a los entornos que tienen un balanceador de carga de aplicaciones o de red.</p>	200	<p>Con un balanceador de carga de aplicaciones: de 200 a 499</p> <p>Con un balanceador de carga de red: de 200 a 399</p>
Port	Puerto en el que escucha el proceso.	80	De 1 a 65535
Protocol	<p>El protocolo utilizado por el proceso.</p> <p>Con un balanceador de carga de aplicaciones solo puede aplicar esta opción a HTTP o HTTPS.</p> <p>Con un balanceador de carga de red solo puede aplicar esta opción a TCP.</p>	<p>Con un balanceador de carga clásico o de aplicaciones: HTTP</p> <p>Con un balanceador de carga de red: TCP</p>	<p>TCP</p> <p>HTTP</p> <p>HTTPS</p>

Nombre	Descripción	Predeterminado	Valores válidos
StickinessEnabled	<p>Establézcalo en true para habilitar las sesiones sticky.</p> <p>Esta opción solo se aplica a los entornos con un balanceador de carga de aplicaciones.</p>	'false'	'false' 'true'
StickinessLBCookieDuration	<p>La duración, en segundos, de la cookie de la sesión sticky.</p> <p>Esta opción solo se aplica a los entornos con un balanceador de carga de aplicaciones.</p>	86400 (un día)	De 1 a 604800
StickinessType	<p>Establézcalo en lb_cookie si desea usar cookies en las sesiones sticky.</p> <p>Esta opción solo se aplica a los entornos con un balanceador de carga de aplicaciones.</p>	lb_cookie	lb_cookie

Nombre	Descripción	Predeterminado	Valores válidos
UnhealthyThresholdCount	El número de solicitudes consecutivas fallidas antes de que Elastic Load Balancing cambie el estado de mantenimiento de la instancia.	5	De 2 a 10

aws:elasticbeanstalk:environment:process:process_name

Configure otros procesos en el entorno.

Espacio de nombres: **aws:elasticbeanstalk:environment:process:process_name**

Nombre	Descripción	Predeterminado	Valores válidos
DeregistrationDelay	La cantidad de tiempo, en segundos, que debe esperar a que se completen las solicitudes activas antes de anular el registro.	20	De 0 a 3600
HealthCheckInterval	Intervalo, en segundos, que Elastic Load Balancing comprueba el estado de las instancias de Amazon EC2 para la aplicación.	Con un balanceador de carga clásico o de aplicaciones: 15 Con un balanceador de carga de red: 30	Con un balanceador de carga clásico o de aplicaciones: 5 a 300 Con un balanceador de carga de red: 10, 30
HealthCheckPath	La ruta de acceso a la que se envían las solicitudes HTTP para	/	Una ruta de direccionamiento.

Nombre	Descripción	Predeterminado	Valores válidos
	comprobaciones de estado.		
HealthCheckTimeout	<p>La cantidad de tiempo, en segundos, que se va a esperar para obtener una respuesta durante una comprobación de estado.</p> <p>Esta opción solo se aplica a los entornos con un balanceador de carga de aplicaciones.</p>	5	De 1 a 60
HealthyThresholdCount	El número de solicitudes consecutivas correctas antes de que Elastic Load Balancing cambie el estado de mantenimiento de la instancia.	<p>Con un balanceador de carga clásico o de aplicaciones: 3</p> <p>Con un balanceador de carga de red: 5</p>	De 2 a 10

Nombre	Descripción	Predeterminado	Valores válidos
MatcherHTTPCode	<p>Una lista separada por comas de códigos HTTP que indica que una instancia tiene un estado correcto.</p> <p>Esta opción solo se aplica a los entornos que tienen un balanceador de carga de aplicaciones o de red.</p>	200	<p>Con un balanceador de carga de aplicaciones: de 200 a 499</p> <p>Con un balanceador de carga de red: de 200 a 399</p>
Port	Puerto en el que escucha el proceso.	80	De 1 a 65535
Protocol	<p>El protocolo utilizado por el proceso.</p> <p>Con un balanceador de carga de aplicaciones solo puede aplicar esta opción a HTTP o HTTPS.</p> <p>Con un balanceador de carga de red solo puede aplicar esta opción a TCP.</p>	<p>Con un balanceador de carga clásico o de aplicaciones: HTTP</p> <p>Con un balanceador de carga de red: TCP</p>	<p>TCP</p> <p>HTTP</p> <p>HTTPS</p>

Nombre	Descripción	Predeterminado	Valores válidos
StickinessEnabled	<p>Establézcalo en true para habilitar las sesiones sticky.</p> <p>Esta opción solo se aplica a los entornos con un balanceador de carga de aplicaciones.</p>	'false'	'false' 'true'
StickinessLBCookieDuration	<p>La duración, en segundos, de la cookie de la sesión sticky.</p> <p>Esta opción solo se aplica a los entornos con un balanceador de carga de aplicaciones.</p>	86400 (un día)	De 1 a 604800
StickinessType	<p>Establézcalo en lb_cookie si desea usar cookies en las sesiones sticky.</p> <p>Esta opción solo se aplica a los entornos con un balanceador de carga de aplicaciones.</p>	lb_cookie	lb_cookie

Nombre	Descripción	Predeterminado	Valores válidos
UnhealthyThresholdCount	El número de solicitud es consecutivas fallidas antes de que Elastic Load Balancing cambie el estado de mantenimiento de la instancia.	5	De 2 a 10

aws:elasticbeanstalk:environment:proxy:staticfiles

Puede utilizar el siguiente espacio de nombres para configurar el servidor proxy de forma que proporcione archivos estáticos. Cuando el servidor proxy recibe una solicitud de un archivo en la ruta especificada, proporciona directamente el archivo en lugar de direccionar la solicitud la aplicación. De este modo, se reduce el número de solicitudes que la aplicación tiene que procesar.

Asigne una ruta proporcionada por el servidor proxy a una carpeta del código fuente que contenga activos estáticos. Cada una de las opciones que defina en este espacio de nombres asignará una ruta diferente.

Note

Este espacio de nombres se aplica a las ramificaciones de la plataforma basadas en Amazon Linux 2 o posterior. Si su entorno utiliza una versión de plataforma basada en la AMI de Amazon Linux (anterior a Amazon Linux 2), consulte [the section called “Opciones específicas de la plataforma”](#) para los espacios de nombres de archivos estáticos específicos de la plataforma.

Espacio de nombres: **aws:elasticbeanstalk:environment:proxy:staticfiles**


Nombre	Valor
Ruta de acceso donde el servidor proxy sirve los archivos. Comience el valor con /.	El nombre de la carpeta que contiene los archivos.



Nombre	Valor
Por ejemplo: especifique <code>/images</code> para servir archivos en <code>subdomain .elasticbeanstalk.com/images</code> .	Por ejemplo: especifique <code>staticimages</code> para servir archivos de una carpeta llamada <code>staticimages</code> en el nivel superior de su paquete fuente.

`aws:elasticbeanstalk:healthreporting:system`

Configure informes de estado avanzados sobre el entorno.

Espacio de nombres: `aws:elasticbeanstalk:healthreporting:system`

Nombre	Descripción	Predeterminado	Valores válidos
SystemType	<p>El sistema de generación de informes de estado (básico o avanzado). Los informes de estado avanzados necesitan un rol de servicio y una versión de la plataforma de versión 2 o más reciente.</p> <div style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>Si utiliza la consola de Elastic Beanstalk o la CLI de EB para crear un entorno, no podrá establecer esta opción en un archivo de configuración. La consola y la CLI de EB invalidan esta opción con un valor recomendado.</p> </div>	basic	basic enhanced
ConfigDocument	Un documento JSON que describe las métricas del entorno y de la instancia en el que se va a publicar. CloudWatch	Ninguna	
EnhancedHealthAuthEnabled	Habilita la autorización para la API interna que utiliza Elastic Beanstalk para comunicar	true	true false

Nombre	Descripción	Predeterminado	Valores válidos
	<p>información de estado mejorada de las instancias del entorno al servicio de Elastic Beanstalk.</p> <p>Para obtener más información, consulte the section called “Roles de estado mejorados”.</p> <div data-bbox="423 510 1122 827" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>Esta opción solo es aplicable a los informes de estado mejorado (tal como cuando <code>SystemType</code> se establece en <code>enhanced</code>).</p> </div>		
HealthCheckSuccessThreshold	<p>Reduce el umbral para que las instancias puedan superar las comprobaciones de estado.</p> <div data-bbox="423 989 1122 1352" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>Si utiliza la consola de Elastic Beanstalk para crear un entorno, no podrá establecer esta opción en un archivo de configuración. La consola invalida esta opción con un valor recomendado.</p> </div>	0k	0k Warning Degraded Severe

aws:elasticbeanstalk:hostmanager

Configure las instancias EC2 de su entorno para que puedan cargarse registros rotados en Amazon S3.

Espacio de nombres: **aws:elasticbeanstalk:hostmanager**

Nombre	Descripción	Predeterminado	Valores válidos
LogPublicationControl	Copie los archivos de registro de las instancias de Amazon EC2 para la aplicación en el bucket de Amazon S3 asociado a la aplicación.	false	true false

aws:elasticbeanstalk:managedactions

Configure las actualizaciones de plataforma administradas del entorno.

Espacio de nombres: **aws:elasticbeanstalk:managedactions**

Nombre	Descripción	Predeterminado	Valores válidos
ManagedActionsEnabled	Habilite las actualizaciones de plataforma administradas . Si establece esta opción en true, también debe especificar PreferredStartTime y UpdateLevel .	false	true false
PreferredStartTime	Configure un periodo de mantenimiento en formato UTC para las acciones administradas. Por ejemplo, "Tue:09:00".	Ninguno	Día y hora del <i>día:hora:minuto</i> formato.
ServiceRoleForManagedUpdates	Nombre de un rol de IAM que Elastic Beanstalk utiliza para realizar actualizaciones de plataforma administradas en el entorno.	Ninguno	Igual que ServiceRole o bien

Nombre	Descripción	Predeterminado	Valores válidos
	<p>Puede utilizar el mismo rol que especificó en la opción <code>ServiceRole</code> del espacio de nombres <code>aws:elasticbeanstalk:environment</code>, o bien el rol vinculado al servicio de actualizaciones administradas de su cuenta. En este último caso, si la cuenta aún no tiene un rol vinculado al servicio de actualizaciones administradas, Elastic Beanstalk lo creará.</p>		<code>AWSServiceRoleForElasticBeanstalkManagedUpdates</code>

`aws:elasticbeanstalk:managedactions:platformupdate`

Configure las actualizaciones de plataforma administradas del entorno.

Espacio de nombres: **`aws:elasticbeanstalk:managedactions:platformupdate`**


Nombre	Descripción	Predeterminado	Valores válidos
<code>UpdateLevel</code>	<p>Nivel de actualización más alto que se va a aplicar con las actualizaciones de plataforma administradas. La versión de las plataformas tiene el formato <i>major.minor.patch</i>. Por ejemplo, en 2.0.8, la versión principal es 2, la versión secundaria es 0 y la versión de parche es 8.</p>	Ninguno	<p><code>patch</code> solamente en las actualizaciones de versión de parche.</p> <p><code>minor</code> tanto para las actualizaciones de versión</p>

Nombre	Descripción	Predeterminado	Valores válidos
			secundaria como para las actualizaciones de versión de parche.
InstanceRefreshEnabled	Habilita la sustitución semanal de instancias. Es necesario que ManagedActionsEnabled se establezca en true.	false	true false

aws:elasticbeanstalk:monitoring

Configure el entorno para terminar las instancias EC2 que no superen las comprobaciones de estado.

Espacio de nombres: **aws:elasticbeanstalk:monitoring**

Nombre	Descripción	Predeterminado	Valores válidos
Automatically Terminate Unhealthy Instances	Termina una instancia si no supera las comprobaciones de estado.	true	true false
	<div data-bbox="527 1549 560 1585" style="float: left; margin-right: 5px;">  </div> Note Esta opción solo se admite en entornos heredados . Determina el estado de una instancia según la posibilidad de llegar a		

Nombre	Descripción	Predeterminado	Valores válidos
	<p>ella y otras métricas basadas en instancias. Elastic Beanstalk no proporciona una forma de terminar instancias automáticamente en función del estado de la aplicación.</p>		

`aws:elasticbeanstalk:sns:topics`

Configure las notificaciones del entorno.

Espacio de nombres: **`aws:elasticbeanstalk:sns:topics`**

Nombre	Descripción	Predeterminado	Valores válidos
Notification Endpoint	<p>El punto de enlace donde desea que se notifiquen los eventos importantes que afectan a la aplicación.</p> <div data-bbox="380 1360 883 1871" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p>Note</p> <p>Si utiliza la consola de Elastic Beanstalk para crear un entorno, no podrá establecer esta opción en un archivo de configuración. La consola invalida esta opción con un valor recomendado.</p> </div>	Ninguna	


Nombre	Descripción	Predeterminado	Valores válidos
Notification Protocol	El protocolo que se utiliza para enviar notificaciones al punto de enlace.	email	http https email email-json sqs
Notification Topic ARN	El nombre de recurso de Amazon (ARN) del tema al que está suscrito.	Ninguno	
Notification Topic Name	El nombre del tema al que está suscrito.	Ninguno	

aws:elasticbeanstalk:sqsd

Configure la cola de Amazon SQS para un entorno de trabajo.


Espacio de nombres: **aws:elasticbeanstalk:sqsd**

Nombre	Descripción	Predeterminado	Valores válidos
WorkerQueueURL	La dirección URL de la cola desde la que el daemon del nivel del entorno de trabajo lee los mensajes.	Se genera automáticamente	Si no se especifica ningún valor, Elastic Beanstalk crea una cola automáticamente.

 **Note**

Cuando no se especifica un valor, la cola que Elastic Beanstalk crea automáticamente

Nombre	Descripción	Predeterminado	Valores válidos
	<p>amente es una cola estándar de Amazon SQS. Cuando indica un valor, puede proporcionar la URL de una cola estándar o FIFO de Amazon SQS. Tenga en cuenta que, si proporciona una cola FIFO, las tareas periódicas no serán compatibles.</p>		
HttpPath	La ruta relativa a la aplicación a la que se le envían los mensajes HTTP POST.	/	
MimeType	El tipo MIME del mensaje que se envía en la solicitud HTTP POST.	application/json	<p>application/json</p> <p>application/x-www-form-urlencoded</p> <p>application/xml</p> <p>text/plain</p> <p>Tipo MIME personalizado.</p>

Nombre	Descripción	Predeterminado	Valores válidos
HttpConnections	<p>El número máximo de conexiones simultáneas a cualquiera de las aplicaciones de una instancia de Amazon EC2.</p> <div data-bbox="380 495 880 999" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p> Note</p> <p>Si utiliza la consola de Elastic Beanstalk para crear un entorno, no podrá establecer esta opción en un archivo de configuración. La consola invalida esta opción con un valor recomendado.</p> </div>	50	De 1 a 100
ConnectTimeout	El periodo de tiempo, en segundos, que se va a esperar a que las conexiones con una aplicación se realicen correctamente.	5	De 1 a 60
InactivityTimeout	<p>El periodo de tiempo, en segundos, que se va a esperar para que llegue la respuesta de una conexión existente con una aplicación.</p> <p>El mensaje se procesa hasta que el daemon recibe una respuesta 200 (OK) de la aplicación en el nivel del entorno de trabajo o hasta que <code>RetentionPeriod</code> expira.</p>	299	De 1 a 36000

Nombre	Descripción	Predeterminado	Valores válidos
VisibilityTimeout	periodo de tiempo, en segundos, durante el que se bloquea el procesamiento de un mensaje entrante procedente de la cola de Amazon SQS. Una vez transcurrido el periodo de tiempo configurado, el mensaje vuelve a hacerse visible en la cola para que cualquier otro daemon pueda leerlo.	300	0 De a 43200
ErrorVisibilityTimeout	periodo de tiempo, en segundos, que transcurre antes de que Elastic Beanstalk devuelva un mensaje a la cola de Amazon SQS cuando se produce un error explícito en un intento de procesamiento.	2 segundos	De 0 a 43200 segundos
RetentionPeriod	El periodo de tiempo, en segundos, durante el cual un mensaje es válido y se procesa de forma activa.	345600	De 60 a 1209600
MaxRetries	Número máximo de intentos que Elastic Beanstalk realiza para enviar el mensaje a la aplicación web que lo va a procesar antes de mover el mensaje a la cola de mensajes fallidos.	10	De 1 a 100

aws:elasticbeanstalk:trafficplitting

Configure las implementaciones de división de tráfico para su entorno.

Este espacio de nombres se aplica cuando se establece la opción `DeploymentPolicy` del espacio de nombres [aws:elasticbeanstalk:command](#) en `TrafficSplitting`. Para obtener más información acerca de las políticas de implementación, consulte [the section called “Opciones de implementación”](#).

Espacio de nombres: **aws:elasticbeanstalk:trafficsplitting**

Nombre	Descripción	Predeterminado	Valores válidos
<code>NewVersionPercent</code>	Porcentaje inicial de tráfico de cliente entrante que Elastic Beanstalk pasa a instancias de entorno que ejecutan la nueva versión de la aplicación que está implementando.	10	De 1 a 100
<code>EvaluationTime</code>	periodo de tiempo, en minutos, que Elastic Beanstalk espera después de una implementación inicial en buen estado antes de proceder a cambiar todo el tráfico de cliente entrante a la nueva versión de la aplicación que está implementando.	5	De 3 a 600

aws:elasticbeanstalk:xray

Ejecute el AWS X-Ray daemon para transmitir información de rastreo desde su aplicación [integrada de X-Ray](#).

Espacio de nombres: **aws:elasticbeanstalk:xray**

Nombre	Descripción	Predeterminado	Valores válidos
<code>XRayEnabled</code>	Establezca esta opción en <code>true</code> para ejecutar el daemon de X-Ray en las instancias del entorno.	<code>false</code>	<code>true</code> <code>false</code>

aws:elb:healthcheck

Configure comprobaciones de estado para un Classic Load Balancer.

Espacio de nombres: **aws:elb:healthcheck**

Nombre	Descripción	Predeterminado	Valores válidos
HealthyThreshold	El número de solicitudes consecutivas correctas antes de que Elastic Load Balancing cambie el estado de mantenimiento de la instancia.	3	De 2 a 10
Interval	El intervalo en el que Elastic Load Balancing comprueba el estado de las instancias de Amazon EC2 de la aplicación en.	10	De 5 a 300
Timeout	La cantidad de tiempo, en segundos, que Elastic Load Balancing espera una respuesta antes de considerar que la instancia no responde.	5	De 2 a 60
UnhealthyThreshold	El número de solicitudes consecutivas fallidas antes de que Elastic Load Balancing cambie el estado de mantenimiento de la instancia.	5	De 2 a 10
(obsoleto) Target	El destino de una instancia de backend al que se envían las comprobaciones de estado. En su lugar, utilice Application Healthcheck URL en el espacio de nombres aws:elasticbeanstalk:application .	TCP:80	Destino con el formato PROTOCOLO:PUERTO/RUTA


aws:elb:loadbalancer

Configure un Classic Load Balancer para el entorno.

Varias de las opciones de este espacio de nombres no son más compatibles en favor de opciones específicas de los agentes de escucha del espacio de nombres [aws:elb:listener](#). Con estas opciones

que ya no son compatibles, solo puede configurar dos agentes de escucha (uno seguro y otro no seguro) en puertos estándar.

Espacio de nombres: **aws:elb:loadbalancer**

Nombre	Descripción	Valor predeterminado	Valores válidos
CrossZone	<p>Configure el balanceador de carga para que el tráfico se distribuya uniformemente entre todas las instancias de todas las zona de disponibilidad y no solo de una determinada zona.</p> <div data-bbox="467 787 1096 1249" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>Si utiliza la consola de Elastic Beanstalk o la CLI de EB para crear un entorno, no podrá establecer esta opción en un archivo de configuración. La consola y la CLI de EB invalidan esta opción con un valor recomendado.</p> </div>	false	true false
SecurityGroups	Asigne uno o varios de los grupos de seguridad creados para el balanceador de carga.	Ninguno	Uno o varios identificadores de grupo de seguridad.
ManagedSecurityGroup	Permite asignar un grupo de seguridad existente al balanceador de carga del entorno en lugar de crear otro nuevo. Para utilizar esta configuración, actualice la SecurityGroups configuración de este espacio de nombres para incluir el ID	Ninguno	Un ID de grupo de seguridad.

Nombre	Descripción	Valor predeterminado	Valores válidos
	<p>del grupo de seguridad y quite el identificador del grupo de seguridad que se creó automáticamente, si se creó uno.</p> <p>Para permitir el tráfico entre el balanceador de carga y las instancias EC2 del entorno, Elastic Beanstalk agrega una regla al grupo de seguridad de las instancias que permite el tráfico entrante procedente del grupo de seguridad administrado.</p>		
(obsoleto) LoadBalancerHTTPPort	El puerto de escucha de un agente de escucha no seguro.	80	OFF 80
(obsoleto) LoadBalancerPortProtocol	El protocolo que se va a utilizar en un agente de escucha no seguro.	HTTP	HTTP TCP
(obsoleto) LoadBalancerHTTPSPort	El puerto de escucha de un agente de escucha seguro.	OFF	OFF 443 8443
(obsoleto) LoadBalancerSSLPortProtocol	El protocolo que se va a utilizar en un agente de escucha seguro.	HTTPS	HTTPS SSL
(obsoleto) SSLCertificateId	El nombre de recurso de Amazon (ARN) de un certificado SSL para enlazar con el agente de escucha seguro.	Ninguno	

aws:elb:listener

Configure el agente de escucha predeterminado (puerto 80) en un Classic Load Balancer.

Espacio de nombres: **aws:elb:listener**

Nombre	Descripción	Valor predeterminado	Valores válidos
ListenerProtocol	Protocolo utilizado por el agente de escucha.	HTTP	HTTP TCP
InstancePort	Puerto que este agente de escucha utiliza para comunicarse con las instancias EC2.	80	De 1 a 65535
InstanceProtocol	<p>Protocolo que este agente de escucha utiliza para comunicarse con las instancias EC2.</p> <p>Debe encontrarse en la misma capa del protocolo de Internet que el ListenerProtocol. También debe tener el mismo nivel de seguridad que cualquier otro agente de escucha que use el mismo InstancePort que este agente.</p> <p>Por ejemplo, si ListenerProtocol es HTTPS (capa de aplicación, que usa una conexión segura) puede elegir el InstanceProtocol HTTP (también en la capa de aplicación, pero con una conexión no segura). Si además establece el InstancePort a 80, debe definir InstanceProtocol como HTTP en todos los demás agentes de escucha con InstancePort definido como 80.</p>	<p>HTTP cuando ListenerProtocol es HTTP</p> <p>TCP cuando ListenerProtocol es TCP</p>	<p>HTTP o HTTPS cuando ListenerProtocol es HTTP o HTTPS</p> <p>TCP o SSL cuando ListenerProtocol es TCP o SSL</p>
PolicyNames	Lista separada por comas con los nombres de las políticas que se van a aplicar al puerto de este agente de escucha. En su lugar, le recomendamos que utilice la	Ninguna	

Nombre	Descripción	Valor predeterminado	Valores válidos
	LoadBalancerPorts opción del espacio de nombres aws:elb:policiies .		
ListenerEnabled	Especifica si este agente de escucha está habilitado. Si especifica <code>false</code> , el agente de escucha no se incluye en el balanceador de carga.	<code>true</code>	<code>true</code> <code>false</code>

aws:elb:listener:listener_port

Configure otros agentes de escucha en un Classic Load Balancer.

Espacio de nombres: **aws:elb:listener:*listener_port***

Nombre	Descripción	Valor predeterminado	Valores válidos
ListenerProtocol	Protocolo utilizado por el agente de escucha.	HTTP	HTTP HTTPS TCP SSL
InstancePort	Puerto que este agente de escucha utiliza para comunicarse con las instancias EC2.	El mismo que <i>listener_port</i> .	De 1 a 65535
InstanceProtocol	Protocolo que este agente de escucha utiliza para comunicarse con las instancias EC2. Debe encontrarse en la misma capa del protocolo de Internet que el <code>ListenerProtocol</code> . También debe tener el mismo nivel de seguridad que cualquier otro	HTTP cuando <code>ListenerProtocol</code> es HTTP o HTTPS	HTTP o HTTPS cuando <code>ListenerProtocol</code> es HTTP o HTTPS

Nombre	Descripción	Valor predeterminado	Valores válidos
	<p>agente de escucha que use el mismo <code>InstancePort</code> que este agente.</p> <p>Por ejemplo, si <code>ListenerProtocol</code> es HTTPS (capa de aplicación, que usa una conexión segura) puede elegir el <code>InstanceProtocol</code> HTTP (también en la capa de aplicación, pero con una conexión no segura). Si además establece el <code>InstancePort</code> a 80, debe definir <code>InstanceProtocol</code> como HTTP en todos los demás agentes de escucha con <code>InstancePort</code> definido como 80.</p>	TCP cuando <code>ListenerProtocol</code> es TCP o SSL	TCP o SSL cuando <code>ListenerProtocol</code> es TCP o SSL
<code>PolicyNames</code>	Lista separada por comas con los nombres de las políticas que se van a aplicar al puerto de este agente de escucha. Le sugerimos que utilice la <code>LoadBalancerPorts</code> opción del espacio de aws:elb:policies nombres en su lugar.	Ninguna	
<code>SSLCertificateId</code>	El nombre de recurso de Amazon (ARN) de un certificado SSL para enlazar al agente de escucha.	Ninguno	


Nombre	Descripción	Valor predeterminado	Valores válidos
ListenerEnabled	Especifica si este agente de escucha está habilitado. Si especifica <code>false</code> , el agente de escucha no se incluye en el balanceador de carga.	<code>true</code> si se ha establecido cualquier otra opción. <code>false</code> en caso contrario.	<code>true</code> <code>false</code>

aws:elb:policies


Modifique la persistencia predeterminada y las políticas globales de balanceador de carga de un Classic Load Balancer.

Espacio de nombres: **aws:elb:policies**

Nombre	Descripción	Predeterminado	Valores válidos
ConnectionDrainingEnabled	Especifica si el balanceador de carga va a mantener las conexiones existentes con las instancias que han adoptado un estado incorrecto o cuyo registro se ha anulado para completar las solicitudes en curso.	<code>false</code>	<code>true</code> <code>false</code>

 **Note**

Si utiliza la consola de Elastic Beanstalk o la CLI de EB para crear un entorno, no podrá establecer esta opción en un [archivo de configuración](#).

Nombre	Descripción	Predeterminado	Valores válidos
	<p>La consola y la CLI de EB invalidan esta opción con un valor recomendado.</p>		
ConnectionDrainingTimeout	<p>Número máximo de segundos que el balanceador de carga mantiene las conexiones existentes con una instancia durante el vaciado de conexiones antes de que se fuerce el cierre de las conexiones.</p> <div style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>Si utiliza la consola de Elastic Beanstalk para crear un entorno, no podrá establecer esta opción en un archivo de configuración. La consola invalida esta opción con un valor recomendado.</p> </div>	20	De 1 a 3600
ConnectionSettingIdleTimeout	<p>La cantidad de tiempo, en segundos, en el que el balanceador de carga va a esperar para que se envíen o se reciban datos a través de la conexión. Si una vez transcurrido este periodo de tiempo no se ha enviado ni recibido ningún dato, el balanceador de carga cierra la conexión.</p>	60	De 1 a 3600
LoadBalancerPorts	<p>Lista separada por comas de los puertos de agente de escucha a los que se aplica la política predeterminada (AWSEB-ELB-StickinessPolicy).</p>	Ninguno	Puede utilizar :all para indicar todos los puertos del agente de escucha.

Nombre	Descripción	Predeterminado	Valores válidos
Stickiness Cookie Expiration	periodo en segundos, durante el que es válida cada cookie. Utiliza la política predeterminada (AWSEB-ELB-StickinessPolicy).	0	De 0 a 1000000
Stickiness Policy	Vincula la sesión de un usuario con una instancia de servidor específica para que todas las solicitudes que provengan de ese usuario durante la sesión se envíen a la misma instancia de servidor. Utiliza la política predeterminada (AWSEB-ELB-StickinessPolicy).	false	true false

`aws:elb:policies:policy_name`

Cree políticas de balanceador de carga adicionales para un Classic Load Balancer.

Espacio de nombres: `aws:elb:policies:policy_name`

Nombre	Descripción	Predeterminado	Valores válidos
CookieName	Nombre de la cookie generada por la aplicación que controla la duración de las sesiones de una política AppCookie StickinessPolicyType. Esta política solo se puede asociar a agentes de escucha HTTP/HTTPS.	Ninguno	
InstancePorts	Lista separada por comas de los puertos de instancia a los que se aplica esta política.	Ninguno	Una lista de puertos o :all

Nombre	Descripción	Predeterminado	Valores válidos
LoadBalancerPorts	Lista separada por comas de los puertos de agente de escucha a los que se aplica esta política.	Ninguno	Una lista de puertos o :all
ProxyProtocol	En el caso de una política ProxyProtocolPolicyType, especifica si se va a incluir la dirección IP y el puerto de la solicitud de origen en los mensajes TCP. Esta política solo se puede asociar a agentes de escucha TCP/SSL.	Ninguna	true false
PublicKey	El contenido de una clave pública de una política PublicKeyPolicyType que se va a usar al autenticar el servidor o servidores backend. Esta política no se puede aplicar directamente a servidores backend o a agentes de escucha. Debe ser parte de una política BackendServerAuthenticationPolicyType.	Ninguna	
PublicKeyPolicyNames	Una lista separada por comas de los nombres de políticas (procedentes de las políticas PublicKeyPolicyType) de una política BackendServerAuthenticationPolicyType que controla la autenticación en un servidor o servidores backend. Esta política solo puede asociarse con servidores backend que utilicen HTTPS/SSL.	Ninguna	

Nombre	Descripción	Predeterminado	Valores válidos
SSLProtocols	Una lista separada por comas de los protocolos SSL que se van a habilitar en una política <code>SSLNegotiationPolicyType</code> donde se definen los cifrados y los protocolos aceptados por el balanceador de carga. Esta política solo se puede asociar a agentes de escucha HTTPS/SSL.	Ninguna	
SSLReferencePolicy	El nombre de una política de seguridad predefinida que cumpla con las prácticas recomendadas de AWS seguridad y que desees activar para una <code>SSLNegotiationPolicyType</code> política que defina los cifrados y protocolos que acepta el balanceador de cargas. Esta política solo se puede asociar a agentes de escucha HTTPS/SSL.	Ninguno	
Stickiness Cookie Expiration	periodo en segundos, durante el que es válida cada cookie.	0	De 0 a 1000000
Stickiness Policy	Vincula la sesión de un usuario con una instancia de servidor específica para que todas las solicitudes que provengan de ese usuario durante la sesión se envíen a la misma instancia de servidor.	false	true false

aws:elbv2:listener:default

Configure el agente de escucha predeterminado (puerto 80) en un balanceador de carga de aplicaciones o en un balanceador de carga de red.

Este espacio de nombres no se aplica a un entorno que utiliza un balanceador de carga compartido. Los balanceadores de carga compartidos no tienen un agente de escucha predeterminado.

Espacio de nombres: **aws:elbv2:listener:default**

Nombre	Descripción	Predeterminado	Valores válidos
DefaultProcess	El nombre del proceso para reenviar el tráfico cuando no coincida ninguna regla.	default	Un nombre de proceso.
ListenerEnabled	Establézcalo en false para deshabilitar el agente de escucha. Puede utilizar esta opción para deshabilitar el agente de escucha predeterminado del puerto 80.	true	true false
Protocol	El protocolo de tráfico que se va a procesar.	Con un balanceador de carga de aplicaciones: HTTP Con un balanceador de carga de red: TCP	Con un balanceador de carga de aplicaciones: HTTP, HTTPS Con un balanceador de carga de red: TCP
Rules	Una lista de reglas que se van a aplicar al agente de escucha. Esta opción solo se aplica a los entornos con un balanceador de carga de aplicaciones.	Ninguno	Una lista separada por comas de nombres de reglas.

Nombre	Descripción	Predeterminado	Valores válidos
SSLCertificateArns	<p>Nombre de recurso de Amazon (ARN) del certificado SSL para enlazar al agente de escucha.</p> <p>Esta opción solo se aplica a los entornos con un balanceador de carga de aplicaciones.</p>	Ninguno	ARN de un certificado almacenado en IAM o ACM.
SSLPolicy	<p>Especifique la política de seguridad que desea aplicar al agente de escucha.</p> <p>Esta opción solo se aplica a los entornos con un balanceador de carga de aplicaciones.</p>	Ninguno (valor predeterminado de ELB)	Nombre de la política de seguridad del balanceador de carga.

aws:elbv2:listener:listener_port

Configure los agentes de escucha adicionales en un balanceador de carga de aplicaciones o en un balanceador de carga de red.

Note

Para un balanceador de carga de aplicaciones compartido, solo puede especificar la opción `Rule`. Las demás opciones no se aplican a los balanceadores de carga compartidos.

Espacio de nombres: **aws:elbv2:listener:***listener_port*

Nombre	Descripción	Predeterminado	Valores válidos
DefaultProcess	El nombre del proceso donde se reenvía el tráfico cuando no coincide ninguna regla.	default	Un nombre de proceso.
ListenerEnabled	Establézcalo en false para deshabilitar el agente de escucha. Puede utilizar esta opción para deshabilitar el agente de escucha predeterminado del puerto 80.	true	true false
Protocol	El protocolo de tráfico que se va a procesar.	Con un balanceador de carga de aplicaciones: HTTP Con un balanceador de carga de red: TCP	Con un balanceador de carga de aplicaciones: HTTP, HTTPS Con un balanceador de carga de red: TCP
Rules	Lista de reglas que se van a aplicar al agente de escucha. Esta opción solo se aplica a los entornos con un balanceador de carga de aplicaciones. Si su entorno utiliza un balanceador de	Ninguno	Una lista separada por comas de nombres de reglas.

Nombre	Descripción	Predeterminado	Valores válidos
	carga de aplicaciones compartido y no especifica esta opción para ningún agente de escucha, Elastic Beanstalk asocia automáticamente la regla default con un agente de escucha del puerto 80.		
SSLCertificateArns	<p>Nombre de recurso de Amazon (ARN) del certificado SSL para enlazar al agente de escucha.</p> <p>Esta opción solo se aplica a los entornos con un balanceador de carga de aplicaciones.</p>	Ninguno	ARN de un certificado almacenado en IAM o ACM.
SSLPolicy	<p>Especifique la política de seguridad que desea aplicar al agente de escucha.</p> <p>Esta opción solo se aplica a los entornos con un balanceador de carga de aplicaciones.</p>	Ninguno (valor predeterminado de ELB)	Nombre de la política de seguridad del balanceador de carga.

aws:elbv2:listenerrule:rule_name

Defina reglas de agente de escucha para un balanceador de carga de aplicaciones. Si una solicitud se corresponde con los nombres de host o las rutas de una regla, el balanceador de carga la reenvía al proceso especificado. Para utilizar una regla, agréguela a un agente de escucha con la opción Rules del espacio de nombres [aws:elbv2:listener:listener_port](#).

Note

Este espacio de nombres no es aplicable a entornos con un balanceador de carga de red.

Espacio de nombres: **aws:elbv2:listenerrule:rule_name**

Nombre	Descripción	Predeterminado	Valores válidos
HostHeaders	Una lista de nombres de host con los que buscar correspondencia. Por ejemplo, <code>my.example.com</code> .	Balanceador de carga dedicado: Ninguno Balanceador de carga compartido: el CNAME del entorno	Cada nombre puede contener hasta 128 caracteres. Un patrón puede incluir letras mayúsculas y minúsculas, números, guiones (—) y hasta tres caracteres comodín (* coincide con cero o más caracteres; ? coincide exactamente con un carácter). Puede enumerar más de un nombre separados por coma. El balanceador de carga de aplicaciones admite hasta cinco reglas HostHead

Nombre	Descripción	Predeterminado	Valores válidos
			<p>r y PathPattern combinadas.</p> <p>Para obtener más información, consulte Condiciones de host en la Guía del usuario de balanceadores de carga de aplicaciones.</p>

Nombre	Descripción	Predeterminado	Valores válidos
PathPatterns	<p>Los patrones de ruta para coincidir (por ejemplo: /img/*).</p> <p>Esta opción solo se aplica a los entornos con un balanceador de carga de aplicaciones.</p>	Ninguno	<p>Cada patrón puede contener hasta 128 caracteres. Un patrón puede incluir letras mayúsculas y minúsculas, números, guiones (—) y hasta tres caracteres comodín (* coincide con cero o más caracteres; ? coincide exactamente con un carácter). Puede agregar varios patrones de ruta separados por comas. El balanceador de carga de aplicaciones admite hasta cinco reglas HostHeader y PathPattern combinadas.</p> <p>Para obtener más información, consulte Condiciones de ruta en la Guía del usuario de balanceadores de carga de aplicaciones.</p>

Nombre	Descripción	Predeterminado	Valores válidos
Priority	<p>La precedencia de esta regla cuando coinciden varias reglas. Cuanto menor sea el número, mayor será la prioridad. No puede haber dos reglas con la misma prioridad.</p> <p>Con un balanceador de carga compartido, Elastic Beanstalk trata las prioridades de regla como relativas en todos los entornos de uso compartido y las asigna a prioridades absolutas durante la creación.</p>	1	De 1 a 1000
Process	El nombre del proceso para reenviar el tráfico cuando esta regla coincida con la solicitud.	default	Un nombre de proceso.

aws:elbv2:loadbalancer

Configure un Application Load Balancer.

Para un balanceador de carga compartido, solo las opciones SharedLoadBalancer y SecurityGroups son válidas.

Note

Este espacio de nombres no es aplicable a entornos con un Network Load Balancer.

Espacio de nombres: **aws:elbv2:loadbalancer**

Nombre	Descripción	Predeterminado	Valores válidos
AccessLogsS3Bucket	El bucket de Amazon S3 donde se almacenan los registros de acceso.	Ninguno	Un nombre de bucket.

Nombre	Descripción	Predeterminado	Valores válidos
	El bucket debe estar en la misma región que el entorno y permitir que el balanceador de carga tenga acceso de escritura.		
AccessLogsS3Enabled	Habilita el almacenamiento de los logs de acceso.	false	true false
AccessLogsS3Prefix	Un prefijo que se va a agregar a los nombres de los logs de acceso. De forma predeterminada, el balanceador de cargas carga los registros en un directorio con el nombre AWSLogs del bucket que especifique. Especifique un prefijo para colocar el AWSLogs directorio dentro de otro directorio.	Ninguna	
IdleTimeout	La cantidad de tiempo, en segundos, que debe esperar a que se complete una solicitud antes de cerrar las conexiones con el cliente y la instancia.	Ninguno	De 1 a 3600

Nombre	Descripción	Predefinido	Valores válidos
ManagedSecurityGroup	<p>Permite asignar un grupo de seguridad existente al balanceador de carga del entorno en lugar de crear otro nuevo. Si desea utilizar esta configuración, actualice la opción <code>SecurityGroups</code> de este espacio de nombres para incluir el ID del grupo de seguridad y para que, si procede, se elimine el ID de grupo de seguridad que se creó automáticamente.</p> <p>Para permitir el tráfico entre el balanceador de carga y las instancias EC2 del entorno, Elastic Beanstalk agrega una regla al grupo de seguridad de las instancias que permite el tráfico entrante procedente del grupo de seguridad administrado.</p>	Grupo de seguridad que Elastic Beanstalk crea para el balanceador de carga.	Un ID de grupo de seguridad.

Nombre	Descripción	Predeterminado	Valores válidos
SecurityGroups	<p>Una lista de grupos de seguridad que se va a adjuntar al balanceador de carga.</p> <p>En el caso de un balanceador de carga compartido, si no especifica este valor, Elastic Beanstalk comprueba si un grupo de seguridad existente que administra ya está asociado al balanceador de carga. Si uno no está conectado al balanceador de carga, Elastic Beanstalk crea un grupo de seguridad y lo conecta al balanceador de carga. Elastic Beanstalk elimina este grupo de seguridad cuando finaliza el último entorno que comparte el balanceador de carga.</p> <p>Los grupos de seguridad del balanceador de carga se utilizan para configurar la regla de ingreso del grupo de seguridad de Amazon EC2 instance.</p>	Grupo de seguridad que Elastic Beanstalk crea para el balanceador de carga.	Lista separada por comas de ID de grupos de seguridad.

Nombre	Descripción	Predeterminado	Valores válidos
SharedLoadBalancer	<p>El nombre de recurso de Amazon (ARN) del balanceador de carga. Esta opción solo es relevante para un Application Load Balancer. Es necesario cuando la opción <code>LoadBalancerIsShared</code> del espacio de nombres aws:elasticbeanstalk:environment se establece en <code>true</code>. No puede cambiar el ARN del balanceador de carga compartido después de la creación del entorno.</p> <p>Criterios para un valor válido:</p> <ul style="list-style-type: none"> • Debe ser un balanceador de cargas activo y válido en la AWS región en la que se encuentra el entorno. • Debe estar en la misma Amazon Virtual Private Cloud (Amazon VPC) que el entorno. • No puede ser un balanceador de carga creado por Elastic Beanstalk como balanceador de carga específico para otro entorno. Puede identificar estos balanceadores de carga dedicados mediante el prefijo <code>awseb-</code>. <p>Ejemplo:</p> <pre>arn:aws:elasticloadbalancing:us-east-2:123456789012:lo</pre>	Ninguno	ARN de un balanceador de carga válido que cumple todos los criterios descritos aquí.


Nombre	Descripción	Predeterminado	Valores válidos
	adbalancer/app/FrontEndLB/0dbf78d8ad96abbc		

aws:rds:dbinstance

Configure una instancia de base de datos de Amazon RDS adjuntada.


Espacio de nombres: **aws:rds:dbinstance**

Nombre	Descripción	Predeterminado	Valores válidos
DBAllocatedStorage	Tamaño de almacenamiento asignado en la base de datos especificado en gigabytes	MySQL: 5 Oracle: 10 sqlserver-se: 200 sqlserver-ex: 30 sqlserver-web: 30	MySQL: 5-1024 Oracle: 10-1024 sqlserver: no se puede modificar
DBDeletionPolicy	Especifica si se va a retener, eliminar o realizar una instantánea de la instancia de base de datos cuando se termina un entorno. Esta opción funciona en conjunto con <code>HasCoupledDatabase</code> , también una opción de este espacio de nombres.	Delete	Delete Retain Snapshot

Nombre	Descripción	Predeterminado	Valores válidos
	<div style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; background-color: #fff9f9;"> <p> Warning</p> <p>Si se elimina una instancia de base de datos, los datos se perderán permanentemente.</p> </div>		
DBEngine	Nombre del motor de base de datos que se va a usar con esta instancia.	mysql	mysql oracle-se1 sqlserver-ex sqlserver-web sqlserver-se postgres
DBEngineVersion	Número de versión del motor de base de datos.	5.5	
DBInstanceClass	Tipo de instancia de base de datos.	db.t2.micro (db.m1.large para un entorno que no se ejecuta en Amazon VPC)	Para obtener más información, consulte Clase de instancia de base de datos en la Guía del usuario de Amazon Relational Database Service.

Nombre	Descripción	Predeterminado	Valores válidos
DBPassword	Nombre de la contraseña de usuario maestra de la instancia de base de datos.	Ninguno	
DBSnapshotIdentifier	Identificador de la instantánea de base de datos desde la que se va a realizar la restauración.	Ninguno	
DBUser	Nombre del usuario maestro de la instancia de base de datos.	ebroot	

Nombre	Descripción	Predeterminado	Valores válidos
HasCoupledDatabase	<p>Especifica si una instancia de base de datos está acoplada al entorno. Si se cambia a <code>true</code>, Elastic Beanstalk crea una nueva instancia de base de datos acoplada a su entorno. Si cambia a <code>false</code>, Elastic Beanstalk inicia el desacoplamiento de la instancia de base de datos de su entorno.</p> <p>Esta opción funciona en conjunto con <code>DBDeletionPolicy</code>, también una opción de este espacio de nombres.</p>	<code>false</code>	<code>true</code> <code>false</code>

 **Note**

Nota: si vuelve a cambiar este valor a `true` tras desacoplar la base de datos anterior, Elastic Beanstalk crea una nueva base de datos con la configuración de opciones de base de datos anterior. Sin embargo, para mantener la seguridad de su entorno, no retiene las configuraciones `DBUser` y `DBPassword` existentes. Debe especificar `DBUser` y `DBPassword` nuevamente.

Nombre	Descripción	Predeterminado	Valores válidos
MultiAZDatabase	Especifica si debe crearse un Despliegue Multi-AZ en la instancia de base de datos. Para obtener más información sobre las implementaciones Multi-AZ con Amazon Relational Database Service (RDS), consulte Regiones y zonas de disponibilidad en la Guía del usuario de Amazon Relational Database Service.	false	true false

Opciones específicas de la plataforma

Algunas plataformas de Elastic Beanstalk definen espacios de nombres de opciones que son específicos de la plataforma. A continuación se enumeran estos espacios de nombres y sus opciones para cada plataforma.

Note

Anteriormente, en versiones de plataforma basadas en la AMI de Amazon Linux (anterior a Amazon Linux 2), las dos características siguientes y sus respectivos espacios de nombres se consideraban características específicas de la plataforma y se han enumerado aquí por plataforma:

- Configuración del proxy para archivos estáticos – [aws:elasticbeanstalk:environment:proxy:staticfiles](#)
- AWS X-Ray Compatibilidad con – [aws:elasticbeanstalk:xray](#)

En las versiones de la plataforma de Amazon Linux 2, Elastic Beanstalk implementa estas características de manera coherente en todas las plataformas compatibles. El espacio de nombres relacionado se muestra ahora en la página [the section called “Opciones generales”](#). Solo hicimos mención en esta página para las plataformas que tenían espacios de nombres diferentes.

Plataformas

- [Opciones de la plataforma Docker](#)
- [Opciones de la plataforma Go](#)
- [Opciones de la plataforma Java SE](#)
- [Opciones de la plataforma Java con Tomcat](#)
- [Historial de plataformas de .NET Core en Linux](#)
- [Opciones de la plataforma .NET](#)
- [Opciones de la plataforma Node.js](#)
- [Opciones de la plataforma PHP](#)
- [Opciones de la plataforma Python](#)
- [Opciones de la plataforma Ruby](#)

Opciones de la plataforma Docker

Las siguientes opciones de configuración específicas de Docker se aplican a las plataformas de Docker y Preconfigured Docker.

Note

Estas opciones de configuración no se aplican a

- la plataforma Docker (Amazon Linux 2) con Docker Compose
- la plataforma Multicontainer Docker (AMI de Amazon Linux)

Espacio de nombres: **aws:elasticbeanstalk:environment:proxy**

Nombre	Descripción	Valor predeterminado	Valores válidos
ProxyServer	Especifica el servidor web que se usará como proxy.	nginx	nginx none — Amazon Linux AM y Docker w/DC solamente

Opciones de la plataforma Go

Opciones de plataforma AMI de Amazon Linux (anteriores a Amazon Linux 2)

Espacio de nombres: **aws:elasticbeanstalk:container:golang:staticfiles**

Puede utilizar el siguiente espacio de nombres para configurar el servidor proxy de forma que proporcione archivos estáticos. Cuando el servidor proxy recibe una solicitud de un archivo en la ruta especificada, proporciona directamente el archivo en lugar de direccionar la solicitud a la aplicación. De este modo, se reduce el número de solicitudes que la aplicación tiene que procesar.

Asigne una ruta proporcionada por el servidor proxy a una carpeta del código fuente que contenga activos estáticos. Cada una de las opciones que defina en este espacio de nombres asignará una ruta diferente.

Nombre	Valor
Ruta donde el servidor proxy distribuirá los archivos.	Nombre de la carpeta que contiene los archivos.
Ejemplo: <code>/images</code> para distribuir archivos en <code>subdomain .eleasticbeanstalk.com/images</code> .	Ejemplo: <code>staticimages</code> para distribuir archivos de una carpeta con el nombre <code>staticimages</code> en la parte superior del paquete de código fuente.

Opciones de la plataforma Java SE

Opciones de plataforma AMI de Amazon Linux (anteriores a Amazon Linux 2)

Espacio de nombres: **aws:elasticbeanstalk:container:java:staticfiles**

Puede utilizar el siguiente espacio de nombres para configurar el servidor proxy de forma que proporcione archivos estáticos. Cuando el servidor proxy recibe una solicitud de un archivo en la ruta especificada, proporciona directamente el archivo en lugar de direccionar la solicitud a la aplicación. De este modo, se reduce el número de solicitudes que la aplicación tiene que procesar.

Asigne una ruta proporcionada por el servidor proxy a una carpeta del código fuente que contenga activos estáticos. Cada una de las opciones que defina en este espacio de nombres asignará una ruta diferente.

Nombre	Valor
Ruta donde el servidor proxy distribuirá los archivos.	Nombre de la carpeta que contiene los archivos.
Ejemplo: <code>/images</code> para distribuir archivos en <code>subdomain .elasticbeanstalk.com/images</code> .	Ejemplo: <code>staticimages</code> para distribuir archivos de una carpeta con el nombre <code>staticimages</code> en la parte superior del paquete de código fuente.

Opciones de la plataforma Java con Tomcat


Espacio de nombres: **aws:elasticbeanstalk:application:environment**

Nombre	Descripción	Valor predeterminado	Valores válidos
JDBC_CONNECTION_STRING	Cadena de conexión con una base de datos externa.	n/a	n/a

Para obtener más información, consulte [Propiedades del entorno y otras opciones del software](#).

Espacio de nombres: **aws:elasticbeanstalk:container:tomcat:jvmoptions**

Nombre	Descripción	Valor predeterminado	Valores válidos
JVM Options	Pasa opciones de línea de comando a JVM al arrancar.	n/a	n/a
Xmx	Tamaño máximo de los montones de JVM.	256m	n/a
XX:MaxPermSize	Sección del montón de JVM que se utiliza para almacenar definiciones de clases y metadatos asociados.	64m	n/a

Nombre	Descripción	Valor predeterminado	Valores válidos
	<p> Note</p> <p>Esta opción solo se aplica a versiones de Java anteriores a Java 8 y no es compatible con las plataformas Tomcat de Elastic Beanstalk basadas en Amazon Linux 2 y versiones posteriores.</p>		
Xms	Tamaño de los montones de JVM iniciales.	256m	n/a
<i>optionName</i>	Especifique las opciones de JVM arbitrarias, además de las definidas por la plataforma de Tomcat.	n/a	n/a

Espacio de nombres: **aws:elasticbeanstalk:environment:proxy**

Nombre	Descripción	Valor predeterminado	Valores válidos
GzipCompression	<p>Establézcalo en <code>false</code> para deshabilitar la compresión de las respuestas.</p> <p>Sólo es válido en versiones de plataforma AMI de Amazon Linux (anteriores a Amazon Linux 2).</p>	true	true false
ProxyServer	Configure el proxy que se utilizará en las instancias del entorno. Si configura esta opción en <code>apache</code> , Elastic Beanstalk utiliza Apache 2.4 .	nginx (Amazon Linux 2)	apache apache/2.2 – Solo AMI de Amazon Linux nginx

Nombre	Descripción	Valor predeterminado	Valores válidos
	<p>Configúrelo en <code>apache/2.2</code> si la aplicación no lista para migrar desde Apache 2.2 debido a opciones de configuración de proxy incompatibles. Este valor solo es válido en versiones de plataforma de la AMI de Amazon Linux (anterior a Amazon Linux 2).</p> <p>Configúrelo en <code>nginx</code> para utilizar nginx. Este es el valor predeterminado que comienza con las versiones de la plataforma de Amazon Linux 2.</p> <p>Para obtener más información, consulte Configuración del servidor proxy de su entorno Tomcat.</p>	apache(Linux AMI)	

Historial de plataformas de .NET Core en Linux

Espacio de nombres: **aws:elasticbeanstalk:environment:proxy**

Nombre	Descripción	Valor predeterminado	Valores válidos
ProxyServer	Especifica el servidor web que se usará como proxy.	nginx	nginx none

Opciones de la plataforma .NET

Espacio de nombres: **aws:elasticbeanstalk:container:dotnet:apppool**

Nombre	Descripción	Valor predeterminado	Valores válidos
Target Runtime	Elija la versión de .NET Framework de la aplicación.	4.0	2.0 4.0
Enable 32-bit Applications	Establezca en True para ejecutar aplicaciones de 32 bits.	False	True False

Opciones de la plataforma Node.js

Espacio de nombres: **aws:elasticbeanstalk:environment:proxy**


Nombre	Descripción	Valor predeterminado	Valores válidos
ProxyServer	Configure el proxy que se utilizará en las instancias del entorno.	nginx	apache nginx

Opciones de plataforma AMI de Amazon Linux (anteriores a Amazon Linux 2)

Espacio de nombres: **aws:elasticbeanstalk:container:nodejs**

Nombre	Descripción	Valor predeterminado	Valores válidos
NodeCommand	Command que se utiliza para iniciar la aplicación de Node.js. Si se especifica una cadena vacía, se utiliza app.js, luego	""	n/a

Nombre	Descripción	Valor predeterminado	Valores válidos
	<code>server.js</code> y por último <code>npm start</code> (en ese orden).		

Nombre	Descripción	Valor predeterminado	Valores válidos
NodeVersion	<p data-bbox="321 321 893 359">Versión de Node.js. Por ejemplo, 4.4.6</p> <p data-bbox="321 405 911 674">Las versiones de Node.js compatibles pueden variar entre las versiones de la plataforma Node.js. Consulte Node.js en el documento Plataformas de AWS Elastic Beanstalk para ver una lista de las versiones que se admiten actualmente.</p> <div data-bbox="326 716 954 1801" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"> <p data-bbox="350 751 469 789"> Note</p> <p data-bbox="399 814 911 1801">Cuando la compatibilidad de la versión de Node.js que utiliza se elimine de la plataforma, deberá cambiar o eliminar la configuración de la versión antes de realizar una actualización de la plataforma. Esto puede suceder cuando se identifica una vulnerabilidad de seguridad en una o varias versiones de Node.js. Cuando esto sucede, si se intenta actualizar a una nueva versión de la plataforma no compatible con la NodeVersion configurada, se producirá un error. Para evitar la necesidad de crear un nuevo entorno, cambie la opción de configuración NodeVersion por una versión de Node.js compatible con la antigua y la nueva versión de la plataforma, o elimine la opción de configuración y, a continuación,</p> </div>	varía	varía

Nombre	Descripción	Valor predeterminado	Valores válidos
	lleve a cabo la actualización de la plataforma.		
GzipCompression	Especifica si se ha habilitado la compresión gzip. Si se establece ProxyServer en none, se deshabilita la compresión gzip.	false	true false
ProxyServer	Especifica qué servidor web debe utilizarse con las conexiones del proxy en Node.js. Si ProxyServer se establece en none, las asignaciones de archivos estáticos no surten efecto y la compresión gzip se deshabilita.	nginx	apache nginx none

Espacio de nombres: **aws:elasticbeanstalk:container:nodejs:staticfiles**

Puede utilizar el siguiente espacio de nombres para configurar el servidor proxy de forma que proporcione archivos estáticos. Cuando el servidor proxy recibe una solicitud de un archivo en la ruta especificada, proporciona directamente el archivo en lugar de direccionar la solicitud a la aplicación. De este modo, se reduce el número de solicitudes que la aplicación tiene que procesar.

Asigne una ruta proporcionada por el servidor proxy a una carpeta del código fuente que contenga activos estáticos. Cada una de las opciones que defina en este espacio de nombres asignará una ruta diferente.

Note

La configuración de archivos estáticos no se aplica si `aws:elasticbeanstalk:container:nodejs::ProxyFiles` se establece en none.

Nombre	Valor
Ruta donde el servidor proxy distribuirá los archivos.	Nombre de la carpeta que contiene los archivos.
Ejemplo: /images para distribuir archivos en <i>subdomain</i> .elasticbeanstalk.com/images .	Ejemplo: staticimages para distribuir archivos de una carpeta con el nombre staticimages en la parte superior del paquete de código fuente.

Opciones de la plataforma PHP

Espacio de nombres: **aws:elasticbeanstalk:container:php:phpini**

Nombre	Descripción	Valor predeterminado	Valores válidos
document_root	Especifique el directorio secundario del proyecto que se va a utilizar como raíz web pública.	/	Las cadenas vacías se tratan como /. También puede especificar una cadena que comience por /.
memory_limit	Cantidad de memoria asignada al entorno de PHP.	256M	n/a
zlib.output_compression	Especifica si PHP debe utilizar o no la compresión en la salida.	Off	On Off true false
allow_url_fopen	Especifica si las funciones de archivo de PHP tienen permiso para recuperar datos	On	On Off

Nombre	Descripción	Valor predeterminado	Valores válidos
	de ubicaciones remotas, como sitios web o servidores FTP.		true false
display_errors	Especifica si los mensajes de error deben formar parte de la salida.	Off	On Off
max_execution_time	Establece el tiempo máximo, en segundos, que un script puede mantenerse en ejecución antes de que el entorno lo termine.	60	De 0 a 9223372036854775807 (PHP_INT_MAX)
composer_options	Establece las opciones personalizadas que se van a usar al instalar dependencias con Composer a través de composer.phar install. Para obtener más información, incluidas las opciones disponibles, visite http://getcomposer.org/doc/03-cli.md#install .	n/a	n/a

Espacio de nombres: **aws:elasticbeanstalk:environment:proxy**

Nombre	Descripción	Valor predeterminado	Valores válidos
ProxyServer	Configure el proxy que se utilizará en las instancias del entorno.	nginx	apache nginx

Note

Para obtener más información acerca de la plataforma PHP, consulte [Uso de la plataforma PHP de Elastic Beanstalk](#).

Opciones de la plataforma Python

Espacio de nombres: **aws:elasticbeanstalk:application:environment**

Nombre	Descripción	Valor predeterminado	Valores válidos
DJANGO_SETTINGS_MODULE	Especifica qué archivo de configuración se va a utilizar.	n/a	n/a

Para obtener más información, consulte [Propiedades del entorno y otras opciones del software](#).

Espacio de nombres: **aws:elasticbeanstalk:container:python**

Nombre	Descripción	Valor predeterminado	Valores válidos
WSGIPath	Archivo que contiene la aplicación de WSGI. Este archivo debe tener una <code>application</code> que se pueda invocar.	En las versiones de la plataforma Python de Amazon Linux 2: <code>application</code> En las versiones de la	n/a

Nombre	Descripción	Valor predeterminado	Valores válidos
		plataforma Python AMI de Amazon Linux: application.py	
NumProcesos	Número de procesos del daemon que deben iniciarse en el grupo de procesos al ejecutar aplicaciones de WSGI.	1	n/a
NumThreads	Número de subprocesos que se van a crear para administrar las solicitudes de cada proceso daemon del grupo de procesos al ejecutar aplicaciones de WSGI.	15	n/a

Espacio de nombres: **aws:elasticbeanstalk:environment:proxy**

Nombre	Descripción	Valor predeterminado	Valores válidos
ProxyServer	Configure el proxy que se utilizará en las instancias del entorno.	nginx	apache nginx

Opciones de plataforma AMI de Amazon Linux (anteriores a Amazon Linux 2)

Espacio de nombres: **aws:elasticbeanstalk:container:python:staticfiles**

Puede utilizar el siguiente espacio de nombres para configurar el servidor proxy de forma que proporcione archivos estáticos. Cuando el servidor proxy recibe una solicitud de un archivo en la ruta especificada, proporciona directamente el archivo en lugar de direccionar la solicitud a la aplicación. De este modo, se reduce el número de solicitudes que la aplicación tiene que procesar.

Asigne una ruta proporcionada por el servidor proxy a una carpeta del código fuente que contenga activos estáticos. Cada una de las opciones que defina en este espacio de nombres asignará una ruta diferente.

De forma predeterminada, el servidor proxy en un entorno de Python proporciona los archivos en una carpeta llamada `static` situada en la ruta `/static`.

Espacio de nombres: **aws:elasticbeanstalk:container:python:staticfiles**

Nombre	Valor
Ruta donde el servidor proxy distribuirá los archivos.	Nombre de la carpeta que contiene los archivos.
Ejemplo: <code>/images</code> para distribuir archivos en <code>subdomain.eleasticbeanstalk.com/images</code> .	Ejemplo: <code>staticimages</code> para distribuir archivos de una carpeta con el nombre <code>staticimages</code> en la parte superior del paquete de código fuente.

Opciones de la plataforma Ruby

Espacio de nombres: **aws:elasticbeanstalk:application:environment**

Nombre	Descripción	Valor predeterminado	Valores válidos
<code>RAILS_SKIP_MIGRATIONS</code>	Especifica si se va a ejecutar <code>`rake db:migrate`</code> en nombre de las aplicaciones de los usuarios o si debe omitirse. Solo puede utilizarse con aplicaciones de Rails 3.	<code>false</code>	<code>true</code> <code>false</code>
<code>RAILS_SKIP_ASSET_COMPILATION</code>	Especifica si el contenedor debe ejecutar <code>`rake assets:precompile`</code> en nombre de las aplicaciones de los usuarios o si debe omitirse. Del mismo modo, solo puede utilizarse con aplicaciones de Rails 3.	<code>false</code>	<code>true</code> <code>false</code>

Nombre	Descripción	Valor predeterminado	Valores válidos
BUNDLE_WITHOUT	Lista de grupos separados por dos puntos (:) que se van a omitir al instalar dependencias desde un archivo Gemfile.	test:development	n/a
RACK_ENV	Especifica la fase del entorno en la que puede ejecutarse una aplicación. Algunos ejemplos de entornos comunes son el entorno de desarrollo, producción y pruebas.	production	n/a

Para obtener más información, consulte [Propiedades del entorno y otras opciones del software](#).

Opciones personalizadas

Utilice el espacio de nombres `aws:elasticbeanstalk:customoption` para definir opciones y valores que puedan consultarse en los bloques `Resources` de otros archivos de configuración. Utilice las opciones personalizadas para recopilar la configuración especificada por el usuario en un único archivo de configuración.

Por ejemplo, puede tener un archivo de configuración complejo donde se defina un recurso que el usuario pueda configurar al lanzar el entorno. Si utiliza `Fn::GetOptionSetting` para recuperar el valor de una opción personalizada, puede incluir la definición de esa opción en otro archivo de configuración, donde al usuario le resultará más fácil encontrarla y modificarla.

Además, como se trata de opciones de configuración, las opciones personalizadas se pueden configurar en el nivel de la API para anular los valores definidos en un archivo de configuración. Consulte [Prioridad](#) para obtener más información.

Las opciones personalizadas se definen igual que cualquier otra opción:

```
option_settings:
  aws:elasticbeanstalk:customoption:
    option name: option value
```

Por ejemplo, el siguiente archivo de configuración crea una opción denominada `ELBAlarmEmail` y establece su valor en `someone@example.com`:

```
option_settings:
  aws:elasticbeanstalk:customoption:
    ELBAlarmEmail: someone@example.com
```

Por otra parte, un archivo de configuración define un tema de SNS que consulta la opción utilizando `Fn::GetOptionSetting` para rellenar el valor del atributo `Endpoint`:

```
Resources:
  MySNSTopic:
    Type: AWS::SNS::Topic
    Properties:
      Subscription:
        - Endpoint:
            Fn::GetOptionSetting:
              OptionName: ELBAlarmEmail
              DefaultValue: nobody@example.com
          Protocol: email
```

Encontrará otros fragmentos de código de ejemplo en los que se utiliza `Fn::GetOptionSetting` en [Añadir y personalizar recursos del entorno de Elastic Beanstalk](#).

Personalización avanzada de entornos con archivos de configuración (**.ebextensions**)

Puede agregar archivos de AWS Elastic Beanstalk configuración (`.ebextensions`) al código fuente de la aplicación web para configurar el entorno y personalizar los AWS recursos que contiene.

[Los archivos de configuración son documentos con formato YAML o JSON con una extensión de `.config` archivo que se colocan en una carpeta denominada `.ebextensions` y se despliegan en el paquete fuente de la aplicación.](#)

Example `.ebextensions/ .config network-load-balancer`

En este ejemplo se realiza un simple cambio de configuración. Modifica una opción de configuración para establecer el tipo de balanceador de carga del entorno en el balanceador de carga de red.

```
option_settings:
```

```
aws:elasticbeanstalk:environment:  
  LoadBalancerType: network
```

Recomendamos utilizar YAML para los archivos de configuración, ya que es más fácil de leer que JSON. YAML admite comentarios, comandos multilínea, varias alternativas de uso de comillas, etc. Sin embargo, puede realizar cualquier cambio de configuración en los archivos de configuración de Elastic Beanstalk exactamente igual usando YAML o JSON.

Sugerencia

Cuando esté desarrollando o probando nuevos archivos de configuración, lance un entorno limpio que ejecute la aplicación predeterminada e implementelos en este entorno. Si los archivos de configuración no tienen un formato correcto, se producirán errores irreversibles al lanzar un nuevo entorno.

En la sección `option_settings` de un archivo de configuración, se definen los valores de las [opciones de configuración](#). Las opciones de configuración le permiten configurar el entorno de Elastic Beanstalk AWS, los recursos que contiene y el software que ejecuta la aplicación. Los archivos de configuración son solo uno de los diferentes mecanismos para definir las opciones de configuración.

La [Resource](#) sección le permite personalizar aún más los recursos del entorno de su aplicación y definir recursos AWS adicionales además de las funciones que ofrecen las opciones de configuración. Puede añadir y configurar cualquier recurso compatible con Elastic Beanstalk que utilice para crear entornos. AWS CloudFormation

El resto de las secciones de un archivo de configuración (`packages`, `sources`, `files`, `users`, `groups`, `commands`, `container_commands` y `services`) le permiten configurar las instancias EC2 que se lanzan en el entorno. Cuando se lanza un servidor en el entorno, Elastic Beanstalk ejecuta las operaciones definidas en estas secciones para preparar el sistema operativo y el sistema de almacenamiento de la aplicación.

Para obtener ejemplos de `.ebextensions` de uso común, consulte el [Repositorio de archivos de configuración de Elastic Beanstalk](#).

Requisitos

- Ubicación: Elastic Beanstalk `.ebextensions` procesará todas las carpetas presentes en la implementación. Sin embargo, le recomendamos que coloque todos los archivos de configuración

en una sola carpeta, denominada `.ebextensions`, en la raíz del paquete de código fuente. Los exploradores de archivos pueden ocultar las carpetas que comienzan con un punto, así que asegúrese de agregar la carpeta al crear el paquete de código fuente. Para obtener más información, consulte [Cree el paquete de código fuente de la aplicación.](#)

- Naming (Denominación): los archivos de configuración deben tener la extensión de archivo `.config`.
- Formatting (Formato): los archivos de configuración deben cumplir las especificaciones de formato de YAML o JSON.

Si utiliza YAML, use siempre espacios para aplicar sangría a las claves en los diferentes niveles de anidación. Para obtener más información acerca de YAML, consulte [YAML Ain't Markup Language \(YAML™\) \(versión 1.1\)](#).

- Uniqueness (Unicidad): use cada clave una sola vez en cada archivo de configuración.

Advertencia

Si utiliza una clave (por ejemplo, `option_settings`) dos veces en el mismo archivo de configuración, se descartará una de las secciones. Combine secciones duplicadas en una sola sección o colóquelas en archivos de configuración distintos.

El proceso de implementación varía ligeramente en función del cliente que se utilice para administrar los entornos. Consulte las secciones siguientes para obtener más detalles:

- [Consola de Elastic Beanstalk](#)
- [CLI DE EB](#)
- [AWS CLI](#)

Temas

- [Opciones de configuración](#)
- [Personalización de software en servidores Linux](#)
- [Personalización de software en servidores Windows](#)
- [Añadir y personalizar recursos del entorno de Elastic Beanstalk](#)

Opciones de configuración

Puede utilizar la clave `option_settings` para modificar la configuración de Elastic Beanstalk y definir variables que puedan recuperarse de su aplicación mediante variables de entorno. Algunos espacios de nombres le permiten ampliar el número de parámetros y especificar los nombres de parámetro. Para obtener una lista de espacios de nombres y opciones de configuración, consulte [Opciones de configuración](#).

La configuración de opciones se puede aplicar también directamente a un entorno durante la creación o la actualización de un entorno. La configuración se aplica directamente al entorno e invalida la configuración de las mismas opciones en los archivos de configuración. Si elimina los valores de la configuración de un entorno, se aplicarán los valores de los archivos de configuración. Para obtener más información, consulte [Prioridad](#).

Sintaxis

La sintaxis estándar para la configuración de opciones es una matriz de objetos, cada uno de los cuales tiene una clave `namespace`, `option_name` y `value`.

```
option_settings:  
  - namespace: namespace  
    option_name: option name  
    value: option value  
  - namespace: namespace  
    option_name: option name  
    value: option value
```

La clave `namespace` es opcional. Si no especifica un espacio de nombres, se usa el espacio de nombres predeterminado `aws:elasticbeanstalk:application:environment`:

```
option_settings:  
  - option_name: option name  
    value: option value  
  - option_name: option name  
    value: option value
```

Elastic Beanstalk admite también una sintaxis abreviada para la configuración de opciones que le permite especificar las opciones como pares de clave-valor bajo el espacio de nombres:

```
option_settings:
```



```
namespace:  
  option name: option value  
  option name: option value
```

Ejemplos

Los siguientes ejemplos establecen una opción específica de la plataforma Tomcat en el espacio de nombres `aws:elasticbeanstalk:container:tomcat:jvmoptions` y una propiedad de entorno denominada `MYPARAMETER`.

En formato YAML estándar:

Example `.ebextensions/options.config`

```
option_settings:  
  - namespace: aws:elasticbeanstalk:container:tomcat:jvmoptions  
    option_name: Xmx  
    value: 256m  
  - option_name: MYPARAMETER  
    value: parametervalue
```

En formato abreviado:

Example `.ebextensions/options.config`

```
option_settings:  
  aws:elasticbeanstalk:container:tomcat:jvmoptions:  
    Xmx: 256m  
  aws:elasticbeanstalk:application:environment:  
    MYPARAMETER: parametervalue
```

En JSON:

Example `.ebextensions/options.config`

```
{  
  "option_settings": [  
    {  
      "namespace": "aws:elasticbeanstalk:container:tomcat:jvmoptions",  
      "option_name": "Xmx",
```

```
    "value": "256m"
  },
  {
    "option_name": "MYPARAMETER",
    "value": "parametervalue"
  }
]
```

Personalización de software en servidores Linux

Tal vez desee personalizar y configurar el software del que depende la aplicación. Puede agregar comandos que se ejecutarán durante el aprovisionamiento de instancias; definir usuarios y grupos de Linux; y descargar o crear archivos directamente en las instancias de entorno. Estos archivos podrían ser dependencias necesarias para la aplicación (por ejemplo, paquetes adicionales del repositorio de yum) o archivos de configuración, por ejemplo, un archivo sustituto de un archivo de configuración del proxy que anulara ciertos valores específicos establecidos de forma predeterminada por Elastic Beanstalk).

En esta sección, se describe el tipo de información que se puede incluir en un archivo de configuración para personalizar el software de las instancias EC2 que ejecutan Linux. Para obtener información general acerca de cómo personalizar y configurar los entornos de Elastic Beanstalk, consulte [Configuración de entornos de Elastic Beanstalk](#). Para obtener más información sobre la personalización del software de las instancias EC2 que ejecutan Windows, consulte [Personalización de software en servidores Windows](#).

Notas

- En las plataformas de Amazon Linux 2, en lugar de proporcionar archivos y comandos en los archivos de configuración .ebextensions, recomendamos encarecidamente que utilice Buildfile, Procfile, y enlaces de la plataforma siempre que sea posible para configurar y ejecutar código personalizado en las instancias de entorno durante el aprovisionamiento de instancias. Para obtener más información sobre estos mecanismos, consulte [the section called “Ampliación de plataformas Linux”](#).
- YAML usa la sangría uniforme. Utilice el mismo nivel de sangría cuando sustituya el contenido en el archivo de configuración de ejemplo y asegúrese de que el editor de texto utiliza espacios para la sangría, no tabuladores.

Los archivos de configuración admiten las siguientes claves que afectan al servidor Linux en el que se ejecuta la aplicación.

Claves

- [Paquetes](#)
- [Grupos](#)
- [Usuarios](#)
- [Orígenes](#)
- [Archivos](#)
- [Comandos](#)
- [Servicios](#)
- [Comandos de contenedor](#)
- [Ejemplo: usar CloudWatch métricas personalizadas de Amazon](#)

Las claves se procesan en el orden en el que se muestran.

Vigile los [eventos](#) de su entorno mientras desarrolle y pruebe archivos de configuración. Elastic Beanstalk hará caso omiso de un archivo de configuración si este contiene errores de validación, como una clave no válida, por ejemplo, y no procesará ninguna de las claves contenidas en ese mismo archivo. Si esto sucede, Elastic Beanstalk agrega un evento de advertencia al registro de eventos.

Paquetes

Puede utilizar la clave `packages` para descargar e instalar aplicaciones y componentes previamente empaquetados.

Sintaxis

```
packages:  
  name of package manager:  
    package name: version  
    ...  
  name of package manager:  
    package name: version  
    ...  
  ...
```

Puede especificar varios paquetes con la clave de cada administrador de paquetes.

Formatos de paquetes admitidos

En la actualidad, Elastic Beanstalk es compatible con los siguientes administradores de paquetes: yum, rubygems, python y rpm. Los paquetes se procesan en el orden siguiente: rpm, yum y, después, rubygems y python. No hay ningún orden entre rubygems y python. Dentro de cada administrador de paquetes, no está garantizado el orden de instalación de paquetes. Utilice un administrador de paquetes que sea compatible con el sistema operativo.

Note

Elastic Beanstalk es compatible con dos administradores de paquetes subyacentes para Python: pip y easy_install. Sin embargo, en la sintaxis del archivo de configuración, debe especificarse el nombre del administrador de paquetes como python. Si utiliza un archivo de configuración para especificar un administrador de paquetes de Python, Elastic Beanstalk utilizará Python 2.7. Si la aplicación se basa en una versión diferente de Python, puede especificar los paquetes que se van a instalar en un archivo `requirements.txt`. Para obtener más información, consulte [Especificar dependencias mediante un archivo de requisitos](#).

Especificación de versiones

Dentro de cada administrador de paquetes, cada paquete se especifica con un nombre de paquete y una lista de versiones. La versión puede ser una cadena, una lista de versiones o una cadena o lista vacía. Una cadena o lista vacía indica que se debe usar la versión más reciente. Para el administrador de rpm, la versión se especifica como una ruta a un archivo en el disco o una URL. No se admiten rutas relativas.

Si especifica una versión de un paquete, Elastic Beanstalk intentará instalar esa versión incluso si ya hay una versión más reciente del paquete instalada en la instancia. Si ya hay una versión más reciente instalada, la implementación fallará. Algunos administradores de paquetes admiten varias versiones, pero puede haber otros que no. Consulte la documentación del administrador de paquetes para obtener más información. Si no se especifica ninguna versión y ya hay una versión del paquete instalada, Elastic Beanstalk no instalará una nueva versión, sino que presupondrá que se desea mantener y utilizar la versión existente.

Fragmento de código de ejemplo

En el siguiente fragmento de código, se especifica la URL de una versión de rpm y se solicita la última versión de yum y la versión 0.10.2 de chef de rubygems.

```
packages:
  yum:
    libmemcached: []
    ruby-devel: []
    gcc: []
  rpm:
    epel: http://download.fedoraproject.org/pub/epel/5/i386/epel-release-5-4.noarch.rpm
  rubygems:
    chef: '0.10.2'
```

Grupos

Puede utilizar la clave `groups` para crear grupos de Linux/UNIX y asignar identificadores de grupo. Para crear un grupo, añada un nuevo par clave-valor que asigne un nuevo nombre de grupo a un ID de grupo opcional. La clave “groups” puede contener uno o varios nombres de grupo. En la tabla siguiente, se muestran las claves disponibles.

Sintaxis

```
groups:
  name of group: {}
  name of group:
    gid: "group id"
```

Opciones

gid

Número de ID de grupo.

Si se especifica un ID de grupo y el nombre del grupo ya existe, se producirá un error al crear el grupo. Si otro grupo tuviera el ID de grupo especificado, el sistema operativo podría rechazar la creación del grupo.

Fragmento de código de ejemplo

En el siguiente fragmento de código, se especifica un grupo denominado `groupOne` sin asignar un ID de grupo y un grupo denominado `groupTwo` cuyo ID de grupo es 45.

```
groups:
  groupOne: {}
  groupTwo:
    gid: "45"
```

Usuarios

Puede utilizar la clave `users` para crear usuarios de Linux/UNIX en la instancia EC2.

Sintaxis

```
users:
  name of user:
    groups:
      - name of group
    uid: "id of the user"
    homeDir: "user's home directory"
```

Opciones

uid

ID de usuario. El proceso de creación falla si existe un nombre de usuario con otro ID. Si el ID de usuario ya estuviera asignado a un usuario existente, el sistema operativo podría rechazar la solicitud de creación.

groups

Lista de nombres de grupos. El usuario se agrega a cada grupo de la lista.

homeDir

Directorio de inicio del usuario.

Los usuarios se crean como usuarios de un sistema no interactivo con el shell `/sbin/nologin`. Esto es así por diseño y no se puede modificar.

Fragmento de código de ejemplo

```
users:
  myuser:
    groups:
      - group1
      - group2
    uid: "50"
    homeDir: "/tmp"
```

Orígenes

Puede utilizar la clave `sources` para descargar un archivo de almacenamiento de una URL pública y extraerlo en un directorio de destino de la instancia EC2.

Sintaxis

```
sources:
  target directory: location of archive file
```

Formatos admitidos

Los formatos admitidos son: tar, tar+gzip, tar+bz2 y zip. Puede hacer referencia a ubicaciones externas, como Amazon Simple Storage Service (Amazon S3) (por ejemplo, `https://mybucket.s3.amazonaws.com/myobject`) siempre que la URL esté accesible públicamente.

Fragmento de código de ejemplo

En el siguiente ejemplo, se descarga un archivo .zip público de un bucket de Amazon S3 y se desempaqueta en `/etc/myapp`:

```
sources:
  /etc/myapp: https://mybucket.s3.amazonaws.com/myobject
```

Note

Varias extracciones no deben reutilizar la misma ruta de destino. La extracción de otro origen en la misma ruta de destino reemplazará en lugar de añadirse al contenido.

Archivos

Puede utilizar la clave `files` para crear archivos en la instancia de EC2. El contenido puede estar insertado en el archivo de configuración o extraerse de una URL. Los archivos se escriben en el disco en orden lexicográfico.

Puede utilizar la clave `files` para descargar archivos privados de Amazon S3 si proporciona un perfil de instancia para su autorización.

Si la ruta de archivo especificada ya existe en la instancia, el archivo existente se conserva con la extensión `.bak` anexada a su nombre.

Sintaxis

```
files:
  "target file location on disk":
    mode: "six-digit octal value"
    owner: name of owning user for file
    group: name of owning group for file
    source: URL
    authentication: authentication name:

  "target file location on disk":
    mode: "six-digit octal value"
    owner: name of owning user for file
    group: name of owning group for file
    content: |
      # this is my
      # file content
    encoding: encoding format
    authentication: authentication name:
```

Opciones

content

Contenido de la cadena que se va a agregar al archivo. Especifique `content` o `source`, pero no ambos.

source

URL de un archivo que se va a descargar. Especifique `content` o `source`, pero no ambos.

encoding

Formato de codificación de la cadena especificada con la opción `content`.

Valores válidos: `plain` | `base64`

group

Grupo de Linux al que pertenece el archivo.

owner

Usuario de Linux al que pertenece el archivo.

mode

Un valor octal de seis dígitos que representa el modo para este archivo. No es compatible con sistemas Windows. Utilice los tres primeros dígitos para symlinks y los últimos tres dígitos para la configuración de permisos. Para crear un symlink, especifique `120xxx`, donde `xxx` define los permisos del archivo de destino. Para especificar los permisos de un archivo, utilice los tres últimos dígitos, como `000644`.

authentication

Nombre del [método de autenticación de AWS CloudFormation](#) que se utilizará. Puede añadir métodos de autenticación a los metadatos del grupo de Auto Scaling con la clave `Resources`. Consulte a continuación un ejemplo.

Fragmento de código de ejemplo

```
files:
  "/home/ec2-user/myfile" :
    mode: "000755"
    owner: root
    group: root
    source: http://foo.bar/myfile

  "/home/ec2-user/myfile2" :
    mode: "000755"
    owner: root
    group: root
    content: |
      this is my
      file content
```

En el ejemplo se utiliza un objeto symlink. Este crea un enlace, `/tmp/myfile2.txt`, que apunta al archivo existente `/tmp/myfile1.txt`.

```
files:
  "/tmp/myfile2.txt" :
    mode: "120400"
    content: "/tmp/myfile1.txt"
```

En el siguiente ejemplo, se utiliza la clave `Resources` para añadir un método de autenticación denominado `S3Auth` que se emplea para descargar un archivo privado en el bucket de Amazon S3:

```
Resources:
  AWSEBAutoScalingGroup:
    Metadata:
      AWS::CloudFormation::Authentication:
        S3Auth:
          type: "s3"
          buckets: ["elasticbeanstalk-us-west-2-123456789012"]
          roleName:
            "Fn::GetOptionSetting":
              Namespace: "aws:autoscaling:launchconfiguration"
              OptionName: "IamInstanceProfile"
              DefaultValue: "aws-elasticbeanstalk-ec2-role"

files:
  "/tmp/data.json" :
    mode: "000755"
    owner: root
    group: root
    authentication: "S3Auth"
    source: https://elasticbeanstalk-us-west-2-123456789012.s3-us-west-2.amazonaws.com/
data.json
```

Comandos

Puede utilizar la clave `commands` para ejecutar comandos en la instancia EC2. Los comandos se ejecutan antes de que se configuren la aplicación y el servidor web y se extraiga el archivo de versión de la aplicación.

Los comandos especificados se ejecutan como el usuario `root` y se procesan en orden alfabético por nombre. De forma predeterminada, los comandos se ejecutan en el directorio raíz. Para ejecutar comandos en otro directorio, utilice la opción `cwd`.

Para solucionar problemas con los comandos, puede encontrar su salida en los [registros de instancia](#).

Sintaxis

```
commands:
  command name:
    command: command to run
    cwd: working directory
    env:
      variable name: variable value
    test: conditions for command
    ignoreErrors: true
```

Opciones

command

Puede ser una matriz ([colección de secuencias de bloques](#) en la sintaxis YAML) o una cadena que especifique el comando que hay que ejecutar. Notas importantes:

- Si utiliza una cadena, no es preciso que la cadena completa vaya incluida entre comillas. Si aun así utiliza comillas, deberá utilizar el carácter de escape para las comillas literales del mismo tipo que aparezcan en la cadena.
- Si utiliza una matriz, no es preciso que utilizar el carácter de escape para los caracteres de espacio ni incluir los parámetros de comandos entre comillas. Cada elemento de la matriz es un solo argumento de comando. No utilice una matriz para especificar varios comandos.

Los siguientes ejemplos son equivalentes:

```
commands:
  command1:
    command: git commit -m "This is a comment."
  command2:
    command: "git commit -m \"This is a comment.\""
  command3:
    command: 'git commit -m "This is a comment."'
  command4:
    command:
      - git
      - commit
      - -m
```

```
- This is a comment.
```

Para especificar varios comandos, use un [escalar de bloque literal](#), tal y como se muestra en el siguiente ejemplo.

```
commands:
  command block:
    command: |
      git commit -m "This is a comment."
      git push
```

env

(Opcional) Establece las variables de entorno del comando. Esta propiedad sobrescribe, en lugar de anexar, el entorno existente.

cwd

(Opcional) El directorio de trabajo. Si no se especifica, los comandos se ejecutan en el directorio raíz (/).

test

(Opcional) Comando que debe devolver el valor `true` (código de salida 0) para que Elastic Beanstalk pueda procesar el comando (por ejemplo, un script de shell) incluido en la clave `command`.

ignoreErrors

(Opcional) Valor booleano que determina si otros comandos deben ejecutarse en caso de que se produzca un error (se devuelva un valor distinto de cero) en el comando incluido en la clave `command`. Establezca este valor en `true` si desea continuar ejecutando comandos aunque se produzca un error. Establézcalo en `false` si desea detener la ejecución de comandos en caso de que se produzca un error. El valor predeterminado es `false`.

Fragmento de código de ejemplo

En el siguiente fragmento de ejemplo se ejecuta un script de Python.

```
commands:
  python_install:
```

```
command: myscript.py
cwd: /home/ec2-user
env:
  myvarname: myvarvalue
test: "[ -x /usr/bin/python ]"
```

Servicios

Puede utilizar la clave `services` para definir qué servicios deben iniciarse o detenerse cuando se lance la instancia. La clave `services` también le permite especificar dependencias en orígenes, paquetes y archivos, de manera que si es necesario reiniciar debido a los archivos que se están instalado, Elastic Beanstalk se encargará de reiniciar el servicio.

Sintaxis

```
services:
  sysvinit:
    name of service:
      enabled: "true"
      ensureRunning: "true"
      files:
        - "file name"
      sources:
        - "directory"
      packages:
        name of package manager:
          "package name[: version]"
      commands:
        - "name of command"
```

Opciones

ensureRunning

Establézcalo en `true` para asegurarse de que el servicio se está ejecutando una vez que Elastic Beanstalk finalice.

Establézcalo en `false` para asegurarse de que el servicio no se está ejecutando una vez que Elastic Beanstalk finalice.

Omita esta clave para no realizar ningún cambio en el estado del servicio.

enabled

Establézcalo en `true` para garantizar que el servicio se iniciará automáticamente al arrancar.

Establézcalo en `false` para garantizar que el servicio no se iniciará automáticamente al arrancar.

Omita esta clave para no realizar ningún cambio a esta propiedad.

files

Una lista de archivos. Si Elastic Beanstalk cambia uno directamente a través del bloque de archivos, el servicio se reinicia.

sources

Una lista de directorios. Si Elastic Beanstalk expande un archivo en uno de estos directorios, el servicio se reinicia.

packages

Una asignación del administrador de paquetes a una lista de nombres de paquetes. Si Elastic Beanstalk instala o actualiza uno de estos paquetes, el servicio se reinicia.

commands

Una lista de nombres de comandos. Si Elastic Beanstalk ejecuta el comando especificado, se reinicia el servicio.

Fragmento de código de ejemplo

A continuación, se muestra un fragmento de código de ejemplo:

```
services:
  sysvinit:
    myservice:
      enabled: true
      ensureRunning: true
```

Comandos de contenedor

Puede utilizar la clave `container_commands` para ejecutar comandos que afecten el código abierto de la aplicación. Estos comandos se ejecutan una vez configurada la aplicación y el servidor

web, y una vez que se ha extraído el archivo de versiones de la aplicación, pero antes de que se implemente la versión de la aplicación. Los comandos que no son de contenedor y otras operaciones de personalización se realizan antes de que se extraiga el código fuente de la aplicación.

Los comandos especificados se ejecutan como el usuario root y se procesan en orden alfabético por nombre. Los comandos de contenedor se ejecutan desde el directorio de ensayo, del que se extrae el código fuente antes de implementarse en el servidor de la aplicación. Todos los cambios que realice en el código fuente en el directorio de ensayo con un comando de contenedor se incluirán cuando se implemente el código fuente en su ubicación final.

Note

Los resultados de los comandos del contenedor se registran en el registro de instancias `cfn-init-cmd.log`. Para obtener más información sobre cómo recuperar y ver los registros de instancias, consulte [Visualización de registros de instancias de Amazon EC2](#).

Puede utilizar `leader_only` para ejecutar el comando en una sola instancia o configurar un objeto `test` para que el comando se ejecute exclusivamente cuando un comando "test" se evalúe como `true`. Los comandos de contenedor "leader-only" solo se ejecutan durante la creación del entorno y las implementaciones, mientras que otros comandos y operaciones de personalización del servidor se ejecutan cada vez que se aprovisiona o se actualiza una instancia. Los comandos de contenedor "leader-only" no se ejecutan debido a los cambios en la configuración de lanzamiento, como un cambio en el ID de AMI o tipo de instancia.

Sintaxis

```
container_commands:
  name_of_container_command:
    command: "command to run"
    leader_only: true
  name_of_container_command:
    command: "command to run"
```

Opciones

command

Cadena o matriz de cadenas que se va a ejecutar.

env

(Opcional) Establezca las variables de entorno antes de ejecutar el comando para invalidar cualquier valor existente.

cwd

(Opcional) El directorio de trabajo. De forma predeterminada, este es el directorio de ensayo de la aplicación sin descomprimir.

leader_only

(Opcional) Ejecute solo el comando en una sola instancia seleccionada por Elastic Beanstalk. Los comandos de contenedor "leader-only" se ejecutan antes que otros comandos de contenedor. Un comando puede ser "leader-only" o tener un objeto `test`, pero no ambas cosas (`leader_only` tiene prioridad).

test

(Opcional) Ejecute un comando de prueba que deba devolver `true` para poder ejecutar el comando de contenedor. Un comando puede ser "leader-only" o tener un objeto `test`, pero no ambas cosas (`leader_only` tiene prioridad).

ignoreErrors

(Opcional) No produce un error en las implementaciones si el comando de contenedor devuelve un valor distinto de 0 (correcto). Establézcalo en `true` para habilitarlo.

Fragmento de código de ejemplo

A continuación, se muestra un fragmento de código de ejemplo.

```
container_commands:
  collectstatic:
    command: "django-admin.py collectstatic --noinput"
  01syncdb:
    command: "django-admin.py syncdb --noinput"
    leader_only: true
  02migrate:
    command: "django-admin.py migrate"
    leader_only: true
  99customize:
    command: "scripts/customize.sh"
```


Ejemplo: usar CloudWatch métricas personalizadas de Amazon

Amazon CloudWatch es un servicio web que le permite supervisar, gestionar y publicar diversas métricas, así como configurar acciones de alarma en función de los datos de las métricas. Puede definir métricas personalizadas para su propio uso y Elastic Beanstalk las enviará a Amazon. CloudWatch Una vez que Amazon CloudWatch contenga tus métricas personalizadas, podrás verlas en la CloudWatch consola de Amazon.

Important

Los scripts de CloudWatch monitoreo de Amazon están obsoletos. El CloudWatch agente ahora ha reemplazado los scripts CloudWatch de monitoreo para recopilar métricas y registros.

Si sigue migrando de los scripts de monitoreo obsoletos al agente y necesita información sobre los scripts de monitoreo, consulte [Obsoleto: recopilar métricas mediante los scripts de CloudWatch monitoreo en la Guía del usuario](#) de Amazon EC2.

El CloudWatch agente de Amazon

El CloudWatch agente de Amazon permite la recopilación de CloudWatch métricas y registros tanto de instancias de Amazon EC2 como de servidores locales en todos los sistemas operativos. El agente admite las métricas recopiladas a nivel del sistema. También admite la recopilación de métricas y registros personalizados desde las aplicaciones o servicios. Para obtener más información sobre el CloudWatch agente de Amazon, consulta [Cómo recopilar estadísticas y registros con el CloudWatch agente](#) en la Guía del CloudWatch usuario de Amazon.

Note

Los informes de salud [mejorados de Elastic Beanstalk](#) cuentan con soporte nativo para publicar una amplia gama de métricas de instancias y entornos en. CloudWatch Para obtener más información, consulte [Publicación de métricas personalizadas de un entorno en Amazon CloudWatch](#).

Temas

- [Archivo de configuración .ebextensions](#)
- [Permisos](#)

- [Visualización de las métricas en la CloudWatch consola](#)

Archivo de configuración .ebextensions

En este ejemplo, se utilizan los archivos y comandos de un archivo de configuración .ebextensions para configurar y ejecutar el CloudWatch agente de Amazon en la plataforma Amazon Linux 2. El agente está preempaquetado con Amazon Linux 2. Si utiliza otro sistema operativo, es posible que sean necesarios pasos adicionales para instalar el agente. Para obtener más información, consulte [Instalación del CloudWatch agente](#) en la Guía del CloudWatch usuario de Amazon.

Para utilizar este ejemplo, guárdelo en un archivo llamado `cloudwatch.config` que esté situado en un directorio denominado `.ebextensions` en el nivel superior del directorio del proyecto. A continuación, implemente la aplicación a través de la consola de Elastic Beanstalk (incluya el directorio `.ebextensions` en el [paquete de código fuente](#)) o la [CLI de EB](#).

Para obtener más información sobre los archivos de configuración, consulte [Personalización avanzada de entornos con archivos de configuración \(.ebextensions\)](#).

.ebextensions/cloudwatch.config

```
files:
  "/opt/aws/amazon-cloudwatch-agent/bin/config.json":
    mode: "000600"
    owner: root
    group: root
    content: |
      {
        "agent": {
          "metrics_collection_interval": 60,
          "run_as_user": "root"
        },
        "metrics": {
          "namespace": "System/Linux",
          "append_dimensions": {
            "AutoScalingGroupName": "${aws:AutoScalingGroupName}"
          },
          "metrics_collected": {
            "mem": {
              "measurement": [
                "mem_used_percent"
              ]
            }
          }
        }
      }
```

```
        }
      }
    }
  }
  container_commands:
    start_cloudwatch_agent:
      command: /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
config -m ec2 -s -c file:/opt/aws/amazon-cloudwatch-agent/bin/config.json
```

Este archivo tiene dos secciones:

- **files:** en esta sección se agrega el archivo de configuración del agente. Indica qué métricas y registros debe enviar el agente a Amazon CloudWatch. En este ejemplo, solo enviamos la métrica `mem_used_percent`. Para obtener una lista completa de las métricas a nivel del sistema admitidas por el CloudWatch agente de Amazon, consulta [las métricas recopiladas por el CloudWatch agente](#) en la Guía del CloudWatch usuario de Amazon.
- **container_commands:** esta sección contiene el comando que inicia el agente al transferir el archivo de configuración como parámetro. Para obtener información detallada acerca de `container_commands`, consulte [Comandos de contenedor](#).

Permisos

Las instancias de su entorno necesitan los permisos de IAM adecuados para poder publicar CloudWatch métricas de Amazon personalizadas mediante el CloudWatch agente de Amazon. Los permisos para las instancias del entorno se conceden agregándolas al [perfil de instancias](#) del entorno. Los permisos pueden agregarse al perfil de instancia antes o después de implementar la aplicación.

Para conceder permisos para publicar métricas CloudWatch

1. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
2. Seleccione Roles en el panel de navegación.
3. Seleccione el rol del perfil de instancia del entorno. De forma predeterminada, cuando se crea un entorno con la consola de Elastic Beanstalk o la [CLI de EB](#), el rol es `aws-elasticbeanstalk-ec2-role`.
4. Elija la pestaña Permisos.
5. Bajo Permissions Policies (Políticas de permisos), en la sección Permissions (Permisos), elija Attach policies (Adjuntar políticas).

6. En Adjuntar permisos, selecciona la política AWS gestionada CloudWatchAgentServerPolicy. Haga clic en Attach Policy (Adjuntar política).

Para obtener más información acerca de la administración de políticas, consulte [Uso de las políticas](#) en la Guía del usuario de IAM.

Visualización de las métricas en la CloudWatch consola

Tras implementar el archivo de CloudWatch configuración en su entorno, consulte la [CloudWatch consola de Amazon](#) para ver las métricas. Las métricas personalizadas se ubicarán en el espacio de nombres CWAgent.

Para obtener más información, consulta Cómo [ver las métricas disponibles](#) en la Guía del CloudWatch usuario de Amazon.

Personalización de software en servidores Windows

Tal vez desee personalizar y configurar el software del que depende la aplicación. Estos archivos puede ser dependencias requeridas por la aplicación, como paquetes adicionales o servicios que necesite ejecutar. Para obtener información general acerca de cómo personalizar y configurar los entornos de Elastic Beanstalk, consulte [Configuración de entornos de Elastic Beanstalk](#).

Note

YAML usa la sangría uniforme. Utilice el mismo nivel de sangría cuando sustituya el contenido en el archivo de configuración de ejemplo y asegúrese de que el editor de texto utiliza espacios para la sangría, no tabuladores.

Los archivos de configuración admiten las siguientes claves que afectan al servidor Windows en el que se ejecuta la aplicación.

Claves

- [Paquetes](#)
- [Orígenes](#)
- [Archivos](#)
- [Comandos](#)
- [Servicios](#)

- [Comandos de contenedor](#)

Las claves se procesan en el orden en el que se muestran.

Note

Las versiones más antiguas (sin control de versiones) de la plataforma .NET no procesan los archivos de configuración en el orden correcto. Encontrará más información en [Migración entre las versiones principales de la plataforma Windows Server de Elastic Beanstalk](#).

Vigile los [eventos](#) de su entorno mientras desarrolle y pruebe archivos de configuración. Elastic Beanstalk hará caso omiso de un archivo de configuración si este contiene errores de validación, como una clave no válida, por ejemplo, y no procesará ninguna de las claves contenidas en ese mismo archivo. Si esto sucede, Elastic Beanstalk agrega un evento de advertencia al registro de eventos.

Paquetes

Utilice la clave `packages` para descargar e instalar aplicaciones y componentes previamente empaquetados.

En entornos de Windows, Elastic Beanstalk admite la descarga e instalación de paquetes MSI. (Los entornos Linux admiten administradores de paquetes adicionales. Para obtener más información, consulte [Paquetes](#) en la página Personalización de software en servidores Linux.)

Puede hacer referencia a cualquier ubicación externa, como un objeto de Amazon Simple Storage Service (Amazon S3), siempre que la URL sea accesible públicamente.

Si especifica varios paquetes `msi:`, su orden de instalación no está garantizado.

Sintaxis

Elija un nombre como el nombre del paquete y una URL a una ubicación de archivo MSI como el valor. Puede especificar varios paquetes en la clave `msi:`.

```
packages:  
  msi:  
    package name: package url  
    ...
```

Ejemplos

El siguiente ejemplo especifica una URL para descargar mysql de `https://dev.mysql.com/`.

```
packages:
  msi:
    mysql: https://dev.mysql.com/get/Downloads/Connector-Net/mysql-connector-
net-8.0.11.msi
```

En el siguiente ejemplo se especifica un objeto de Amazon S3 como la ubicación del archivo MSI.

```
packages:
  msi:
    mymsi: https://mybucket.s3.amazonaws.com/myobject.msi
```

Orígenes

Utilice la clave `sources` para descargar un archivo de almacenamiento de una URL pública y extraerlo en un directorio de destino de la instancia EC2.

Sintaxis

```
sources:
  target directory: location of archive file
```

Formatos admitidos

En entornos Windows, Elastic Beanstalk admite el formato .zip. (Los entornos Linux admiten formatos adicionales. Para obtener más información, consulte [Orígenes](#) en la página Personalización de software en servidores Linux.)

Puede hacer referencia a cualquier ubicación externa, como un objeto de Amazon Simple Storage Service (Amazon S3), siempre que la URL sea accesible públicamente.

Ejemplo

En el siguiente ejemplo, se descarga un archivo .zip público de un bucket de Amazon S3 y se desempaqueta en `c:/myproject/myapp`.

```
sources:
```

```
"c:/myproject/myapp": https://mybucket.s3.amazonaws.com/myobject.zip
```

Archivos

Utilice la clave `files` para crear archivos en la instancia EC2. El contenido puede estar insertado en el archivo de configuración o extraerse de una URL. Los archivos se escriben en el disco en orden lexicográfico. Para descargar archivos privados de Amazon S3 proporcione un perfil de instancia para su autorización.

Sintaxis

```
files:
  "target file location on disk":
    source: URL
    authentication: authentication name:

  "target file location on disk":
    content: |
      this is my content
    encoding: encoding format
```

Opciones

content

(Opcional) Cadena A.

source

(Opcional) La dirección URL desde la que se carga el archivo. Esta opción no se puede especificar con la clave de contenido.

encoding

(Opcional) El formato de la codificación. Esta opción solo se utiliza para un valor de clave de contenido proporcionado. El valor predeterminado es `plain`.

Valores válidos: `plain` | `base64`

authentication

(Opcional) el nombre del [método de autenticación de AWS CloudFormation](#) que se va a utilizar. Puede añadir métodos de autenticación a los metadatos del grupo de Auto Scaling con la clave `Resources`.

Ejemplos

En el siguiente ejemplo se muestran las dos formas de proporcionar el contenido del archivo: desde una URL o en línea en el archivo de configuración.

```
files:
  "c:\\targetdirectory\\targetfile.txt":
    source: http://foo.bar/myfile

  "c:/targetdirectory/targetfile.txt":
    content: |
      # this is my file
      # with content
```

Note

Si utiliza una barra diagonal inversa (\) en la ruta del archivo, debe anteponer otra barra diagonal inversa (el carácter de escape), tal y como se muestra en el ejemplo anterior.

En el siguiente ejemplo, se utiliza la clave Resources para añadir un método de autenticación denominado S3Auth que se emplea para descargar un archivo privado en el bucket de Amazon S3:

```
files:
  "c:\\targetdirectory\\targetfile.zip":
    source: https://elasticbeanstalk-us-east-2-123456789012.s3.amazonaws.com/prefix/myfile.zip
    authentication: S3Auth

Resources:
  AWSEBAutoScalingGroup:
    Metadata:
      AWS::CloudFormation::Authentication:
        S3Auth:
          type: "s3"
          buckets: ["elasticbeanstalk-us-east-2-123456789012"]
          roleName:
            "Fn::GetOptionSetting":
              Namespace: "aws:autoscaling:launchconfiguration"
              OptionName: "IamInstanceProfile"
              DefaultValue: "aws-elasticbeanstalk-ec2-role"
```


Comandos

Utilice la clave `commands` para ejecutar comandos en la instancia EC2. Los comandos se procesan en orden alfabético por nombre, y se ejecutan antes de que se configure la aplicación y el servidor web y se extraiga el archivo de versión de la aplicación.

Los comandos especificados se ejecutan como usuario administrador.

Para solucionar problemas con los comandos, puede encontrar su salida en los [registros de instancia](#).

Sintaxis

```
commands :  
  command name :  
    command: command to run
```

Opciones

`command`

Una matriz o bien una cadena que especifica el comando que se va a ejecutar. Si utiliza una matriz, no es preciso que utilice el carácter de escape para los caracteres de espacio ni que incluya los parámetros de los comandos entre comillas.

`cwd`

(Opcional) El directorio de trabajo. De forma predeterminada, Elastic Beanstalk intenta encontrar la ubicación del directorio del proyecto. Si no la encuentra, utiliza `c:\Windows\System32` de forma predeterminada.

`env`

(Opcional) Establece las variables de entorno del comando. Esta propiedad sobrescribe, en lugar de anexar, el entorno existente.

`ignoreErrors`

(Opcional) Valor booleano que determina si otros comandos deben ejecutarse en caso de que se produzca un error (se devuelva un valor distinto de cero) en el comando incluido en la clave `command`. Establezca este valor en `true` si desea continuar ejecutando comandos aunque se produzca un error. Establézcalo en `false` si desea detener la ejecución de comandos en caso de que se produzca un error. El valor predeterminado es `false`.

test

(Opcional) Un comando que debe devolver el valor `true` (código de salida 0) para que Elastic Beanstalk pueda procesar el comando incluido en la clave `command`.

waitAfterCompletion

(Opcional) Segundos que se deben esperar a que se complete el comando antes de ejecutar el siguiente comando. Si el sistema debe reiniciarse después de que se complete el comando, se reinicia después del número especificado de segundos. Si el sistema se reinicia como resultado de un comando, Elastic Beanstalk restablecerá el sistema al punto después del comando del archivo de configuración. El valor predeterminado es **60** segundos. También puede especificar **forever**, pero el sistema debe reiniciarse para poder ejecutar otro comando.

Ejemplo

El siguiente ejemplo guarda el resultado del comando `set` en el archivo especificado. Si hay un comando posterior, Elastic Beanstalk ejecuta ese comando inmediatamente después de que se complete este comando. Si este comando requiere un reinicio, Elastic Beanstalk reinicia la instancia inmediatamente después de que se complete el comando.

```
commands:
  test:
    command: set > c:\\myapp\\set.txt
    waitAfterCompletion: 0
```

Servicios

Utilice la clave `services` para definir qué servicios deben iniciarse o detenerse cuando se lance la instancia. La clave `services` también le permite especificar dependencias en orígenes, paquetes y archivos, de manera que si es necesario reiniciar debido a los archivos que se están instalando, Elastic Beanstalk se encargará de reiniciar el servicio.

Sintaxis

```
services:
  windows:
    name of service:
      files:
        - "file name"
```

```
sources:
  - "directory"
packages:
  name of package manager:
    "package name[: version]"
commands:
  - "name of command"
```

Opciones

ensureRunning

(Opcional) Establézcalo en `true` para asegurarse de que el servicio se está ejecutando una vez que Elastic Beanstalk finalice.

Establézcalo en `false` para asegurarse de que el servicio no se está ejecutando una vez que Elastic Beanstalk finalice.

Omita esta clave para no realizar ningún cambio en el estado del servicio.

enabled

(Opcional) Establézcalo en `true` para garantizar que el servicio se iniciará automáticamente al arrancar.

Establézcalo en `false` para garantizar que el servicio no se iniciará automáticamente al arrancar.

Omita esta clave para no realizar ningún cambio a esta propiedad.

files

Una lista de archivos. Si Elastic Beanstalk cambia uno directamente a través del bloque de archivos, el servicio se reinicia.

sources

Una lista de directorios. Si Elastic Beanstalk expande un archivo en uno de estos directorios, el servicio se reinicia.

packages

Una asignación del administrador de paquetes a una lista de nombres de paquetes. Si Elastic Beanstalk instala o actualiza uno de estos paquetes, el servicio se reinicia.

commands

Una lista de nombres de comandos. Si Elastic Beanstalk ejecuta el comando especificado, se reinicia el servicio.

Ejemplo

```
services:
  windows:
    myservice:
      enabled: true
      ensureRunning: true
```

Comandos de contenedor

Utilice la clave `container_commands` para ejecutar comandos que afecten al código fuente de la aplicación. Estos comandos se ejecutan una vez configurada la aplicación y el servidor web, y una vez que se ha extraído el archivo de versiones de la aplicación, pero antes de que se implemente la versión de la aplicación. Los comandos que no son de contenedor y otras operaciones de personalización se realizan antes de que se extraiga el código fuente de la aplicación.

Los comandos de contenedor se ejecutan desde el directorio de ensayo, del que se extrae el código fuente antes de implementarse en el servidor de la aplicación. Todos los cambios que realice en el código fuente en el directorio de ensayo con un comando de contenedor se incluirán cuando se implemente el código fuente en su ubicación final.

Para solucionar problemas con los comandos de contenedor, puede encontrar su salida en los [registros de instancia](#).

Utilice la opción `leader_only` para ejecutar el comando en una sola instancia o configure un objeto `test` para que el comando se ejecute exclusivamente cuando un comando "test" se evalúe como `true`. Los comandos de contenedor "leader-only" solo se ejecutan durante la creación del entorno y las implementaciones, mientras que otros comandos y operaciones de personalización del servidor se ejecutan cada vez que se aprovisiona o se actualiza una instancia. Los comandos de contenedor "leader-only" no se ejecutan debido a los cambios en la configuración de lanzamiento, como un cambio en el ID de AMI o tipo de instancia.

Sintaxis

```
container_commands:
```

```
name of container_command:  
  command: command to run
```

Opciones

command

Cadena o matriz de cadenas que se va a ejecutar.

env

(Opcional) Establezca las variables de entorno antes de ejecutar el comando para invalidar cualquier valor existente.

cwd

(Opcional) El directorio de trabajo. De forma predeterminada, este es el directorio de ensayo de la aplicación sin descomprimir.

leader_only

(Opcional) Ejecute solo el comando en una sola instancia seleccionada por Elastic Beanstalk. Los comandos de contenedor "leader-only" se ejecutan antes que otros comandos de contenedor. Un comando puede ser "leader-only" o tener un objeto `test`, pero no ambas cosas (`leader_only` tiene prioridad).

test

(Opcional) Ejecute un comando de prueba que deba devolver `true` para poder ejecutar el comando de contenedor. Un comando puede ser "leader-only" o tener un objeto `test`, pero no ambas cosas (`leader_only` tiene prioridad).

ignoreErrors

(Opcional) No produce un error en las implementaciones si el comando de contenedor devuelve un valor distinto de 0 (correcto). Establézcalo en `true` para habilitarlo.

waitAfterCompletion

(Opcional) Segundos que se deben esperar a que se complete el comando antes de ejecutar el siguiente comando. Si el sistema debe reiniciarse después de que se complete el comando, se reinicia después del número especificado de segundos. Si el sistema se reinicia como resultado de un comando, Elastic Beanstalk restablecerá el sistema al punto después del comando del

archivo de configuración. El valor predeterminado es **60** segundos. También puede especificar **forever**, pero el sistema debe reiniciarse para poder ejecutar otro comando.

Ejemplo

El siguiente ejemplo guarda el resultado del comando `set` en el archivo especificado. Elastic Beanstalk ejecuta el comando en una sola instancia y reinicia la instancia inmediatamente después de que se complete el comando.

```
container_commands:
  foo:
    command: set > c:\\myapp\\set.txt
    leader_only: true
    waitAfterCompletion: 0
```

Añadir y personalizar recursos del entorno de Elastic Beanstalk

Es posible que desee personalizar los recursos que forman parte de su entorno de Elastic Beanstalk. Por ejemplo, es posible que desee agregar una cola de Amazon SQS y una alarma sobre la profundidad de cola o un clúster de Amazon ElastiCache. Puede personalizar fácilmente el entorno al mismo tiempo que implementa la versión de la aplicación incluyendo un archivo de configuración con el paquete de código fuente.

Puede utilizar la clave `Resources` de un [archivo de configuración](#) para crear y personalizar recursos de AWS en el entorno. Los recursos definidos en los archivos de configuración se añaden a la plantilla de AWS CloudFormation utilizada para lanzar el entorno. Se admiten todos los AWS CloudFormation tipos de recursos [de](#).

Note

Siempre que agregue un recurso que no esté administrado por Elastic Beanstalk, asegúrese de agregar una política de usuario con los permisos adecuados para los usuarios de AWS Identity and Access Management (IAM). Las [políticas de usuario administradas](#) que Elastic Beanstalk proporciona solo incluyen permisos para los recursos administrados por Elastic Beanstalk.

Por ejemplo, el siguiente archivo de configuración añade un enlace de ciclo de vida de Auto Scaling al grupo de Auto Scaling creado por Elastic Beanstalk:

~/my-app/.ebextensions/as-hook.config

```
Resources:
  hookrole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument: {
        "Version" : "2012-10-17",
        "Statement": [ {
          "Effect": "Allow",
          "Principal": {
            "Service": [ "autoscaling.amazonaws.com" ]
          },
          "Action": [ "sts:AssumeRole" ]
        } ]
      }
    Policies: [ {
      "PolicyName": "SNS",
      "PolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [{
          "Effect": "Allow",
          "Resource": "*",
          "Action": [
            "sqs:SendMessage",
            "sqs:GetQueueUrl",
            "sns:Publish"
          ]
        } ]
      }
    ]
  } ]

  hooktopic:
    Type: AWS::SNS::Topic
    Properties:
      Subscription:
        - Endpoint: "my-email@example.com"
          Protocol: email

  lifecyclehook:
    Type: AWS::AutoScaling::LifecycleHook
    Properties:
      AutoScalingGroupName: { "Ref" : "AWSEBAutoScalingGroup" }
      LifecycleTransition: autoscaling:EC2_INSTANCE_TERMINATING
      NotificationTargetARN: { "Ref" : "hooktopic" }
```

```
RoleARN: { "Fn::GetAtt" : [ "hookrole", "Arn"] }
```

Este ejemplo define tres recursos, `hookrole`, `hooktopic` y `lifecyclehook`. Los dos primeros recursos son un rol de IAM, que concede permiso a Amazon EC2 Auto Scaling para publicar mensajes en Amazon SNS, y un tema SNS, que retransmite mensajes del grupo de Auto Scaling a una dirección de correo electrónico. Elastic Beanstalk crea estos recursos con los tipos y propiedades especificados.

El recurso final, `lifecyclehook`, es el propio enlace de ciclo de vida:

```
lifecyclehook:
  Type: AWS::AutoScaling::LifecycleHook
  Properties:
    AutoScalingGroupName: { "Ref" : "AWSEBAutoScalingGroup" }
    LifecycleTransition: autoscaling:EC2_INSTANCE_TERMINATING
    NotificationTargetARN: { "Ref" : "hooktopic" }
    RoleARN: { "Fn::GetAtt" : [ "hookrole", "Arn"] }
```

La definición del enlace de ciclo de vida usa dos [funciones](#) para rellenar los valores de las propiedades del enlace. `{ "Ref" : "AWSEBAutoScalingGroup" }` recupera el nombre del grupo de auto Scaling creado por Elastic Beanstalk para el entorno. `AWSEBAutoScalingGroup` es uno de los [nombres de recurso](#) estándar proporcionados por Elastic Beanstalk.

Para [AWS::IAM::Role](#), `Ref` solo devuelve el nombre del rol, no el ARN. Para obtener el ARN del parámetro `RoleARN`, se utiliza otra función intrínseca, `Fn::GetAtt`, en su lugar, que puede obtener cualquier atributo de un recurso. `RoleARN: { "Fn::GetAtt" : ["hookrole", "Arn"] }` obtiene el atributo `Arn` del recurso `hookrole`.

`{ "Ref" : "hooktopic" }` obtiene el ARN del tema de Amazon SNS creado anteriormente en el archivo de configuración. El valor devuelto por `Ref` varía en función del recurso y puede encontrarse en el AWS CloudFormation tema del tipo de recurso `AWS::SNS::Topic` [de la Guía del usuario de](#) .

Modificar los recursos que Elastic Beanstalk crea para su entorno

Los recursos que Elastic Beanstalk crea para su entorno tienen nombres. Puede utilizar estos nombres para obtener información sobre los recursos con una [función](#) o modificar las propiedades de los recursos para personalizar su comportamiento. Este tema describe los recursos de AWS que Elastic Beanstalk utiliza en los diferentes tipos de entornos.

Note

El tema anterior, [Recursos personalizados](#), proporciona algunos casos de uso y ejemplos para personalizar los recursos del entorno. También puede encontrar más ejemplos de archivos de configuración en el tema posterior [Ejemplos de recursos personalizados](#).

Los entornos del servidor web tienen los siguientes recursos.

Entornos del servidor web

- `AWSEBAutoScalingGroup` ([AWS::AutoScaling::AutoScalingGroup](#)): el grupo de Auto Scaling asociado a su entorno.
- Uno de los dos recursos siguientes.
 - `AWSEBAutoScalingLaunchConfiguration` ([AWS::AutoScaling::LaunchConfiguration](#)): la configuración de lanzamiento asociada al grupo de Auto Scaling de su entorno.
 - `AWSEBEC2LaunchTemplate` ([AWS::EC2::LaunchTemplate](#)): la plantilla de inicio de Amazon EC2 que utiliza el grupo de Auto Scaling de su entorno.

Note

Si el entorno utiliza una funcionalidad que necesita plantillas de lanzamiento de Amazon EC2 y la política del usuario carece de los permisos necesarios, es posible que se produzca un error al crear o actualizar el entorno. Utilice la política de usuario administrada [AdministratorAccess-AWSElasticBeanstalk](#) o agregue los permisos requeridos a la [política personalizada](#).

- `AWSEBEnvironmentName` ([AWS::ElasticBeanstalk::Environment](#)): su entorno.
- `AWSEBSecurityGroup` ([AWS::EC2::SecurityGroup](#)): el grupo de seguridad asociado al grupo de Auto Scaling.
- `AWSEBRDSDatabase` ([AWS::RDS::DBInstance](#)): la instancia de base de datos de Amazon RDS asociada a su entorno (si procede).

En un entorno con balanceo de carga, puede obtener acceso a recursos adicionales relacionados con el balanceador de carga. Los Classic Load Balancers tienen un recurso para el balanceador de carga y otro para el grupo de seguridad asociado. Los balanceadores de carga de aplicaciones y de

red disponen de recursos adicionales para el agente de escucha, la regla para el agente de escucha y el grupo de destino predeterminados.

Entornos con balanceo de carga

- `AWSEBLoadBalancer` ([AWS::ElasticLoadBalancing::LoadBalancer](#)): balanceador de carga clásico de su entorno.
- `AWSEBV2LoadBalancer` ([AWS::ElasticLoadBalancingV2::LoadBalancer](#)): balanceador de carga de la aplicación o de la red de su entorno.
- `AWSEBLoadBalancerSecurityGroup` ([AWS::EC2::SecurityGroup](#)): solo en una nube privada personalizada de [Amazon Virtual Private Cloud](#) (Amazon VPC), el nombre del grupo de seguridad que Elastic Beanstalk crea para el balanceador de carga. En una VPC predeterminada o instancia de EC2 Classic, Elastic Load Balancing asigna un grupo de seguridad predeterminado al balanceador de carga.
- `AWSEBV2LoadBalancerListener` ([AWS::ElasticLoadBalancingV2::Listener](#)): un agente de escucha que permite que el balanceador de carga compruebe las solicitudes de conexión y las reenvía a uno o varios grupos de destino.
- `AWSEBV2LoadBalancerListenerRule` ([AWS::ElasticLoadBalancingV2::ListenerRule](#)): define las solicitudes sobre las que un agente de escucha de Elastic Load Balancing actúa y la acción que realiza.
- `AWSEBV2LoadBalancerTargetGroup` ([AWS::ElasticLoadBalancingV2::TargetGroup](#)): grupo de destino Elastic Load Balancing que enruta las solicitudes a uno o varios destinos registrados, como instancias de Amazon EC2.

Los entornos de trabajo tienen recursos para la cola de SQS que almacena en búfer solicitudes entrantes y una tabla de Amazon DynamoDB que las instancias utilizan para elegir un líder.

Entornos de trabajo

- `AWSEBWorkerQueue` ([AWS::SQS::Queue](#)): la cola de Amazon SQS desde la que el daemon obtiene las solicitudes que deben procesarse.
- `AWSEBWorkerDeadLetterQueue` ([AWS::SQS::Queue](#)): la cola de Amazon SQS que almacena los mensajes que no se pueden entregar o que el daemon no pudo procesar correctamente.
- `AWSEBWorkerCronLeaderRegistry` ([AWS::DynamoDB::Table](#)): la tabla de Amazon DynamoDB que es el registro interno usado por el daemon para las tareas periódicas.

Otras claves AWS CloudFormation de plantilla

Ya hemos introducido claves de archivos de configuración AWS CloudFormation como `ResourcesFiles`, y `packages`. Elastic Beanstalk agrega el contenido de los archivos de configuración AWS CloudFormation a la plantilla que admite su entorno, de modo que puede AWS CloudFormation usar otras secciones para realizar tareas avanzadas en los archivos de configuración.

Claves

- [Parámetros](#)
- [Salidas](#)
- [Mapeos](#)

Parámetros

Los parámetros son una alternativa a las [opciones personalizadas](#) de Elastic Beanstalk. Puede utilizarlos para definir los valores que se aplicarán en otras partes de sus archivos de configuración. Al igual que las opciones personalizadas, puede utilizar parámetros para reunir los valores que puede configurar el usuario en un único lugar. A diferencia de las opciones personalizadas, no puede usar la API de Elastic Beanstalk para establecer los valores de los parámetros y el número de parámetros que puede definir en una plantilla está limitado por AWS CloudFormation

Una de las razones por las que puede querer usar parámetros es hacer que los archivos de configuración sirvan como plantillas. AWS CloudFormation Si usa parámetros en lugar de opciones personalizadas, puede usar el archivo de configuración para crear el mismo recurso AWS CloudFormation como su propia pila. Por ejemplo, podría disponer de un archivo de configuración que agregue un sistema de archivos de Amazon EFS a su entorno para pruebas y, a continuación, utilizar el mismo archivo para crear un sistema de archivos independiente con fines de producción, que no esté vinculado al ciclo de vida de dicho entorno.

El siguiente ejemplo muestra el uso de parámetros para reunir los valores configurables por el usuario en la parte superior de un archivo de configuración.

Example [Loadbalancer-accesslogs-existingbucket .config](#) — Parámetros

```
Parameters:
  bucket:
    Type: String
```

```

    Description: "Name of the Amazon S3 bucket in which to store load balancer logs"
    Default: "DOC-EXAMPLE-BUCKET"
  bucketprefix:
    Type: String
    Description: "Optional prefix. Can't start or end with a /, or contain the word
AWSLogs"
    Default: ""

```

Salidas

Puede utilizar un bloque `Outputs` para exportar información sobre los recursos creados a AWS CloudFormation. A continuación, puede usar la `Fn::ImportValue` función para incluir el valor en una AWS CloudFormation plantilla fuera de Elastic Beanstalk.

En el siguiente ejemplo, se crea un tema de Amazon SNS y se exporta su ARN a con AWS CloudFormation ese nombre. `NotificationTopicArn`

Example [sns-topic.config](#)

```

Resources:
  NotificationTopic:
    Type: AWS::SNS::Topic

Outputs:
  NotificationTopicArn:
    Description: Notification topic ARN
    Value: { "Ref" : "NotificationTopic" }
    Export:
      Name: NotificationTopicArn

```

En un archivo de configuración para un entorno diferente o en una AWS CloudFormation plantilla ajena a Elastic Beanstalk, puede `Fn::ImportValue` usar la función para obtener el ARN exportado. Este ejemplo asigna el valor exportado a una propiedad del entorno llamada `TOPIC_ARN`.

Example `env.config`

```

option_settings:
  aws:elasticbeanstalk:application:environment:
    TOPIC_ARN: ``{ "Fn::ImportValue" : "NotificationTopicArn" }``

```

Mapeos

Puede utilizar una asignación para almacenar pares de clave-valor organizados por espacio de nombres. Una asignación puede ayudarle a organizar los valores que utiliza en todas las configuraciones o cambiar el valor de un parámetro que depende de otro valor. Por ejemplo, la siguiente configuración establece el valor de un parámetro de ID de cuenta en función de la región actual.

Example [L .config — Mapeos oadbalancer-accesslogs-newbucket](#)

```

Mappings:
  Region2ELBAccountId:
    us-east-1:
      AccountId: "111122223333"
    us-west-2:
      AccountId: "444455556666"
    us-west-1:
      AccountId: "123456789012"
    eu-west-1:
      AccountId: "777788889999"
    ...
  Principal:
    AWS:
      ? "Fn::FindInMap"
      :
        - Region2ELBAccountId
        -
          Ref: "AWS::Region"
        - AccountId

```

Funciones

Puede utilizar funciones en los archivos de configuración para rellenar valores de propiedades de recursos con información de otros recursos o de otros valores de opciones de configuración de Elastic Beanstalk. Elastic Beanstalk admite funciones de AWS CloudFormation (`Ref`, `Fn::GetAtt`, `Fn::Join`) y una función específica de Elastic Beanstalk, `Fn::GetOptionSetting`.

Funciones

- [Ref](#).
- [Fn::GetAtt](#)

- [Fn::Join](#)
- [Fn::GetOptionSetting](#)

Ref.

Utilice Ref para recuperar la representación predeterminada en forma de cadena de un recurso de AWS. El valor devuelto por Ref depende del tipo de recurso y a veces depende también de otros factores. Por ejemplo, un grupo de seguridad ([AWS::EC2::SecurityGroup](#)) devuelve el nombre o ID del grupo de seguridad, en función de si el grupo de seguridad se encuentra en una [Amazon Virtual Private Cloud](#) (Amazon VPC) predeterminada, una instancia de EC2 Classic o una VPC personalizada.

```
{ "Ref" : "resource name" }
```

Note

Para obtener información detallada sobre cada tipo de recurso, incluidos los valores que devuelve Ref, consulte [Referencia de tipos de recursos de AWS](#) en la Guía del usuario de AWS CloudFormation.

Del ejemplo de [enlace del ciclo de vida de Auto Scaling](#):

```
Resources:
  lifecyclehook:
    Type: AWS::AutoScaling::LifecycleHook
    Properties:
      AutoScalingGroupName: { "Ref" : "AWSEBAutoScalingGroup" }
```

También puede utilizar Ref para recuperar el valor de un parámetro de AWS CloudFormation definido en otro lugar en el mismo archivo o en un archivo de configuración diferente.

Fn::GetAtt

Use Fn::GetAtt para recuperar el valor de un atributo de un recurso de AWS.

```
{ "Fn::GetAtt" : [ "resource name", "attribute name" ] }
```

Del ejemplo de [enlace del ciclo de vida de Auto Scaling](#):

```
Resources:
  lifecyclehook:
    Type: AWS::AutoScaling::LifecycleHook
    Properties:
      RoleARN: { "Fn::GetAtt" : [ "hookrole", "Arn" ] }
```

Consulte [Fn::GetAtt](#) para obtener más información.

Fn::Join

Utilice `Fn::Join` para combinar cadenas con un delimitador. Las cadenas pueden ser de codificación fija o se puede usar el resultado de `Fn::GetAtt` o `Ref`.

```
{ "Fn::Join" : [ "delimiter", [ "string1", "string2" ] ] }
```

Consulte [Fn::Join](#) para obtener más información.

Fn::GetOptionSetting

Utilice `Fn::GetOptionSetting` para recuperar el valor de una [opción de configuración](#) aplicada al entorno.

```
"Fn::GetOptionSetting":
  Namespace: "namespace"
  OptionName: "option name"
  DefaultValue: "default value"
```

Del ejemplo de [almacenamiento de claves privadas](#):

```
Resources:
  AWSEBAutoScalingGroup:
    Metadata:
      AWS::CloudFormation::Authentication:
        S3Auth:
          type: "s3"
          buckets: ["elasticbeanstalk-us-west-2-123456789012"]
          roleName:
            "Fn::GetOptionSetting":
```

```
Namespace: "aws:autoscaling:launchconfiguration"  
OptionName: "IamInstanceProfile"  
DefaultValue: "aws-elasticbeanstalk-ec2-role"
```

Ejemplos de recursos personalizados

A continuación se incluye una lista de archivos de configuración de ejemplo que puede utilizar para personalizar sus entornos de Elastic Beanstalk:

- [DynamoDB, CloudWatch y SNS](#)
- [Elastic Load Balancing y CloudWatch](#)
- [ElastiCache](#)
- [RDS y CloudWatch](#)
- [SQS, SNS y CloudWatch](#)

Los subtemas de esta página proporcionan algunos ejemplos ampliados sobre cómo añadir y configurar recursos personalizados en un entorno de Elastic Beanstalk.

Ejemplos

- [Ejemplo: ElastiCache](#)
- [Ejemplo: SQS, CloudWatch y SNS](#)
- [Ejemplo: DynamoDB, CloudWatch y SNS](#)

Ejemplo: ElastiCache

Los ejemplos a continuación añaden un clúster de Amazon ElastiCache a las plataformas de EC2-Classical y EC2-VPC (ambas predeterminadas y personalizadas de [Amazon Virtual Private Cloud](#) (Amazon VPC)). Para obtener más información sobre estas plataformas y cómo puede determinar cuáles son las que EC2 admite para su región y su cuenta de AWS, consulte <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-supported-platforms.html>. A continuación, consulte la sección de este tema que sea aplicable a su plataforma.

- [Plataformas EC2-Classical](#)
- [EC2-VPC \(predeterminada\)](#)
- [EC2-VPC \(personalizada\)](#)

Plataformas EC2-Classic

En este ejemplo, se agrega un clúster de Amazon ElastiCache a un entorno con instancias lanzadas en la plataforma EC2-Classic. Todas las propiedades que se muestran en este ejemplo son las propiedades mínimas necesarias que deben establecerse en cada tipo de recurso. Puede descargar el ejemplo en [Ejemplo de ElastiCache](#).

Note

En este ejemplo se crean recursos de AWS, los cuales podrían incurrir gastos. Para obtener más información sobre los precios de AWS, consulte <https://aws.amazon.com/pricing/>. Algunos servicios forman parte del nivel de uso gratuito de AWS. Si es un cliente nuevo, puede probar estos servicios de forma gratuita. Para obtener más información, consulte <https://aws.amazon.com/free/>.

Para usar este ejemplo, haga lo siguiente:

1. Cree un directorio `.ebextensions` en el directorio de nivel superior del paquete de código fuente.
2. Cree dos archivos de configuración con la extensión `.config` y colóquelos en el directorio `.ebextensions`. En un archivo de configuración se definen los recursos y en el otro, las opciones.
3. Implemente su aplicación en Elastic Beanstalk.

YAML usa la sangría uniforme. Utilice el mismo nivel de sangría cuando sustituya el contenido en el archivo de configuración de ejemplo y asegúrese de que el editor de texto utiliza espacios para la sangría, no tabuladores.

Cree un archivo de configuración (por ejemplo, `elasticache.config`) que defina los recursos. En este ejemplo, vamos a crear el clúster de ElastiCache especificando el nombre del recurso (`MyElastiCache`), declarando el tipo y configurando las propiedades del clúster. En el ejemplo, se hace referencia al recurso de grupo de seguridad de ElastiCache que se crea y se define en este archivo de configuración. A continuación, crearemos un grupo de seguridad de ElastiCache. Definiremos el nombre de este recurso, declararemos su tipo y agregaremos una descripción del grupo de seguridad. Por último, definiremos las reglas de entrada del grupo de seguridad de ElastiCache para permitir el acceso únicamente al tráfico procedente de las instancias del

grupo de ElastiCache (MyCacheSecurityGroup) y del grupo de seguridad de Elastic Beanstalk (AWSEBSecurityGroup). El nombre del parámetro, AWSEBSecurityGroup, es un nombre de recurso fijo proporcionado por Elastic Beanstalk. Debe agregar AWSEBSecurityGroup a las reglas de entrada del grupo de seguridad de ElastiCache para que la aplicación de Elastic Beanstalk pueda conectarse a las instancias del clúster de ElastiCache).

```
#This sample requires you to create a separate configuration file that defines the
  custom option settings for CacheCluster properties.
```

Resources:

MyElastiCache:

Type: AWS::ElastiCache::CacheCluster

Properties:

CacheNodeType:

Fn::GetOptionSetting:

OptionName : CacheNodeType

DefaultValue: cache.m1.small

NumCacheNodes:

Fn::GetOptionSetting:

OptionName : NumCacheNodes

DefaultValue: 1

Engine:

Fn::GetOptionSetting:

OptionName : Engine

DefaultValue: memcached

CacheSecurityGroupNames:

- Ref: MyCacheSecurityGroup

MyCacheSecurityGroup:

Type: AWS::ElastiCache::SecurityGroup

Properties:

Description: "Lock cache down to webserver access only"

MyCacheSecurityGroupIngress:

Type: AWS::ElastiCache::SecurityGroupIngress

Properties:

CacheSecurityGroupName:

Ref: MyCacheSecurityGroup

EC2SecurityGroupName:

Ref: AWSEBSecurityGroup

Para obtener más información sobre los recursos que se utilizan en este archivo de configuración de ejemplo, consulte las siguientes referencias:

- [AWS::ElastiCache::CacheCluster](#)
- [AWS::ElastiCache::SecurityGroup](#)
- [AWS::ElastiCache::SecurityGroupIngress](#)

Cree un archivo de configuración distinto llamado `options.config` y defina la configuración de opciones personalizadas.

```
option_settings:
  "aws:elasticbeanstalk:customoption":
    CacheNodeType : cache.m1.small
    NumCacheNodes : 1
    Engine : memcached
```

Estas líneas indican a Elastic Beanstalk que obtenga los valores de las propiedades `CacheNodeType`, `NumCacheNodes` y `Engine` a partir de los valores `CacheNodeType`, `NumCacheNodes` y `Engine` de un archivo de configuración (`options.config` en nuestro ejemplo) que contiene una sección `option_settings` con una sección `aws:elasticbeanstalk:customoption` que, a su vez, contiene el par nombre-valor con el valor real que se va a usar. En el ejemplo anterior, esto significa que se utilizará `cache.m1.small`, `1` y `memcached` para los valores. Para obtener más información acerca de `Fn::GetOptionSetting`, consulte [Funciones](#).

EC2-VPC (predeterminada)

En este ejemplo, se agrega un clúster de Amazon ElastiCache a un entorno con instancias lanzadas en la plataforma EC2-VPC. En concreto, la información de esta sección se aplica a los casos en los que las instancias EC2 se lanzan en la VPC predeterminada. Todas las propiedades de este ejemplo son las propiedades mínimas necesarias que deben establecerse en cada tipo de recurso. Para obtener más información acerca de las VPC predeterminadas, consulte [VPC y subredes predeterminadas](#).

Note

En este ejemplo se crean recursos de AWS, los cuales podrían incurrir gastos. Para obtener más información sobre los precios de AWS, consulte <https://aws.amazon.com/pricing/>. Algunos servicios forman parte del nivel de uso gratuito de AWS. Si es un cliente nuevo, puede probar estos servicios de forma gratuita. Para obtener más información, consulte <https://aws.amazon.com/free/>.

Para usar este ejemplo, haga lo siguiente:

1. Cree un directorio `.ebextensions` en el directorio de nivel superior del paquete de código fuente.
2. Cree dos archivos de configuración con la extensión `.config` y colóquelos en el directorio `.ebextensions`. En un archivo de configuración se definen los recursos y en el otro, las opciones.
3. Implemente su aplicación en Elastic Beanstalk.

YAML usa la sangría uniforme. Utilice el mismo nivel de sangría cuando sustituya el contenido en el archivo de configuración de ejemplo y asegúrese de que el editor de texto utiliza espacios para la sangría, no tabuladores.

Ahora, asigne un nombre al archivo de configuración de recursos `elasticache.config`. En este ejemplo, para crear el clúster de ElastiCache, se especifica el nombre del recurso del clúster de ElastiCache (`MyElastiCache`), se declara el tipo y se configuran las propiedades del clúster. En el ejemplo, se hace referencia al ID del recurso del grupo de seguridad que creamos y definimos en este archivo de configuración.

A continuación, vamos a crear un grupo de seguridad de EC2. Vamos a definir el nombre de este recurso, a declarar el tipo, a agregar una descripción y a definir las reglas de entrada del grupo de seguridad para permitir únicamente el acceso del tráfico procedente de las instancias del grupo de seguridad de Elastic Beanstalk (`AWSEBSecurityGroup`). (El nombre del parámetro, `AWSEBSecurityGroup`, es un nombre de recurso fijo proporcionado por Elastic Beanstalk. Debe agregar `AWSEBSecurityGroup` a las reglas de entrada del grupo de seguridad de ElastiCache para que la aplicación de Elastic Beanstalk pueda conectarse a las instancias del clúster de ElastiCache).

Las reglas de entrada del grupo de seguridad de EC2 también definen el protocolo IP y los números de puerto en los que los nodos de caché pueden aceptar conexiones. En Redis, el número de puerto predeterminado es 6379.

```
#This sample requires you to create a separate configuration file that defines the
  custom option settings for CacheCluster properties.
```

```
Resources:
```

```
  MyCacheSecurityGroup:
```

```
    Type: "AWS::EC2::SecurityGroup"
```

```
    Properties:
```

```
      GroupDescription: "Lock cache down to webserver access only"
```

```
SecurityGroupIngress :
  - IpProtocol : "tcp"
    FromPort :
      Fn::GetOptionSetting:
        OptionName : "CachePort"
        DefaultValue: "6379"
    ToPort :
      Fn::GetOptionSetting:
        OptionName : "CachePort"
        DefaultValue: "6379"
    SourceSecurityGroupName:
      Ref: "AWSEBSecurityGroup"
MyElastiCache:
  Type: "AWS::ElastiCache::CacheCluster"
  Properties:
    CacheNodeType:
      Fn::GetOptionSetting:
        OptionName : "CacheNodeType"
        DefaultValue : "cache.t2.micro"
    NumCacheNodes:
      Fn::GetOptionSetting:
        OptionName : "NumCacheNodes"
        DefaultValue : "1"
    Engine:
      Fn::GetOptionSetting:
        OptionName : "Engine"
        DefaultValue : "redis"
    VpcSecurityGroupIds:
      -
        Fn::GetAtt:
          - MyCacheSecurityGroup
          - GroupId

Outputs:
  ElastiCache:
    Description : "ID of ElastiCache Cache Cluster with Redis Engine"
    Value :
      Ref : "MyElastiCache"
```

Para obtener más información sobre los recursos que se utilizan en este archivo de configuración de ejemplo, consulte las siguientes referencias:

- [AWS::ElastiCache::CacheCluster](#)

- [AWS::EC2::SecurityGroup](#)

A continuación, asigne un nombre al archivo de configuración de opciones, `options.config`, y defina los ajustes predeterminados.

```
option_settings:
  "aws:elasticbeanstalk:customoption":
    CacheNodeType : cache.t2.micro
    NumCacheNodes : 1
    Engine : redis
    CachePort : 6379
```

Estas líneas le indican a Elastic Beanstalk que obtenga los valores de las propiedades `CacheNodeType`, `NumCacheNodes`, `Engine` y `CachePort` a partir de los valores `CacheNodeType`, `NumCacheNodes`, `Engine` y `CachePort` del archivo de configuración (`options.config` en nuestro ejemplo). Este archivo contiene una sección `aws:elasticbeanstalk:customoption` (bajo `option_settings`) que, a su vez, contiene pares nombre-valor con los valores reales que se van a usar. En el ejemplo anterior, se utilizaría `cache.t2.micro`, `1`, `redis` y `6379` para los valores. Para obtener más información acerca de `Fn::GetOptionSetting`, consulte [Funciones](#).

EC2-VPC (personalizada)

Si crea una VPC personalizada en la plataforma EC2-VPC y la define como la VPC en la que EC2 lanza las instancias, el proceso para agregar un clúster de Amazon ElastiCache al entorno será diferente que si se utiliza la VPC predeterminada. La principal diferencia es que debe crear un grupo de subredes para el clúster de ElastiCache. Todas las propiedades de este ejemplo son las propiedades mínimas necesarias que deben establecerse en cada tipo de recurso.

Note

En este ejemplo se crean recursos de AWS, los cuales podrían incurrir gastos. Para obtener más información sobre los precios de AWS, consulte <https://aws.amazon.com/pricing/>. Algunos servicios forman parte del nivel de uso gratuito de AWS. Si es un cliente nuevo, puede probar estos servicios de forma gratuita. Para obtener más información, consulte <https://aws.amazon.com/free/>.

Para usar este ejemplo, haga lo siguiente:

1. Cree un directorio `.ebextensions` en el directorio de nivel superior del paquete de código fuente.
2. Cree dos archivos de configuración con la extensión `.config` y colóquelos en el directorio `.ebextensions`. En un archivo de configuración se definen los recursos y en el otro, las opciones.
3. Implemente su aplicación en Elastic Beanstalk.

YAML usa la sangría uniforme. Utilice el mismo nivel de sangría cuando sustituya el contenido en el archivo de configuración de ejemplo y asegúrese de que el editor de texto utiliza espacios para la sangría, no tabuladores.

Ahora, asigne un nombre al archivo de configuración de recursos `elasticache.config`. En este ejemplo, para crear el clúster de ElastiCache, se especifica el nombre del recurso del clúster de ElastiCache (`MyElasticache`), se declara el tipo y se configuran las propiedades del clúster. Las propiedades del ejemplo hacen referencia al nombre del grupo de subredes del clúster de ElastiCache, así como al ID del recurso de grupo de seguridad que creamos y definimos en este archivo de configuración.

A continuación, vamos a crear un grupo de seguridad de EC2. Vamos a definir el nombre de este recurso, a declarar el tipo, a agregar una descripción, a especificar el ID de la VPC y a definir las reglas de entrada del grupo de seguridad para permitir únicamente el acceso del tráfico procedente de las instancias del grupo de seguridad de Elastic Beanstalk (`AWSEBSecurityGroup`). (El nombre del parámetro, `AWSEBSecurityGroup`, es un nombre de recurso fijo proporcionado por Elastic Beanstalk. Debe agregar `AWSEBSecurityGroup` a las reglas de entrada del grupo de seguridad de ElastiCache para que la aplicación de Elastic Beanstalk pueda conectarse a las instancias del clúster de ElastiCache).

Las reglas de entrada del grupo de seguridad de EC2 también definen el protocolo IP y los números de puerto en los que los nodos de caché pueden aceptar conexiones. En Redis, el número de puerto predeterminado es 6379. Por último, en este ejemplo se crea un grupo de subredes para el clúster de ElastiCache. Definiremos el nombre de este recurso, declararemos su tipo y agregaremos una descripción y un ID de la subred en el grupo de subredes.

Note

Le recomendamos que utilice subredes privadas con el clúster de ElastiCache. Para obtener más información sobre las VPC con subredes privadas, consulte https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Scenario2.html.

#This sample requires you to create a separate configuration file that defines the custom option settings for CacheCluster properties.

Resources:**MyElastiCache:**

Type: "AWS::ElastiCache::CacheCluster"

Properties:**CacheNodeType:****Fn::GetOptionSetting:**

OptionName : "CacheNodeType"

DefaultValue : "cache.t2.micro"

NumCacheNodes:**Fn::GetOptionSetting:**

OptionName : "NumCacheNodes"

DefaultValue : "1"

Engine:**Fn::GetOptionSetting:**

OptionName : "Engine"

DefaultValue : "redis"

CacheSubnetGroupName:

Ref: "MyCacheSubnets"

VpcSecurityGroupIds:

- Ref: "MyCacheSecurityGroup"

MyCacheSecurityGroup:

Type: "AWS::EC2::SecurityGroup"

Properties:

GroupDescription: "Lock cache down to webserver access only"

VpcId:**Fn::GetOptionSetting:**

OptionName : "VpcId"

SecurityGroupIngress :

- IpProtocol : "tcp"

FromPort :**Fn::GetOptionSetting:**

OptionName : "CachePort"


```

        DefaultValue: "6379"
    ToPort :
        Fn::GetOptionSetting:
            OptionName : "CachePort"
            DefaultValue: "6379"
    SourceSecurityGroupId:
        Ref: "AWSEBSecurityGroup"
MyCacheSubnets:
    Type: "AWS::ElastiCache::SubnetGroup"
    Properties:
        Description: "Subnets for ElastiCache"
        SubnetIds:
            Fn::GetOptionSetting:
                OptionName : "CacheSubnets"
Outputs:
    ElastiCache:
        Description : "ID of ElastiCache Cache Cluster with Redis Engine"
        Value :
            Ref : "MyElastiCache"

```

Para obtener más información sobre los recursos que se utilizan en este archivo de configuración de ejemplo, consulte las siguientes referencias:

- [AWS::ElastiCache::CacheCluster](#)
- [AWS::EC2::SecurityGroup](#)
- [AWS::ElastiCache::SubnetGroup](#)

A continuación, asigne un nombre al archivo de configuración de opciones, `options.config`, y defina los ajustes predeterminados.

Note

En el siguiente ejemplo, sustituya los valores `CacheSubnets` y `VpcId` por sus propias subredes y VPC.

```

option_settings:
  "aws:elasticbeanstalk:customoption":
    CacheNodeType : cache.t2.micro
    NumCacheNodes : 1

```

```
Engine : redis
CachePort : 6379
CacheSubnets:
  - subnet-1a1a1a1a
  - subnet-2b2b2b2b
  - subnet-3c3c3c3c
VpcId: vpc-4d4d4d4d
```

Estas líneas le indican a `CacheNodeType` que obtenga los valores de las propiedades `NumCacheNodes`, `Engine`, `CachePort`, `CacheSubnets` y `VpcId` a partir de los valores `CacheNodeType`, `NumCacheNodes`, `Engine`, `CachePort`, `CacheSubnets` y `VpcId` del archivo de configuración (`options.config` en nuestro ejemplo). Este archivo contiene una sección `aws:elasticbeanstalk:customoption` (bajo `option_settings`) que, a su vez, contiene pares nombre-valor con valores de ejemplo. En el ejemplo anterior, se utilizaría `cache.t2.micro`, `1`, `redis`, `6379`, `subnet-1a1a1a1a`, `subnet-2b2b2b2b`, `subnet-3c3c3c3c` y `vpc-4d4d4d4d` para los valores. Para obtener más información acerca de `Fn::GetOptionSetting`, consulte [Funciones](#).

Ejemplo: SQS, CloudWatch y SNS

Este ejemplo añade una cola de Amazon SQS y una alarma sobre la longitud de la cola al entorno. Las propiedades que se muestran en este ejemplo son las propiedades mínimas necesarias que debe establecer para cada uno de estos recursos. Puede descargar el ejemplo en [SQS, SNS y CloudWatch](#).

Note

En este ejemplo se crean recursos de AWS, los cuales podrían incurrir gastos. Para obtener más información sobre los precios de AWS, consulte <https://aws.amazon.com/pricing/>. Algunos servicios forman parte del nivel de uso gratuito de AWS. Si es un cliente nuevo, puede probar estos servicios de forma gratuita. Para obtener más información, consulte <https://aws.amazon.com/free/>.

Para usar este ejemplo, haga lo siguiente:

1. Cree un directorio [.ebextensions](#) en el directorio de nivel superior del paquete de código fuente.

2. Cree dos archivos de configuración con la extensión `.config` y colóquelos en el directorio `.ebextensions`. En un archivo de configuración se definen los recursos y en el otro, las opciones.
3. Implemente su aplicación en Elastic Beanstalk.

YAML usa la sangría uniforme. Utilice el mismo nivel de sangría cuando sustituya el contenido en el archivo de configuración de ejemplo y asegúrese de que el editor de texto utiliza espacios para la sangría, no tabuladores.

Cree un archivo de configuración (por ejemplo, `sqs.config`) que defina los recursos. En este ejemplo, crearemos una cola de SQS y definiremos la propiedad `VisibilityTimeout` del recurso `MySQSQueue`. A continuación, crearemos un `Topic` de SNS y especificaremos que se envíe un correo electrónico a `someone@example.com` cuando se active la alarma. Por último, crearemos una alarma de `CloudWatch` si la cola tiene más de 10 mensajes. En la propiedad `Dimensions`, especificamos el nombre de la dimensión y el valor que representa la medición de la dimensión. Utilizamos `Fn::GetAtt` para devolver el valor de `QueueName` de `MySQSQueue`.

```
#This sample requires you to create a separate configuration file to define the custom
options for the SNS topic and SQS queue.
Resources:
  MySQLSQueue:
    Type: AWS::SQS::Queue
    Properties:
      VisibilityTimeout:
        Fn::GetOptionSetting:
          OptionName: VisibilityTimeout
          DefaultValue: 30
  AlarmTopic:
    Type: AWS::SNS::Topic
    Properties:
      Subscription:
        - Endpoint:
            Fn::GetOptionSetting:
              OptionName: AlarmEmail
              DefaultValue: "nobody@amazon.com"
            Protocol: email
  QueueDepthAlarm:
    Type: AWS::CloudWatch::Alarm
    Properties:
      AlarmDescription: "Alarm if queue depth grows beyond 10 messages"
```

```

Namespace: "AWS/SQS"
MetricName: ApproximateNumberOfMessagesVisible
Dimensions:
  - Name: QueueName
    Value : { "Fn::GetAtt" : [ "MySQSQueue", "QueueName"] }
Statistic: Sum
Period: 300
EvaluationPeriods: 1
Threshold: 10
ComparisonOperator: GreaterThanThreshold
AlarmActions:
  - Ref: AlarmTopic
InsufficientDataActions:
  - Ref: AlarmTopic

```

Outputs :

```

QueueURL:
  Description : "URL of newly created SQS Queue"
  Value : { Ref : "MySQSQueue" }
QueueARN :
  Description : "ARN of newly created SQS Queue"
  Value : { "Fn::GetAtt" : [ "MySQSQueue", "Arn"] }
QueueName :
  Description : "Name newly created SQS Queue"
  Value : { "Fn::GetAtt" : [ "MySQSQueue", "QueueName"] }

```

Para obtener más información sobre los recursos que se utilizan en este archivo de configuración de ejemplo, consulte las siguientes referencias:

- [AWS::SQS::Queue](#)
- [AWS::SNS::Topic](#)
- [AWS::CloudWatch::Alarm](#)

Cree un archivo de configuración distinto llamado `options.config` y defina la configuración de opciones personalizadas.

```

option_settings:
  "aws:elasticbeanstalk:customoption":
    VisibilityTimeout : 30
    AlarmEmail : "nobody@example.com"

```

Estas líneas indican a Elastic Beanstalk que obtenga los valores de las propiedades `VisibilityTimeout` y `SubscriptionEndpoint` de los valores `VisibilityTimeout` y `SubscriptionEndpoint` de un archivo de configuración (`options.config` en nuestro ejemplo) que contiene una sección `option_settings` con una sección `aws:elasticbeanstalk:customoption` que contiene el par de nombre-valor con el valor real que se va a usar. En el ejemplo anterior, este valor es 30 y se usaría "nobody@amazon.com" para los valores. Para obtener más información acerca de `Fn::GetOptionSetting`, consulte [the section called "Funciones"](#).

Ejemplo: DynamoDB, CloudWatch y SNS

Este archivo de configuración configura la tabla de DynamoDB como controlador de sesión en una aplicación basada en PHP a través del SDK de AWS para PHP 2. Para utilizar este ejemplo, debe tener un perfil de instancia de IAM, que se agrega a las instancias del entorno y que se utiliza para obtener acceso a la tabla de DynamoDB.

Puede descargar el ejemplo que utilizaremos en este paso en [DynamoDB Session Support Example](#). El ejemplo contiene los siguientes archivos:

- La aplicación de ejemplo, `index.php`
- Un archivo de configuración, `dynamodb.config`, para crear y configurar una tabla de DynamoDB y otros recursos de AWS e instalar software en las instancias de EC2 que alojan la aplicación en un entorno de Elastic Beanstalk
- Un archivo de configuración, `options.config`, que sustituye los valores predeterminados de `dynamodb.config` por otros específicos para esta instalación en particular

index.php

```
<?php

// Include the SDK using the Composer autoloader
require '../vendor/autoload.php';

use Aws\DynamoDb\DynamoDbClient;

// Grab the session table name and region from the configuration file
list($tableName, $region) = file(__DIR__ . '/../sessiontable');
$tableName = rtrim($tableName);
$region = rtrim($region);

// Create a DynamoDB client and register the table as the session handler
```

```

$dynamodb = DynamoDbClient::factory(array('region' => $region));
$handler = $dynamodb->registerSessionHandler(array('table_name' => $tableName,
  'hash_key' => 'username'));

// Grab the instance ID so we can display the EC2 instance that services the request
$instanceId = file_get_contents("http://169.254.169.254/latest/meta-data/instance-id");
?>
<h1>Elastic Beanstalk PHP Sessions Sample</h1>
<p>This sample application shows the integration of the Elastic Beanstalk PHP
container and the session support for DynamoDB from the AWS SDK for PHP 2.
Using DynamoDB session support, the application can be scaled out across
multiple web servers. For more details, see the
<a href="https://aws.amazon.com/php/">PHP Developer Center</a>.</p>

<form id="SimpleForm" name="SimpleForm" method="post" action="index.php">
<?php
echo 'Request serviced from instance ' . $instanceId . '<br/>';
echo '<br/>';

if (isset($_POST['continue'])) {
    session_start();
    $_SESSION['visits'] = $_SESSION['visits'] + 1;
    echo 'Welcome back ' . $_SESSION['username'] . '<br/>';
    echo 'This is visit number ' . $_SESSION['visits'] . '<br/>';
    session_write_close();
    echo '<br/>';
    echo '<input type="Submit" value="Refresh" name="continue" id="continue"/>';
    echo '<input type="Submit" value="Delete Session" name="killsession"
id="killsession"/>';
} elseif (isset($_POST['killsession'])) {
    session_start();
    echo 'Goodbye ' . $_SESSION['username'] . '<br/>';
    session_destroy();
    echo 'Username: <input type="text" name="username" id="username" size="30"/><br/>';
    echo '<br/>';
    echo '<input type="Submit" value="New Session" name="newsession" id="newsession"/>';
} elseif (isset($_POST['newsession'])) {
    session_start();
    $_SESSION['username'] = $_POST['username'];
    $_SESSION['visits'] = 1;
    echo 'Welcome to a new session ' . $_SESSION['username'] . '<br/>';
    session_write_close();
    echo '<br/>';
    echo '<input type="Submit" value="Refresh" name="continue" id="continue"/>';

```

```

    echo '<input type="Submit" value="Delete Session" name="killsession"
id="killsession"/>';
} else {
    echo 'To get started, enter a username.<br/>';
    echo '<br/>';
    echo 'Username: <input type="text" name="username" id="username" size="30"/><br/>';
    echo '<input type="Submit" value="New Session" name="newsession" id="newsession"/>';
}
?>
</form>

```

.ebextensions/dynamodb.config

Resources:

SessionTable:

Type: AWS::DynamoDB::Table

Properties:

KeySchema:

HashKeyElement:

AttributeName:

Fn::GetOptionSetting:

OptionName : SessionHashKeyName

DefaultValue: "username"

AttributeType:

Fn::GetOptionSetting:

OptionName : SessionHashKeyType

DefaultValue: "S"

ProvisionedThroughput:

ReadCapacityUnits:

Fn::GetOptionSetting:

OptionName : SessionReadCapacityUnits

DefaultValue: 1

WriteCapacityUnits:

Fn::GetOptionSetting:

OptionName : SessionWriteCapacityUnits

DefaultValue: 1

SessionWriteCapacityUnitsLimit:

Type: AWS::CloudWatch::Alarm

Properties:

AlarmDescription: { "Fn::Join" : ["", [{ "Ref" : "AWSEBEnvironmentName" }], " write capacity limit on the session table."] }

Namespace: "AWS/DynamoDB"

```
MetricName: ConsumedWriteCapacityUnits
Dimensions:
  - Name: TableName
    Value: { "Ref" : "SessionTable" }
Statistic: Sum
Period: 300
EvaluationPeriods: 12
Threshold:
  Fn::GetOptionSetting:
    OptionName : SessionWriteCapacityUnitsAlarmThreshold
    DefaultValue: 240
ComparisonOperator: GreaterThanThreshold
AlarmActions:
  - Ref: SessionAlarmTopic
InsufficientDataActions:
  - Ref: SessionAlarmTopic

SessionReadCapacityUnitsLimit:
  Type: AWS::CloudWatch::Alarm
  Properties:
    AlarmDescription: { "Fn::Join" : [ "", [ { "Ref" : "AWSEBEnvironmentName" }, " read
capacity limit on the session table." ] ] }
    Namespace: "AWS/DynamoDB"
    MetricName: ConsumedReadCapacityUnits
    Dimensions:
      - Name: TableName
        Value: { "Ref" : "SessionTable" }
    Statistic: Sum
    Period: 300
    EvaluationPeriods: 12
    Threshold:
      Fn::GetOptionSetting:
        OptionName : SessionReadCapacityUnitsAlarmThreshold
        DefaultValue: 240
    ComparisonOperator: GreaterThanThreshold
    AlarmActions:
      - Ref: SessionAlarmTopic
    InsufficientDataActions:
      - Ref: SessionAlarmTopic

SessionThrottledRequestsAlarm:
  Type: AWS::CloudWatch::Alarm
  Properties:
```



```

AlarmDescription: { "Fn::Join" : [ "", [ { "Ref" : "AWSEBEnvironmentName" }, " :
requests are being throttled." ] ] }
Namespace: AWS/DynamoDB
MetricName: ThrottledRequests
Dimensions:
  - Name: TableName
    Value: { "Ref" : "SessionTable" }
Statistic: Sum
Period: 300
EvaluationPeriods: 1
Threshold:
  Fn::GetOptionSetting:
    OptionName: SessionThrottledRequestsThreshold
    DefaultValue: 1
ComparisonOperator: GreaterThanThreshold
AlarmActions:
  - Ref: SessionAlarmTopic
InsufficientDataActions:
  - Ref: SessionAlarmTopic

```

```

SessionAlarmTopic:
  Type: AWS::SNS::Topic
  Properties:
    Subscription:
      - Endpoint:
          Fn::GetOptionSetting:
            OptionName: SessionAlarmEmail
            DefaultValue: "nobody@amazon.com"
          Protocol: email

```

```

files:
  "/var/app/sessiontable":
    mode: "000444"
    content: |
      `{"Ref" : "SessionTable"}`
      `{"Ref" : "AWS::Region"}`

  "/var/app/composer.json":
    mode: "000744"
    content:
      {
        "require": {
          "aws/aws-sdk-php": "*"
        }
      }

```

```

    }

container_commands:
  "1-install-composer":
    command: "cd /var/app; curl -s http://getcomposer.org/installer | php"
  "2-install-dependencies":
    command: "cd /var/app; php composer.phar install"
  "3-cleanup-composer":
    command: "rm -Rf /var/app/composer.*"

```

En el archivo de configuración de ejemplo, primero crearemos la tabla de DynamoDB y configuraremos la clave principal de la tabla y las unidades de capacidad para asignar recursos suficientes como para obtener el desempeño solicitado. A continuación, crearemos alarmas de CloudWatch para `WriteCapacity` y `ReadCapacity`. Crearemos un tema de SNS que envíe un email a "nobody@amazon.com" si se superan los umbrales de alarma.

Después de crear y configurar nuestros recursos de AWS para nuestro entorno, necesitamos personalizar las instancias de EC2. Utilizamos la clave `files` con el fin de pasar los detalles de la tabla de DynamoDB a las instancias EC2 de nuestro entorno, así como de agregar un "requisito" en el archivo `composer.json` para el SDK de AWS para PHP 2. Por último, ejecutaremos comandos de contenedor para instalar Composer, las dependencias requeridas y eliminar después el instalador.

.ebextensions/options.config

```

option_settings:
  "aws:elasticbeanstalk:customoption":
    SessionHashKeyName           : username
    SessionHashKeyType           : S
    SessionReadCapacityUnits     : 1
    SessionReadCapacityUnitsAlarmThreshold : 240
    SessionWriteCapacityUnits    : 1
    SessionWriteCapacityUnitsAlarmThreshold : 240
    SessionThrottledRequestsThreshold : 1
    SessionAlarmEmail            : me@example.com

```

Sustituya el valor de `SessionAlarmEmail` por la dirección de email a la que quiere enviar las notificaciones de alarma. El archivo `options.config` contiene los valores que se utilizan en algunas de las variables definidas en `dynamodb.config`. Por ejemplo, `dynamodb.config` contiene las siguientes líneas:

```
Subscription:
```

```
- Endpoint:
  Fn::GetOptionSetting:
    OptionName: SessionAlarmEmail
    DefaultValue: "nobody@amazon.com"
```

Estas líneas indican a Elastic Beanstalk que obtenga el valor de la propiedad `Endpoint` a partir del valor de `SessionAlarmEmail` en un archivo de configuración (`options.config` en nuestra aplicación de ejemplo) que contiene una sección `option_settings` con una sección `aws:elasticbeanstalk:customoption` que, a su vez, contiene un par nombre-valor con el valor real a utilizar. En el ejemplo anterior, esto significa que a `SessionAlarmEmail` se le asigna el valor `nobody@amazon.com`.

Para obtener más información sobre los recursos de CloudFormation utilizados en este ejemplo, consulte las siguientes referencias:

- [AWS::DynamoDB::Table](#)
- [AWS::CloudWatch::Alarm](#)
- [AWS::SNS::Topic](#)

Uso de configuraciones guardadas de Elastic Beanstalk

Puede guardar la configuración de su entorno como un objeto en Amazon Simple Storage Service (Amazon S3) que pueda aplicarse a otros entornos cuando estos se crean o a un entorno en ejecución. Las configuraciones guardadas son plantillas con formato YAML que definen la [versión de la plataforma](#), la [capa](#), los valores de las [opciones de configuración](#) y las etiquetas de un entorno.

Puede aplicar etiquetas a una configuración guardada cuando la cree y editar etiquetas de configuraciones guardadas existentes. La etiquetas aplicadas a una configuración guardada no están relacionadas con las etiquetas especificadas en una configuración guardada con la clave `Tags` : . Estas últimas se aplican a un entorno cuando se establece la configuración guardada en el entorno. Para obtener más información, consulte [Etiquetado de configuraciones guardadas](#).

Note

Puede crear y aplicar configuraciones guardadas a sus entornos de Elastic Beanstalk mediante varios métodos. Estos métodos incluyen la consola de Elastic Beanstalk, la CLI de EB y la AWS CLI.


Consulte los siguientes temas para ver ejemplos de otros métodos de creación y aplicación de configuraciones guardadas:

- [Ajuste de opciones de configuración antes de crear el entorno](#)
- [Definición de opciones de configuración durante la creación del entorno](#)
- [Ajuste de opciones de configuración después de crear el entorno](#)

Cree una configuración guardada del estado actual de su entorno en la consola de administración de Elastic Beanstalk.

Para guardar la configuración de un entorno

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

 Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. Elija Actions (Acciones) y, a continuación, seleccione Save configuration (Guardar configuración).
4. Utilice el formulario en pantalla para asignar un nombre a la configuración guardada. Si lo desea, proporcione una breve descripción y añada las claves y los valores de etiqueta.
5. Seleccione Save.

Elastic Beanstalk > Environments > GettingStartedApp-env

Save Configuration

Save this environment's current configuration.

Environment:
GettingStartedApp-env

Configuration name:

Description:

Tags

Apply up to 50 tags. You can use tags to group and filter your resources. A tag is a key-value pair. The key must be unique within the resource and is case-sensitive. [Learn more](#)

Key	Value	
<input type="text" value="mytag1"/>	<input type="text" value="value1"/>	<input type="button" value="Remove tag"/>

49 remaining


La configuración guardada incluye todas las opciones que ha aplicado al entorno con la consola o con cualquier otro cliente que utilice la API de Elastic Beanstalk. Más adelante, puede aplicar la configuración guardada a su entorno para restablecerlo a su estado anterior o aplicarla a un nuevo entorno durante la [creación del entorno](#).

Puede descargar una configuración con el comando [the section called “eb config”](#) de la CLI de EB, tal y como se muestra en el siguiente ejemplo. *NAME* es el nombre de su configuración guardada.

```
eb config get NAME
```

Para aplicar una configuración guardada durante la creación del entorno (consola de Elastic Beanstalk)

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Applications (Aplicaciones) y, a continuación, elija el nombre de la aplicación de la lista.

 Note

Si tiene muchas aplicaciones, utilice la barra de búsqueda para filtrar la lista de aplicaciones.

3. En el panel de navegación, busque el nombre de la aplicación y elija Saved configurations (Configuraciones guardadas).
4. Seleccione la configuración guardada que desea aplicar y, a continuación, elija Launch environment (Iniciar entorno).
5. Avance por el asistente para crear su entorno.

Las configuraciones guardadas no incluyen las opciones aplicadas con [archivos de configuración](#) del código fuente de su aplicación. Si la misma opción se aplica tanto en un archivo de configuración como en una configuración guardada, tiene prioridad la opción de la configuración guardada. Del mismo modo, las opciones especificadas en la consola de Elastic Beanstalk invalidan las opciones de las configuraciones guardadas. Para obtener más información, consulte [Prioridad](#).

Las configuraciones guardadas se almacenan en el bucket de S3 de Elastic Beanstalk en una carpeta en la que está incluida el nombre de la aplicación. Por ejemplo, las configuraciones de una aplicación llamada my-app en la región us-west-2 para el número de cuenta 123456789012 se encuentran en `s3://elasticbeanstalk-us-west-2-123456789012/resources/templates/my-app/`.

Puede ver el contenido de una configuración guardada abriéndola en un editor de texto. La siguiente configuración de ejemplo muestra la configuración de un entorno de servidor web lanzado con la consola de administración de Elastic Beanstalk.

```
EnvironmentConfigurationMetadata:
  Description: Saved configuration from a multicontainer Docker environment created
with the Elastic Beanstalk Management Console
  DateCreated: '1520633151000'
  DateModified: '1520633151000'
Platform:
  PlatformArn: arn:aws:elasticbeanstalk:us-east-2::platform/Java 8 running on 64bit
Amazon Linux/2.5.0
OptionSettings:
  aws:elasticbeanstalk:command:
    BatchSize: '30'
    BatchSizeType: Percentage
  aws:elasticbeanstalk:sns:topics:
    Notification Endpoint: me@example.com
  aws:elb:policies:
    ConnectionDrainingEnabled: true
    ConnectionDrainingTimeout: '20'
  aws:elb:loadbalancer:
    CrossZone: true
  aws:elasticbeanstalk:environment:
    ServiceRole: aws-elasticbeanstalk-service-role
  aws:elasticbeanstalk:application:
    Application Healthcheck URL: /
  aws:elasticbeanstalk:healthreporting:system:
    SystemType: enhanced
  aws:autoscaling:launchconfiguration:
    IamInstanceProfile: aws-elasticbeanstalk-ec2-role
    InstanceType: t2.micro
    EC2KeyName: workstation-uswest2
  aws:autoscaling:updatepolicy:rollingupdate:
    RollingUpdateType: Health
    RollingUpdateEnabled: true
EnvironmentTier:
  Type: Standard
  Name: WebServer
AWSConfigurationTemplateVersion: 1.1.0.0
Tags:
  Cost Center: WebApp Dev
```

Puede modificar el contenido de una configuración guardada y guardarlo en la misma ubicación en Amazon S3. Todas las configuraciones guardadas con el formato correcto almacenadas en la ubicación correcta se pueden aplicar a un entorno con la consola de administración de Elastic Beanstalk.

Se admiten las siguientes claves.

- `AWSConfigurationTemplateVersion` (obligatoria): la versión de la plantilla de configuración (1.1.0.0).

```
AWSConfigurationTemplateVersion: 1.1.0.0
```

- `Platform`: el nombre de recurso de Amazon (ARN) de la versión de la plataforma del entorno. Puede especificar la plataforma por ARN o el nombre de una pila de soluciones.

```
Platform:
```

```
PlatformArn: arn:aws:elasticbeanstalk:us-east-2::platform/Java 8 running on 64bit Amazon Linux/2.5.0
```

- `SolutionStack`: el nombre completo de la [pila de soluciones](#) usada para crear el entorno.

```
SolutionStack: 64bit Amazon Linux 2017.03 v2.5.0 running Java 8
```

- `OptionSettings`: los valores de las [opciones de configuración](#) que se van a aplicar al entorno. Por ejemplo, la siguiente entrada establece el tipo de instancia en t2.micro.

```
OptionSettings:
```

```
aws:autoscaling:launchconfiguration:  
InstanceType: t2.micro
```

- `Tags`: etiquetas (hasta un máximo de 47) que se aplican a los recursos creados en el entorno.

```
Tags:
```

```
Cost Center: WebApp Dev
```

- `EnvironmentTier`: el tipo de entorno que se va a crear. Para un entorno de servidor web, puede excluir esta sección (servidor web es el valor predeterminado). Para un entorno de trabajo, use lo siguiente.

```
EnvironmentTier:
```

```
Name: Worker
```


Type: SQS/HTTP

Note

Puede crear y aplicar configuraciones guardadas a sus entornos de Elastic Beanstalk mediante varios métodos. Estos métodos incluyen la consola de Elastic Beanstalk, la CLI de EB y la AWS CLI.

Consulte los siguientes temas para ver ejemplos de otros métodos de creación y aplicación de configuraciones guardadas:

- [Ajuste de opciones de configuración antes de crear el entorno](#)
- [Definición de opciones de configuración durante la creación del entorno](#)
- [Ajuste de opciones de configuración después de crear el entorno](#)

Etiquetado de configuraciones guardadas

Puede aplicar etiquetas a las configuraciones guardadas de AWS Elastic Beanstalk. Las etiquetas son pares de clave-valor asociadas a los recursos de AWS. Para obtener información sobre el etiquetado de recursos de Elastic Beanstalk, los casos de uso, las restricciones de las claves y los valores de las etiquetas y los tipos de recursos admitidos, consulte [Etiquetar recursos de la aplicación Elastic Beanstalk](#).

Puede especificar etiquetas cuando cree una configuración guardada. En una configuración guardada existente, puede añadir o eliminar etiquetas, y actualizar los valores de etiquetas existentes. Puede añadir hasta 50 etiquetas a cada configuración guardada.

Adición de etiquetas durante la creación de configuraciones guardadas

Cuando se utiliza la consola de Elastic Beanstalk para [guardar una configuración](#), puede especificar claves y valores de etiqueta en la página Save Configuration (Guardar configuración).

Si utiliza la CLI de EB para guardar una configuración, use la opción `--tags` con [eb config](#) para añadir etiquetas.

```
~/workspace/my-app$ eb config --tags mytag1=value1,mytag2=value2
```

Con la AWS CLI u otros clientes basados en API, agregue etiquetas con el parámetro `--tags` en el comando [create-configuration-template](#).

```
$ aws elasticbeanstalk create-configuration-template \  
  --tags Key=mytag1,Value=value1 Key=mytag2,Value=value2 \  
  --application-name my-app --template-name my-template --solution-stack-  
name solution-stack
```

Administración de etiquetas de una configuración guardada existente

Puede agregar, actualizar y eliminar etiquetas en una configuración guardada de Elastic Beanstalk existente.

Para administrar las etiquetas de una configuración guardada mediante la consola de Elastic Beanstalk

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Applications (Aplicaciones) y, a continuación, elija el nombre de la aplicación de la lista.

Note

Si tiene muchas aplicaciones, utilice la barra de búsqueda para filtrar la lista de aplicaciones.

3. En el panel de navegación, busque el nombre de la aplicación y elija Saved configurations (Configuraciones guardadas).
4. Seleccione la configuración guardada que desea administrar.
5. Elija Actions (Acciones) y, a continuación, elija Manage Tags (Administrar etiquetas).
6. Utilice el formulario en pantalla para añadir, actualizar o eliminar etiquetas.
7. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

Si utiliza la CLI de EB para actualizar su configuración guardada, utilice [eb tags](#) para añadir, actualizar, eliminar o enumerar etiquetas.

Por ejemplo, el siguiente comando muestra las etiquetas en una configuración guardada.

```
~/workspace/my-app$ eb tags --list --resource "arn:aws:elasticbeanstalk:us-east-2:my-account-id:configurationtemplate/my-app/my-template"
```

El siguiente comando actualiza la etiqueta mytag1 y elimina la etiqueta mytag2.

```
~/workspace/my-app$ eb tags --update mytag1=newvalue --delete mytag2 \
  --resource "arn:aws:elasticbeanstalk:us-east-2:my-account-id:configurationtemplate/my-app/my-template"
```

Para obtener una lista de las opciones y más ejemplos, consulte [eb tags](#).

Con la AWS CLI u otros clientes basados en API, utilice el comando [list-tags-for-resource](#) para enumerar las etiquetas de una configuración guardada.

```
$ aws elasticbeanstalk list-tags-for-resource --resource-arn
"arn:aws:elasticbeanstalk:us-east-2:my-account-id:configurationtemplate/my-app/my-template"
```

Utilice el comando [update-tags-for-resource](#) para añadir, actualizar o eliminar etiquetas en una configuración guardada.

```
$ aws elasticbeanstalk update-tags-for-resource \
  --tags-to-add Key=mytag1,Value=newvalue --tags-to-remove mytag2 \
  --resource-arn "arn:aws:elasticbeanstalk:us-east-2:my-account-id:configurationtemplate/my-app/my-template"
```

Especifique las etiquetas que desea añadir y las que desea actualizar en el parámetro `--tags-to-add` de `update-tags-for-resource`. Se añade una etiqueta no existente y se actualiza el valor de una etiqueta existente.

Note

Para utilizar algunos de los comandos de la CLI de EB y los comandos de la AWS CLI con una configuración guardada de Elastic Beanstalk, necesita el ARN de la configuración guardada. Para construir el ARN, en primer lugar, utilice el siguiente comando para recuperar el nombre de la configuración guardada.

```
$ aws elasticbeanstalk describe-applications --application-names my-app
```

Busque la clave `ConfigurationTemplates` en la salida del comando. Este elemento muestra el nombre de la configuración guardada. Utilice este nombre en el que se especifica *my-template* en los comandos mencionados en esta página.

Manifiesto del entorno (`env.yaml`)

Puede incluir un manifiesto del entorno con formato YAML en la raíz de su paquete de código fuente de la aplicación para configurar el nombre del entorno, la pila de soluciones y los [enlaces del entorno](#) que desee usar al crear el entorno.

Este formato de archivo incluye compatibilidad con los grupos de entornos. Para utilizar grupos, especifique el nombre del entorno en el archivo de manifiesto con un signo + al final. Cuando cree o actualice el entorno, especifique el nombre de grupo con `--group-name` (AWS CLI) o `--env-group-suffix` (CLI de EB). Para obtener más información acerca de los grupos, consulte [Creación y actualización de grupos de entornos de Elastic Beanstalk](#).

El siguiente manifiesto de ejemplo define un entorno de servidor web con un enlace a un componente del entorno de trabajo del que depende. El manifiesto utiliza grupos para permitir la creación de varios entornos con el mismo paquete de código fuente:

`~/myapp/frontend/env.yaml`

```
AWSConfigurationTemplateVersion: 1.1.0.0
SolutionStack: 64bit Amazon Linux 2015.09 v2.0.6 running Multi-container Docker 1.7.1
(Generic)
OptionSettings:
  aws:elasticbeanstalk:command:
    BatchSize: '30'
    BatchSizeType: Percentage
  aws:elasticbeanstalk:sns:topics:
    Notification Endpoint: me@example.com
  aws:elb:policies:
    ConnectionDrainingEnabled: true
    ConnectionDrainingTimeout: '20'
  aws:elb:loadbalancer:
    CrossZone: true
  aws:elasticbeanstalk:environment:
    ServiceRole: aws-elasticbeanstalk-service-role
  aws:elasticbeanstalk:application:
```

```

Application Healthcheck URL: /
aws:elasticbeanstalk:healthreporting:system:
  SystemType: enhanced
aws:autoscaling:launchconfiguration:
  IamInstanceProfile: aws-elasticbeanstalk-ec2-role
  InstanceType: t2.micro
  EC2KeyName: workstation-uswest2
aws:autoscaling:updatepolicy:rollingupdate:
  RollingUpdateType: Health
  RollingUpdateEnabled: true
Tags:
  Cost Center: WebApp Dev
CName: front-A08G28LG+
EnvironmentName: front+
EnvironmentLinks:
  "WORKERQUEUE" : "worker+"

```

Se admiten las siguientes claves.

- **AWSConfigurationTemplateVersion** (obligatoria): la versión de la plantilla de configuración (1.1.0.0).

```
AWSConfigurationTemplateVersion: 1.1.0.0
```

- **Platform**: el nombre de recurso de Amazon (ARN) de la versión de la plataforma del entorno. Puede especificar la plataforma por ARN o el nombre de una pila de soluciones.

```

Platform:
  PlatformArn: arn:aws:elasticbeanstalk:us-east-2::platform/Java 8 running on 64bit
  Amazon Linux/2.5.0

```

- **SolutionStack**: el nombre completo de la [pila de soluciones](#) usada para crear el entorno.

```
SolutionStack: 64bit Amazon Linux 2017.03 v2.5.0 running Java 8
```

- **OptionSettings**: los valores de las [opciones de configuración](#) que se van a aplicar al entorno. Por ejemplo, la siguiente entrada establece el tipo de instancia en t2.micro.

```

OptionSettings:
  aws:autoscaling:launchconfiguration:
    InstanceType: t2.micro

```

- **Tags:** etiquetas (hasta un máximo de 47) que se aplican a los recursos creados en el entorno.

```
Tags:
  Cost Center: WebApp Dev
```

- **EnvironmentTier:** el tipo de entorno que se va a crear. Para un entorno de servidor web, puede excluir esta sección (servidor web es el valor predeterminado). Para un entorno de trabajo, use lo siguiente.

```
EnvironmentTier:
  Name: Worker
  Type: SQS/HTTP
```

- **CName:** el CNAME del entorno. Incluya un carácter + al final del nombre para habilitar grupos.

```
CName: front-A08G28LG+
```

- **EnvironmentName:** el nombre del entorno que se va a crear. Incluya un carácter + al final del nombre para habilitar grupos.

```
EnvironmentName: front+
```

Con los grupos habilitados, debe especificar un nombre de grupo al crear los entornos. Elastic Beanstalk añade el nombre de grupo al nombre del entorno con un guion. Por ejemplo, con el nombre de entorno `front+` y el nombre de grupo `dev`, Elastic Beanstalk creará el entorno con el nombre `front-dev`.

- **EnvironmentLinks:** un mapa de variables de entorno y nombres de dependencias del entorno. El siguiente ejemplo incluye en el entorno `worker+` una dependencia e indica a Elastic Beanstalk que guarde la información del enlace en una variable denominada `WORKERQUEUE`.

```
EnvironmentLinks:
  "WORKERQUEUE" : "worker+"
```

El valor de la variable de enlace varía según el tipo de entorno vinculado. Para un entorno de servidor web, el enlace es el CNAME del entorno. Para un entorno de trabajo, el vínculo es el nombre de la cola del servicio Amazon Simple Queue (Amazon SQS) del entorno.

Las claves CName, EnvironmentName y EnvironmentLinks se pueden usar para crear [grupos de entornos](#) y [enlaces a otros entornos](#). Estas características se admiten actualmente cuando se utiliza la CLI de EB, la AWS CLI o un SDK.

Uso de una imagen de máquina de Amazon (AMI) personalizada

Al crear un AWS Elastic Beanstalk entorno, puede especificar una imagen de máquina de Amazon (AMI) para utilizarla en lugar de la AMI de Elastic Beanstalk estándar incluida en la versión de la plataforma. Una AMI personalizada puede mejorar los tiempos de aprovisionamiento cuando se lanzan las instancias de su entorno si necesita instalar una gran cantidad de software que no está incluido en las AMI estándar.

El uso de [archivos de configuración](#) es ideal para configurar y personalizar su entorno de forma rápida y coherente. La aplicación de configuraciones, sin embargo, puede tardar mucho tiempo durante la creación del entorno y las actualizaciones. Si realiza muchas tareas de configuración del servidor con los archivos de configuración, puede reducir este tiempo creando una AMI personalizada que ya tenga el software y la configuración que necesita.

Una AMI personalizada también le permite realizar cambios en los componentes de bajo nivel, como el kernel de Linux, que son difíciles de implementar o que tardan mucho tiempo en aplicarse mediante archivos de configuración. Para crear una AMI personalizada, lance una AMI de plataforma de Elastic Beanstalk en Amazon EC2 personalice el software y la configuración según sus necesidades y, a continuación, detenga la instancia y guarde una AMI.

Creación de una AMI personalizada

Para identificar la base AMI de Elastic Beanstalk

1. En una ventana del sistema, ejecute un comando como el siguiente. Para obtener más información, consulte la Referencia de [describe-platform-version](#) comandos.AWS CLI

Especifique la AWS región en la que desea utilizar la AMI personalizada y sustituya el ARN y el número de versión de la plataforma por la plataforma Elastic Beanstalk en la que se basa la aplicación.

Example - Mac OS/sistema operativo Linux

```
$ aws elasticbeanstalk describe-platform-version --region us-east-2 \
```

```

--platform-arn "arn:aws:elasticbeanstalk:us-east-2::platform/Tomcat 8.5 with
Java 8 running on 64bit Amazon Linux/3.1.6" \
--query PlatformDescription.CustomAmiList
[
  {
    "VirtualizationType": "pv",
    "ImageId": ""
  },
  {
    "VirtualizationType": "hvm",
    "ImageId": "ami-020ae06fdda6a0f66"
  }
]

```

Example - Sistema operativo Windows

```

C:\> aws elasticbeanstalk describe-platform-version --region us-east-2 --platform-
arn"arn:aws:elasticbeanstalk:us-east-2::platform/IIS 10.0 running on 64bit Windows
Server 2019/2.6.4" --query PlatformDescription.CustomAmiList
[
  {
    "VirtualizationType": "pv",
    "ImageId": ""
  },
  {
    "VirtualizationType": "hvm",
    "ImageId": "ami-020ae06fdda6a0f66"
  }
]

```

2. Tome nota del valor ImageId que es similar a `ami-020ae06fdda6a0f66` en el resultado.

El valor es la AMI de Elastic Beanstalk original para la versión de plataforma, la arquitectura AWS de la instancia EC2 y la región que sean relevantes para su aplicación. Si necesita crear AMI para varias plataformas, arquitecturas o AWS regiones, repita este proceso para identificar la AMI base correcta para cada combinación.

Notas

- No cree una AMI a partir de una instancia que se ha lanzado en un entorno de Elastic Beanstalk. Elastic Beanstalk realiza cambios en las instancias durante el

aprovisionamiento que pueden provocar problemas en la AMI guardada. Al guardar una imagen de una instancia en un entorno de Elastic Beanstalk, también se incluye como parte fija de la imagen la versión de la aplicación que se implementó en la instancia.

- Le recomendamos que utilice siempre la versión más reciente de la plataforma. Cuando actualice a una nueva versión de la plataforma, le recomendamos que también modifique su AMI personalizada de tal forma que esté basada en la AMI de la nueva versión de la plataforma. De este modo se minimizan los errores de implementación debidos a incompatibilidades de versiones de paquetes o bibliotecas.

Para Linux, también es posible crear una AMI personalizada a partir de una AMI de comunidad que Elastic Beanstalk no publicó. Puede utilizar la última AMI de [Amazon Linux](#) como punto de partida. Cuando se lanza un entorno con una AMI de Linux que no administra Elastic Beanstalk, Elastic Beanstalk intenta instalar el software de la plataforma (idioma, plataforma, servidor proxy, etc.) y los componentes adicionales para admitir características como los [informes de estado mejorados](#).

Note

Las AMI personalizadas basadas en Windows Server requieren la AMI de Elastic Beanstalk de stock devuelta `describe-platform-version`, como se muestra anteriormente en el paso 1.

Aunque Elastic Beanstalk puede utilizar una AMI no administrada por Elastic Beanstalk, el aumento del tiempo de aprovisionamiento resultante de la instalación de los componentes que faltan por parte de Elastic Beanstalk puede reducir o eliminar los beneficios de la creación de una AMI personalizada. Otras distribuciones de Linux podrían funcionar realizando algunas soluciones de problemas, pero no están admitidas oficialmente. Si la aplicación necesita una distribución de Linux específica, una alternativa puede ser crear una imagen de Docker y ejecutarla en la [plataforma de Docker](#) en Elastic Beanstalk o en la [plataforma Multicontainer Docker](#).

Para crear una AMI personalizada

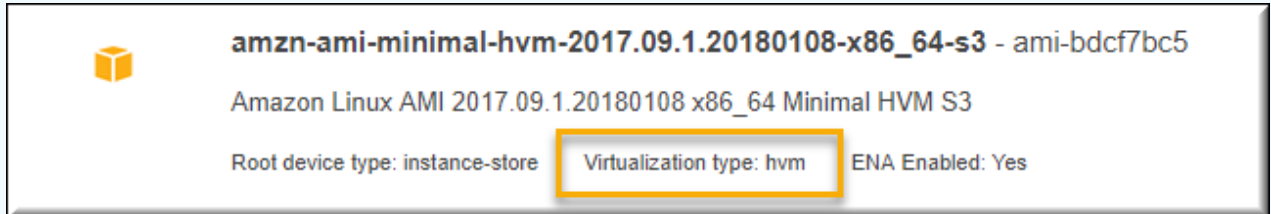
1. Abra la consola de Amazon EC2 en <https://console.aws.amazon.com/ec2/>.
2. Elija Iniciar instancia.
3. Elija AMI de comunidad.

- Si identificó una AMI base de Elastic Beanstalk (con `describe-platform-version`) o una AMI de Amazon Linux, escriba su ID de AMI en el cuadro de búsqueda. Luego pulse Intro.

También puede buscar en la lista otra AMI de la comunidad que se adapte a sus necesidades.

Note

Le recomendamos que elija una AMI que utilice virtualización HVM. Estas AMI muestran `Virtualization type: hvm` en su descripción.



Para obtener más información sobre los tipos de virtualización de instancias, consulte [Tipos de virtualización de AMI de Linux](#) en la Guía del usuario de Amazon EC2 o [Tipos de virtualización de AMI](#) de Windows en la Guía del usuario de Amazon EC2.

- Elija Select para seleccionar la AMI.
- Seleccione un tipo de instancia y, a continuación, elija Next: Configure Instance Details.
- (Para plataformas Linux) Expanda la sección Detalles avanzados y pegue el texto siguiente en el campo Datos de usuario.

```
#cloud-config
repo_releasever: repository version number
repo_upgrade: none
```

El número de versión del repositorio es la versión de año y mes en el nombre de la AMI. Por ejemplo, las AMI basadas en la versión de marzo de 2015 de Amazon Linux tienen el número de versión de repositorio `2015.03`. Para una imagen de Elastic Beanstalk, coincide con la fecha mostrada en el nombre de la pila de soluciones para la [versión de su plataforma](#) basada en la AMI de Amazon Linux (anterior a Amazon Linux 2).

Note

La `repo_releasever` configuración configura la lock-on-launch función de una AMI de Amazon Linux. Esto hace que la AMI utilice una versión de repositorio fija y específica cuando se lanza. Esta función no es compatible con Amazon Linux 2: no la especifique


si su entorno utiliza una rama actual de plataforma Amazon Linux 2. La configuración es obligatoria si utiliza una AMI personalizada con Elastic Beanstalk solo en las ramas de la plataforma AMI de Amazon Linux (antes de Amazon Linux 2).

La configuración de `repo_upgrade` desactiva la instalación automática de actualizaciones de seguridad. Es necesario usar una AMI personalizada con Elastic Beanstalk.

8. Avance por el asistente hasta [lanzar la instancia EC2](#). Cuando se le solicite, seleccione un par de claves a las que tenga acceso para poder conectarse a la instancia para los pasos siguientes.
9. [Conéctese a la instancia](#) con SSH o RDP.
10. Realice las personalizaciones que desee.
11. (Plataformas Windows) Ejecute el servicio de EC2Config Sysprep. Para obtener más información acerca de EC2Config, consulte [Configuración de una instancia de Windows con el servicio EC2Config](#). Asegúrese de que Sysprep está configurado para generar una contraseña aleatoria que pueda recuperarse en la AWS Management Console.
12. En la consola de Amazon EC2, detenga la instancia EC2. A continuación, en el menú Instance Actions (Acciones de instancia), elija Create Image (EBS AMI) (Crear imagen [AMI de EBS]).
13. Para evitar incurrir en AWS cargos adicionales, [cancele la instancia EC2](#).

Para utilizar su AMI personalizada en un entorno de Elastic Beanstalk

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

 Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Capacity (Capacidad), elija Edit (Editar).
5. En AMI ID (ID de AMI), especifique su ID de AMI personalizado.
6. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

Cuando crea un nuevo entorno con la AMI personalizada, debe utilizar la misma versión de la plataforma que utilizó como punto de partida para crear la AMI. Si posteriormente aplica una [actualización de la plataforma](#) a un entorno utilizando una AMI personalizada, Elastic Beanstalk intenta aplicar la biblioteca y las actualizaciones de la configuración durante el proceso de arranque.

Limpieza de una AMI personalizada

Cuando haya terminado una AMI personalizada y ya no la necesite para lanzar entornos de Elastic Beanstalk, considere la posibilidad de limpiarla para minimizar el costo de almacenamiento. La limpieza de una AMI personalizada conlleva anular el registro de Amazon EC2 y eliminar los demás recursos asociados. Para obtener detalles, consulte [Anulación del registro de una AMI de Linux](#) o [Anulación del registro de una AMI de Windows](#).

Conservar el acceso a una Imagen de máquina de Amazon (AMI) de una plataforma retirada

Elastic Beanstalk establece el estado de una ramificación de la plataforma en retirada cuando el sistema operativo o el componente principal utilizado por la ramificación llega al final de su vida útil. La AMI base de Elastic Beanstalk para la ramificación de la plataforma también se puede convertir en privada para evitar el uso de esta AMI desactualizada. Los entornos que utilizan AMI que se han convertido en privadas no podrán seguir lanzando instancias.

Si no puede migrar la aplicación a un entorno compatible antes de que se retire, es posible que su entorno se encuentre en esta situación. Puede surgir la necesidad de actualizar un entorno para una ramificación de la plataforma Beanstalk, en la que la AMI base de Elastic Beanstalk se haya convertido en privada. Hay disponible un enfoque alternativo. Puede actualizar un entorno existente en función de una copia de la AMI base de Elastic Beanstalk utilizada en su entorno.

Este apartado ofrece algunos pasos y un script independiente para actualizar un entorno existente en función de una copia de la AMI base de Elastic Beanstalk utilizada en su entorno. Una vez que pueda migrar la aplicación a una plataforma compatible, podrá seguir utilizando los procedimientos estándar para mantener la aplicación y los entornos compatibles.

Procedimiento manual

Para actualizar un entorno basado en una AMI (copia de la AMI básica de Elastic Beanstalk)

1. Determine qué AMI utiliza tu entorno. Este comando devuelve la AMI utilizada por el entorno de Elastic Beanstalk que proporcione en los parámetros. El valor devuelto se utiliza como el valor de `source-ami-id` en el siguiente paso.

En una ventana del sistema, ejecute un comando como el siguiente. Para obtener más información, consulte [describe-configuration-settings](#) en la Referencia de comandos de la AWS CLI.

Especifique la región de AWS que almacena la AMI de origen que desea copiar. Sustituya el nombre de la aplicación y el nombre del entorno por aquellos basados en la AMI de origen. Ingrese el texto del parámetro de consulta como se muestra.

Example

```
>aws elasticbeanstalk describe-configuration-settings \  
  --application-name my-application \  
  --environment-name my-environment \  
  --region us-east-2 \  
  --query "ConfigurationSettings[0].OptionSettings[?OptionName=='ImageId'] |  
  [0].Value"
```

2. Copie la AMI en su cuenta. Este comando devuelve la nueva AMI que resulta al copiar el parámetro `source-ami-id` que se devolvió en el paso anterior.

Note

Asegúrese de anotar el nuevo identificador de AMI que genera este comando. Deberá ingresarlo en el siguiente paso y reemplazar el valor de `copied-ami-id` en el comando de ejemplo.

En una ventana del sistema, ejecute un comando como el siguiente. Para obtener más información, consulte [copy-image](#) en la Referencia de comandos de la AWS CLI.

Especifique la región de AWS de la AMI de origen que desea copiar (`--source-region`) y la región en la que quiere usar la nueva AMI personalizada (`--region`). Sustituya `source-ami-id` por la AMI de la imagen que está copiando. El comando devolvió el parámetro `source-ami-id` en el paso anterior. Sustituya `new-ami-name` por un nombre que describa la nueva AMI en la región de destino. El script que sigue a este procedimiento genera el nuevo nombre de la AMI al agregar la cadena "Copy of" al principio del nombre de `source-ami-id`.

```
>aws ec2 copy-image \  
  --region us-east-2 \  
  --source-image-id source-ami-id \  
  --source-region us-east-2 \  
  --name new-ami-name
```

3. Actualice un entorno para utilizar la AMI copiada. El comando devuelve el estado del entorno tras su ejecución.

En una ventana del sistema, ejecute un comando como el siguiente. Para obtener más información, consulte [update-environment](#) en la Referencia de comandos de la AWS CLI.

Especifique la región de AWS del entorno y la aplicación que necesita actualizar. Sustituya el nombre de la aplicación y el nombre del entorno por los que necesite asociar al parámetro `copied-ami-id` del paso anterior. Para el parámetro `--option-setttings`, sustituya `copied-ami-id` por el identificador de AMI que indicó en el resultado del comando anterior.

```
>aws elasticbeanstalk update-environment \  
  --application-name my-application \  
  --environment-name my-environment \  
  --region us-east-2 \  
  --option-settings  
  "Namespace=aws:autoscaling:launchconfiguration,OptionName=ImageId,Value=copied-ami-id"
```

Note

Para minimizar los costos de almacenamiento, considere limpiar la AMI personalizada cuando ya no la necesite para lanzar entornos de Elastic Beanstalk. Para obtener más información, consulte [Limpieza de una AMI personalizada](#).

Script independiente

El siguiente script proporciona los mismos resultados que los pasos manuales anteriores. Seleccione el siguiente enlace para descargar el script: [copy_ami_and_update_env.zip](#).

Origen del script: `copy_ami_and_update_env.sh`

```
#!/bin/bash

set -ue

USAGE="This script is used to copy an AMI used by your Elastic Beanstalk environment
into your account to use in your environment.\n\n"
USAGE+="Usage:\n\n"
USAGE+="./$(basename $0) [OPTIONS]\n"
USAGE+="OPTIONS:\n"
USAGE+="\t--application-name <application-name>\tThe name of your Elastic Beanstalk
application.\n"
USAGE+="\t--environment-name <environment-name>\tThe name of your Elastic Beanstalk
environment.\n"
USAGE+="\t--region <region> \t\t\tThe AWS region your Elastic Beanstalk environment is
deployed to.\n"
USAGE+="\n\n"
USAGE+="Script Usage Example(s):\n"
USAGE+="./$(basename $0) --application-name my-application --environment-name my-
environment --region us-east-1\n"

if [ $# -eq 0 ]; then
    echo -e $USAGE
    exit
fi

while [[ $# -gt 0 ]]; do
    case $1 in
        --application-name)    APPLICATION_NAME="$2"; shift ;;
        --environment-name)    ENVIRONMENT_NAME="$2"; shift ;;
        --region)              REGION="$2"; shift ;;
        *)                      echo "Unknown option $1" ; echo -e $USAGE ; exit ;;
    esac
    shift
done

aws_cli_version="$(aws --version)"
```

```
if [ $? -ne 0 ]; then
    echo "aws CLI not found. Please install it: https://docs.aws.amazon.com/cli/latest/
    userguide/getting-started-install.html. Exiting."
    exit 1
fi
echo "Using aws CLI version: ${aws_cli_version}"

account=$(aws sts get-caller-identity --query "Account" --output text)
echo "Using account ${account}"

environment_ami_id=$(aws elasticbeanstalk describe-configuration-settings \
    --application-name "$APPLICATION_NAME" \
    --environment-name "$ENVIRONMENT_NAME" \
    --region "$REGION" \
    --query "ConfigurationSettings[0].OptionSettings[?OptionName=='ImageId'] | [0].Value" \
    --output text)
echo "Image associated with environment ${ENVIRONMENT_NAME} is ${environment_ami_id}"

owned_image=$(aws ec2 describe-images \
    --owners self \
    --image-ids "$environment_ami_id" \
    --region "$REGION" \
    --query "Images[0]" \
    --output text)
if [ "$owned_image" != "None" ]; then
    echo "${environment_ami_id} is already owned by account ${account}. Exiting."
    exit
fi

source_image_name=$(aws ec2 describe-images \
    --image-ids "$environment_ami_id" \
    --region "$REGION" \
    --query "Images[0].Name" \
    --output text)
if [ "$source_image_name" = "None" ]; then
    echo "Cannot find ${environment_ami_id}. Please contact AWS support if you need
    additional help: https://aws.amazon.com/support."
    exit 1
fi

copied_image_name="Copy of ${source_image_name}"
copied_ami_id=$(aws ec2 describe-images \
    --owners self \
```



```
--filters Name=name,Values="{copied_image_name}" \  
--region "$REGION" \  
--query "Images[0].ImageId" \  
--output text)  
if [ "$copied_ami_id" != "None" ]; then  
    echo "Detected that ${environment_ami_id} has already been copied by account  
    ${account}. Skipping image copy."  
else  
    echo "Copying ${environment_ami_id} to account ${account} with name  
    ${copied_image_name}"  
    copied_ami_id=$(aws ec2 copy-image \  
        --source-image-id "$environment_ami_id" \  
        --source-region "$REGION" \  
        --name "$copied_image_name" \  
        --region "$REGION" \  
        --query "ImageId" \  
        --output text)  
    echo "New AMI ID is ${copied_ami_id}"  
  
    echo "Waiting for ${copied_ami_id} to become available"  
    aws ec2 wait image-available \  
        --image-ids "$copied_ami_id" \  
        --region "$REGION"  
    echo "${copied_ami_id} is now available"  
fi  
  
echo "Updating environment ${ENVIRONMENT_NAME} to use ${copied_ami_id}"  
environment_status=$(aws elasticbeanstalk update-environment \  
    --application-name "$APPLICATION_NAME" \  
    --environment-name "$ENVIRONMENT_NAME" \  
    --option-settings  
    "Namespace=aws:autoscaling:launchconfiguration,OptionName=ImageId,Value=  
    ${copied_ami_id}" \  
    --region "$REGION" \  
    --query "Status" \  
    --output text)  
echo "Environment ${ENVIRONMENT_NAME} is now ${environment_status}"  
  
echo "Waiting for environment ${ENVIRONMENT_NAME} update to complete"  
aws elasticbeanstalk wait environment-updated \  
    --application-name "$APPLICATION_NAME" \  
    --environment-names "$ENVIRONMENT_NAME" \  
    --region "$REGION"
```

```
echo "Environment ${ENVIRONMENT_NAME} update complete"
```

Note

Debe tener la AWS CLI instalada para ejecutar el script. Para obtener instrucciones de instalación, consulte [Instalación o actualización de la versión más reciente de AWS CLI](#) en la Guía del usuario de AWS Command Line Interface.

Tras instalar la AWS CLI, también debe configurarla para que utilice la cuenta de AWS propietaria del entorno. Para obtener más información, consulte [Configuración de AWS CLI](#) en la Guía del usuario de AWS Command Line Interface. La cuenta también debe tener permisos para crear una AMI y actualizar el entorno de Elastic Beanstalk.

Estos pasos describen el proceso que sigue el script.

1. Imprima la cuenta en uso.
2. Determine qué AMI utiliza el entorno (AMI de origen).
3. Compruebe si la AMI de origen ya es propiedad de la cuenta. En caso afirmativo, salga.
4. Determine el nombre de la AMI de origen para que pueda usarse en el nombre de la nueva AMI. Esto también sirve para confirmar el acceso a la AMI de origen.
5. Compruebe si la AMI de origen ya se ha copiado en la cuenta. Esto se hace buscando las AMI con el nombre de la AMI copiada que es propiedad de la cuenta. Si el nombre de la AMI se ha cambiado entre las ejecuciones del script, volverá a copiar la imagen.
6. Si la AMI de origen aún no se ha copiado, cópiela en la cuenta y espere a que la nueva AMI esté disponible.
7. Actualice la configuración del entorno para utilizar la nueva AMI.
8. Espere a que se complete la actualización del entorno.

Tras extraer el script del archivo [copy_ami_and_update_env.zip](#), ejecútelo mediante la ejecución del siguiente ejemplo. Sustituya el nombre de la aplicación y el nombre del entorno del ejemplo con sus propios valores.

```
>sh copy_ami_and_update_env.sh \  
  --application-name my-application \  
  --environment-name my-environment \  
  --region us-east-1
```

Note

Para minimizar los costos de almacenamiento, considere limpiar la AMI AMI personalizada cuando ya no la necesite para lanzar entornos de Elastic Beanstalk. Para obtener más información, consulte [Limpieza de una AMI personalizada](#).

Distribución de archivos estáticos

Para mejorar el desempeño, puede configurar el servidor proxy para servir archivos estáticos de servidor (por ejemplo, HTML o imágenes) desde un conjunto de directorios incluidos en su aplicación web. Cuando el servidor proxy recibe una solicitud de un archivo en la ruta especificada, proporciona directamente el archivo en lugar de direccionar la solicitud a la aplicación.

Elastic Beanstalk admite la configuración del proxy para que sirva archivos estáticos en la mayoría de las sucursales de plataforma basadas en Amazon Linux 2. La única excepción es Docker.

Note

En las plataformas Python y Ruby, Elastic Beanstalk configura algunas carpetas de archivos estáticos de forma predeterminada. Para obtener más detalles, consulte las secciones de configuración de archivos estáticos de [Python](#) y [Ruby](#). Como se explica en esta página, puede configurar carpetas adicionales.

Configurar archivos estáticos mediante la consola


Para configurar el servidor proxy de forma que proporcione archivos estáticos

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Actualizaciones, supervisión y registro, seleccione Editar.
5. Vaya a la sección Software de la plataforma y busque el grupo Archivos estáticos.
 - a. Para agregar una asignación de archivos estáticos, seleccione Agregar archivos estáticos. En la fila adicional, ingrese una ruta para el uso de los archivos estáticos y el directorio que contiene los archivos estáticos que se van a usar.
 - En el campo Ruta, comience el nombre de la ruta con una barra (/) (por ejemplo, “/imágenes”).
 - En el campo Directorio, especifique un nombre de directorio ubicado en la raíz del código fuente de la aplicación. No lo empiece con una barra (por ejemplo, “estáticos/archivos-de-imagen”).
 - b. Para eliminar una asignación, seleccione Eliminar.
6. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

 Note

Si no ve la sección Static Files (Archivos estáticos), debe añadir al menos un mapeo mediante un [archivo de configuración](#). Para obtener más información, consulte [the section called “Configurar archivos estáticos mediante opciones de configuración”](#) en esta página.

Configurar archivos estáticos mediante opciones de configuración

Puede utilizar un [archivo de configuración](#) para configurar rutas estáticas de archivos y ubicaciones de directorio mediante opciones de configuración. Puede agregar un archivo de configuración al paquete de origen de la aplicación e implementarlo durante la creación del entorno o una implementación posterior.

Si su entorno utiliza una rama de plataforma basada en Amazon Linux 2, use el espacio de nombres de [aws:elasticbeanstalk:environment:proxy:staticfiles](#).

El siguiente archivo de configuración de ejemplo indica al servidor proxy que sirva los archivos de la carpeta `statichtml` en la ruta `/html` y los archivos de la carpeta `staticimages` en la ruta `/images`.

Example `.ebextensions/static-files.config`

```
option_settings:
  aws:elasticbeanstalk:environment:proxy:staticfiles:
    /html: statichtml
    /images: staticimages
```

Si su entorno de Elastic Beanstalk utiliza una versión de la plataforma AMI de Amazon Linux (anterior a Amazon Linux 2), lea la siguiente información adicional:

Espacios de nombres específicos de la plataforma AMI de Amazon Linux

En las ramas de la plataforma de la AMI de Amazon Linux, los espacios de nombres de configuración de archivos estáticos varían según la plataforma. Para obtener más información, consulte una de las páginas siguientes:

- [Espacios de nombres de configuración de Go](#)
- [Espacios de nombres de configuración de Java SE](#)
- [Espacios de nombres de configuración de Tomcat](#)
- [Espacio de nombres de configuración de Node.js](#)
- [Espacios de nombres de configuración de Python](#)

Configuración de HTTPS para su entorno de Elastic Beanstalk

Si ha adquirido y configurado un [nombre de dominio personalizado](#) para su entorno de Elastic Beanstalk, puede usar HTTPS para permitir a los usuarios conectarse a su sitio web de forma segura. Si no posee un nombre de dominio, puede utilizar igualmente HTTPS con un certificado autofirmado para las tareas de desarrollo y las pruebas. HTTPS es imprescindible para cualquier aplicación que transmita datos de los usuarios o información de inicio de sesión.

La forma más sencilla de utilizar HTTPS con un entorno de Elastic Beanstalk consiste en [asignar un certificado de servidor al balanceador de carga de su entorno](#). Cuando configure su balanceador de carga para que termine las conexiones HTTPS, la conexión entre el cliente y el balanceador de carga

estará protegida. Las conexiones de backend entre el balanceador de carga y las instancias EC2 utilizan HTTP, por lo que las instancias no requieren configuración adicional.

Note

Con [AWS Certificate Manager \(ACM\)](#), puede crear de forma gratuita un certificado de confianza para los nombres de dominio. Los certificados de ACM se pueden utilizar únicamente con los balanceadores de carga de AWS y las distribuciones de Amazon CloudFront, y ACM solo está [disponible en determinadas regiones de AWS](#).

Para utilizar un certificado de ACM con Elastic Beanstalk, consulte [Configuración del balanceador de carga del entorno de Elastic Beanstalk para terminar conexiones HTTPS](#).

Si ejecuta su aplicación en un entorno de una sola instancia o necesita proteger toda la trayectoria de conexión con las instancias EC2 detrás del balanceador de carga, puede [configurar el servidor proxy que se ejecuta en la instancia para que termine las conexiones HTTPS](#). La configuración de las instancias para terminar conexiones HTTPS requiere el uso de [archivos de configuración](#) para modificar el software que se ejecuta en las instancias y para modificar los grupos de seguridad que permiten las conexiones seguras.

Para las conexiones HTTPS de punto a punto en un entorno con balanceo de carga, puede [combinar la terminación de las instancias y del balanceador de carga](#) para cifrar ambas conexiones. De forma predeterminada, si configura el balanceador de carga para que reenvíe el tráfico mediante HTTPS, este confiará en cualquier certificado que las instancias del backend le presenten. Para disfrutar de la máxima seguridad, puede asociar políticas al balanceador de carga que le impidan conectarse a instancias que no presenten un certificado público en el que confíe.

Note

También puede configurar el balanceador de carga para que [transmita el tráfico HTTPS sin descifrarlo](#). El inconveniente de este método es que el balanceador de carga no puede ver las solicitudes y, por lo tanto, no puede optimizar el enrutamiento ni registrar métricas de respuesta.

Si ACM no está disponible en su región, puede comprar un certificado de confianza a un tercero. Se puede usar un certificado de terceros para descifrar el tráfico HTTPS en el balanceador de carga, en las instancias del backend o en ambos.

Para el desarrollo y las pruebas, puede [crear y firmar un certificado](#) usted mismo con herramientas de código abierto. Los certificados autofirmados son gratuitos y fáciles de crear, pero no se pueden usar para el descifrado front-end en sitios públicos. Si intenta usar un certificado autofirmado para una conexión HTTPS con un cliente, el navegador del usuario muestra un mensaje de error que indica que el sitio web no es seguro. No obstante, puede utilizar un certificado autofirmado para proteger las conexiones del backend sin problema.

ACM es la herramienta preferida para aprovisionar, administrar e implementar los certificados de servidor mediante programación o la AWS CLI. Si ACM no está [disponible en su región de AWS](#), puede [cargar un certificado de terceros o autofirmado y la clave privada](#) en AWS Identity and Access Management (IAM) mediante la AWS CLI. Los certificados almacenados en IAM se pueden usar con balanceadores de carga y distribuciones de CloudFront.

Note

La aplicación de muestra [Does it have Snakes?](#) de GitHub incluye archivos de configuración e instrucciones para cada método de configuración de HTTPS con una aplicación web Tomcat. Consulte el [archivo readme](#) y las [instrucciones de HTTPS](#) para obtener información detallada.

Temas

- [Crear y firmar un certificado X509](#)
- [Carga de un certificado en IAM](#)
- [Configuración del balanceador de carga del entorno de Elastic Beanstalk para terminar conexiones HTTPS](#)
- [Configurar su aplicación para terminar las conexiones HTTPS en la instancia](#)
- [Configuración del cifrado integral en un entorno de Elastic Beanstalk con balanceo de carga](#)
- [Configuración del balanceador de carga del entorno para el paso a través de TCP](#)
- [Almacenamiento seguro de claves privadas en Amazon S3](#)
- [Configuración del redireccionamiento de HTTP a HTTPS](#)

Crear y firmar un certificado X509

Puede crear un certificado X509 para su aplicación con OpenSSL. OpenSSL es una biblioteca de código abierto estándar que admite un amplio conjunto de funciones criptográficas, incluida la

creación y firma de certificados x509. Para obtener más información acerca de OpenSSL, visite www.openssl.org.

Note

Solo necesita crear un certificado localmente si desea [utilizar HTTPS en un entorno de una sola instancia](#) o [volver a cifrar en el backend](#) con un certificado autofirmado. Si dispone de un nombre de dominio, puede crear un certificado en AWS y utilizarlo de forma gratuita con un entorno de carga balanceada mediante AWS Certificate Manager (ACM). Para obtener instrucciones, consulte [Solicitar un certificado](#) en la Guía del usuario de AWS Certificate Manager.

Ejecute `openssl version` en la línea de comandos para saber si ya tiene instalado OpenSSL. Si no lo tiene, puede crear e instalar el código fuente mediante las instrucciones del [repositorio de GitHub público](#) o utilizar su administrador de paquetes favorito. OpenSSL se instala también en imágenes Linux de Elastic Beanstalk, por lo que es una buena alternativa para conectarse a una instancia EC2 en un entorno en ejecución a través del comando `eb ssh` de la [CLI de EB](#):

```
~/eb$ eb ssh
[ec2-user@ip-255-55-55-255 ~]$ openssl version
OpenSSL 1.0.1k-fips 8 Jan 2015
```

Necesita crear una clave privada RSA para crear la solicitud de firma del certificado (CSR). Para crear su clave privada, utilice el comando `openssl genrsa`:

```
[ec2-user@ip-255-55-55-255 ~]$ openssl genrsa 2048 > privatekey.pem
Generating RSA private key, 2048 bit long modulus
.....
+++
.....+++
e is 65537 (0x10001)
```

privatekey.pem

El nombre del archivo en el que desea guardar la clave privada. Normalmente, el comando `openssl genrsa` muestra el contenido de la clave privada en la pantalla, pero este comando envía el resultado a un archivo. Elija un nombre de archivo y almacene el archivo en un lugar

seguro para que pueda recuperarlo más adelante. Si pierde su clave privada, no podrá utilizar su certificado.

Una CSR es un archivo que se envía a una entidad de certificación (CA) para solicitar un certificado digital de servidor. Para crear un CSR, utilice el comando `openssl req`:

```
$ openssl req -new -key privatekey.pem -out csr.pem
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
```

Introduzca la información solicitada y pulse Intro. En la siguiente tabla se describen y muestran ejemplos para cada campo.

Nombre	Descripción	Ejemplo
Country Name	La abreviatura ISO de dos letras del país.	US = Estados Unidos
State or Province	Nombre del país o de la región donde se ubica su organización. Este nombre no se puede abreviar.	Washington
Locality Name	Nombre de la ciudad donde se ubica su organización.	Seattle
Organization Name	Nombre legal completo de su organización. No abrevie el nombre de la organización.	Empresa de ejemplo
Organizational Unit	Información adicional sobre la organización (opcional).	Marketing
Common Name	Nombre completo del dominio de su sitio web. Debe coincidir con el nombre de dominio que los usuarios ven cuando visitan su sitio; de lo contrario, se mostrarán los errores del certificado.	www.example.com

Nombre	Descripción	Ejemplo
Email address	Dirección de correo electrónico del administrador del sitio.	alguien@ejemplo.com

Puede enviar la solicitud de firma a un tercero para que la firme o firmarla usted mismo para las tareas de desarrollo y pruebas. Los certificados autofirmados se pueden utilizar también para la conexión HTTPS del backend entre un balanceador de carga e instancias EC2.

Para firmar el certificado, use el comando `openssl x509`. En el siguiente ejemplo se utiliza la clave privada del paso anterior (*privatekey.pem*) y la solicitud de firma (*csr.pem*) para crear un certificado público denominado *public.crt* válido durante **365** días.

```
$ openssl x509 -req -days 365 -in csr.pem -signkey privatekey.pem -out public.crt
Signature ok
subject=/C=us/ST=Washington/L=Seattle/O=example corporation/OU=marketing/
CN=www.example.com/emailAddress=someone@example.com
Getting Private key
```

Mantenga la clave privada y el certificado público para usarlos más adelante. Puede desechar la solicitud de firma. [Almacene siempre la clave privada en un lugar seguro](#) y evite añadirla a su código fuente.

Para utilizar el certificado con la plataforma Windows Server, debe convertirlo a un formato PFX. Utilice el siguiente comando para crear un certificado PFX a partir de los archivos de clave privada y certificado público:

```
$ openssl pkcs12 -export -out example.com.pfx -inkey privatekey.pem -in public.crt
Enter Export Password: password
Verifying - Enter Export Password: password
```

Ahora que ya tiene un certificado, puede [cargarlo en IAM](#) para su uso con un balanceador de carga o [configurar las instancias de su entorno para terminar HTTPS](#).

Carga de un certificado en IAM

Si desea utilizar el certificado con un balanceador de carga del entorno de Elastic Beanstalk, cargue el certificado y la clave privada en AWS Identity and Access Management (IAM). Puede

utilizar un certificado almacenado en IAM con balanceadores de carga de Elastic Load Balancing y distribuciones de Amazon CloudFront.

Note

AWS Certificate Manager (ACM) es la herramienta preferida para aprovisionar, administrar e implementar los certificados de servidor. Para obtener más información sobre cómo solicitar un certificado de ACM, consulte [Solicitar un certificado](#) en la Guía del usuario de AWS Certificate Manager. Para obtener más información sobre la importación de certificados de terceros en ACM, consulte [Importar certificados](#) en la Guía del usuario de AWS Certificate Manager. Use IAM para cargar un certificado solo si ACM no está [disponible en su región de AWS](#).

Puede utilizar la [AWS Command Line Interface](#) (AWS CLI) para cargar el certificado. El comando siguiente carga un certificado autofirmado llamado *https-cert.crt* con una clave privada denominada *private-key.pem*:

```
$ aws iam upload-server-certificate --server-certificate-name elastic-beanstalk-x509 --
certificate-body file://https-cert.crt --private-key file://private-key.pem
{
  "ServerCertificateMetadata": {
    "ServerCertificateId": "AS5YBEI0N02Q7CAIHKNGC",
    "ServerCertificateName": "elastic-beanstalk-x509",
    "Expiration": "2017-01-31T23:06:22Z",
    "Path": "/",
    "Arn": "arn:aws:iam::123456789012:server-certificate/elastic-beanstalk-x509",
    "UploadDate": "2016-02-01T23:10:34.167Z"
  }
}
```

El prefijo `file://` le indica a la AWS CLI que cargue el contenido de un archivo en el directorio actual. *elastic-beanstalk-x509* especifica el nombre con el que se llamará al certificado de IAM.

Si compró un certificado en una entidad de certificación y recibió un archivo de cadena de certificados, cárguelo también con la opción `--certificate-chain`:

```
$ aws iam upload-server-certificate --server-certificate-name elastic-beanstalk-x509 --  
certificate-chain file://certificate-chain.pem --certificate-body file://https-cert.crt  
--private-key file://private-key.pem
```

Anote el nombre de recurso de Amazon (ARN) del certificado. Tendrá que utilizarlo cuando actualice la configuración del balanceador de carga para poder usar HTTPS.

Note

Un certificado cargado en IAM sigue almacenado incluso después de que no se utilice en cualquier balanceador de carga del entorno. Contiene información confidencial. Cuando ya no necesite el certificado para ningún entorno, asegúrese de eliminarlo. Para obtener detalles acerca de cómo eliminar un certificado de IAM, consulte https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_server-certs.html#delete-server-certificate.

Para obtener más información sobre los certificados de servidor en IAM, consulte [Uso de certificados de servidor](#) en la Guía del usuario de IAM.

Configuración del balanceador de carga del entorno de Elastic Beanstalk para terminar conexiones HTTPS

Para actualizar su AWS Elastic Beanstalk entorno para que use HTTPS, debe configurar un detector HTTPS para el balanceador de cargas de su entorno. Dos tipos de balanceadores de carga admiten un agente de escucha HTTPS: el balanceador de carga clásico y el balanceador de carga de aplicaciones.

Puede utilizar la consola de Elastic Beanstalk o un archivo de configuración para configurar un agente de escucha seguro y asignar el certificado.

Note

Los entornos de una sola instancia no tienen balanceador de carga ni admiten la terminación HTTPS en el balanceador de carga.

Configuración de un agente de escucha seguro mediante la consola de Elastic Beanstalk

Para asignar un certificado al balanceador de carga del entorno

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.


3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Load balancer (Balanceador de carga), elija Edit (Editar).

Note

Si la categoría de configuración Load balancer (Balanceador de carga) no tiene un botón Edit (Editar), el entorno no tiene un [balanceador de carga](#).

5. En la página Modify load balancer (Modificar balanceador de carga), el procedimiento varía en función del tipo de balanceador de carga asociado al entorno.
 - Equilibrador de carga clásico
 - a. Elija Añadir oyente.
 - b. En el cuadro de diálogo Classic Load Balancer listener (Agente de escucha del balanceador de carga clásico), configure las siguientes opciones:
 - En Listener port (Puerto del agente de escucha), escriba el puerto del tráfico entrante, normalmente 443.
 - En Listener protocol (Protocolo del agente de escucha), seleccione HTTPS.
 - En Instance port (Puerto de instancia), escriba 80.
 - En Instance protocol (Protocolo de instancia), elija HTTP.
 - En SSL certificate (Certificado SSL), elija su certificado.
 - c. Elija Añadir.

- Application Load Balancer
 - a. Elija Añadir oyente.
 - b. En el cuadro de diálogo Application Load Balancer listener (Agente de escucha del balanceador de carga de aplicaciones), configure las siguientes opciones:
 - En Port (Puerto), escriba el puerto del tráfico entrante, normalmente 443.
 - En Protocol (Protocolo), seleccione HTTPS.
 - En SSL certificate (Certificado SSL), elija su certificado.
 - c. Elija Añadir.

 Note

Para Classic Load Balancer y Application Load Balancer, si el menú desplegable no muestra ningún certificado, debe crear o cargar un certificado para su [nombre de dominio personalizado](#) en [AWS Certificate Manager \(ACM\)](#) (opción preferida). De forma alternativa, cargue un certificado en IAM con la AWS CLI.

- Network Load Balancer
 - a. Elija Añadir oyente.
 - b. En el cuadro de diálogo Network Load Balancer listener (Agente de escucha del Network Load Balancer), en Port (Puerto), escriba el puerto del tráfico entrante, normalmente 443.
 - c. Elija Añadir.
6. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

Configuración de un agente de escucha seguro mediante un archivo de configuración

Puede configurar un agente de escucha seguro en el balanceador de carga con uno de estos [archivos de configuración](#).

Example `.ebextensions/securelistener-clb.config`

Utilice este ejemplo si su entorno tiene un Classic Load Balancer. En el ejemplo se utilizan las opciones del espacio de nombres `aws:elb:listener` para configurar un agente de escucha

HTTPS en el puerto 443 con el certificado especificado y para que reenvíe el tráfico descifrado a las instancias del entorno en el puerto 80.

```
option_settings:
  aws:elb:listener:443:
    SSLCertificateId: arn:aws:acm:us-east-2:1234567890123:certificate/
    #####
    ListenerProtocol: HTTPS
    InstancePort: 80
```

Sustituya el texto resaltado por el ARN del certificado. El certificado puede ser uno que haya creado o cargado en AWS Certificate Manager (ACM) (preferiblemente) o uno que haya cargado en IAM con el AWS CLI

Para obtener más información acerca de las opciones de configuración del Classic Load Balancer, consulte [Espacios de nombres de la configuración del balanceador de carga clásico](#).

Example `.ebextensions/securelistener-alb.config`

Utilice este ejemplo si su entorno tiene un Application Load Balancer. El ejemplo utiliza las opciones del espacio de nombres `aws:elbv2:listener` para configurar un agente de escucha HTTPS en el puerto 443 con el certificado especificado. El agente de escucha dirige el tráfico al proceso predeterminado.

```
option_settings:
  aws:elbv2:listener:443:
    ListenerEnabled: 'true'
    Protocol: HTTPS
    SSLCertificateArns: arn:aws:acm:us-east-2:1234567890123:certificate/
    #####
```

Example `.ebextensions/securelistener-nlb.config`

Utilice este ejemplo si su entorno tiene un Network Load Balancer. El ejemplo utiliza opciones en el espacio de nombres `aws:elbv2:listener` para configurar un agente de escucha en el puerto 443. El agente de escucha dirige el tráfico al proceso predeterminado.

```
option_settings:
  aws:elbv2:listener:443:
    ListenerEnabled: 'true'
```

Configuración de un grupo de seguridad

Si configura el balanceador de carga para que reenvíe el tráfico a un puerto de la instancia que no sea el 80, debe agregar una regla al grupo de seguridad que permita el tráfico procedente del balanceador de carga en el puerto de la instancia. Si crea el entorno en una VPC personalizada, Elastic Beanstalk agrega esta regla por usted.

Para incorporar esta regla, puede agregar una clave `Resources` a un [archivo de configuración](#) del directorio `.ebextensions` de la aplicación.

El siguiente archivo de configuración de ejemplo agrega una regla de entrada al grupo de seguridad `AWSEBSecurityGroup`. Esto permite el tráfico en el puerto 1000 procedente del grupo de seguridad del balanceador de carga.

Example `.ebextensions/sg-ingressfromlb.config`

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 1000
      FromPort: 1000
      SourceSecurityGroupId: {"Fn::GetAtt" : ["AWSEBLoadBalancerSecurityGroup",
"GroupId"]}]}
```

Configurar su aplicación para terminar las conexiones HTTPS en la instancia

Puede utilizar [archivos de configuración](#) para configurar el servidor proxy que pasa el tráfico a la aplicación de forma que termine las conexiones HTTPS. Esto resulta útil si desea utilizar HTTPS con un entorno de una sola instancia o si configura el balanceador de carga para que transfiera el tráfico sin descifrarlo.

Para habilitar HTTPS, debe permitir el tráfico entrante en el puerto 443 dirigido a la instancia EC2 en la que se ejecuta la aplicación de Elastic Beanstalk. Para ello, debe usar la clave `Resources` del archivo de configuración para agregar una regla sobre el puerto 443 a las reglas de entrada del grupo de seguridad `AWSEBSecurityGroup`.

El siguiente fragmento de código agrega una regla de entrada al grupo de seguridad `AWSEBSecurityGroup` que abre el puerto 443 a todo el tráfico en un entorno formado por una sola instancia:

`.ebextensions/https-instance-securitygroup.config`

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      CidrIp: 0.0.0.0/0
```

En un entorno con balanceo de carga de una [Amazon Virtual Private Cloud](#) (Amazon VPC) predeterminada, puede modificar esta política para que solo acepte el tráfico procedente del balanceador de carga. Consulte [Configuración del cifrado integral en un entorno de Elastic Beanstalk con balanceo de carga](#) para ver un ejemplo.

Plataformas

- [Terminación de HTTPS en instancias EC2 que ejecutan Docker](#)
- [Terminación de HTTPS en instancias EC2 que ejecutan Go](#)
- [Terminación de HTTPS en instancias EC2 que ejecutan Java SE](#)
- [Terminación de HTTPS en instancias EC2 que ejecutan Node.js](#)
- [Terminación de HTTPS en instancias EC2 que ejecutan PHP](#)
- [Terminación de HTTPS en instancias EC2 que ejecutan Python](#)
- [Terminación de HTTPS en instancias EC2 que ejecutan Ruby](#)
- [Terminación de HTTPS en instancias EC2 que ejecutan Tomcat](#)
- [Terminación de HTTPS en instancias de Amazon EC2 que ejecutan .NET Core en Linux](#)
- [Terminación de HTTPS en instancias de Amazon EC2 que ejecutan .NET](#)

Terminación de HTTPS en instancias EC2 que ejecutan Docker

Para los contenedores Docker, utiliza un [archivo de configuración](#) para habilitar HTTPS.

Añada el siguiente fragmento al archivo de configuración, sustituyendo los marcadores de posición de certificado y las claves privadas tal como se indica, y guarde el archivo en el directorio `.ebextensions` del paquete de código fuente. El archivo de configuración realiza las siguientes tareas:


- La clave `files` crea los siguientes archivos en la instancia:

```
/etc/nginx/conf.d/https.conf
```

Configura el servidor de nginx. Este archivo se carga cuando se inicia el servicio de nginx.

```
/etc/pki/tls/certs/server.crt
```

Crea el archivo de certificado en la instancia. Sustituya *certificate file contents* por el contenido del certificado.

 Note

YAML usa la sangría uniforme. Utilice el mismo nivel de sangría cuando sustituya el contenido en el archivo de configuración de ejemplo y asegúrese de que el editor de texto utiliza espacios para la sangría, no tabuladores.

Si tiene certificados intermedios, inclúyalos en `server.crt` después del certificado del sitio.

```
-----BEGIN CERTIFICATE-----  
certificate file contents  
-----END CERTIFICATE-----  
-----BEGIN CERTIFICATE-----  
first intermediate certificate  
-----END CERTIFICATE-----  
-----BEGIN CERTIFICATE-----  
second intermediate certificate  
-----END CERTIFICATE-----
```

```
/etc/pki/tls/certs/server.key
```

Crea el archivo de clave privada en la instancia. Sustituya *private key contents* por el contenido de la clave privada usada para crear la solicitud del certificado o el certificado autofirmado.

Example .ebextensions/https-instance.config

```
files:
  /etc/nginx/conf.d/https.conf:
    mode: "000644"
    owner: root
    group: root
    content: |
      # HTTPS Server

      server {
        listen 443;
        server_name localhost;

        ssl on;
        ssl_certificate /etc/pki/tls/certs/server.crt;
        ssl_certificate_key /etc/pki/tls/certs/server.key;

        ssl_session_timeout 5m;

        ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
        ssl_prefer_server_ciphers on;

        location / {
          proxy_pass http://docker;
          proxy_http_version 1.1;

          proxy_set_header Connection "";
          proxy_set_header Host $host;
          proxy_set_header X-Real-IP $remote_addr;
          proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
          proxy_set_header X-Forwarded-Proto https;
        }
      }

  /etc/pki/tls/certs/server.crt:
    mode: "000400"
    owner: root
    group: root
    content: |
      -----BEGIN CERTIFICATE-----
      certificate file contents
      -----END CERTIFICATE-----
```

```
/etc/pki/tls/certs/server.key:
mode: "000400"
owner: root
group: root
content: |
  -----BEGIN RSA PRIVATE KEY-----
  private key contents # See note below.
  -----END RSA PRIVATE KEY-----
```

Note

Evite confirmar un archivo de configuración que contenga su clave privada en el control de código fuente. Una vez comprobada la configuración y confirmado su funcionamiento, almacene la clave privada en Amazon S3 y modifique la configuración para descargarla durante la implementación. Para obtener instrucciones, consulte [Almacenamiento seguro de claves privadas en Amazon S3](#).

En un entorno de instancia única, también debe modificar el grupo de seguridad de la instancia para permitir el tráfico en el puerto 443. El siguiente archivo de configuración recupera el ID del grupo de seguridad con una [función](#) de AWS CloudFormation y le agrega una regla.

Example `.ebextensions/https-instance-single.config`

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      CidrIp: 0.0.0.0/0
```

En un entorno de balanceador de carga, el balanceador de carga se configura [para el paso a través del tráfico seguro](#) o [para descifrar y volver a cifrar](#) en el cifrado integral.

Terminación de HTTPS en instancias EC2 que ejecutan Go


Para los tipos de contenedores Go, habilite HTTPS con un [archivo de configuración](#) y un archivo de configuración nginx que configure el servidor nginx para usar HTTPS.

Añada el siguiente fragmento a su archivo de configuración, sustituyendo los marcadores de posición de certificado y clave privada tal como se indica, y guarde el archivo en el directorio `.ebextensions` del paquete de código fuente. El archivo de configuración realiza las siguientes tareas:

- La clave `Resources` habilita el puerto 443 en el grupo de seguridad utilizado por la instancia del entorno.
- La clave `files` crea los siguientes archivos en la instancia:

```
/etc/pki/tls/certs/server.crt
```

Crea el archivo de certificado en la instancia. Sustituya *certificate file contents* por el contenido del certificado.

 Note

YAML usa la sangría uniforme. Utilice el mismo nivel de sangría cuando sustituya el contenido en el archivo de configuración de ejemplo y asegúrese de que el editor de texto utiliza espacios para la sangría, no tabuladores.

Si tiene certificados intermedios, inclúyalos en `server.crt` después del certificado del sitio.

```
-----BEGIN CERTIFICATE-----  
certificate file contents  
-----END CERTIFICATE-----  
-----BEGIN CERTIFICATE-----  
first intermediate certificate  
-----END CERTIFICATE-----  
-----BEGIN CERTIFICATE-----  
second intermediate certificate  
-----END CERTIFICATE-----
```

```
/etc/pki/tls/certs/server.key
```

Crea el archivo de clave privada en la instancia. Sustituya *private key contents* por el contenido de la clave privada usada para crear la solicitud del certificado o el certificado autofirmado.


- La clave `container_commands` reinicia el servidor `nginx` una vez que está todo configurado para que el servidor cargue el archivo de configuración de `nginx`.

Example .ebextensions/https-instance.config

```
files:
  /etc/pki/tls/certs/server.crt:
    content: |
      -----BEGIN CERTIFICATE-----
      certificate file contents
      -----END CERTIFICATE-----

  /etc/pki/tls/certs/server.key:
    content: |
      -----BEGIN RSA PRIVATE KEY-----
      private key contents # See note below.
      -----END RSA PRIVATE KEY-----

container_commands:
  01restart_nginx:
    command: "service nginx restart"
```

 Note

Evite confirmar un archivo de configuración que contenga su clave privada en el control de código fuente. Una vez comprobada la configuración y confirmado su funcionamiento, almacene la clave privada en Amazon S3 y modifique la configuración para descargarla durante la implementación. Para obtener instrucciones, consulte [Almacenamiento seguro de claves privadas en Amazon S3](#).

Incluya lo siguiente en un archivo con la extensión `.conf` en el directorio `.ebextensions/nginx/conf.d/` del paquete de código fuente (por ejemplo, `.ebextensions/nginx/conf.d/https.conf`). Sustituya `app_port` por el número de puerto en el que su aplicación está a la escucha. Este ejemplo configura el servidor nginx para ponerlo a la escucha en el puerto 443 usando SSL. Para obtener más información acerca de estos archivos de configuración en la plataforma Go, consulte [Configuración del proxy inverso](#).

Example .ebextensions/nginx/conf.d/https.conf

```
# HTTPS server

server {
```

```

listen      443;
server_name localhost;

ssl         on;
ssl_certificate      /etc/pki/tls/certs/server.crt;
ssl_certificate_key  /etc/pki/tls/certs/server.key;

ssl_session_timeout 5m;

ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
ssl_prefer_server_ciphers on;

location / {
    proxy_pass http://localhost:app_port;
    proxy_set_header    Connection "";
    proxy_http_version  1.1;
    proxy_set_header    Host      $host;
    proxy_set_header    X-Real-IP $remote_addr;
    proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header    X-Forwarded-Proto https;
}
}

```

En un entorno de instancia única, también debe modificar el grupo de seguridad de la instancia para permitir el tráfico en el puerto 443. El siguiente archivo de configuración recupera el ID del grupo de seguridad con una [función](#) de AWS CloudFormation y le agrega una regla.

Example `.ebextensions/https-instance-single.config`

```

Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      CidrIp: 0.0.0.0/0

```

En un entorno de balanceador de carga, el balanceador de carga se configura [para el paso a través del tráfico seguro](#) o [para descifrar y volver a cifrar](#) en el cifrado integral.

Terminación de HTTPS en instancias EC2 que ejecutan Java SE

Para los tipos de contenedores Java SE, habilite HTTPS con un [archivo de configuración](#) `.ebextensions` y un archivo de configuración nginx que configure el servidor nginx para usar HTTPS.

Todas las plataformas AL2023/AL2 son compatibles con una característica de configuración de proxy uniforme. Para obtener más información sobre la configuración del servidor proxy en las versiones de su plataforma que ejecutan AL2023/AL2, expanda la sección Configuración del proxy inverso de [the section called “Ampliación de plataformas Linux”](#).

Añada el siguiente fragmento a su archivo de configuración, sustituyendo los marcadores de posición de certificado y clave privada tal como se indica, y guárdelo en el directorio `.ebextensions`. El archivo de configuración realiza las siguientes tareas:

- La clave `files` crea los siguientes archivos en la instancia:

```
/etc/pki/tls/certs/server.crt
```

Crea el archivo de certificado en la instancia. Sustituya *certificate file contents* por el contenido del certificado.

Note

YAML usa la sangría uniforme. Utilice el mismo nivel de sangría cuando sustituya el contenido en el archivo de configuración de ejemplo y asegúrese de que el editor de texto utiliza espacios para la sangría, no tabuladores.

Si tiene certificados intermedios, inclúyalos en `server.crt` después del certificado del sitio.

```
-----BEGIN CERTIFICATE-----  
certificate file contents  
-----END CERTIFICATE-----  
-----BEGIN CERTIFICATE-----  
first intermediate certificate  
-----END CERTIFICATE-----  
-----BEGIN CERTIFICATE-----  
second intermediate certificate  
-----END CERTIFICATE-----
```



```
/etc/pki/tls/certs/server.key
```

Crea el archivo de clave privada en la instancia. Sustituya *private key contents* por el contenido de la clave privada usada para crear la solicitud del certificado o el certificado autofirmado.

- La clave `container_commands` reinicia el servidor nginx una vez que está todo configurado para que el servidor cargue el archivo de configuración de nginx.

Example `.ebextensions/https-instance.config`

```
files:
  /etc/pki/tls/certs/server.crt:
    content: |
      -----BEGIN CERTIFICATE-----
      certificate file contents
      -----END CERTIFICATE-----

  /etc/pki/tls/certs/server.key:
    content: |
      -----BEGIN RSA PRIVATE KEY-----
      private key contents # See note below.
      -----END RSA PRIVATE KEY-----

container_commands:
  01restart_nginx:
    command: "service nginx restart"
```

Note

Evite confirmar un archivo de configuración que contenga su clave privada en el control de código fuente. Una vez comprobada la configuración y confirmado su funcionamiento, almacene la clave privada en Amazon S3 y modifique la configuración para descargarla durante la implementación. Para obtener instrucciones, consulte [Almacenamiento seguro de claves privadas en Amazon S3](#).

Incluya lo siguiente en un archivo con la extensión `.conf` en el directorio `.ebextensions/nginx/conf.d/` del paquete de código fuente (por ejemplo, `.ebextensions/nginx/conf.d/https.conf`). Sustituya *app_port* por el número de puerto en el que su aplicación está a la

escucha. Este ejemplo configura el servidor nginx para ponerlo a la escucha en el puerto 443 usando SSL. Para obtener más información acerca de estos archivos de configuración en la plataforma Java SE, consulte [Configuración del proxy inverso](#).

Example `.ebextensions/nginx/conf.d/https.conf`

```
# HTTPS server

server {
    listen      443;
    server_name localhost;

    ssl         on;
    ssl_certificate      /etc/pki/tls/certs/server.crt;
    ssl_certificate_key  /etc/pki/tls/certs/server.key;

    ssl_session_timeout 5m;

    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
    ssl_prefer_server_ciphers on;

    location / {
        proxy_pass http://localhost:app_port;
        proxy_set_header    Connection "";
        proxy_http_version  1.1;
        proxy_set_header    Host      $host;
        proxy_set_header    X-Real-IP $remote_addr;
        proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header    X-Forwarded-Proto https;
    }
}
```

En un entorno de instancia única, también debe modificar el grupo de seguridad de la instancia para permitir el tráfico en el puerto 443. El siguiente archivo de configuración recupera el ID del grupo de seguridad con una [función](#) de AWS CloudFormation y le agrega una regla.

Example `.ebextensions/https-instance-single.config`

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
```

```
IpProtocol: tcp
ToPort: 443
FromPort: 443
CidrIp: 0.0.0.0/0
```

En un entorno de balanceador de carga, el balanceador de carga se configura [para el paso a través del tráfico seguro](#) o [para descifrar y volver a cifrar](#) en el cifrado integral.

Terminación de HTTPS en instancias EC2 que ejecutan Node.js

El siguiente archivo de configuración de ejemplo [amplía la configuración de nginx predeterminada](#) para estar a la escucha en el puerto 443 y terminar las conexiones SSL/TLS con un certificado público y una clave privada.

Si configuró el entorno para [informes de estado mejorados](#), necesita configurar nginx para generar los registros de acceso. Para ello, quite el comentario del bloque de líneas debajo del comentario que `# For enhanced health...` quitando los caracteres `#` del principio.

Example `.ebextensions/https-instance.config`

```
files:
  /etc/nginx/conf.d/https.conf:
    mode: "000644"
    owner: root
    group: root
    content: |
      # HTTPS server

      server {
        listen      443;
        server_name localhost;

        ssl          on;
        ssl_certificate      /etc/pki/tls/certs/server.crt;
        ssl_certificate_key  /etc/pki/tls/certs/server.key;

        ssl_session_timeout 5m;

        ssl_protocols  TLSv1 TLSv1.1 TLSv1.2;
        ssl_prefer_server_ciphers  on;

        # For enhanced health reporting support, uncomment this block:
```

```

    #if ($time_iso8601 ~ "^(\\d{4})-(\\d{2})-(\\d{2})T(\\d{2})") {
    #   set $year $1;
    #   set $month $2;
    #   set $day $3;
    #   set $hour $4;
    #}
    #access_log /var/log/nginx/healthd/application.log.$year-$month-$day-$hour
healthd;
    #access_log /var/log/nginx/access.log main;

    location / {
        proxy_pass http://nodejs;
        proxy_set_header    Connection "";
        proxy_http_version 1.1;
        proxy_set_header    Host          $host;
        proxy_set_header    X-Real-IP    $remote_addr;
        proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header    X-Forwarded-Proto https;
    }
}

/etc/pki/tls/certs/server.crt:
mode: "000400"
owner: root
group: root
content: |
    -----BEGIN CERTIFICATE-----
    certificate file contents
    -----END CERTIFICATE-----

/etc/pki/tls/certs/server.key:
mode: "000400"
owner: root
group: root
content: |
    -----BEGIN RSA PRIVATE KEY-----
    private key contents # See note below.
    -----END RSA PRIVATE KEY-----

```

La clave files crea los siguientes archivos en la instancia:

`/etc/nginx/conf.d/https.conf`

Configura el servidor de nginx. Este archivo se carga cuando se inicia el servicio de nginx.

```
/etc/pki/tls/certs/server.crt
```

Crea el archivo de certificado en la instancia. Sustituya *certificate file contents* por el contenido del certificado.

Note

YAML usa la sangría uniforme. Utilice el mismo nivel de sangría cuando sustituya el contenido en el archivo de configuración de ejemplo y asegúrese de que el editor de texto utiliza espacios para la sangría, no tabuladores.

Si tiene certificados intermedios, inclúyalos en `server.crt` después del certificado del sitio.

```
-----BEGIN CERTIFICATE-----  
certificate file contents  
-----END CERTIFICATE-----  
-----BEGIN CERTIFICATE-----  
first intermediate certificate  
-----END CERTIFICATE-----  
-----BEGIN CERTIFICATE-----  
second intermediate certificate  
-----END CERTIFICATE-----
```

```
/etc/pki/tls/certs/server.key
```

Crea el archivo de clave privada en la instancia. Sustituya *private key contents* por el contenido de la clave privada usada para crear la solicitud del certificado o el certificado autofirmado.

Note

Evite confirmar un archivo de configuración que contenga su clave privada en el control de código fuente. Una vez comprobada la configuración y confirmado su funcionamiento, almacene la clave privada en Amazon S3 y modifique la configuración para descargarla durante la implementación. Para obtener instrucciones, consulte [Almacenamiento seguro de claves privadas en Amazon S3](#).

En un entorno de instancia única, también debe modificar el grupo de seguridad de la instancia para permitir el tráfico en el puerto 443. El siguiente archivo de configuración recupera el ID del grupo de seguridad con una [función](#) de AWS CloudFormation y le agrega una regla.

Example `.ebextensions/https-instance-single.config`

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      CidrIp: 0.0.0.0/0
```

En un entorno de balanceador de carga, el balanceador de carga se configura [para el paso a través del tráfico seguro](#) o [para descifrar y volver a cifrar](#) en el cifrado integral.

Terminación de HTTPS en instancias EC2 que ejecutan PHP

Para los tipos de contenedores PHP, usa un [archivo de configuración](#) para habilitar el servidor HTTP Apache para que utilice HTTPS.

Añada el siguiente fragmento al archivo de configuración, sustituyendo los marcadores de posición de certificado y las claves privadas tal como se indica, y guarde el archivo en el directorio `.ebextensions` del paquete de código fuente.

El archivo de configuración realiza las siguientes tareas:

- La clave `packages` usa `yum` para instalar `mod24_ssl`.
- La clave `files` crea los siguientes archivos en la instancia:

```
/etc/httpd/conf.d/ssl.conf
```

Configura el servidor de Apache. Este archivo se carga cuando se inicia el servicio de Apache.

```
/etc/pki/tls/certs/server.crt
```

Crea el archivo de certificado en la instancia. Sustituya *certificate file contents* por el contenido del certificado.

Note

YAML usa la sangría uniforme. Utilice el mismo nivel de sangría cuando sustituya el contenido en el archivo de configuración de ejemplo y asegúrese de que el editor de texto utiliza espacios para la sangría, no tabuladores.

Si tiene certificados intermedios, inclúyalos en `server.crt` después del certificado del sitio.

```

-----BEGIN CERTIFICATE-----
certificate file contents
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
first intermediate certificate
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
second intermediate certificate
-----END CERTIFICATE-----

```

`/etc/pki/tls/certs/server.key`

Crea el archivo de clave privada en la instancia. Sustituya *private key contents* por el contenido de la clave privada usada para crear la solicitud del certificado o el certificado autofirmado.

Example `.ebextensions/https-instance.config`

```

packages:
  yum:
    mod24_ssl : []

files:
  /etc/httpd/conf.d/ssl.conf:
    mode: "000644"
    owner: root
    group: root
    content: |
      LoadModule ssl_module modules/mod_ssl.so
      Listen 443
      <VirtualHost *:443>

```

```
<Proxy *>
  Order deny,allow
  Allow from all
</Proxy>

SSLEngine          on
SSLCertificateFile  "/etc/pki/tls/certs/server.crt"
SSLCertificateKeyFile "/etc/pki/tls/certs/server.key"
SSLCipherSuite     EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH
SSLProtocol        All -SSLv2 -SSLv3
SSLHonorCipherOrder On
SSLSessionTickets  Off

Header always set Strict-Transport-Security "max-age=63072000;
includeSubdomains; preload"
Header always set X-Frame-Options DENY
Header always set X-Content-Type-Options nosniff

ProxyPass / http://localhost:80/ retry=0
ProxyPassReverse / http://localhost:80/
ProxyPreserveHost on
RequestHeader set X-Forwarded-Proto "https" early

</VirtualHost>

/etc/pki/tls/certs/server.crt:
mode: "000400"
owner: root
group: root
content: |
  -----BEGIN CERTIFICATE-----
  certificate file contents
  -----END CERTIFICATE-----

/etc/pki/tls/certs/server.key:
mode: "000400"
owner: root
group: root
content: |
  -----BEGIN RSA PRIVATE KEY-----
  private key contents # See note below.
  -----END RSA PRIVATE KEY-----
```


Note

Evite confirmar un archivo de configuración que contenga su clave privada en el control de código fuente. Una vez comprobada la configuración y confirmado su funcionamiento, almacene la clave privada en Amazon S3 y modifique la configuración para descargarla durante la implementación. Para obtener instrucciones, consulte [Almacenamiento seguro de claves privadas en Amazon S3](#).

En un entorno de instancia única, también debe modificar el grupo de seguridad de la instancia para permitir el tráfico en el puerto 443. El siguiente archivo de configuración recupera el ID del grupo de seguridad con una [función](#) de AWS CloudFormation y le agrega una regla.

Example `.ebextensions/https-instance-single.config`

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      CidrIp: 0.0.0.0/0
```

En un entorno de balanceador de carga, el balanceador de carga se configura [para el paso a través del tráfico seguro](#) o [para descifrar y volver a cifrar](#) en el cifrado integral.

Terminación de HTTPS en instancias EC2 que ejecutan Python

En los tipos de contenedores de Python que utilizan Apache HTTP Server con Web Server Gateway Interface (WSGI), puede utilizar un [archivo de configuración](#) para permitir que Apache HTTP Server use HTTPS.

Añada el siguiente fragmento al [archivo de configuración](#), sustituyendo los marcadores de posición de certificado y las claves privadas tal como se indica, y guarde el archivo en el directorio `.ebextensions` del paquete de código fuente. El archivo de configuración realiza las siguientes tareas:

- La clave `packages` usa `yum` para instalar `mod24_ssl`.

- La clave `files` crea los siguientes archivos en la instancia:


```
/etc/httpd/conf.d/ssl.conf
```

Configura el servidor de Apache. Si la aplicación no se llama `application.py`, cambie el texto resaltado en el valor de `WSGIScriptAlias` por la ruta local de la aplicación. Por ejemplo, una aplicación de django podría estar en `django/wsgi.py`. La ubicación debe coincidir con el valor de la opción `WSGIPath` que estableció en el entorno.

En función de los requisitos de la aplicación, es posible que tenga que agregar también otros directorios al parámetro `python-path`.

```
/etc/pki/tls/certs/server.crt
```

Crea el archivo de certificado en la instancia. Sustituya *certificate file contents* por el contenido del certificado.

 Note

YAML usa la sangría uniforme. Utilice el mismo nivel de sangría cuando sustituya el contenido en el archivo de configuración de ejemplo y asegúrese de que el editor de texto utiliza espacios para la sangría, no tabuladores.

Si tiene certificados intermedios, inclúyalos en `server.crt` después del certificado del sitio.

```
-----BEGIN CERTIFICATE-----
certificate file contents
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
first intermediate certificate
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
second intermediate certificate
-----END CERTIFICATE-----
```

```
/etc/pki/tls/certs/server.key
```

Crea el archivo de clave privada en la instancia. Sustituya *private key contents* por el contenido de la clave privada usada para crear la solicitud del certificado o el certificado autofirmado.

- La clave `container_commands` detiene el servicio `httpd` una vez que está todo configurado para que el servicio use el nuevo archivo `https.conf` y certificado.

Note

El ejemplo solo funciona en entornos que utilizan la plataforma [Python](#).

Example `.ebextensions/https-instance.config`

```
packages:
  yum:
    mod24_ssl : []

files:
  /etc/httpd/conf.d/ssl.conf:
    mode: "000644"
    owner: root
    group: root
    content: |
      LoadModule wsgi_module modules/mod_wsgi.so
      WSGIPythonHome /opt/python/run/baselinenv
      WSGISocketPrefix run/wsgi
      WSGIRestrictEmbedded On
      Listen 443
      <VirtualHost *:443>
        SSLEngine on
        SSLCertificateFile "/etc/pki/tls/certs/server.crt"
        SSLCertificateKeyFile "/etc/pki/tls/certs/server.key"

        Alias /static/ /opt/python/current/app/static/
        <Directory /opt/python/current/app/static>
          Order allow,deny
          Allow from all
        </Directory>

        WSGIScriptAlias / /opt/python/current/app/application.py

        <Directory /opt/python/current/app>
          Require all granted
        </Directory>
```

```
WSGIDaemonProcess wsgi-ssl processes=1 threads=15 display-name=%{GROUP} \  
  python-path=/opt/python/current/app \  
  python-home=/opt/python/run/venv \  
  home=/opt/python/current/app \  
  user=wsgi \  
  group=wsgi  
WSGIProcessGroup wsgi-ssl  
  
</VirtualHost>  
  
/etc/pki/tls/certs/server.crt:  
mode: "000400"  
owner: root  
group: root  
content: |  
  -----BEGIN CERTIFICATE-----  
  certificate file contents  
  -----END CERTIFICATE-----  
  
/etc/pki/tls/certs/server.key:  
mode: "000400"  
owner: root  
group: root  
content: |  
  -----BEGIN RSA PRIVATE KEY-----  
  private key contents # See note below.  
  -----END RSA PRIVATE KEY-----  
  
container_commands:  
  01killhttpd:  
    command: "killall httpd"  
  02waitforhttpddeath:  
    command: "sleep 3"
```

Note

Evite confirmar un archivo de configuración que contenga su clave privada en el control de código fuente. Una vez comprobada la configuración y confirmado su funcionamiento, almacene la clave privada en Amazon S3 y modifique la configuración para descargarla durante la implementación. Para obtener instrucciones, consulte [Almacenamiento seguro de claves privadas en Amazon S3](#).

En un entorno de instancia única, también debe modificar el grupo de seguridad de la instancia para permitir el tráfico en el puerto 443. El siguiente archivo de configuración recupera el ID del grupo de seguridad con una [función](#) de AWS CloudFormation y le agrega una regla.

Example `.ebextensions/https-instance-single.config`

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      CidrIp: 0.0.0.0/0
```

En un entorno de balanceador de carga, el balanceador de carga se configura [para el paso a través del tráfico seguro](#) o [para descifrar y volver a cifrar](#) en el cifrado integral.

Terminación de HTTPS en instancias EC2 que ejecutan Ruby

En los tipos de contenedores de Ruby, la forma de habilitar HTTPS depende del tipo de servidor de aplicaciones utilizado.

Temas

- [Configuración de HTTPS para Ruby con Puma](#)
- [Configuración de HTTPS para Ruby con Passenger](#)

Configuración de HTTPS para Ruby con Puma

En los tipos de contenedores de Ruby que utilizan Puma como servidor de aplicaciones, debe usar un [archivo de configuración](#) para habilitar el servidor HTTPS.

Añada el siguiente fragmento al archivo de configuración, sustituyendo los marcadores de posición de certificado y las claves privadas tal como se indica, y guarde el archivo en el directorio `.ebextensions` del paquete de código fuente. El archivo de configuración realiza las siguientes tareas:


- La clave `files` crea los siguientes archivos en la instancia:

```
/etc/nginx/conf.d/https.conf
```

Configura el servidor de nginx. Este archivo se carga cuando se inicia el servicio de nginx.

```
/etc/pki/tls/certs/server.crt
```

Crea el archivo de certificado en la instancia. Sustituya *certificate file contents* por el contenido del certificado.

 Note

YAML usa la sangría uniforme. Utilice el mismo nivel de sangría cuando sustituya el contenido en el archivo de configuración de ejemplo y asegúrese de que el editor de texto utiliza espacios para la sangría, no tabuladores.

Si tiene certificados intermedios, inclúyalos en `server.crt` después del certificado del sitio.

```

-----BEGIN CERTIFICATE-----
certificate file contents
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
first intermediate certificate
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
second intermediate certificate
-----END CERTIFICATE-----

```

```
/etc/pki/tls/certs/server.key
```

Crea el archivo de clave privada en la instancia. Sustituya *private key contents* por el contenido de la clave privada usada para crear la solicitud del certificado o el certificado autofirmado.

- La clave `container_commands` reinicia el servidor nginx una vez que todo está configurado para que el servidor use el nuevo archivo `https.conf`.

Example `.ebextensions/https-instance.config`

```
files:
  /etc/nginx/conf.d/https.conf:
```

```
content: |
  # HTTPS server

  server {
    listen      443;
    server_name localhost;

    ssl         on;
    ssl_certificate      /etc/pki/tls/certs/server.crt;
    ssl_certificate_key  /etc/pki/tls/certs/server.key;

    ssl_session_timeout 5m;

    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
    ssl_prefer_server_ciphers on;

    location / {
      proxy_pass http://my_app;
      proxy_set_header    Host          $host;
      proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;
      proxy_set_header    X-Forwarded-Proto https;
    }

    location /assets {
      alias /var/app/current/public/assets;
      gzip_static on;
      gzip on;
      expires max;
      add_header Cache-Control public;
    }

    location /public {
      alias /var/app/current/public;
      gzip_static on;
      gzip on;
      expires max;
      add_header Cache-Control public;
    }
  }

/etc/pki/tls/certs/server.crt:
content: |
  -----BEGIN CERTIFICATE-----
  certificate file contents
```

```

-----END CERTIFICATE-----

/etc/pki/tls/certs/server.key:
content: |
-----BEGIN RSA PRIVATE KEY-----
private key contents # See note below.
-----END RSA PRIVATE KEY-----

container_commands:
  01restart_nginx:
    command: "service nginx restart"

```

Note

Evite confirmar un archivo de configuración que contenga su clave privada en el control de código fuente. Una vez comprobada la configuración y confirmado su funcionamiento, almacene la clave privada en Amazon S3 y modifique la configuración para descargarla durante la implementación. Para obtener instrucciones, consulte [Almacenamiento seguro de claves privadas en Amazon S3](#).

En un entorno de instancia única, también debe modificar el grupo de seguridad de la instancia para permitir el tráfico en el puerto 443. El siguiente archivo de configuración recupera el ID del grupo de seguridad con una [función](#) de AWS CloudFormation y le agrega una regla.

Example `.ebextensions/https-instance-single.config`

```

Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      CidrIp: 0.0.0.0/0

```

En un entorno de balanceador de carga, el balanceador de carga se configura [para el paso a través del tráfico seguro](#) o [para descifrar y volver a cifrar](#) en el cifrado integral.

Configuración de HTTPS para Ruby con Passenger

En los tipos de contenedores de Passenger que utilizan Puma como servidor de aplicaciones, debe usar un archivo de configuración y un archivo JSON para habilitar el servidor HTTPS.

Si desea configurar HTTPS para Ruby con Passenger

1. Añada el siguiente fragmento al archivo de configuración, sustituyendo los marcadores de posición de certificado y las claves privadas tal como se indica, y guarde el archivo en el directorio `.ebextensions` del paquete de código fuente. El archivo de configuración realiza las siguientes tareas:

- La clave `files` crea los siguientes archivos en la instancia:

```
/etc/pki/tls/certs/server.crt
```

Crea el archivo de certificado en la instancia. Sustituya *certificate file contents* por el contenido del certificado.

Note

YAML usa la sangría uniforme. Utilice el mismo nivel de sangría cuando sustituya el contenido en el archivo de configuración de ejemplo y asegúrese de que el editor de texto utiliza espacios para la sangría, no tabuladores.

Si tiene certificados intermedios, inclúyalos en `server.crt` después del certificado del sitio.

```
-----BEGIN CERTIFICATE-----  
certificate file contents  
-----END CERTIFICATE-----  
-----BEGIN CERTIFICATE-----  
first intermediate certificate  
-----END CERTIFICATE-----  
-----BEGIN CERTIFICATE-----  
second intermediate certificate  
-----END CERTIFICATE-----
```

```
/etc/pki/tls/certs/server.key
```

Crea el archivo de clave privada en la instancia. Sustituya *private key contents* por el contenido de la clave privada usada para crear la solicitud del certificado o el certificado autofirmado.

Example Fragmento de código .ebextensions sobre la configuración de HTTPS para Ruby con Passenger

```
files:
  /etc/pki/tls/certs/server.crt:
    content: |
      -----BEGIN CERTIFICATE-----
      certificate file contents
      -----END CERTIFICATE-----

  /etc/pki/tls/certs/server.key:
    content: |
      -----BEGIN RSA PRIVATE KEY-----
      private key contents # See note below.
      -----END RSA PRIVATE KEY-----
```

Note

Evite confirmar un archivo de configuración que contenga su clave privada en el control de código fuente. Una vez comprobada la configuración y confirmado su funcionamiento, almacene la clave privada en Amazon S3 y modifique la configuración para descargarla durante la implementación. Para obtener instrucciones, consulte [Almacenamiento seguro de claves privadas en Amazon S3](#).

2. Cree un archivo de texto y agregue el siguiente JSON al archivo. Guárdelo en el directorio raíz del paquete de código fuente con el nombre `passenger-standalone.json`. Este archivo JSON configura Passenger para usarlo con HTTPS.

⚠ Important

Este archivo JSON no debe contener ninguna marca de orden de bytes (BOM). De lo contrario, la biblioteca JSON de Passenger no leerá el archivo correctamente y no se iniciará el servicio Passenger.

Example passenger-standalone.json

```
{
  "ssl" : true,
  "ssl_port" : 443,
  "ssl_certificate" : "/etc/pki/tls/certs/server.crt",
  "ssl_certificate_key" : "/etc/pki/tls/certs/server.key"
}
```

En un entorno de instancia única, también debe modificar el grupo de seguridad de la instancia para permitir el tráfico en el puerto 443. El siguiente archivo de configuración recupera el ID del grupo de seguridad con una [función](#) de AWS CloudFormation y le agrega una regla.

Example .ebextensions/https-instance-single.config

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      CidrIp: 0.0.0.0/0
```

En un entorno de balanceador de carga, el balanceador de carga se configura [para el paso a través del tráfico seguro](#) o [para descifrar y volver a cifrar](#) en el cifrado integral.

Terminación de HTTPS en instancias EC2 que ejecutan Tomcat


Para los tipos de contenedores Tomcat, utilice un [archivo de configuración](#) para habilitar el servidor HTTP Apache para que use HTTPS cuando actúe como proxy inverso para Tomcat.

Añada el siguiente fragmento al archivo de configuración, sustituyendo los marcadores de posición de certificado y las claves privadas tal como se indica, y guarde el archivo en el directorio `.ebextensions` del paquete de código fuente. El archivo de configuración realiza las siguientes tareas:

- La clave `files` crea los siguientes archivos en la instancia:

```
/etc/pki/tls/certs/server.crt
```

Crea el archivo de certificado en la instancia. Sustituya *certificate file contents* por el contenido del certificado.

 Note

YAML usa la sangría uniforme. Utilice el mismo nivel de sangría cuando sustituya el contenido en el archivo de configuración de ejemplo y asegúrese de que el editor de texto utiliza espacios para la sangría, no tabuladores.

```
/etc/pki/tls/certs/server.key
```

Crea el archivo de clave privada en la instancia. Sustituya *private key contents* por el contenido de la clave privada usada para crear la solicitud del certificado o el certificado autofirmado.

```
/opt/elasticbeanstalk/hooks/appdeploy/post/99_start_httpd.sh
```

Crea un script de enlace después de la implementación para reiniciar el servicio `httpd`.

Example `.ebextensions/https-instance.config`

```
files:
  /etc/pki/tls/certs/server.crt:
    mode: "000400"
    owner: root
    group: root
    content: |
      -----BEGIN CERTIFICATE-----
      certificate file contents
      -----END CERTIFICATE-----
```

```
/etc/pki/tls/certs/server.key:
mode: "000400"
owner: root
group: root
content: |
    -----BEGIN RSA PRIVATE KEY-----
    private key contents # See note below.
    -----END RSA PRIVATE KEY-----

/opt/elasticbeanstalk/hooks/appdeploy/post/99_start_httpd.sh:
mode: "000755"
owner: root
group: root
content: |
    #!/usr/bin/env bash
    sudo service httpd restart
```

También debe configurar el servidor proxy del entorno para estar a la escucha en el puerto 443. La siguiente configuración de Apache 2.4 agrega un agente de escucha en el puerto 443. Para obtener más información, consulte [Configuración del servidor proxy de su entorno Tomcat](#).

Example `.ebextensions/httpd/conf.d/ssl.conf`

```
Listen 443
<VirtualHost *:443>
    ServerName server-name
    SSLEngine on
    SSLCertificateFile "/etc/pki/tls/certs/server.crt"
    SSLCertificateKeyFile "/etc/pki/tls/certs/server.key"

    <Proxy *>
        Require all granted
    </Proxy>
    ProxyPass / http://localhost:8080/ retry=0
    ProxyPassReverse / http://localhost:8080/
    ProxyPreserveHost on

    ErrorLog /var/log/httpd/elasticbeanstalk-ssl-error_log

</VirtualHost>
```

Su proveedor de certificados puede incluir certificados intermedios que puede instalar para mejorar la compatibilidad con los clientes móviles. Configure Apache con un paquete de entidades de certificados intermedios (CA) añadiendo lo siguiente a su archivo de configuración de SSL (consulte [Ampliación y anulación de la configuración predeterminada de Apache: AMI de Amazon Linux \(AL1\)](#) para conocer la ubicación):

- En el contenido del archivo `ssl.conf`, especifique el archivo de cadena:

```
SSLCertificateKeyFile "/etc/pki/tls/certs/server.key"
SSLCertificateChainFile "/etc/pki/tls/certs/gd_bundle.crt"
SSLCipherSuite      EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH
```

- Añada una nueva entrada a la clave `files` con el contenido de los certificados intermedios:

```
files:
  /etc/pki/tls/certs/gd_bundle.crt:
    mode: "000400"
    owner: root
    group: root
    content: |
      -----BEGIN CERTIFICATE-----
      First intermediate certificate
      -----END CERTIFICATE-----
      -----BEGIN CERTIFICATE-----
      Second intermediate certificate
      -----END CERTIFICATE-----
```

Note

Evite confirmar un archivo de configuración que contenga su clave privada en el control de código fuente. Una vez comprobada la configuración y confirmado su funcionamiento, almacene la clave privada en Amazon S3 y modifique la configuración para descargarla durante la implementación. Para obtener instrucciones, consulte [Almacenamiento seguro de claves privadas en Amazon S3](#).

En un entorno de instancia única, también debe modificar el grupo de seguridad de la instancia para permitir el tráfico en el puerto 443. El siguiente archivo de configuración recupera el ID del grupo de seguridad con una [función](#) de AWS CloudFormation y le agrega una regla.

Example .ebextensions/https-instance-single.config

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      CidrIp: 0.0.0.0/0
```

En un entorno de balanceador de carga, el balanceador de carga se configura [para el paso a través del tráfico seguro](#) o [para descifrar y volver a cifrar](#) en el cifrado integral.

Terminación de HTTPS en instancias de Amazon EC2 que ejecutan .NET Core en Linux

Para los tipos de contenedores .NET Core en Linux, habilite HTTPS con un [archivo de configuración](#) .ebextensions y un archivo de configuración nginx que configura el servidor nginx para usar HTTPS.

Añada el siguiente fragmento a su archivo de configuración, sustituyendo los marcadores de posición de certificado y clave privada tal como se indica, y guárdelo en el directorio .ebextensions. El archivo de configuración realiza las siguientes tareas:

- La clave files crea los siguientes archivos en la instancia:

```
/etc/pki/tls/certs/server.crt
```

Crea el archivo de certificado en la instancia. Sustituya *certificate file contents* por el contenido del certificado.

Note

YAML usa la sangría uniforme. Utilice el mismo nivel de sangría cuando sustituya el contenido en el archivo de configuración de ejemplo y asegúrese de que el editor de texto utiliza espacios para la sangría, no tabuladores.

Si tiene certificados intermedios, inclúyalos en `server.crt` después del certificado del sitio.

```

-----BEGIN CERTIFICATE-----
certificate file contents
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
first intermediate certificate
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
second intermediate certificate
-----END CERTIFICATE-----

```

/etc/pki/tls/certs/server.key

Crea el archivo de clave privada en la instancia. Sustituya *private key contents* por el contenido de la clave privada usada para crear la solicitud del certificado o el certificado autofirmado.

- La clave `container_commands` reinicia el servidor nginx una vez que está todo configurado para que el servidor cargue el archivo de configuración de nginx.

Example `.ebextensions/https-instance.config`

```

files:
  /etc/pki/tls/certs/server.crt:
    content: |
      -----BEGIN CERTIFICATE-----
      certificate file contents
      -----END CERTIFICATE-----

  /etc/pki/tls/certs/server.key:
    content: |
      -----BEGIN RSA PRIVATE KEY-----
      private key contents # See note below.
      -----END RSA PRIVATE KEY-----

container_commands:
  01restart_nginx:
    command: "systemctl restart nginx"

```


Note

Evite confirmar un archivo de configuración que contenga su clave privada en el control de código fuente. Una vez comprobada la configuración y confirmado su funcionamiento, almacene la clave privada en Amazon S3 y modifique la configuración para descargarla durante la implementación. Para obtener instrucciones, consulte [Almacenamiento seguro de claves privadas en Amazon S3](#).

Incluya lo siguiente en un archivo con la extensión `.conf` en el directorio `.platform/nginx/conf.d/` del paquete de código fuente (por ejemplo, `.platform/nginx/conf.d/https.conf`). Sustituya `app_port` por el número de puerto en el que su aplicación está a la escucha. Este ejemplo configura el servidor nginx para ponerlo a la escucha en el puerto 443 usando SSL. Para obtener más información acerca de estos archivos de configuración en la plataforma .NET Core en Linux, consulte [the section called “Servidor proxy”](#).

Example `.platform/nginx/conf.d/https.conf`

```
# HTTPS server

server {
    listen      443 ssl;
    server_name localhost;

    ssl_certificate      /etc/pki/tls/certs/server.crt;
    ssl_certificate_key  /etc/pki/tls/certs/server.key;

    ssl_session_timeout 5m;

    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
    ssl_prefer_server_ciphers on;

    location / {
        proxy_pass http://localhost:app_port;
        proxy_set_header    Connection "";
        proxy_http_version  1.1;
        proxy_set_header    Host      $host;
        proxy_set_header    X-Real-IP $remote_addr;
        proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header    X-Forwarded-Proto https;
    }
}
```

```
}
```

En un entorno de instancia única, también debe modificar el grupo de seguridad de la instancia para permitir el tráfico en el puerto 443. El siguiente archivo de configuración recupera el ID del grupo de seguridad con una [función](#) de AWS CloudFormation y le agrega una regla.

Example .ebextensions/https-instance-single.config

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      CidrIp: 0.0.0.0/0
```

En un entorno de balanceador de carga, el balanceador de carga se configura [para el paso a través del tráfico seguro](#) o [para descifrar y volver a cifrar](#) en el cifrado integral.

Terminación de HTTPS en instancias de Amazon EC2 que ejecutan .NET

El siguiente [archivo de configuración](#) crea y ejecuta un PowerShell script de Windows que realiza las siguientes tareas:

- Comprueba si hay un certificado HTTPS existente vinculado al puerto 443.
- Obtiene el [certificado PFX](#) de un bucket de Amazon S3.

Note

Añada una AmazonS3ReadOnlyAccess política `aws-elasticbeanstalk-ec2-role` para acceder al certificado SSL del bucket de Amazon S3.

- Obtiene la contraseña de AWS Secrets Manager.

Note

Agregue una declaración `aws-elasticbeanstalk-ec2-role` que permita la `secretsmanager:GetSecretValue` acción del secreto que contiene la contraseña del certificado

- Instala el certificado.
- Enlaza el certificado al puerto 443.

Note

Para quitar el punto de enlace HTTP (puerto 80), incluya el comando `Remove-WebBinding` en la sección `Remove the HTTP binding` (Quitar la vinculación con HTTP) del ejemplo.

Example `.ebextensions/ .config https-instance-dotnet`

```
files:
  "C:\\certs\\install-cert.ps1":
    content: |
      import-module webadministration
      ## Settings - replace the following values with your own
      $bucket = "DOC-EXAMPLE-BUCKET" ## S3 bucket name
      $certkey = "example.com.pfx" ## S3 object key for your PFX certificate
      $secretname = "example_secret" ## AWS Secrets Manager name for a secret that
      contains the certificate's password
      ##

      # Set variables
      $certfile = "C:\cert.pfx"
      $pwd = Get-SECSecretValue -SecretId $secretname | select -expand SecretString

      # Clean up existing binding
      if ( Get-WebBinding "Default Web Site" -Port 443 ) {
        Echo "Removing WebBinding"
        Remove-WebBinding -Name "Default Web Site" -BindingInformation *:443:
      }
      if ( Get-Item -path IIS:\SslBindings\0.0.0.0!443 ) {
        Echo "Deregistering WebBinding from IIS"
```

```

    Remove-Item -path IIS:\SslBindings\0.0.0.0!443
  }

  # Download certificate from S3
  Read-S3Object -BucketName $bucket -Key $certkey -File $certfile

  # Install certificate
  Echo "Installing cert..."
  $securepwd = ConvertTo-SecureString -String $pwd -Force -AsPlainText
  $cert = Import-PfxCertificate -FilePath $certfile cert:\localMachine\my -Password
$securepwd

  # Create site binding
  Echo "Creating and registering WebBinding"
  New-WebBinding -Name "Default Web Site" -IP "*" -Port 443 -Protocol https
  New-Item -path IIS:\SslBindings\0.0.0.0!443 -value $cert -Force

  ## Remove the HTTP binding
  ## (optional) Uncomment the following line to unbind port 80
  # Remove-WebBinding -Name "Default Web Site" -BindingInformation *:80:
  ##

  # Update firewall
  netsh advfirewall firewall add rule name="Open port 443" protocol=TCP
localport=443 action=allow dir=OUT

commands:
  00_install_ssl:
    command: powershell -NoProfile -ExecutionPolicy Bypass -file C:\\certs\\install-
cert.ps1

```

En un entorno de instancia única, también debe modificar el grupo de seguridad de la instancia para permitir el tráfico en el puerto 443. El siguiente archivo de configuración recupera el ID del grupo de seguridad mediante una AWS CloudFormation [función y le añade una regla](#).

Example .ebextensions/ .config https-instance-single

```

Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp

```

```
ToPort: 443
FromPort: 443
CidrIp: 0.0.0.0/0
```

[Para un entorno con equilibrio de carga, se configura el equilibrador de carga para que el tráfico seguro pase intacto o para que lo descifre y vuelva a cifrar para cifrarlo.](#) end-to-end

Configuración del cifrado integral en un entorno de Elastic Beanstalk con balanceo de carga

Es posible que, en el caso de su aplicación, sea suficiente terminar las conexiones seguras en el balanceador de carga y utilizar HTTP en el backend. El tráfico de red entre los recursos de AWS no puede ser interceptado por las instancias que no forman parte de la conexión, aunque se ejecuten en la misma cuenta.

Sin embargo, si está desarrollando una aplicación que debe cumplir estrictas normativas externas, es posible que necesite proteger todas las conexiones de red. Puede utilizar la consola de Elastic Beanstalk o [archivos de configuración](#) para que el balanceador de carga del entorno de Elastic Beanstalk se conecte de forma segura a las instancias del backend y cumplir así con estos requisitos. El procedimiento siguiente se centra en los archivos de configuración.

En primer lugar, [agregue un agente de escucha seguro al balanceador de carga](#), si aún no lo ha hecho.

Asimismo, debe configurar las instancias del entorno para que reciban las solicitudes en el puerto seguro y terminen las conexiones HTTPS. La configuración varía en función de la plataforma. Para obtener instrucciones, consulte [Configurar su aplicación para terminar las conexiones HTTPS en la instancia](#). Puede utilizar un [certificado autofirmado](#) para las instancias EC2 sin problemas.

A continuación, configure el agente de escucha para que reenvíe el tráfico utilizando HTTPS en el puerto seguro que se utiliza en la aplicación. Utilice uno de los archivos de configuración siguientes, según el tipo de balanceador de carga que su entorno utiliza.

.ebextensions/https-reencrypt-clb.config

Utilice este archivo de configuración con un Classic Load Balancer. Además de configurar el balanceador de carga, el archivo de configuración también cambia la comprobación de estado predeterminada para que utilice el puerto 443 y HTTPS, garantizando así que el balanceador de carga se conecte de forma segura.

```
option_settings:
  aws:elb:listener:443:
    InstancePort: 443
    InstanceProtocol: HTTPS
  aws:elasticbeanstalk:application:
    Application Healthcheck URL: HTTPS:443/
```

.ebextensions/https-reencrypt-alb.config

Utilice este archivo de configuración con un Application Load Balancer.

```
option_settings:
  aws:elbv2:listener:443:
    DefaultProcess: https
    ListenerEnabled: 'true'
    Protocol: HTTPS
  aws:elasticbeanstalk:environment:process:https:
    Port: '443'
    Protocol: HTTPS
```

.ebextensions/https-reencrypt-nlb.config

Utilice este archivo de configuración con un Network Load Balancer.

```
option_settings:
  aws:elbv2:listener:443:
    DefaultProcess: https
    ListenerEnabled: 'true'
  aws:elasticbeanstalk:environment:process:https:
    Port: '443'
```

La opción `DefaultProcess` se denomina de esta forma debido a Application Load Balancers, que pueden tener agentes de escucha no predeterminados en el mismo puerto correspondiente al tráfico a rutas específicas (consulte [Equilibrador de carga de aplicación](#) para obtener más detalles). Para un Network Load Balancer, la opción especifica el único proceso de destino de este agente de escucha.

En este ejemplo, hemos denominado al proceso `https` porque escucha el tráfico seguro (HTTPS). El agente de escucha envía el tráfico al proceso en el puerto designado mediante el protocolo TCP, porque un Network Load Balancer solo funciona con TCP. Esto es correcto, porque el tráfico de red HTTP y HTTPS está implementado sobre TCP.

Note

La CLI de EB y la consola de Elastic Beanstalk aplican los valores recomendados a las opciones anteriores. Debe eliminar estos ajustes si desea usar archivos de configuración para configurarlos. Para obtener más información, consulte [Valores recomendados](#).

En la siguiente tarea, debe modificar el grupo de seguridad del balanceador de carga para permitir el tráfico. En función de cuál sea la [Amazon Virtual Private Cloud](#) (Amazon VPC) en la que lance su entorno (la VPC predeterminada o una VPC personalizada), el grupo de seguridad del balanceador de carga variará. En una VPC predeterminada, Elastic Load Balancing cuenta con un grupo de seguridad predeterminado que pueden utilizar todos los balanceadores de carga. En una Amazon VPC personalizada, Elastic Beanstalk crea un grupo de seguridad para que lo use el balanceador de carga.

Para admitir los dos escenarios, puede crear un grupo de seguridad e indicarle a Elastic Beanstalk que lo use. El archivo de configuración siguiente crea un grupo de seguridad y lo adjunta al balanceador de carga.

.ebextensions/https-lbsecuritygroup.config

```
option_settings:
  # Use the custom security group for the load balancer
  aws:elb:loadbalancer:
    SecurityGroups: ``{ "Ref" : "loadbalancersg" }``
    ManagedSecurityGroup: ``{ "Ref" : "loadbalancersg" }``

Resources:
  loadbalancersg:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: load balancer security group
      VpcId: vpc-#####
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 443
          ToPort: 443
          CidrIp: 0.0.0.0/0
        - IpProtocol: tcp
          FromPort: 80
          ToPort: 80
```

```

    CidrIp: 0.0.0.0/0
  SecurityGroupEgress:
  - IpProtocol: tcp
    FromPort: 80
    ToPort: 80
    CidrIp: 0.0.0.0/0

```

Cambie el texto resaltado por el ID de la VPC predeterminada o personalizada. El ejemplo anterior incluye la entrada y salida a través del puerto 80 para permitir las conexiones HTTP. Puede eliminar estas propiedades si quiere permitir solo conexiones seguras.

Por último, agregue reglas de entrada y salida que permitan la comunicación a través del puerto 443 entre el grupo de seguridad del balanceador de carga y el grupo de seguridad de las instancias.

.ebextensions/https-backendsecurity.config

```

Resources:
  # Add 443-inbound to instance security group (AWSEBSecurityGroup)
  httpsFromLoadBalancerSG:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      SourceSecurityGroupId: {"Fn::GetAtt" : ["loadbalancersg", "GroupId"]}
  # Add 443-outbound to load balancer security group (loadbalancersg)
  httpsToBackendInstances:
    Type: AWS::EC2::SecurityGroupEgress
    Properties:
      GroupId: {"Fn::GetAtt" : ["loadbalancersg", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      DestinationSecurityGroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}

```

Si hace esto en una operación distinta a la creación del grupo de seguridad, podrá restringir los grupos de seguridad de origen y destino sin necesidad de crear una dependencia circular.

Después de haber completado todas las tareas anteriores, el balanceador de carga se conecta a las instancias backend de forma segura mediante HTTPS. Al balanceador de carga no le importa si el

certificado de la instancia es autofirmado o lo ha emitido una autoridad de certificación de confianza; aceptará cualquier certificado que se le presente.

Puede cambiar este comportamiento agregando políticas al balanceador de carga que le indiquen que puede confiar solo en un certificado específico. El siguiente archivo de configuración crea dos políticas. Una política especifica un certificado público y la otra le indica al balanceador de carga que solo debe confiar en ese certificado para las conexiones con el puerto 443 de las instancias.

.ebextensions/https-backendauth.config

```
option_settings:
  # Backend Encryption Policy
  aws:elb:policies:backendencryption:
    PublicKeyPolicyNames: backendkey
    InstancePorts: 443
  # Public Key Policy
  aws:elb:policies:backendkey:
    PublicKey: |
      -----BEGIN CERTIFICATE-----
      #####
      #####
      #####
      #####
      #####
      -----END CERTIFICATE-----
```

Cambie el texto resaltado con el contenido del certificado público de la instancia EC2.

Configuración del balanceador de carga del entorno para el paso a través de TCP

Si no desea que el balanceador de carga de su entorno de AWS Elastic Beanstalk descifre el tráfico HTTPS, puede configurar el agente de escucha seguro para que transmita las solicitudes a instancias del backend tal como llegan.

En primer lugar, [configure las instancias EC2 de su entorno para que terminen las conexiones HTTPS](#). Pruebe la configuración en un entorno de una sola instancia para asegurarse de que todo funciona antes de añadir un balanceador de carga.

Añada un [archivo de configuración](#) a su proyecto para configurar un agente de escucha en el puerto 443 que pase los paquetes TCP tal como llegan al puerto 443 en instancias del backend:

.ebextensions/https-lb-passthrough.config

```
option_settings:
  aws:elb:listener:443:
    ListenerProtocol: TCP
    InstancePort: 443
    InstanceProtocol: TCP
```

En una [Amazon Virtual Private Cloud](#) (Amazon VPC) predeterminada, también deberá añadir una regla al grupo de seguridad de las instancias para permitir el tráfico entrante en el puerto 443 desde el balanceador de carga:

.ebextensions/https-instance-securitygroup.config

```
Resources:
  443inboundfromloadbalancer:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      SourceSecurityGroupName: { "Fn::GetAtt": ["AWSEBLoadBalancer",
"SourceSecurityGroup.GroupName"] }
```

En una VPC personalizada, Elastic Beanstalk actualiza la configuración del grupo de seguridad por usted.

Almacenamiento seguro de claves privadas en Amazon S3

La clave privada que utiliza para firmar el certificado público es privada y no debe confirmarse en el código fuente. Para evitar que las claves privadas se guarden en los archivos de configuración, puede cargarlas en Amazon S3 y configurar Elastic Beanstalk para descargar el archivo de Amazon S3 durante la implementación de la aplicación.

En el siguiente ejemplo, se muestran las secciones [Resources \(Recursos\)](#) y [files \(archivos\)](#) de un [archivo de configuración](#) que descarga un archivo de claves privadas de un bucket de Amazon S3.

Example `.ebextensions/privatekey.config`

```
Resources:
```

```
AWSEBAutoScalingGroup:
  Metadata:
    AWS::CloudFormation::Authentication:
      S3Auth:
        type: "s3"
        buckets: ["elasticbeanstalk-us-west-2-123456789012"]
        roleName:
          "Fn::GetOptionSetting":
            Namespace: "aws:autoscaling:launchconfiguration"
            OptionName: "IamInstanceProfile"
            DefaultValue: "aws-elasticbeanstalk-ec2-role"
  files:
    # Private key
    "/etc/pki/tls/certs/server.key":
      mode: "000400"
      owner: root
      group: root
      authentication: "S3Auth"
      source: https://elasticbeanstalk-us-west-2-123456789012.s3.us-west-2.amazonaws.com/
server.key
```

En el ejemplo, sustituya el nombre y la URL del bucket por sus propios valores. La primera entrada de este archivo agrega un método de autenticación llamado S3Auth a los metadatos del grupo de Auto Scaling del entorno. Si ha configurado un [perfil de instancia](#) personalizado para su entorno, se utilizará dicho perfil; de lo contrario, se aplicará el valor predeterminado: `aws-elasticbeanstalk-ec2-role`. El perfil de instancia predeterminado tiene permiso para leer el contenido del bucket de almacenamiento de Elastic Beanstalk. Si utiliza otro bucket, [agregue permisos al perfil de instancia](#).

La segunda entrada utiliza el método de autenticación S3Auth para descargar la clave privada de la URL especificada y guardarla en `/etc/pki/tls/certs/server.key`. El servidor proxy puede leer la clave privada en esta ubicación para [terminar las conexiones HTTPS en la instancia](#).

El perfil de instancia asignado a las instancias EC2 del entorno deben tener permiso para leer el objeto de clave en el bucket especificado. [Compruebe que el perfil de instancia tiene permiso](#) para leer el objeto en IAM y que los permisos del bucket y del objeto no prohíben el perfil de instancia.

Para ver los permisos de un bucket

1. Abra la [consola de administración de Amazon S3](#).
2. Elija un bucket.
3. Seleccione Properties y después Permissions.

4. Compruebe que su cuenta es beneficiaria del bucket con permisos de lectura.
5. Si hay una política de bucket asociada, seleccione Bucket policy (Política de bucket) para ver los permisos asignados al bucket.

Configuración del redireccionamiento de HTTP a HTTPS

En [Configuración de HTTPS para su entorno de Elastic Beanstalk](#) y sus temas, analizamos la configuración del entorno de Elastic Beanstalk para que utilice HTTPS a fin de garantizar el cifrado del tráfico en su aplicación. En este tema se describe cómo gestionar con elegancia el tráfico HTTP a la aplicación si los usuarios finales todavía la inician. Para ello, configure el redireccionamiento HTTP a HTTPS, que a veces se denomina forzar HTTPS.

Para configurar el redireccionamiento, primero debe configurar el entorno para gestionar el tráfico HTTPS. A continuación, redirige el tráfico HTTP a HTTPS. Estos dos pasos se describen en las subsecciones siguientes.

Configurar el entorno para controlar el tráfico HTTPS

En función de la configuración de balanceo de carga del entorno, realice una de las siguientes acciones:

- Load-balanced environment (Entorno con balanceo de carga): [configure el balanceador de carga para terminar las conexiones HTTPS](#).
- Single-instance environment (Entorno de una sola instancia): [configure la aplicación para terminar las conexiones HTTPS en la instancia](#). Esta configuración depende de la plataforma de su entorno.

Redirigir el tráfico HTTP a HTTPS

Puede configurar los servidores web en las instancias del entorno o el Application Load Balancer del entorno para redirigir el tráfico HTTP a HTTPS. Realice alguna de las siguientes acciones:

- Configure instance web servers (Configurar servidores web de instancias): este método funciona en cualquier entorno de servidor web. Configure los servidores web en sus instancias de Amazon Elastic Compute Cloud (Amazon EC2) para que respondan al tráfico HTTP con un estado de respuesta de redirección HTTP. Esta configuración depende de la plataforma de su entorno. Busque la carpeta para su plataforma en la colección [https-redirect](#) en GitHub y utilice el archivo de configuración de ejemplo en esa carpeta.

Si su entorno utiliza las [comprobaciones de estado de Elastic Load Balancing](#), el balanceador de carga espera que una instancia en buen estado responda a los mensajes de comprobación de estado HTTP con respuestas HTTP 200 (OK). Por lo tanto, el servidor web no debería redireccionar estos mensajes a HTTPS. Los archivos de configuración de ejemplo en [https-redirect](#) gestionan este requisito correctamente.

- Configure load balancer (Configurar balanceador de carga): este método funciona si tiene un entorno con balanceo de carga que utiliza un [Application Load Balancer](#). El Application Load Balancer puede enviar respuestas de redirección a medida que entra el tráfico HTTP. En este caso, no es necesario configurar la redirección en las instancias de su entorno. Tenemos dos archivos de configuración de ejemplo en GitHub que muestran cómo configurar el Application Load Balancer para la redirección. El archivo de configuración [alb-http-to-https-redirection-full.config](#) crea un agente de escucha HTTPS en el puerto 443 y modifica el agente de escucha predeterminado del puerto 80 para redirigir el tráfico HTTP entrante a HTTPS. El archivo de configuración [alb-http-to-https-redirection.config](#) espera que se defina el agente de escucha 443 (puede utilizar espacios de nombres de configuración estándar de Elastic Beanstalk o la consola de Elastic Beanstalk). Luego se encarga de modificar el listener del puerto 80 para la redirección.

Monitorización de un entorno

Cuando ejecuta un sitio web de producción, es importante saber que su aplicación está disponible y responde a las solicitudes. Para ayudarle a monitorear la capacidad de respuesta y el estado de su aplicación, Elastic Beanstalk proporciona características que le permiten monitorear las estadísticas de la aplicación y crear alertas que se activen cuando se superen los umbrales.

Temas

- [Monitoreo del estado del entorno en la consola de administración de AWS](#)
- [Informes de estado básicos](#)
- [Informes y monitorización de estado mejorados](#)
- [Administrar alarmas](#)
- [Visualización del historial de cambios de un entorno de Elastic Beanstalk](#)
- [Visualización de la transmisión de eventos de un entorno de Elastic Beanstalk](#)
- [Listado y conexión a instancias de servidor](#)
- [Visualización de registros de instancias de Amazon EC2 en su entorno de Elastic Beanstalk](#)

Monitoreo del estado del entorno en la consola de administración de AWS

Puede consultar información sobre las operaciones de la aplicación en la consola de Elastic Beanstalk. En la consola se muestra el estado del entorno y de la aplicación de un solo vistazo. En la página Environments (Entornos) de la consola y en la página de cada aplicación, los entornos de la lista están codificados por colores para indicar el estado.

Para supervisar un entorno en la consola de Elastic Beanstalk

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, seleccione Monitoring (Monitoreo).

En la página de monitorización, se muestran estadísticas generales sobre el entorno, como el uso de CPU y la latencia media. Además de estadísticas generales, puede ver gráficos de monitorización que representan el uso de los recursos a lo largo del tiempo. Puede hacer clic en cualquiera de los gráficos para ver información más detallada.

Note

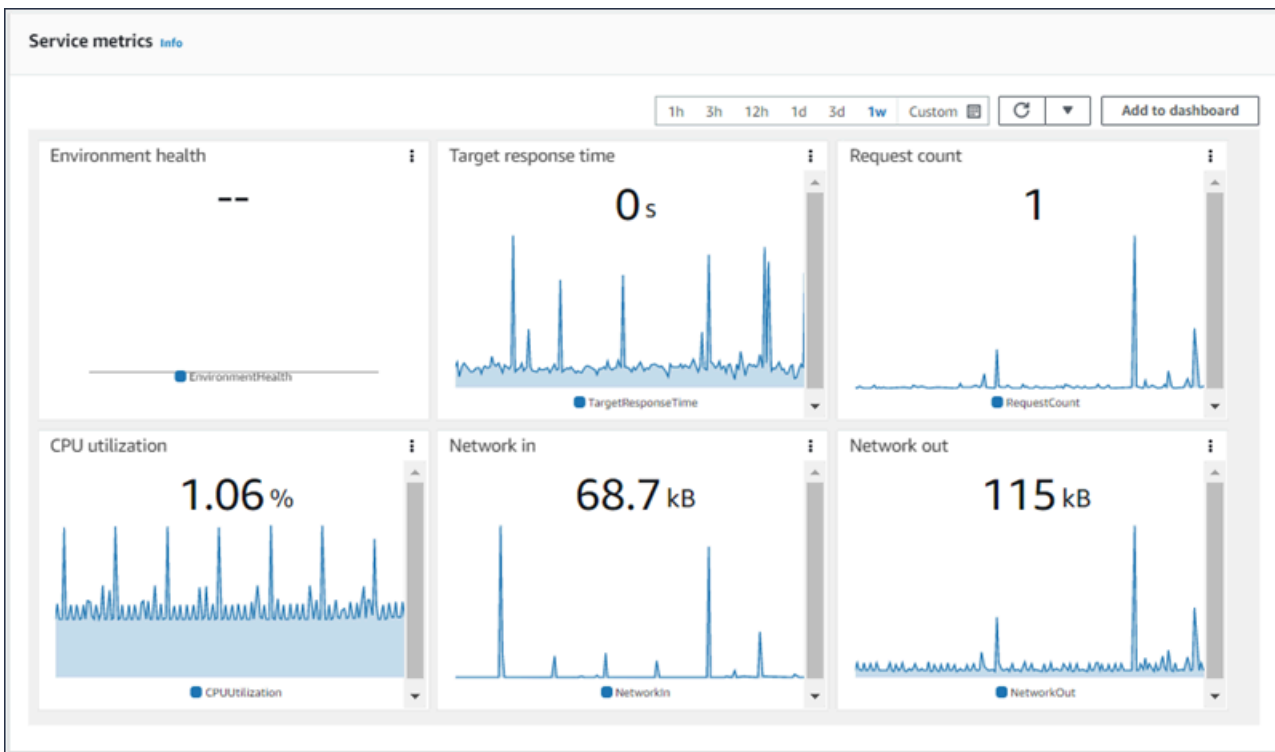
De manera predeterminada, solo están habilitadas las métricas básicas de CloudWatch, que devuelven datos en periodos de cinco minutos. Puede habilitar métricas de CloudWatch más detalladas de un minuto de duración modificando la configuración del entorno.

Gráficos de monitorización

La página Monitorización muestra información general sobre el estado del entorno. Esta información incluye el conjunto predeterminado de métricas que proporcionan Elastic Load Balancing y Amazon EC2, así como gráficos que muestran cómo ha cambiado el estado del entorno a lo largo del tiempo.

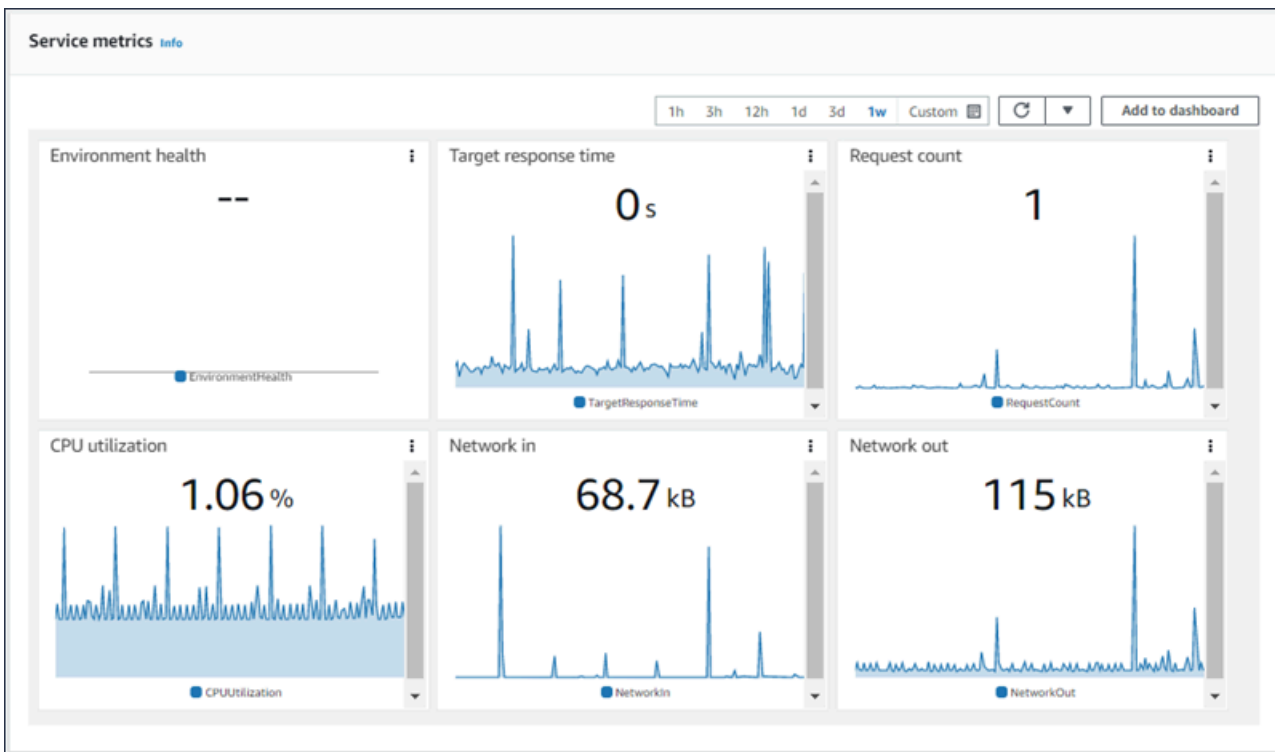
La barra situada por encima de los gráficos proporciona una variedad de intervalos de tiempo que se pueden seleccionar. Por ejemplo, seleccione 1s para mostrar la información que abarca la última semana. O seleccione 3h para mostrar la información de las últimas tres horas.

Para una mayor variedad de selecciones de intervalos de tiempo, elija Personalizado. Desde aquí tiene dos opciones de rango: Absoluto o Relativo. La opción Absoluto le permite especificar un intervalo de fechas específico, p. ej., del 1 de enero de 2023 al 30 de junio de 2023. La opción Relativo permite seleccionar un número entero con una unidad de tiempo específica: minutos, horas, días, semanas o meses. Los ejemplos incluyen 10 horas, 10 días y 10 meses.



Personalización de la consola de monitorización

Para crear y ver métricas personalizadas, debe utilizar Amazon CloudWatch. CloudWatch puede crear paneles personalizados para monitorear sus recursos en una sola vista. Seleccione Agregar al panel para ir a la consola de Amazon CloudWatch desde la página Monitorización. Amazon CloudWatch le ofrece la opción de crear un panel nuevo o seleccionar uno ya existente. Para obtener más información, consulte [Uso de los paneles de Amazon CloudWatch](#) en la Guía del usuario de Amazon CloudWatch.



Las métricas de [Elastic Load Balancing](#) y [Amazon EC2](#) están habilitadas para todos los entornos.

Con [estado mejorado](#), se habilita la métrica EnvironmentHealth y se agrega un gráfico automáticamente a la consola de supervisión. El sistema de estado avanzado también agrega la [página Health](#) a la consola de administración. Para ver una lista de las métricas de estado mejorado disponibles, consulte [Publicación de métricas personalizadas de un entorno en Amazon CloudWatch](#).

Informes de estado básicos

AWS Elastic Beanstalk utiliza información de varias fuentes para determinar si su entorno está disponible y procesa las solicitudes de Internet. El estado de un entorno está representado por uno de los cuatro colores y se muestra en la página de [información general del entorno de la consola](#) de Elastic Beanstalk. También está disponible en la [DescribeEnvironmentsAPI](#) y mediante llamadas eb status con la [CLI de EB](#).

Antes de que se publicara la versión 2 de las versiones de plataforma de Linux, el único sistema de informes de estado era el básico. El sistema básico de informes de estado proporciona información sobre el estado de las instancias en un entorno de Elastic Beanstalk basada en comprobaciones de estado realizadas por Elastic Load Balancing para entornos con equilibrio de carga, o Amazon Elastic Compute Cloud para entornos de instancia única.

Además de comprobar el estado de las instancias EC2, Elastic Beanstalk supervisa el resto de recursos del entorno y notifica los recursos que faltan o tienen una configuración incorrecta y que pueden ocasionar que el entorno deje de estar disponible para los usuarios.

Las métricas recopiladas por los recursos de su entorno se publican CloudWatch en Amazon en intervalos de cinco minutos. Esto incluye las métricas del sistema operativo de EC2 y las métricas de solicitudes de Elastic Load Balancing. Puede ver los gráficos basados en estas CloudWatch métricas en la [página de supervisión](#) de la consola del entorno. En los informes básicos, no se utilizan estas métricas para determinar el estado del entorno.

Temas

- [Colores de estado](#)
- [Comprobaciones de estado Elastic Load Balancing](#)
- [Comprobaciones de estado de entornos de la capa de trabajo y de una sola instancia](#)
- [Comprobaciones adicionales](#)
- [CloudWatch Métricas de Amazon](#)

Colores de estado

Elastic Beanstalk indica el estado de un entorno de servidor web en función de cómo la aplicación que se ejecuta en él responde a la comprobación de estado. Elastic Beanstalk utiliza cuatro colores para describir el estado, tal y como se muestra en la siguiente tabla:

Color	Descripción
Gris	El entorno se está actualizando.
Green	El entorno ha superado la comprobación de estado más reciente. Al menos una instancia del entorno está disponible y atendiendo solicitudes.
Yellow	Su entorno no ha superado una o varias comprobaciones de estado. Algunas solicitudes del entorno no se han superado.
Rojo	El entorno no ha superado tres o más comprobaciones de estado o un recurso de entorno ha dejado de estar disponible. Las solicitudes se consideran erróneas.

Estas descripciones solo se aplican a los entornos que utilizan informes de estado básicos. Consulte [Colores y estados](#) para obtener información sobre los informes de estado avanzados.

Comprobaciones de estado Elastic Load Balancing

En un entorno con balanceo de carga, Elastic Load Balancing envía una solicitud a cada instancia del entorno cada 10 segundos para confirmar que se encuentran en buen estado. De forma predeterminada, el balanceador de carga está configurado para abrir una conexión TCP en el puerto 80. Si la instancia reconoce la conexión, se considera que su estado es correcto.

Si lo desea, puede anular este ajuste especificando un recurso existente en la aplicación. Si especifica una ruta; por ejemplo, `/health`, la URL de comprobación de estado se establece en `HTTP:80/health`. La URL de comprobación de estado debe establecerse en una ruta que siempre esté atendida por la aplicación. Si se establece una página estática atendida o almacenada en caché por el servidor web que hay delante de la aplicación, las comprobaciones de estado no detectarán problemas con el servidor de aplicaciones ni el contenedor web. Para obtener instrucciones sobre la modificación de la URL de comprobación de estado, consulte [Comprobación de estado](#).

Si hay una URL de comprobación de estado configurada, Elastic Load Balancing espera que la solicitud GET que envía devuelva como respuesta `200 OK`. La aplicación no superará la comprobación de estado si no responde en un plazo de 5 segundos o si responde con cualquier otro código de estado HTTP. Después de 5 errores consecutivos de comprobación de estado, Elastic Load Balancing elimina la instancia fuera de servicio.

Para obtener más información acerca de las comprobaciones de estado de Elastic Load Balancing, consulte [Health Check \(Comprobación de estado\)](#) en la Guía de usuario de Elastic Load Balancing.

Note

Si se configura una URL de comprobación de estado, no se modifica el comportamiento de la comprobación de estado de los grupos de Auto Scaling del entorno. Una instancia que no está en buen estado se elimina del balanceador de carga, pero no se sustituye automáticamente por Amazon EC2 Auto Scaling, a menos que configure Amazon EC2 Auto Scaling para utilizar la comprobación de estado de Elastic Load Balancing como base para reemplazar instancias. Para configurar Amazon EC2 Auto Scaling para reemplazar las instancias que no superan una comprobación de estado de Elastic Load Balancing, consulte [Configuración de comprobaciones de estado de Auto Scaling](#).

Comprobaciones de estado de entornos de la capa de trabajo y de una sola instancia

En un entorno de nivel de trabajo o instancia única, Elastic Beanstalk determina el estado de la instancia mediante la supervisión de su estado de instancia de Amazon EC2. La configuración de mantenimiento de Elastic Load Balancing, incluidas las direcciones URL de comprobación de estado HTTP, no se puede utilizar en estos tipos de entorno.

Para obtener más información sobre las comprobaciones de estado de las instancias de Amazon EC2, consulte [Supervisión de instancias con comprobaciones de estado](#) en la Guía del usuario de Amazon EC2.

Comprobaciones adicionales

Además de las comprobaciones de estado de Elastic Load Balancing, Elastic Beanstalk supervisa los recursos del entorno y cambia el estado a rojo si no se pueden implementar, no están configurados correctamente o dejan de estar disponibles. Estas comprobaciones confirman que:

- El grupo de Auto Scaling del entorno está disponible y tiene al menos una instancia.
- El grupo de seguridad del entorno está disponible y está configurado para permitir el tráfico entrante en el puerto 80.
- El CNAME del entorno existe y apunta al balanceador de carga correcto.
- En un entorno del trabajador, la cola de Amazon Simple Queue Service (Amazon SQS) se sondea al menos una vez cada tres minutos.

CloudWatch Métricas de Amazon

Con los informes de estado básicos, el servicio Elastic Beanstalk no publica ninguna métrica en Amazon CloudWatch. Los recursos de su entorno publican las CloudWatch métricas utilizadas para generar gráficos en la [página de monitoreo](#) de la consola del entorno.

Por ejemplo, EC2 publica las siguientes métricas sobre las instancias del grupo Auto Scaling del entorno:

CPUUtilization

Porcentaje de unidades de computación que no están en uso actualmente.

DiskReadBytes, DiskReadOps, DiskWriteBytes, DiskWriteOps

Número de bytes leídos y escritos, y número de las operaciones de lectura y escritura.

NetworkIn, NetworkOut

Número de bytes enviados y recibidos.

Elastic Load Balancing publica las siguientes métricas sobre el balanceador de carga del entorno:

BackendConnectionErrors

Número de errores de conexión entre el balanceador de carga y las instancias del entorno.

HTTPCode_Backend_2XX, HTTPCode_Backend_4XX

Número de códigos de respuesta correctos (2XX) y de errores de cliente (4XX) generados por las instancias del entorno.

Latency

Número de segundos entre el momento en el que el balanceador de carga transmite una solicitud a una instancia y el momento en que se recibe la respuesta.

RequestCount

Número de solicitudes completadas.

Estas listas no están completas. Para obtener una lista completa de las métricas que se pueden informar para estos recursos, consulta los siguientes temas de la Guía para CloudWatch desarrolladores de Amazon:

Métricas

Espacio de nombres	Tema
AWS::ElasticLoadBalancing::LoadBalancer	Métricas y recursos de Elastic Load Balancing
AWS::AutoScaling::AutoScalingGrupo	Métricas y recursos de Amazon Elastic Compute Cloud
AWS::SQS::Queue	Dimensiones y métricas de Amazon SQS

Espacio de nombres	Tema
AWS::RDS::DBInstance	Dimensiones y métricas de Amazon RDS

Métrica de estado del entorno de trabajo

Solo para los entornos de trabajo, el daemon de SQS publica una métrica personalizada del estado del entorno hasta CloudWatch, donde el valor 1 es verde. Puede revisar los datos de las métricas de CloudWatch estado de su cuenta mediante el ElasticBeanstalk/SQSD espacio de nombres. La dimensión de la métrica es EnvironmentName y el nombre de la métrica es Health. Todas las instancias publican sus métricas en el mismo espacio de nombres.

Si desea habilitar el daemon para publicar las métricas del entorno, el perfil de instancia del entorno debe tener permisos para llamar a `cloudwatch:PutMetricData`. Este permiso está incluido en el perfil de instancia predeterminado. Para obtener más información, consulte [Administración de perfiles de instancia de Elastic Beanstalk](#).

Informes y monitorización de estado mejorados

Los informes de estado mejorados son una característica que puede habilitar en su entorno para permitir que AWS Elastic Beanstalk recopile información adicional acerca de los recursos de su entorno. Elastic Beanstalk analiza la información recopilada para proporcionar una imagen más completa del estado general del entorno y ayudar a identificar los problemas que pueden causar que su aplicación deje de estar disponible.

Además de cambiar el funcionamiento de la codificación por colores del estado, los informes de estado mejorados agregan un descriptor de estado que proporciona un indicador de la gravedad de los problemas detectados cuando un entorno tiene el color amarillo o rojo. Cuando haya más información disponible sobre el estado actual, puede elegir el botón Causes para ver información detallada sobre el estado en la [página Health](#).

Elastic Beanstalk > Environments > GettingStartedApp-env

Elastic Beanstalk is updating your environment.
To cancel this operation select **Abort Current Operation** from the **Actions** dropdown.
[View Events](#)

GettingStartedApp-env [Refresh](#)

GettingStartedApp-env.bx7dx222kw.us-east-2.elasticbeanstalk.com

Health

Info

Causes

Running version

Sample Application

Upload and deploy

Tomcat (64b)

Recent events

Time	Type	Details
2020-01-28 15:16:51 UTC-0800	INFO	Deploying new version to instance(s).
2020-01-28 15:16:47 UTC-0800	INFO	Environment update is starting.
2020-01-28 12:11:17 UTC-0800	INFO	Environment health has transitioned from Pending to Ok. Initialization completed 9

Para proporcionar información detallada sobre el estado de las instancias de Amazon EC2 que se ejecutan en su entorno, Elastic Beanstalk incluye un [agente de estado](#) en la Imagen de Amazon Machine (AMI) para cada versión de plataforma que admite la funcionalidad de informes de estado avanzados. El agente de estado monitoriza los logs del servidor web y las métricas del sistema, y envía esta información al servicio de Elastic Beanstalk. Elastic Beanstalk analiza estas métricas y datos de Elastic Load Balancing y Amazon EC2 Auto Scaling para proporcionar una imagen general del estado de un entorno.

Además de recopilar y presentar información sobre los recursos de su entorno, Elastic Beanstalk monitoriza los recursos de su entorno en relación con varias condiciones de error y proporciona notificaciones que le ayudan a evitar los errores y resolver los problemas de configuración. Entre los [factores que influyen en el estado del entorno](#) se incluyen los resultados de cada solicitud servida por

la aplicación, las métricas del sistema operativo de las instancias y el estado de la implementación más reciente.

Puede ver el estado en tiempo real mediante la página de [información del entorno](#) de la consola de Elastic Beanstalk o el [comando eb health](#) de la [interfaz de línea de comandos de Elastic Beanstalk \(CLI de EB\)](#). Para registrar y realizar un seguimiento del estado del entorno y de las instancias a lo largo del tiempo, puede configurar el entorno de forma que la información recopilada por Elastic Beanstalk en relación con los informes de estado mejorados se publiquen en Amazon CloudWatch como métricas personalizadas. CloudWatch [cobra](#) por las métricas personalizadas que se aplican a todas las métricas distintas de EnvironmentHealth, que es gratuita.

Los informes de estado mejorados requieren una [versión de la plataforma](#) versión 2 o posterior. Para monitorizar los recursos y publicar las métricas, su entorno debe tener un [perfil de instancia y un rol de servicio](#). La plataforma de Multicontainer Docker no incluye un servidor web de forma predeterminada, pero puede utilizarse con informes de estado avanzados si configura el servidor web para que [proporcione los registros en el formato adecuado](#).

Notas sobre la plataforma de Windows

- Esta característica no está disponible en las [versiones de la plataforma Windows Server](#) anteriores a la versión 2 (v2).
- Al habilitar los informes de estado avanzados en un entorno de Windows Server, no cambie [la configuración de registros de IIS](#). Para que la monitorización de estado avanzados funcione correctamente, el registro IIS debe estar configurado con el formato W3C y los destinos de evento de registro ETW event only (Solo evento ETW) o Both log file and ETW event (Tanto archivo de registro como evento ETW).

Además, no deshabilitar o detener el servicio de Windows [agente de estado de Elastic Beanstalk](#) en cualquiera de las instancias de su entorno. Para recopilar y notificar información de estado avanzados en una instancia, este servicio debe estar habilitado y en ejecución.

El estado mejorado requiere que el entorno tenga un perfil de instancia. El perfil de instancia debe tener roles que proporcionen permisos para que las instancias de entorno puedan recopilar información sobre el estado mejorado y enviar notificaciones. La primera vez que cree un entorno con una versión de la plataforma v2 en la consola de Elastic Beanstalk, se le pedirá que cree los roles necesarios y habilite los informes de estado mejorados de forma predeterminada. Siga leyendo

para obtener más información acerca de cómo funcionan los informes de estado mejorados o consulte [Habilitación de informes de estado mejorado de Elastic Beanstalk](#) para empezar a utilizarlos inmediatamente.

Las plataformas de Amazon Linux 2 necesitan perfiles de instancia, por lo que pueden admitir el estado mejorado sin reservas. Cuando crea un entorno con una plataforma de Amazon Linux 2, Elastic Beanstalk siempre permite mejorar el estado. Esto es así independientemente del modo en que cree el entorno, mediante la consola de Elastic Beanstalk, la CLI de EB, la AWS CLI o la API.

Temas

- [El agente de estado de Elastic Beanstalk](#)
- [Factores que determinan el estado de las instancias y del entorno](#)
- [Personalización de reglas de comprobación de estado](#)
- [Roles de estado mejorados](#)
- [Autorización de estado mejorada](#)
- [Eventos de estado mejorados](#)
- [Comportamiento de los informes de estado mejorados durante las actualizaciones, implementaciones y escala](#)
- [Habilitación de informes de estado mejorado de Elastic Beanstalk](#)
- [Monitorización del estado mejorado con la consola de administración del entorno](#)
- [Colores y estados](#)
- [Métricas de la instancia](#)
- [Configuración de las reglas de informes de estado avanzados de un entorno](#)
- [Publicación de métricas personalizadas de un entorno en Amazon CloudWatch](#)
- [Uso de informes de estado mejorados con la API de Elastic Beanstalk](#)
- [Formato de los registros de estado avanzados](#)
- [Notificaciones y solución de problemas](#)

El agente de estado de Elastic Beanstalk

El agente de estado de Elastic Beanstalk es un proceso daemon (o un servicio, en entornos Windows) que se ejecuta en cada Amazon EC2 instance de su entorno para monitorizar las métricas de estado del sistema operativo y la aplicación y registrar los problemas en Elastic Beanstalk. El

agente de estado se incluye en todas las versiones de la plataforma Linux a partir de la versión 2.0 de cada plataforma.

El agente de estado notifica métricas similares a las [publicadas en CloudWatch](#) por Amazon EC2 Auto Scaling y Elastic Load Balancing como parte de los [informes básicos de estado](#), incluidos la carga de CPU, los códigos HTTP y la latencia. El agente de estado, sin embargo, registra las métricas directamente en Elastic Beanstalk, con mayor detalle y más a menudo que los informes de estado básicos.

Para los informes de estado básicos, estas métricas se publican cada cinco minutos y se pueden monitorear con gráficos en la consola de administración del entorno. Con un estado mejorado, el agente de estado de Elastic Beanstalk notifica métricas a Elastic Beanstalk cada 10 segundos. Elastic Beanstalk utiliza las métricas proporcionadas por el agente de estado para determinar el estado de cada instancia del entorno y, en combinación con otros [factores](#), determinar el estado general del entorno.

El estado general del entorno se puede ver en tiempo real en la página general del entorno de la consola de Elastic Beanstalk y se publica en CloudWatch por Elastic Beanstalk cada 60 segundos. Puede ver métricas detalladas registradas por el agente de estado en tiempo real con el comando [eb health](#) de la [CLI de EB](#).

Abonando una cantidad adicional, puede publicar las métricas de nivel de instancia y entorno en CloudWatch cada 60 segundos. Las métricas publicadas en CloudWatch se pueden utilizar posteriormente para crear [gráficos de monitorización](#) en la [consola de administración del entorno](#).

Los informes de estado mejorados solo se cobran si elige publicar métricas de estado mejoradas en CloudWatch. Cuando se utilizan los informes de estado mejorados, puede publicar igualmente las métricas de estado básicas de forma gratuita, aunque no decida publicar métricas de estado mejoradas.

Consulte [Métricas de la instancia](#) para obtener más información sobre las métricas registradas por el agente de estado. Para obtener más información sobre la publicación de métricas de estado mejoradas en CloudWatch, consulte [Publicación de métricas personalizadas de un entorno en Amazon CloudWatch](#).

Factores que determinan el estado de las instancias y del entorno

Además de las comprobaciones del sistema de informes de estado básicos, incluida [Comprobaciones de estado Elastic Load Balancing](#) y el [monitoreo de recursos](#), los informes de

estado avanzados de Elastic Beanstalk recopilan datos adicionales sobre el estado de las instancias del entorno. Esto incluye las métricas del sistema operativo, los registros del servidor y el estado de las operaciones del entorno en curso, como las implementaciones y las actualizaciones. El servicio de informes de estado de Elastic Beanstalk combina la información de todos los orígenes disponibles y la analiza para determinar el estado general del entorno.

Operaciones y comandos

Cuando realiza una operación en su entorno, como implementar una nueva versión de una aplicación, Elastic Beanstalk realiza varios cambios que afectan al estado del entorno.

Por ejemplo, cuando implementa una nueva versión de una aplicación en un entorno que ejecuta varias instancias, es posible que vea mensajes similares a los siguientes cuando monitorice el estado del entorno [con la CLI de EB](#).

```
id          status  cause
Overall    Info    Command is executing on 3 out of 5 instances
i-bb65c145  Pending 91 % of CPU is in use. 24 % in I/O wait
           Performing application deployment (running for 31 seconds)
i-ba65c144  Pending Performing initialization (running for 12 seconds)
i-f6a2d525  Ok      Application deployment completed 23 seconds ago and took 26
seconds
i-e8a2d53b  Pending 94 % of CPU is in use. 52 % in I/O wait
           Performing application deployment (running for 33 seconds)
i-e81cca40  Ok
```

En este ejemplo, el estado general del entorno es Ok y la causa de este estado es que el comando está ejecutando 3 de 5 instancias. Tres de las instancias del entorno tienen el estado Pending, lo que indica que una operación está en curso.

Cuando se completa una operación, Elastic Beanstalk registra información adicional sobre la operación. En el ejemplo, Elastic Beanstalk muestra la siguiente información sobre una instancia que ya se ha actualizado con la nueva versión de la aplicación:

```
i-f6a2d525  Ok      Application deployment completed 23 seconds ago and took 26
seconds
```

La información de estado de la instancia incluye también detalles sobre la implementación más reciente en cada instancia del entorno. Cada instancia registra un ID y un estado de implementación.

El ID de implementación es un entero que aumenta en uno cada vez que implementa una nueva versión de la aplicación o cambia las opciones de configuración de una instancia como las variables de entorno. Puede utilizar la información de implementación para identificar las instancias que ejecutan la versión incorrecta de la aplicación después de una [implementación continua](#) que ha producido un error.

En la columna de causa, Elastic Beanstalk incluye mensajes informativos sobre las operaciones realizadas correctamente y otros estados correctos de varias comprobaciones de estado, pero esta información no se conserva de forma indefinida. Las causas de los estados del entorno incorrectos se mantienen hasta que el entorno vuelve a tener un estado correcto.

Tiempo de espera del comando

Elastic Beanstalk aplica un tiempo de espera del comando desde el momento en que comienza una operación para permitir la transición de una instancia a un estado correcto. Este tiempo de espera se establece en la configuración de la actualización e implementación del entorno (en el espacio de nombres [aws:elasticbeanstalk:command](#)) y su valor predeterminado es de 10 minutos.

Durante las actualizaciones continuas, Elastic Beanstalk aplica un tiempo de espera distinto a cada lote de la operación. Este tiempo de espera se establece como parte de la configuración de las actualizaciones continuas del entorno (en el espacio de nombres [aws:autoscaling:updatepolicy:rollingupdate](#)). Si todas las instancias del lote están en buen estado dentro del tiempo de espera de la implementación, la operación continúa con el siguiente lote. De lo contrario, la operación produce un error.

Note

Si la aplicación no supera las comprobaciones de estado con el estado OK pero está estable en otro nivel, puede definir la opción `HealthCheckSuccessThreshold` en el espacio de nombres [aws:elasticbeanstalk:command namespace](#) para cambiar el nivel en el que Elastic Beanstalk considera que una instancia está en buen estado.

Para que un entorno de servidor web se considere que está en buen estado, cada instancia del entorno o lote debe superar 12 comprobaciones de estado consecutivas en el transcurso de dos minutos. Para el entorno de una capa de trabajo, cada instancia debe superar 18 comprobaciones de estado. Antes de que se agote el tiempo de espera del comando, Elastic Beanstalk no rebaja el estado del entorno cuando no se superan las comprobaciones de estado. Si las instancias del

entorno estén en buen estado durante el tiempo de espera del comando, la operación se realizará correctamente.

Solicitudes HTTP

Cuando no hay ninguna operación en curso en un entorno, la fuente principal de información sobre el estado de las instancias y del entorno son los logs del servidor web de cada instancia. Para determinar el estado de una instancia y el estado general del entorno, Elastic Beanstalk considera el número de solicitudes, el resultado de cada solicitud y la velocidad a la que se resolvió cada solicitud.

En las plataformas basadas en Linux, Elastic Beanstalk lee y analiza los logs del servidor web para obtener información sobre las solicitudes HTTP. En la plataforma Windows Server, Elastic Beanstalk recibe esta información [directamente del servidor web de IIS](#).

Puede que su entorno no tenga un servidor web activo. Por ejemplo, la plataforma Docker Multicontainer no incluye un servidor web. Otras plataformas incluyen un servidor web y su aplicación podría deshabilitarlo. En estos casos, el entorno requiere configuración adicional para proporcionar al [agente de estado de Elastic Beanstalk](#) los registros en el formato que se necesita para transmitir la información de estado al servicio de Elastic Beanstalk. Para obtener más información, consulte [Formato de los registros de estado avanzados](#).

Métricas del sistema operativo

Elastic Beanstalk monitoriza las métricas del sistema operativo registradas por el agente de estado para identificar las instancias que se quedan sin recursos del sistema de forma sistemática.

Consulte [Métricas de la instancia](#) para obtener más información sobre las métricas registradas por el agente de estado.

Personalización de reglas de comprobación de estado

Los informes de estado avanzados de Elastic Beanstalk se basan en un conjunto de reglas para determinar el estado de su entorno. Algunas de estas reglas podrían no ser adecuadas para su aplicación específica. Un caso común es una aplicación que devuelve frecuentes errores HTTP 4xx por diseño. Elastic Beanstalk con una de sus reglas predeterminadas, llega a la conclusión de que algo va mal y cambia el estado del entorno OK (Correcto) a Warning (Advertencia), Degraded (Degradado) o Severe (Grave), en función del porcentaje de errores. Para gestionar este caso correctamente, Elastic Beanstalk le permite configurar esta regla y hacer caso omiso de los errores HTTP 4xx de la aplicación. Para obtener más información, consulte [Configuración de las reglas de informes de estado avanzados de un entorno](#).

Roles de estado mejorados

Los informes de estado mejorados requieren dos roles: un rol de servicio para Elastic Beanstalk y un perfil de instancia para el entorno. La función del servicio permite a Elastic Beanstalk interactuar con otros servicios de AWS en su nombre para recopilar información sobre los recursos del entorno. El perfil de instancia permite que las instancias del entorno escriban registros en Amazon S3 y comuniquen información de estado mejorada al servicio de Elastic Beanstalk.

Cuando se crea un entorno de Elastic Beanstalk mediante la consola de Elastic Beanstalk o la CLI de EB, Elastic Beanstalk crea un rol de servicio predeterminado y adjunta las políticas administradas necesarias a un perfil de instancia predeterminado para su entorno.

Si utiliza la API, un SDK o la AWS CLI para crear entornos, debe crear estos roles con antelación y especificarlos durante la creación del entorno a fin de usar un estado mejorado. Para obtener instrucciones sobre cómo crear los roles adecuados para sus entornos, consulte [Roles de servicio, perfiles de instancia y políticas de usuario](#).

Se recomienda utilizar políticas administradas para el perfil de instancia y el rol de servicio. Las políticas administradas son políticas de AWS Identity and Access Management (IAM) que mantiene Elastic Beanstalk. El uso de políticas administradas garantiza que el entorno tenga todos los permisos necesarios para funcionar correctamente.

Para el perfil de instancia, puede utilizar la política administrada `AWSElasticBeanstalkWebTier` o `AWSElasticBeanstalkWorkerTier`, para un entorno de [nivel de servidor web](#) o de [nivel de trabajo](#) respectivamente. Para obtener más información acerca de estas dos políticas de perfil de instancia administrada, consulte [the section called "Perfiles de instancias"](#).

Autorización de estado mejorada

Las políticas administradas por el perfil de instancias de Elastic Beanstalk contienen permisos para la `elasticbeanstalk:PutInstanceStatistics` acción. Esta acción no forma parte de la API de Elastic Beanstalk. Forma parte de una API diferente que las instancias de entorno utilizan de manera interna para comunicar información de estado mejorada al servicio de Elastic Beanstalk. No se llama directamente a esta API.

Cuando crea un nuevo entorno, la autorización para la acción `elasticbeanstalk:PutInstanceStatistics` está habilitada de forma predeterminada. Con el fin de aumentar la seguridad del entorno y evitar la suplantación de datos de estado en su nombre, recomendamos mantener habilitada la autorización para esta acción. Si utiliza políticas administradas para el perfil de instancias, esta característica está disponible en el nuevo entorno sin necesidad de

configuraciones adicionales. Sin embargo, si utiliza un perfil de instancias personalizado en lugar de una política administrada, el entorno podría mostrar un estado sin datos. Esto ocurre porque las instancias no están autorizadas para la acción que comunica datos de estado mejorados al servicio.

Para autorizar la acción, incluya la siguiente instrucción en su perfil de instancia.

```
{
  "Sid": "ElasticBeanstalkHealthAccess",
  "Action": [
    "elasticbeanstalk:PutInstanceStatistics"
  ],
  "Effect": "Allow",
  "Resource": [
    "arn:aws:elasticbeanstalk:*:*:application/*",
    "arn:aws:elasticbeanstalk:*:*:environment/*"
  ]
}
```

Si no desea utilizar la autorización de estado mejorada en este momento, desactívela configurando la opción `EnhancedHealthAuthEnabled` en el espacio de nombre [the section called “aws:elasticbeanstalk:healthreporting:system”](#) como `false`. Si esta opción está desactivada, los permisos descritos anteriormente no son necesarios. Puede eliminarlos del perfil de instancia para [acceso a los privilegios mínimos](#) en sus aplicaciones y entornos.

Note

Anteriormente, la configuración predeterminada para `EnhancedHealthAuthEnabled` era `false`, lo que dio lugar a que la autorización de la acción `elasticbeanstalk:PutInstanceStatistics` también se deshabilitara de forma predeterminada. A fin de habilitar esta acción para un entorno existente, configure la opción `EnhancedHealthAuthEnabled` en el espacio de nombres [the section called “aws:elasticbeanstalk:healthreporting:system”](#) en `true`. Puede configurar esta opción utilizando una [configuración de opción](#) en un [archivo de configuración](#).

Eventos de estado mejorados

El sistema de informes de estado mejorados genera eventos cuando un entorno cambia de un estado a otro. El siguiente ejemplo muestra los eventos generados por un entorno que pasa por los estados Info, OK y Severe.

Recent events		
Time	Type	Details
2020-01-28 16:06:04 UTC-0800	INFO	Environment health has transitioned from Severe to Ok.
2020-01-28 16:05:04 UTC-0800	INFO	Added instance [i-03280193ba1ba4171] to your environment.
2020-01-28 16:05:04 UTC-0800	WARN	Removed instance [i-0a4a27bbf9994ba5] from your environment due to a EC2 health check failure.
2020-01-28 16:03:04 UTC-0800	WARN	Environment health has transitioned from Ok to Severe. ELB processes are not healthy on all instances. None of the instances are sending data. ELB health is failing or not available for all instances.
2020-01-28 15:19:06 UTC-0800	INFO	Environment health has transitioned from Info to Ok. Application update completed 75 seconds ago and took 22 seconds.

Cuando el entorno pasa a un estado peor, el evento de estado avanzado incluye un mensaje que indica la causa de la transición.

No todos los cambios en el estado a nivel de instancia hacen que Elastic Beanstalk emita un evento. Para evitar falsas alarmas, Elastic Beanstalk solo genera un evento relacionado con el estado si el problema persiste en varias comprobaciones.

La información de estado del entorno en tiempo real, incluido el estado, el color y la causa, está disponible en la página de [información general del entorno](#) de la consola de Elastic Beanstalk y en la [CLI de EB](#). Si asocia la CLI de EB a su entorno y ejecuta el comando [eb health](#), también puede consultar el estado en tiempo real de cada una de las instancias de su entorno.

Comportamiento de los informes de estado mejorados durante las actualizaciones, implementaciones y escala

Si se habilitan los informes de estado mejorados, el comportamiento del entorno durante las actualizaciones de configuración y las implementaciones puede resultar afectado. Elastic Beanstalk no completará un lote de actualizaciones hasta que todas las instancias superen las comprobaciones de estado de forma consistente. Además, ya que los informes de estado avanzados aplican un listón más alto para considerar que están en buen estado y monitorizan otros factores, las instancias que superen la [comprobación de estado ELB](#) de los informes de estado básicos no superarán necesariamente la comprobación de estado de los informes de estado avanzados. Consulte los temas sobre las [actualizaciones de configuración continuas](#) y las [implementaciones continuas](#) para obtener más información acerca de cómo las comprobaciones de estado afectan al proceso de actualización.

Los informes de estado mejorados también pueden poner de manifiesto la necesidad de definir una [URL de comprobación de estado](#) correcta para Elastic Load Balancing. Cuando su entorno se amplía para satisfacer la demanda, las nuevas instancias comienzan a recibir solicitudes en cuanto superan suficientes comprobaciones de estado de ELB. Si no se configura una URL de comprobación de estado, este plazo de tiempo puede ser de tan solo 20 segundos una vez que la instancia es capaz de aceptar una conexión TCP.

Si la aplicación no ha terminado de iniciarse en el momento en que el balanceador de carga considera que tiene un estado lo suficientemente bueno como para recibir tráfico, verá un flujo de solicitudes fallidas, y su entorno empezará a no superar las comprobaciones de estado. Una URL de comprobación de estado que encuentre una ruta a la que sirve su aplicación puede evitar este problema. Las comprobaciones de estado de ELB no se superarán hasta que una solicitud GET a la URL de comprobación de estado devuelva un código de estado 200.

Habilitación de informes de estado mejorado de Elastic Beanstalk

Los nuevos entornos creados con las [versiones de plataforma](#) más recientes incluyen el AWS Elastic Beanstalk [agente de estado](#), que admite informes de estado mejorado. Si crea el entorno en la consola de Elastic Beanstalk o la CLI de EB, el sistema de estado avanzado está habilitado de forma predeterminada. También puede establecer sus preferencias respecto a los informes de estado en el código fuente de la aplicación a través de los [archivos de configuración](#).

Los informes de estado avanzados necesitan un [perfil de instancia](#) y un [rol de servicio](#) con el conjunto de permisos estándar. Cuando se crea un entorno en la consola de Elastic Beanstalk, Elastic Beanstalk crea automáticamente los roles necesarios. Consulte [Introducción a Elastic Beanstalk](#) para obtener instrucciones sobre cómo crear el primer entorno.

Temas

- [Habilitación de informes de estado mejorados mediante la consola de Elastic Beanstalk](#)
- [Activación de informes de estado avanzados con la CLI de EB](#)
- [Activación de informes de estado avanzados con un archivo de configuración](#)

Habilitación de informes de estado mejorados mediante la consola de Elastic Beanstalk

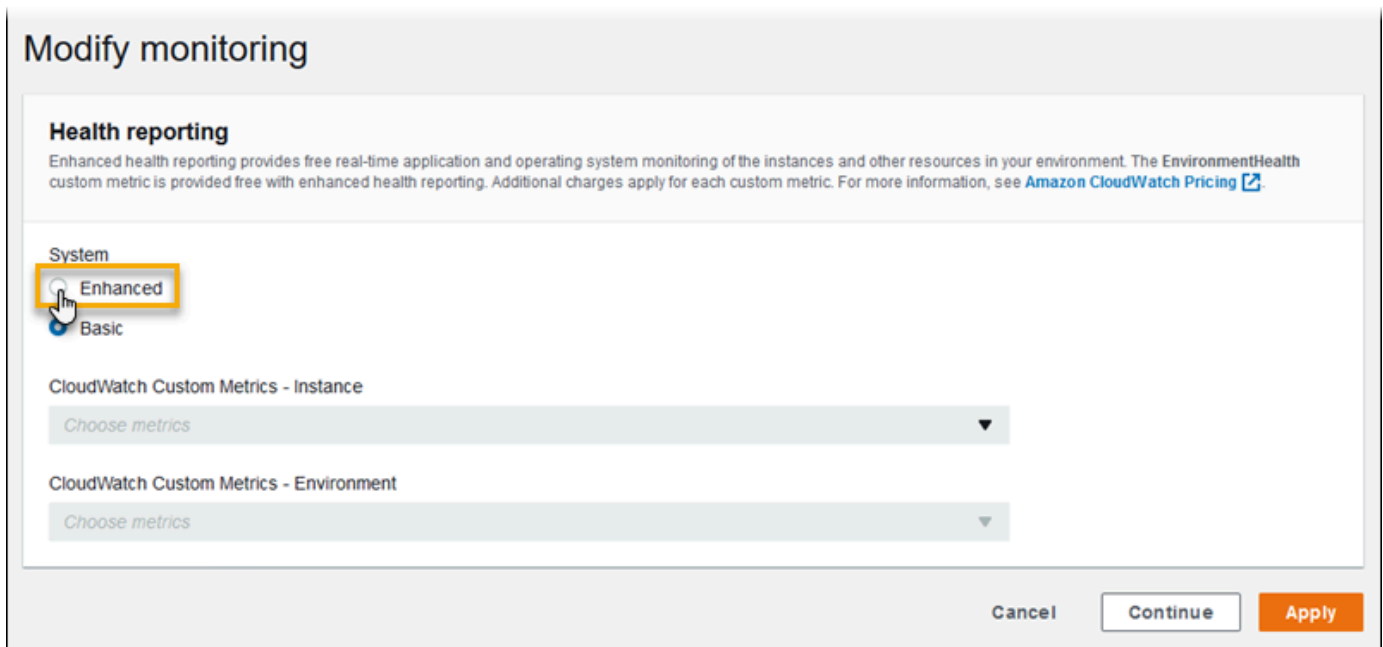
Para habilitar los informes de estado avanzados en un entorno en ejecución con la consola de Elastic Beanstalk

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Monitoring (Monitorización), seleccione Edit (Editar).
5. En Health reporting (Informes de estado), para System (Sistema), seleccione Enhanced (Mejorado).



 Note

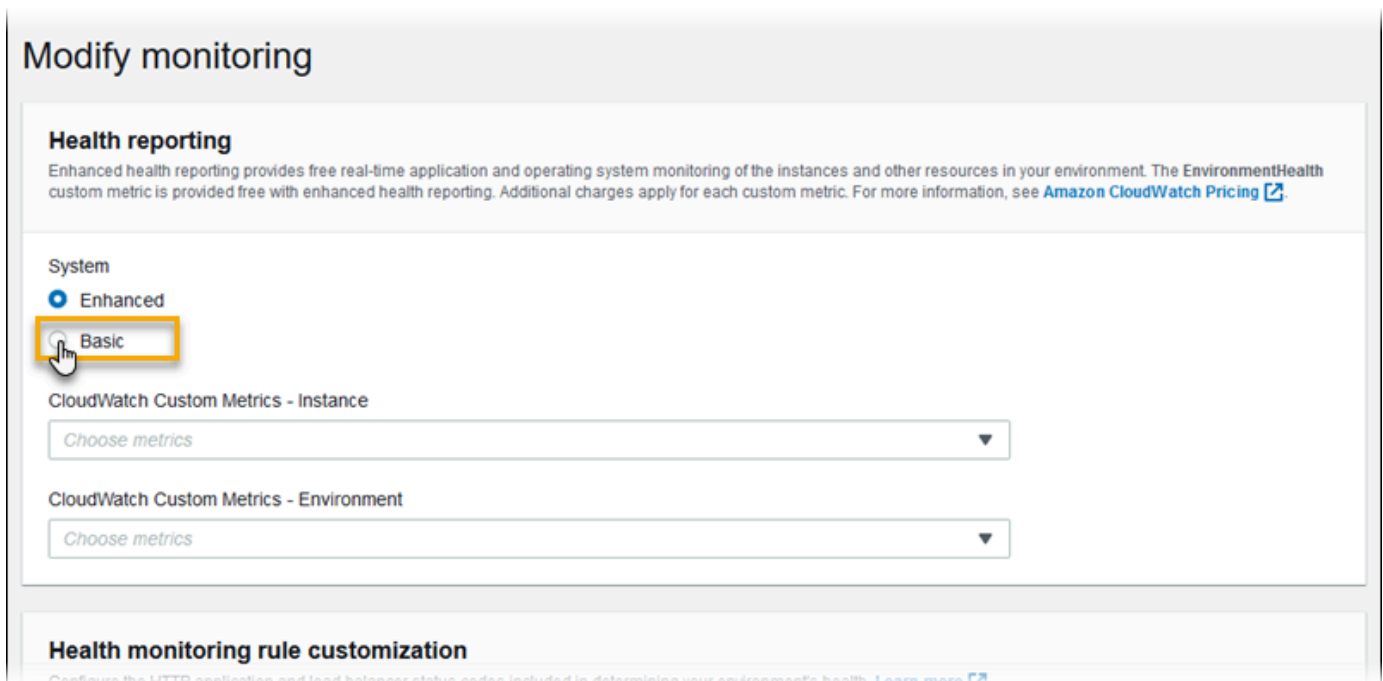
Las opciones para los informes de estado avanzados no aparecerán si utiliza una [plataforma o versión que no es compatible](#).

6. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

La consola de Elastic Beanstalk establece los valores predeterminados de los informes de estado avanzados cuando crea un nuevo entorno con una versión de plataforma número 2 (v2). Puedes desactivar los informes de estado avanzados cambiando la opción correspondiente durante la creación del entorno.

Para deshabilitar los informes de estado avanzados al crear un entorno a través de la consola de Elastic Beanstalk

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. [Cree una aplicación](#) o seleccione una existente.
3. [Cree un entorno](#). En la página Create a new environment (Crear un nuevo entorno), antes de elegir Create environment (Crear entorno), elija Configure more options (Configurar más opciones).
4. En la categoría de configuración Monitoring (Monitorización), seleccione Edit (Editar).
5. En Health reporting (Informes de estado), para System (Sistema), seleccione Basic (Básico).



6. Seleccione Save.

Activación de informes de estado avanzados con la CLI de EB

Si crea un nuevo entorno con el comando `eb create`, la CLI de EB habilitará los informes de estado avanzados de forma predeterminada y aplicará el perfil de instancia y el rol de servicio predeterminados.

Puede especificar un rol de servicio distinto utilizando su nombre con la opción `--service-role`.

Si tiene un entorno de datos que se ejecuta con informes de estado básicos en una versión de plataforma versión 2 y desea cambiar a los informes avanzados, siga estos pasos.

Para habilitar el sistema de estado avanzado en un entorno en ejecución con la [CLI de EB](#)

1. Utilice el comando `eb config` para abrir el archivo de configuración en el editor de texto predeterminado.

```
~/project$ eb config
```

2. Busque el espacio de nombres `aws:elasticbeanstalk:environment` en la sección de configuración. Asegúrese de que el valor de `ServiceRole` no es nulo y que coincide con el nombre del [rol de servicio](#).

```
aws:elasticbeanstalk:environment:  
  EnvironmentType: LoadBalanced  
  ServiceRole: aws-elasticbeanstalk-service-role
```

3. En el espacio de nombres `aws:elasticbeanstalk:healthreporting:system:`, cambie el valor de `SystemType` a **enhanced**.

```
aws:elasticbeanstalk:healthreporting:system:  
  SystemType: enhanced
```

4. Guarde el archivo de configuración y cierre el editor de texto.
5. La CLI de EB inicia una actualización del entorno para aplicar los cambios de configuración. Espere a que la operación se complete o pulse `Ctrl+C` para salir de forma segura.

```
~/project$ eb config  
Printing Status:  
INFO: Environment update is starting.  
INFO: Health reporting type changed to ENHANCED.  
INFO: Updating environment no-role-test's configuration settings.
```

Activación de informes de estado avanzados con un archivo de configuración

Puede habilitar los informes de estado avanzados incluyendo un [archivo de configuración](#) en el paquete de origen. En el ejemplo siguiente, se muestra un archivo de configuración que habilita los informes de estado avanzados y asigna el rol de servicio y el perfil de instancia predeterminados al entorno:

Example `.ebextensions/enhanced-health.config`

```
option_settings:  
  aws:elasticbeanstalk:healthreporting:system:  
    SystemType: enhanced  
  aws:autoscaling:launchconfiguration:  
    IamInstanceProfile: aws-elasticbeanstalk-ec2-role  
  aws:elasticbeanstalk:environment:  
    ServiceRole: aws-elasticbeanstalk-service-role
```

Si creó su propio perfil de instancia o su propio rol de servicio, cambie el texto resaltado por los nombres de estos roles.

Monitorización del estado mejorado con la consola de administración del entorno

Cuando habilita los informes de estado mejorado en AWS Elastic Beanstalk, puede monitorear el estado del entorno en la [consola de administración del entorno](#).

Temas

- [Información general del entorno](#)
- [Página de estado del entorno](#)
- [Página Monitoring \(Monitorización\)](#)

Información general del entorno

La [información general del entorno](#) muestra el [estado](#) del entorno y enumera los eventos que proporcionan información sobre los últimos cambios en el estado.

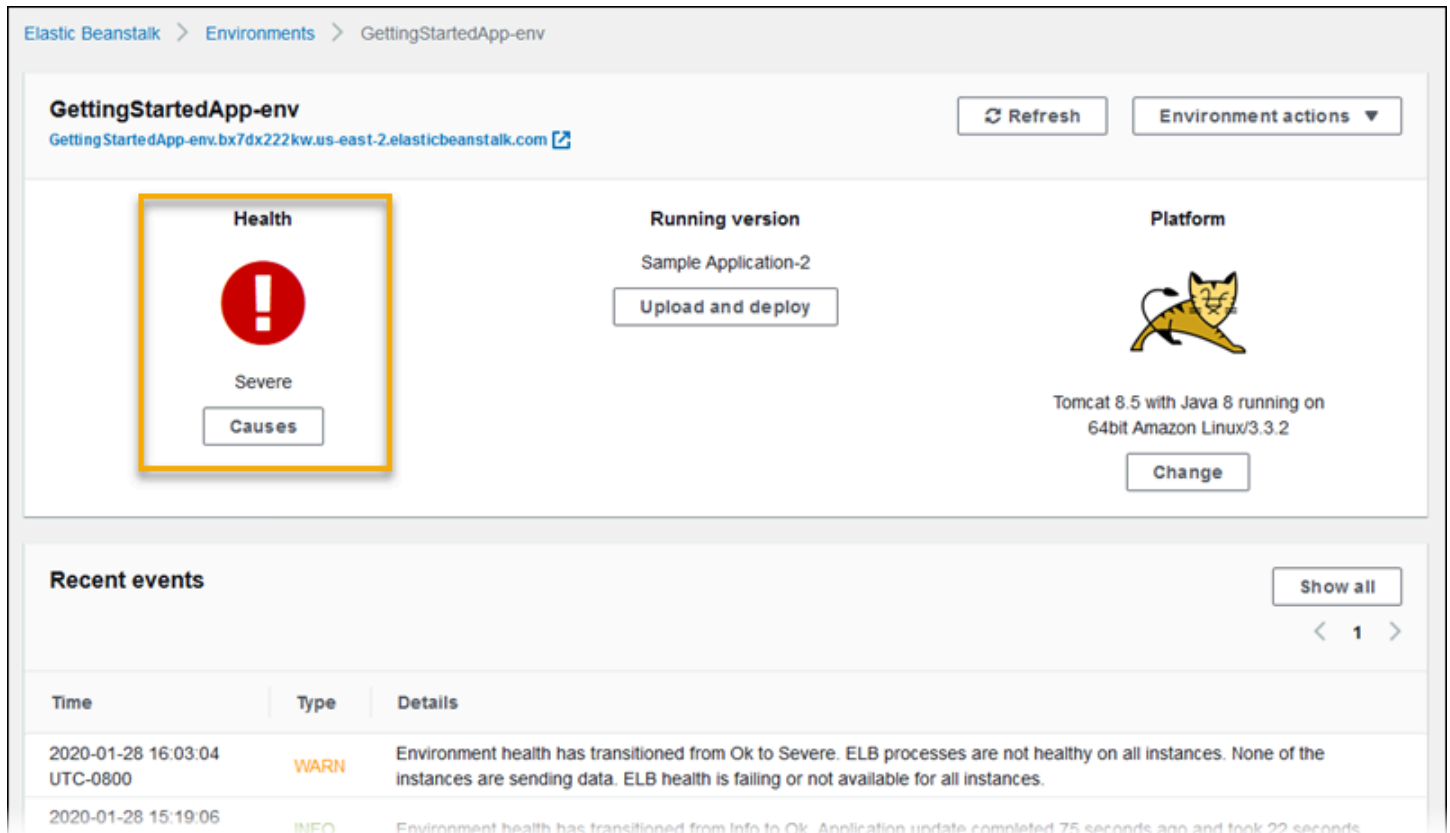
Para ver la información general del entorno

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

Para obtener información detallada sobre el estado del entorno actual, abra la página Health (Estado) eligiendo Causes (Causas). O bien, en el panel de navegación, seleccione Health (Estado).



Página de estado del entorno

La página Health (Estado) muestra el estado, las métricas y las causas del entorno y de cada instancia de Amazon EC2 del entorno.

Note

Elastic Beanstalk muestra la página Health (Estado) solo si ha [habilitado la supervisión de estado mejorado](#) para el entorno.

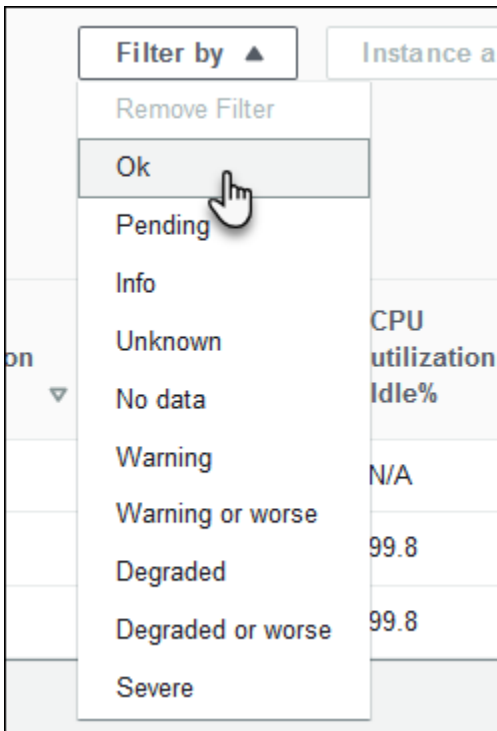
La imagen siguiente muestra la página Health (Estado) de un entorno Linux.

Instance ID	Status	Running	Deployment ID	Requests/sec	2xx Responses	3xx Responses	4xx Responses	5xx Responses	P99 Latency	P90 Latency	P75 Latency	P50 Latency	P10 Latency	Load1 average	Load5 average	CPU utilization User%	CPU utilization Sys%	CPU utilization Idle%	CPU utilization I/O wait%
Overall	Ok	N/A	N/A	0.4	100%	0.0%	0.0%	0.0%	0.002	0.002	0.002	0.002	0.001	N/A	N/A	N/A	N/A	N/A	N/A
i-40227807c4c4a1334	Ok	2 hours	3	0.2	2	0	0	0	0.002	0.002	0.002	0.002	0.002	0.00	0.00	0.0	0.0	99.9	0.0
i-03289193ba1ba4171	Ok	19 days	3	0.2	2	0	0	0	0.001	0.001	0.001	0.001	0.001	0.00	0.00	0.1	0.0	99.9	0.0

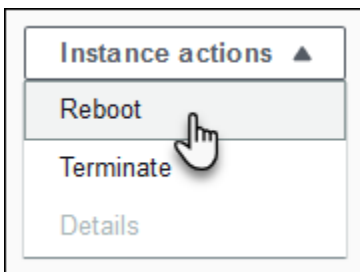
La imagen siguiente muestra la página Health (Estado) de un entorno Windows. Tenga en cuenta que las métricas de CPU son diferentes de las de un entorno Linux.

Instance ID	Status	Running	Deployment ID	Requests/sec	2xx Responses	3xx Responses	4xx Responses	5xx Responses	P99 Latency	P90 Latency	P75 Latency	P50 Latency	P10 Latency	CPU utilization % User Time	CPU utilization % Privileged Time	CPU utilization % Idle Time
Overall	Ok	N/A	N/A	0.2	100%	0.0%	0.0%	0.0%	0.015	0.014	0.011	0.008	0.002	N/A	N/A	N/A
i-04b37b4c983d18af	Ok	20 days	1	0.2	2	0	0	0	0.015	0.014	0.011	0.008	0.002	0.0	0.0	100

En la parte superior de la página puede ver el número total de instancias del entorno, así como el número de instancias por estado. Para mostrar únicamente las instancias que tienen un determinado estado, seleccione Filter By (Filtrar por) y, a continuación, seleccione un [estado](#).



Para reiniciar o terminar una instancia en mal estado, seleccione Instance Actions y después elija Reboot o Terminate.



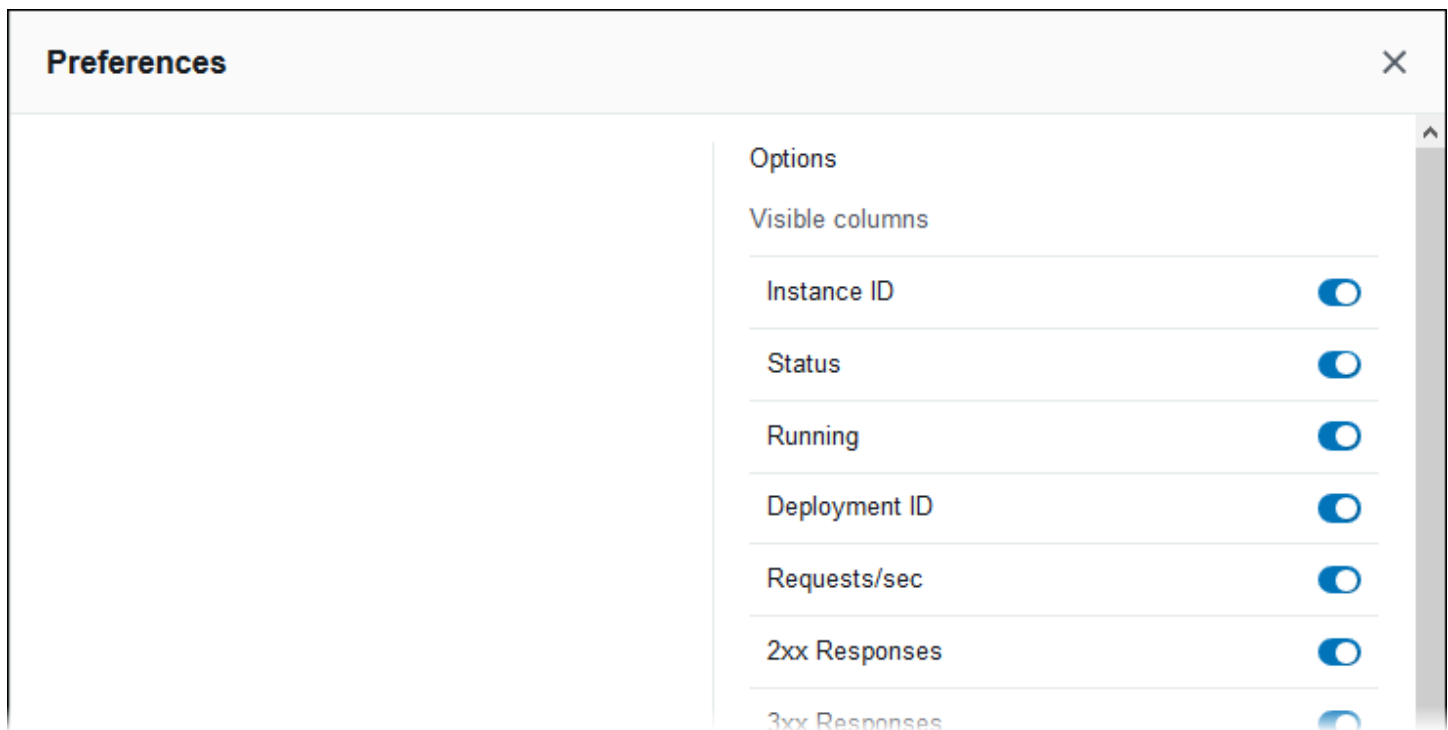
Elastic Beanstalk actualiza la página Estado cada 10 segundos. Registra información sobre el estado del entorno y las instancias.

En cada instancia de Amazon EC2 del entorno, la página muestra el ID y el [estado](#) de la instancia, la cantidad de tiempo transcurrido desde que se lanzó la instancia, el ID de la implementación más reciente ejecutada en la instancia, las respuestas y la latencia de las solicitudes atendidas por la instancia, así como la carga y el uso de la CPU. La fila Overall (Información global) muestra información sobre la respuesta y la latencia promedio de todo el entorno.

La página muestra mucha información en una tabla muy amplia. Para ocultar algunas de las columnas, seleccione



(Preferences (Preferencias)). Marque o desmarque los nombres de las columnas y, a continuación, seleccione Confirm (Confirmar).



Seleccione el Instance ID (ID de instancia) de cualquier instancia para ver más información sobre esta, incluida la zona de disponibilidad y el tipo de instancia.

	Instance ID ▾	Status ▲	Running ▾	Deployment ID ▾	Reque
●	Overall	Ok	N/A	N/A	0.2
○	i-00227807c4c4a1334	Ok	1 day	3	0.1
○	i-03280193ba1ba4171	Ok	20 days	3	0.1



i-00227807c4c4a1334 details

Instance ID: i-00227807c4c4a1334
Instance type: t2.micro
Availability zone: us-east-2b

Seleccione el Deployment ID (ID de implementación) de cualquier instancia para ver información sobre la última [implementación](#) llevada a cabo en la instancia.

	Instance ID ▾	Status ▲	Running ▾	Deployment ID ▾	Reque
●	Overall	Ok	N/A	N/A	0.2
○	i-00227807c4c4a1334	Ok	1 day	3	0.1
○	i-03280193ba1ba4171	Ok	20 days	3	0.1



Deployment details

Deployment ID 3
Version: Sample Application-3
Deployed 1 day ago

La información de implementación incluye lo siguiente:

- **Deployment ID (ID de implementación):** el identificador único de la [implementación](#). Los identificadores de implementación empiezan en 1 y aumentan uno cada vez que se implementa una nueva versión de la aplicación o cambia los ajustes de configuración que afectan al software o al sistema operativo que se ejecutan en las instancias de su entorno.
- **Version (Versión):** la etiqueta de versión del código fuente de la aplicación usada en la implementación.
- **Status (Estado):** el estado de la implementación, que puede ser In Progress, Deployed o Failed.
- **Time (Hora):** para las implementaciones en curso, la hora a la que se inició la implementación. Para las implementaciones completadas, el momento en que finalizó la implementación.

Si [habilita la integración de X-Ray](#) en su entorno y configura su aplicación con el SDK de AWS X-Ray, la página Health (Estado) agrega enlaces a la consola de AWS X-Ray en la fila de información general.

Requests/sec	2xx Responses	3xx Responses	4xx Responses	5xx Responses	P99 Latency	P90 Latency	P75 Latency	P50 Latency	P10 Latency	Location
100%	0.0%	0.0%	0.0%	0.0%	0.002	0.002	0.002	0.002	0.001	N/A
1	0	0	0	0	0.002	0.002	0.002	0.002	0.002	0.00
1	0	0	0	0	0.001	0.001	0.001	0.001	0.001	0.00

Elija un enlace para ver los rastros relacionados con la estadística destacada en la consola de AWS X-Ray.

Página Monitoring (Monitorización)

La página Monitoring (Supervisión) muestra las estadísticas de resumen y los gráficos de las métricas de Amazon CloudWatch personalizadas generados por el sistema de informes de estado mejorados. Consulte [Monitoreo del estado del entorno en la consola de administración de AWS](#) para obtener instrucciones sobre cómo añadir gráficos y estadísticas a esta página.

Colores y estados

Los informes de estado avanzados representan el estado de la instancia y el estado general del entorno con cuatro colores, al igual que en los [informes de estado básicos](#). Los informes de estado

avanzados también cuentan con siete estados, que son descriptores formados por una palabra que proporcionan una indicación más detallada del estado del entorno.

Estado de la instancia y estado del entorno

Cada vez que Elastic Beanstalk ejecuta una comprobación de estado en el entorno, los informes de estado avanzados comprueban el estado de cada instancia del entorno analizando todos [los datos](#) disponibles. Si falla alguna comprobación de nivel inferior, Elastic Beanstalk rebaja la salud de la instancia.

Elastic Beanstalk muestra la información de estado de todo el entorno (color, estado y causa) en la [consola de gestión del entorno](#). Esta información también está disponible en la CLI de EB. Los mensajes sobre el estado y las causas de cada instancia se actualizan cada 10 segundos y están disponibles en la [CLI de EB](#) cuando se consulta el estado con [eb health](#).

Elastic Beanstalk utiliza los cambios en los estados de las instancias para evaluar el estado del entorno, pero no modifica inmediatamente el estado de este. Cuando una instancia no supera una comprobación de estado al menos tres veces en un minuto, Elastic Beanstalk rebaja el estado del entorno. En función del número de instancias del entorno y el problema identificado, una instancia con un estado incorrecto puede hacer que Elastic Beanstalk muestre un mensaje informativo o que cambie el estado del entorno de verde (OK (Bueno)) a amarillo (Warning (Alerta)) o a rojo (Degraded (Degradado) o Severe (Grave)).

OK (verde)

Este estado se muestra cuando:

- Una instancia está superando las comprobaciones de estado y el agente de estado no informa de ningún problema.
- La mayoría de las instancias del entorno están superando las comprobaciones de estado y el agente de estado no informa de ningún problema importante.
- Una instancia está superando las comprobaciones de estado y completando las solicitudes con normalidad.

Ejemplo: el entorno se ha implementado recientemente y está aceptando las solicitudes con normalidad. El cinco por ciento de las solicitudes devuelve errores de la serie 400. La implementación se completó normalmente en cada instancia.

Mensaje (instancia): la implementación de la aplicación se completó hace 23 segundos y tardó 26 segundos.

Warning (amarillo)

Este estado se muestra cuando:

- El agente de estado está notificando un número moderado de solicitudes erróneas u otros problemas en una instancia o en el entorno.
- Hay una operación en curso en una instancia y está tardando mucho.

Ejemplo: una instancia del entorno tiene el estado Severe.

Mensaje (entorno): hay servicios dañados en 1 de 5 instancias.

Degraded (rojo)

El estado se muestra cuando el agente de estado está notificando un número elevado de solicitudes erróneas u otros problemas en una instancia o en el entorno.

Ejemplo: el entorno está en un proceso de escala vertical de hasta 5 instancias.

Mensaje (entorno): el número de instancias activas (4) es inferior al tamaño mínimo del grupo de Auto Scaling (5).

Severe (rojo)

El estado se muestra cuando el agente de estado está notificando un número muy elevado de solicitudes erróneas u otros problemas en una instancia o en el entorno.

Ejemplo: Elastic Beanstalk no puede ponerse en contacto con el balanceador de carga para obtener el estado de la instancia.

Mensaje (entorno): el estado de ELB no es correcto o no está disponible en todas las instancias. Ninguna de las instancias está enviando datos. No se puede asumir el rol "arn:aws:iam::123456789012:role/aws-elasticbeanstalk-service-role". Compruebe que el rol existe y está configurado correctamente.

Mensaje (instancias): hace 37 minutos que el estado de ELB de la instancia no está disponible. No hay datos. La última consulta es de hace 37 minutos.

Info (verde)

Este estado se muestra cuando:

- Hay una operación en curso en una instancia.
- Una operación está en curso en varias instancias de un entorno.

Ejemplo: se está implementando una nueva versión de la aplicación en instancias que están en ejecución.

Mensaje (entorno): el comando se está ejecutando en 3 de 5 instancias.

Mensaje (instancia): implementación de la aplicación en curso (en ejecución durante 3 segundos).

Pending (gris)

Este estado se muestra cuando hay una operación en curso en una instancia dentro del [periodo de espera del comando](#).

Ejemplo: ha creado recientemente el entorno y se están arrancando las instancias.

Mensaje: inicialización en curso (en ejecución durante 12 segundos).

Unknown (gris)

Este estado se muestra cuando Elastic Beanstalk y el agente de estado notifican que hay una cantidad insuficiente de datos en una instancia.

Ejemplo: no se están recibiendo datos.

Suspendido (gris)

Este estado se muestra cuando Elastic Beanstalk detiene la supervisión del estado del entorno. Es posible que el entorno no funcione correctamente. Algunas condiciones de estado graves, si duran mucho tiempo, provocan que Elastic Beanstalk cambie el entorno al estado Suspendido.

Ejemplo: Elastic Beanstalk no puede acceder al [rol de servicio](#) del entorno.

Ejemplo: se ha eliminado el [grupo de Auto Scaling](#) creado por Elastic Beanstalk para el entorno.

Mensaje: El estado del entorno ha cambiado de OK a Severe. No hay instancias. La capacidad deseada del grupo de Auto Scaling se ha configurado en 1.

Métricas de la instancia

Las métricas de instancia proporcionan información acerca del estado de las instancias en el entorno. El [agente de estado de Elastic Beanstalk](#) se ejecuta en cada instancia. Recopila métricas de las instancias y las envía a Elastic Beanstalk, que las analiza para determinar el estado de las instancias de los entornos.

El agente de estado de Elastic Beanstalk en la instancia recopila métricas sobre las instancias de los servidores web y del sistema operativo. Para obtener información de servidor web sobre plataformas basadas en Linux, Elastic Beanstalk lee y analiza los registros del servidor web. En la plataforma Windows Server, Elastic Beanstalk recibe esta información directamente del servidor web de IIS. Los servidores web proporcionan información sobre las solicitudes HTTP entrantes: cuántas solicitudes llegaron, cuántas generaron errores y cuánto tiempo tardaron en resolverse. El sistema operativo proporciona información instantánea sobre el estado de los recursos de las instancias: la carga de CPU y la distribución del tiempo empleado en cada tipo de proceso.

El agente de estado recopila las métricas del servidor web y el sistema operativo y las transmite a Elastic Beanstalk cada 10 segundos. Elastic Beanstalk analiza los datos y utiliza los resultados para actualizar el estado de cada instancia y del entorno.

Temas

- [Métricas del servidor web](#)
- [Métricas del sistema operativo](#)
- [Captura de métricas de servidor web en IIS en Windows Server](#)

Métricas del servidor web

En las plataformas basadas en Linux, el agente de estado de Elastic Beanstalk consulta las métricas del servidor web en los registros generados por el contenedor web o el servidor que procesa las solicitudes en cada instancia del entorno. Las plataformas Elastic Beanstalk están configuradas para generar dos registros: uno en lenguaje natural y otro en lenguaje de máquina. El agente de estado retransmite registros legibles por máquina a Elastic Beanstalk cada 10 segundos.

Para obtener más información sobre el formato de registro utilizado por Elastic Beanstalk, consulte [Formato de los registros de estado avanzados](#).

En la plataforma Windows Server, Elastic Beanstalk agrega un módulo a la canalización de peticiones del servidor web IIS y captura métricas sobre los tiempos de petición HTTP y los códigos

de respuesta. El módulo envía estas métricas al agente de estado en la instancia mediante un canal de comunicación entre procesos (IPC) de alto rendimiento. Para conocer los detalles de implementación, consulte [Captura de métricas de servidor web en IIS en Windows Server](#).

Métricas del servidor web notificadas

RequestCount

Número de solicitudes administradas por el servidor web por segundo durante los últimos 10 segundos. Se muestra como un valor medio \bar{x} /sec (solicitudes por segundo) en la CLI de EB y en [Página de estado del entorno](#).

Status2xx, Status3xx, Status4xx, Status5xx

Número de solicitudes que generaron cada tipo de código de estado durante los últimos 10 segundos. Por ejemplo, las solicitudes correctas devuelven 200 OK, las solicitudes redireccionadas devuelven 301 y, si la URL especificada no coincide con ningún recurso de la aplicación, se devuelve 404.

En la CLI de EB y en [Página de estado del entorno](#), estas métricas se muestran como la cantidad total de solicitudes de las instancias y como un porcentaje de las solicitudes totales de los entornos.

p99.9, p99, p95, p90, p85, p75, p50, p10

Latencia media del porcentaje x de solicitudes más lentas durante los últimos 10 segundos, donde x es la diferencia entre el número y 100. Por ejemplo, p99 1.403 indica que la latencia del 1 % más lento de las solicitudes durante los últimos 10 segundos tuvo una latencia media de 1,403 segundos.

Métricas del sistema operativo

El agente de estado de Elastic Beanstalk notifica las siguientes métricas del sistema operativo. Elastic Beanstalk utiliza estas métricas para identificar las instancias que tienen una carga pesada durante periodos prolongados. Las métricas difieren según el sistema operativo.

Métricas del sistema operativo notificadas: Linux

Running

Cantidad de tiempo que ha transcurrido desde que se lanzó la instancia.

Load 1, Load 5

Carga media durante el último minuto y los últimos cinco minutos. Se muestra como un valor decimal que indica el número medio de procesos que estaban en ejecución en ese momento. Si el número que se muestra es mayor que el número de vCPU (subprocesos) disponibles, el resto es el número medio de procesos que estaban a la espera.

Por ejemplo, si el tipo de instancia tiene cuatro vCPU y la carga es 4,5, hubo una media de 0,5 procesos en espera durante ese periodo de tiempo. En un proceso completo, equivaldría al 50 por ciento del tiempo en espera.

User %, Nice %, System %, Idle %, I/O Wait %

Porcentaje de tiempo que la CPU ha pasado en cada estado durante los últimos 10 segundos.

Métricas del sistema operativo notificadas: Windows

Running

Cantidad de tiempo que ha transcurrido desde que se lanzó la instancia.

% User Time, % Privileged Time, % Idle Time

Porcentaje de tiempo que la CPU ha pasado en cada estado durante los últimos 10 segundos.

Captura de métricas de servidor web en IIS en Windows Server

En la plataforma Windows Server, Elastic Beanstalk agrega un módulo a la canalización de peticiones del servidor web IIS y captura métricas sobre los tiempos de petición HTTP y los códigos de respuesta. El módulo envía estas métricas al agente de estado en la instancia mediante un canal de comunicación entre procesos (IPC) de alto rendimiento. El agente de estado agrega estas métricas, las combina con las métricas del sistema operativo y las envía al servicio de Elastic Beanstalk.

Detalles de la implementación

Para capturar las métricas de IIS, Elastic Beanstalk implementa un [IHttpModule](#) administrado y se suscribe a los eventos [BeginRequest](#) y [EndRequest](#). Esto permite al módulo notificar la latencia de solicitudes HTTP y los códigos de respuesta de todas las peticiones web tratadas por IIS. Para agregar el módulo a la canalización de solicitudes de IIS, Elastic Beanstalk registra el módulo en la

sección [<modules>](#) del archivo de configuración de IIS, %windir%\System32\inetsrv\config\applicationHost.config.

El módulo de Elastic Beanstalk en IIS envía las métricas de solicitud web capturadas al agente de estado en la instancia, que es un servicio de Windows llamado HealthD. Para enviar estos datos, el módulo utiliza [NetNamedPipeBinding](#), que proporciona un enlace seguro y de confianza que está optimizado para la comunicación en la máquina.

Configuración de las reglas de informes de estado avanzados de un entorno

Los informes de estado mejorado de AWS Elastic Beanstalk se basan en un conjunto de reglas para determinar el estado del entorno. Algunas de estas reglas podrían no ser adecuadas para su aplicación específica. A continuación, se presentan algunos ejemplos comunes:

- Utiliza herramientas de prueba del lado del cliente. En este caso, se esperan errores de cliente HTTP (4xx) frecuentes.
- Utilice [AWS WAF](#) junto con el Application Load Balancer de su entorno para bloquear el tráfico entrante no deseado. En este caso, el balanceador de carga de aplicaciones devuelve HTTP 403 por cada mensaje entrante rechazado.

De forma predeterminada, Elastic Beanstalk incluye todos los errores HTTP 4xx de la aplicación al determinar el estado del entorno. Cambia el estado de su entorno de Correcto a Advertencia, Degradado o Grave, en función de la tasa de errores. Para gestionar correctamente casos como los ejemplos indicados, Elastic Beanstalk permite configurar algunas reglas de informes de estado avanzados. Puede optar por omitir los errores HTTP 4xx de la aplicación en las instancias del entorno u omitir los errores HTTP 4xx devueltos por el balanceador de carga del entorno. En este tema se describe cómo realizar estos cambios de configuración.

Note

En la actualidad, estas son las únicas personalizaciones de reglas de informes de estado avanzados disponibles. No puede configurar los informes de estado avanzados para ignorar otros errores HTTP además de 4xx.

Configuración de reglas de mantenimiento mejoradas mediante la consola de Elastic Beanstalk

Puede utilizar la consola de Elastic Beanstalk para configurar reglas de estado avanzado en su entorno.

Para configurar la comprobación de código de estado HTTP 4xx mediante la consola de Elastic Beanstalk


1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Monitoring (Monitorización), seleccione Edit (Editar).
5. En Health monitoring rule customization (Personalización de la regla de monitoreo de estado), habilite o deshabilite las opciones Ignore (Omitir).

Health monitoring rule customization

Configure the HTTP application and load balancer status codes included in determining your environment's health. [Learn more](#) 

Ignore application 4xx

Enabled

Ignore load balancer 4xx

Enabled

6. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

Configuración de reglas de informes de estado avanzados mediante la CLI de EB

Puede utilizar la CLI de EB para configurar reglas de estado avanzado guardando localmente la configuración del entorno, añadiendo una entrada que configure las reglas de estado avanzado y cargando la configuración en Elastic Beanstalk. Puede aplicar la configuración guardada al entorno durante o después de la creación.

Para configurar la comprobación de código de estado HTTP 4xx mediante la CLI de EB y las configuraciones guardadas

1. Inicialice la carpeta del proyecto con [eb init](#).
2. Cree un entorno ejecutando el comando [eb create](#).
3. Guarde una plantilla de configuración localmente ejecutando el comando `eb config save`. En el ejemplo siguiente, se utiliza la opción `--cfg` para especificar el nombre de la configuración.

```
$ eb config save --cfg 01-base-state
Configuration saved at: ~/project/.elasticbeanstalk/saved_configs/01-base-state.cfg.yml
```

4. Abra el archivo de configuración guardado en un editor de texto.
5. En `OptionSettings > aws:elasticbeanstalk:healthreporting:system:`, agregue una clave `ConfigDocument` para mostrar cada regla de estado avanzado que se va a configurar. El siguiente `ConfigDocument` deshabilita la comprobación de los códigos de estado HTTP 4xx de la aplicación, mientras mantiene habilitada la comprobación del código HTTP 4xx del balanceador de carga.

```
OptionSettings:
  ...
  aws:elasticbeanstalk:healthreporting:system:
    ConfigDocument:
      Rules:
        Environment:
          Application:
            ApplicationRequests4xx:
              Enabled: false
          ELB:
            ELBRequests4xx:
              Enabled: true
      Version: 1
    SystemType: enhanced
```

...

Note

Puede combinar Rules y CloudWatchMetrics en el mismo ajuste de opción de ConfigDocument. CloudWatchMetrics se describen en [Publicación de métricas personalizadas de un entorno en Amazon CloudWatch](#).

Si ha habilitado previamente CloudWatchMetrics, el archivo de configuración que recupere mediante el comando `eb config save` ya tiene una clave ConfigDocument con una sección CloudWatchMetrics. No lo elimine; agrega una sección Rules en el mismo valor de la opción ConfigDocument.

6. Guarde el archivo de configuración y cierre el editor de texto. En este ejemplo, el archivo de configuración actualizado se ha guardado con otro nombre (`02-cloudwatch-enabled.cfg.yml`) que sea diferente al del archivo de configuración que se descargó. De este modo, se crea una configuración guardada distinta cuando se cargue el archivo. Puede utilizar el mismo nombre que el del archivo descargado para sobrescribir la configuración existente sin crear una nueva.
7. Utilice el comando `eb config put` para cargar el archivo de configuración actualizado a Elastic Beanstalk.

```
$ eb config put 02-cloudwatch-enabled
```

Si utiliza los comandos `eb config get` y `put` con configuraciones guardadas, no incluya la extensión de nombre de archivo.

8. Aplique la configuración guardada en el entorno que está en ejecución.

```
$ eb config --cfg 02-cloudwatch-enabled
```

La opción `--cfg` especifica el archivo de configuración designado que se aplica al entorno. Puede guardar el archivo de configuración localmente o en Elastic Beanstalk. Si hay un archivo de configuración con el nombre especificado en las dos ubicaciones, la CLI de EB utiliza el archivo local.

Configuración de reglas de informes de estado avanzados mediante un documento de configuración

El documento de configuración (config) para reglas de estado avanzado es un documento JSON que contiene una lista de las reglas que se van a configurar.

El siguiente ejemplo muestra un documento de configuración que deshabilita la comprobación de los códigos de estado HTTP 4xx de la aplicación y habilita la comprobación de los códigos de estado HTTP 4xx del balanceador de carga.

```
{
  "Rules": {
    "Environment": {
      "Application": {
        "ApplicationRequests4xx": {
          "Enabled": false
        }
      },
      "ELB": {
        "ELBRequests4xx": {
          "Enabled": true
        }
      }
    }
  },
  "Version": 1
}
```

En el caso de la AWS CLI, debe pasar el documento como valor de la clave Value en un argumento de configuración de opciones, que en sí mismo es un objeto JSON. En este caso, las comillas del documento insertado deben escribirse con caracteres de escape. El siguiente comando comprueba si los ajustes de configuración son válidos.

```
$ aws elasticbeanstalk validate-configuration-settings --application-name my-app --
environment-name my-env --option-settings '[
  {
    "Namespace": "aws:elasticbeanstalk:healthreporting:system",
    "OptionName": "ConfigDocument",
    "Value": "{\"Rules\": { \"Environment\": { \"Application\":
{ \"ApplicationRequests4xx\": { \"Enabled\": false } }, \"ELB\": { \"ELBRequests4xx\":
{ \"Enabled\": true } } } }, \"Version\": 1 }"
```

```
}  
]'
```

En los archivos de configuración `.ebextensions` de YAML, puede proporcionar el documento JSON tal y como está.

```
option_settings:  
  - namespace: aws:elasticbeanstalk:healthreporting:system  
    option_name: ConfigDocument  
    value: {  
"Rules": {  
  "Environment": {  
    "Application": {  
      "ApplicationRequests4xx": {  
        "Enabled": false  
      }  
    },  
    "ELB": {  
      "ELBRequests4xx": {  
        "Enabled": true  
      }  
    }  
  }  
},  
  "Version": 1  
}
```

Publicación de métricas personalizadas de un entorno en Amazon CloudWatch

Puede publicar los datos recopilados mediante informes de estado mejorado de AWS Elastic Beanstalk en Amazon CloudWatch como métricas personalizadas. Gracias a la publicación de métricas en CloudWatch, puede supervisar los cambios que se producen con el tiempo en el rendimiento de la aplicación e identificar posibles problemas, ya que puede hacer un seguimiento del uso de los recursos y la escala de latencia de las solicitudes con carga.

Cuando se publican métricas en CloudWatch, también se pueden utilizar con [gráficos de supervisión](#) y [alarmas](#). Al utilizar informes de estado avanzados, se habilita automáticamente una métrica gratuita: EnvironmentHealth. Todas las métricas personalizadas, excepto EnvironmentHealth, generan [cargas en CloudWatch](#).

Para publicar métricas personalizadas sobre un entorno en CloudWatch, primero deben habilitarse los informes de estado avanzados en el entorno. Para obtener instrucciones, consulte [Habilitación de informes de estado mejorado de Elastic Beanstalk](#).

Temas

- [Métricas de informes de estado avanzados](#)
- [Configuración de métricas de CloudWatch mediante la consola de Elastic Beanstalk](#)
- [Configuración de métricas personalizadas de CloudWatch mediante la CLI de EB](#)
- [Proporcionar documentos de configuración de métricas personalizadas](#)

Métricas de informes de estado avanzados

Cuando habilite los informes de estado avanzados en el entorno, el sistema de informes de estado avanzados publica automáticamente una [métrica personalizada de CloudWatch](#), EnvironmentHealth. Para publicar otras métricas en CloudWatch, configure el entorno con dichas métricas utilizando la [consola de Elastic Beanstalk](#), [la CLI de EB](#) o [.ebextensions](#).

Puede publicar las siguientes métricas de estado avanzado desde su entorno en CloudWatch.

Métricas disponibles: todas las plataformas

EnvironmentHealth

Solo se aplica al entorno. Si no se configuran otras métricas, esta es la única métrica de CloudWatch que publica el sistema de informes de estado avanzado. El estado del entorno está representado por uno de los siete [estados](#). En la consola de CloudWatch, estos estados se corresponden con los siguientes valores:

- 0: OK
- 1: Info
- 5: Unknown
- 10: No data
- 15: Warning
- 20: Degraded
- 25: Severe

InstancesSevere, InstancesDegraded, InstancesWarning, InstancesInfo, InstancesOk, InstancesPending, InstancesUnknown, InstancesNoData

Solo se aplica al entorno. Estas métricas indican el número de instancias del entorno y el estado de cada una. InstancesNoData indica el número de instancias para las que no se reciben datos.

ApplicationRequestsTotal, ApplicationRequests5xx, ApplicationRequests4xx, ApplicationRequests3xx, ApplicationRequests2xx

Se aplica a la instancia y al entorno. Indica el número total de solicitudes completado por la instancia o el entorno, así como el número de solicitudes completadas con el código de estado de cada categoría.

ApplicationLatencyP10, ApplicationLatencyP50, ApplicationLatencyP75, ApplicationLatencyP85, ApplicationLatencyP90, ApplicationLatencyP95, ApplicationLatencyP99, ApplicationLatencyP99.9

Se aplica a la instancia y al entorno. Indica el período medio de tiempo, en segundos, que tarda en completarse el porcentaje x de las solicitudes más rápidas.

InstanceHealth

Se aplica solo a la instancia. Indica el estado actual de la instancia. El estado de la instancia está representado por uno de los siete [estados](#). En la consola de CloudWatch, estos estados se corresponden con los siguientes valores:

- 0: OK
- 1: Info
- 5: Unknown
- 10: No data
- 15: Warning
- 20: Degraded
- 25: Severe

Métricas disponibles: Linux

CPUIrq, CPUIdle, CPUUser, CPUSystem, CPUSoftirq, CPUIowait, CPUNice

Se aplica solo a la instancia. Indica el porcentaje de tiempo que la CPU ha pasado en cada estado durante el último minuto.

LoadAverage1min

Se aplica solo a la instancia. Carga media de la CPU de la instancia durante el último minuto.

RootFilesystemUtil

Se aplica solo a la instancia. Indica el porcentaje de espacio en disco que está en uso.

Métricas disponibles: Windows

CPUIdle, CPUUser, CPUPrivileged

Se aplica solo a la instancia. Indica el porcentaje de tiempo que la CPU ha pasado en cada estado durante el último minuto.

Configuración de métricas de CloudWatch mediante la consola de Elastic Beanstalk

Puede utilizar la consola de Elastic Beanstalk para configurar el entorno de forma que pueda publicar métricas de los informes de estado avanzados en CloudWatch y puedan utilizarse con gráficos de supervisión y alarmas.

Para configurar métricas personalizadas de CloudWatch en la consola de Elastic Beanstalk

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Monitoring (Monitorización), seleccione Edit (Editar).
5. En Health reporting (Informe de estado), seleccione las métricas de instancia y de entorno que desee publicar en CloudWatch. Para seleccionar varias métricas, presione la tecla Ctrl mientras hace la selección.
6. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

Al habilitar las métricas personalizadas de CloudWatch, estas se agregan a la lista de métricas disponibles en la página [Monitoring \(Supervisión\)](#).

Configuración de métricas personalizadas de CloudWatch mediante la CLI de EB

Puede utilizar la CLI de EB para configurar métricas personalizadas. Para ello, guarde la configuración del entorno localmente, agregue una entrada que defina las métricas que se van a publicar y cargue la configuración en Elastic Beanstalk. Puede aplicar la configuración guardada al entorno durante o después de la creación.

Para configurar las métricas personalizadas de CloudWatch con la CLI de EB y las configuraciones guardadas

1. Inicialice la carpeta del proyecto con [eb init](#).
2. Cree un entorno ejecutando el comando [eb create](#).
3. Guarde una plantilla de configuración localmente ejecutando el comando `eb config save`. En el ejemplo siguiente, se utiliza la opción `--cfg` para especificar el nombre de la configuración.

```
$ eb config save --cfg 01-base-state
Configuration saved at: ~/project/.elasticbeanstalk/saved_configs/01-base-state.cfg.yml
```

4. Abra el archivo de configuración guardado en un editor de texto.
5. En `OptionSettings > aws:elasticbeanstalk:healthreporting:system:`, agregue una clave `ConfigDocument` para habilitar cada una de las métricas de CloudWatch que desee. Por ejemplo, en el ejemplo siguiente, `ConfigDocument` publica las métricas `ApplicationRequests5xx` y `ApplicationRequests4xx` en el nivel del entorno y la métrica `ApplicationRequestsTotal` en el nivel de la instancia.

```
OptionSettings:
  ...
  aws:elasticbeanstalk:healthreporting:system:
    ConfigDocument:
      CloudWatchMetrics:
        Environment:
          ApplicationRequests5xx: 60
          ApplicationRequests4xx: 60
        Instance:
          ApplicationRequestsTotal: 60
      Version: 1
```

```
SystemType: enhanced
```

```
...
```

En el ejemplo, 60 indica el número de segundos entre las mediciones. En la actualidad, este es el único valor que se permite.

Note

Puede combinar `CloudWatchMetrics` y `Rules` en el mismo ajuste de opción de `ConfigDocument`. `Rules` se describen en [Configuración de las reglas de informes de estado avanzados de un entorno](#).

Si ha utilizado previamente `Rules` para configurar las reglas de estado avanzado, a continuación, el archivo de configuración que recupere mediante el comando `eb config save` ya tiene una clave `ConfigDocument` con una sección `Rules`. No lo elimine; agrega una sección `CloudWatchMetrics` en el mismo valor de la opción `ConfigDocument`.

6. Guarde el archivo de configuración y cierre el editor de texto. En este ejemplo, el archivo de configuración actualizado se ha guardado con otro nombre (`02-cloudwatch-enabled.cfg.yml`) que sea diferente al del archivo de configuración que se descargó. De este modo, se crea una configuración guardada distinta cuando se cargue el archivo. Puede utilizar el mismo nombre que el del archivo descargado para sobrescribir la configuración existente sin crear una nueva.
7. Utilice el comando `eb config put` para cargar el archivo de configuración actualizado a Elastic Beanstalk.

```
$ eb config put 02-cloudwatch-enabled
```

Si utiliza los comandos `eb config get` y `put` con configuraciones guardadas, no incluya la extensión de archivo.

8. Aplique la configuración guardada en el entorno que está en ejecución.

```
$ eb config --cfg 02-cloudwatch-enabled
```

La opción `--cfg` especifica el archivo de configuración designado que se aplica al entorno. Puede guardar el archivo de configuración localmente o en Elastic Beanstalk. Si hay un archivo

de configuración con el nombre especificado en las dos ubicaciones, la CLI de EB utiliza el archivo local.

Proporcionar documentos de configuración de métricas personalizadas

El documento de configuración (config) de métricas personalizadas de Amazon CloudWatch es un documento JSON donde se muestran las métricas que se van a publicar en los niveles del entorno y de la instancia. En el ejemplo siguiente, se muestra un documento de configuración que habilita todas las métricas personalizadas disponibles en Linux.

```
{
  "CloudWatchMetrics": {
    "Environment": {
      "ApplicationLatencyP99.9": 60,
      "InstancesSevere": 60,
      "ApplicationLatencyP90": 60,
      "ApplicationLatencyP99": 60,
      "ApplicationLatencyP95": 60,
      "InstancesUnknown": 60,
      "ApplicationLatencyP85": 60,
      "InstancesInfo": 60,
      "ApplicationRequests2xx": 60,
      "InstancesDegraded": 60,
      "InstancesWarning": 60,
      "ApplicationLatencyP50": 60,
      "ApplicationRequestsTotal": 60,
      "InstancesNoData": 60,
      "InstancesPending": 60,
      "ApplicationLatencyP10": 60,
      "ApplicationRequests5xx": 60,
      "ApplicationLatencyP75": 60,
      "InstancesOk": 60,
      "ApplicationRequests3xx": 60,
      "ApplicationRequests4xx": 60
    },
    "Instance": {
      "ApplicationLatencyP99.9": 60,
      "ApplicationLatencyP90": 60,
      "ApplicationLatencyP99": 60,
      "ApplicationLatencyP95": 60,
      "ApplicationLatencyP85": 60,
    }
  }
}
```

```

    "CPUUser": 60,
    "ApplicationRequests2xx": 60,
    "CPUIdle": 60,
    "ApplicationLatencyP50": 60,
    "ApplicationRequestsTotal": 60,
    "RootFilesystemUtil": 60,
    "LoadAverage1min": 60,
    "CPUirq": 60,
    "CPUNice": 60,
    "CPUiowait": 60,
    "ApplicationLatencyP10": 60,
    "LoadAverage5min": 60,
    "ApplicationRequests5xx": 60,
    "ApplicationLatencyP75": 60,
    "CPUSystem": 60,
    "ApplicationRequests3xx": 60,
    "ApplicationRequests4xx": 60,
    "InstanceHealth": 60,
    "CPUsoftirq": 60
  }
},
"Version": 1
}

```

En el caso de la AWS CLI, debe pasar el documento como valor de la clave `Value` en un argumento de configuración de opciones, que en sí mismo es un objeto JSON. En este caso, las comillas del documento insertado deben escribirse con caracteres de escape.

```

$ aws elasticbeanstalk validate-configuration-settings --application-name my-app --
environment-name my-env --option-settings '[
  {
    "Namespace": "aws:elasticbeanstalk:healthreporting:system",
    "OptionName": "ConfigDocument",
    "Value": "{\"CloudWatchMetrics\": {\"Environment\":
{\"ApplicationLatencyP99.9\": 60,\"InstancesSevere\": 60,\"ApplicationLatencyP90\":
60,\"ApplicationLatencyP99\": 60,\"ApplicationLatencyP95\": 60,\"InstancesUnknown
\": 60,\"ApplicationLatencyP85\": 60,\"InstancesInfo\": 60,\"ApplicationRequests2xx
\": 60,\"InstancesDegraded\": 60,\"InstancesWarning\": 60,\"ApplicationLatencyP50\":
60,\"ApplicationRequestsTotal\": 60,\"InstancesNoData\": 60,\"InstancesPending
\": 60,\"ApplicationLatencyP10\": 60,\"ApplicationRequests5xx\": 60,
\"ApplicationLatencyP75\": 60,\"InstancesOk\": 60,\"ApplicationRequests3xx\": 60,
\"ApplicationRequests4xx\": 60},\"Instance\": {\"ApplicationLatencyP99.9\": 60,
\"ApplicationLatencyP90\": 60,\"ApplicationLatencyP99\": 60,\"ApplicationLatencyP95\":

```

```
60,\"ApplicationLatencyP85\": 60,\"CPUUser\": 60,\"ApplicationRequests2xx\":
60,\"CPUIdle\": 60,\"ApplicationLatencyP50\": 60,\"ApplicationRequestsTotal\":
60,\"RootFilesystemUtil\": 60,\"LoadAverage1min\": 60,\"CPUirq\": 60,\"CPUNice
\": 60,\"CPUiowait\": 60,\"ApplicationLatencyP10\": 60,\"LoadAverage5min\": 60,
\"ApplicationRequests5xx\": 60,\"ApplicationLatencyP75\": 60,\"CPUSystem\": 60,
\"ApplicationRequests3xx\": 60,\"ApplicationRequests4xx\": 60,\"InstanceHealth\": 60,
\"CPUsoftirq\": 60}},\"Version\": 1}"
  }
]'
```

En los archivos de configuración `.ebextensions` de YAML, puede proporcionar el documento JSON tal y como está.

```
option_settings:
  - namespace: aws:elasticbeanstalk:healthreporting:system
    option_name: ConfigDocument
    value: {
"CloudWatchMetrics": {
  "Environment": {
    "ApplicationLatencyP99.9": 60,
    "InstancesSevere": 60,
    "ApplicationLatencyP90": 60,
    "ApplicationLatencyP99": 60,
    "ApplicationLatencyP95": 60,
    "InstancesUnknown": 60,
    "ApplicationLatencyP85": 60,
    "InstancesInfo": 60,
    "ApplicationRequests2xx": 60,
    "InstancesDegraded": 60,
    "InstancesWarning": 60,
    "ApplicationLatencyP50": 60,
    "ApplicationRequestsTotal": 60,
    "InstancesNoData": 60,
    "InstancesPending": 60,
    "ApplicationLatencyP10": 60,
    "ApplicationRequests5xx": 60,
    "ApplicationLatencyP75": 60,
    "InstancesOk": 60,
    "ApplicationRequests3xx": 60,
    "ApplicationRequests4xx": 60
  },
  "Instance": {
    "ApplicationLatencyP99.9": 60,
```

```
"ApplicationLatencyP90": 60,  
"ApplicationLatencyP99": 60,  
"ApplicationLatencyP95": 60,  
"ApplicationLatencyP85": 60,  
"CPUUser": 60,  
"ApplicationRequests2xx": 60,  
"CPUIidle": 60,  
"ApplicationLatencyP50": 60,  
"ApplicationRequestsTotal": 60,  
"RootFilesystemUtil": 60,  
"LoadAverage1min": 60,  
"CPUirq": 60,  
"CPUNice": 60,  
"CPUiowait": 60,  
"ApplicationLatencyP10": 60,  
"LoadAverage5min": 60,  
"ApplicationRequests5xx": 60,  
"ApplicationLatencyP75": 60,  
"CPUSystem": 60,  
"ApplicationRequests3xx": 60,  
"ApplicationRequests4xx": 60,  
"InstanceHealth": 60,  
"CPUSoftirq": 60  
}  
,  
"Version": 1  
}
```

Uso de informes de estado mejorados con la API de Elastic Beanstalk

Como los informes de estado mejorados de AWS Elastic Beanstalk tienen requisitos de rol y de pila de soluciones, debe actualizar los scripts y el código que usó antes de que se lanzara la funcionalidad mejorada de informes de estado para poder usarlos. Para mantener la compatibilidad con versiones anteriores, los informes de estado mejorados no están habilitados de forma predeterminada cuando crea un entorno utilizando la API de Elastic Beanstalk.

Los informes de estado mejorados se configuran definiendo el rol del servicio, el perfil de instancia y las opciones de configuración de Amazon CloudWatch para su entorno. Puede hacer esto de tres formas: mediante las opciones de configuración de la carpeta `.ebextensions`, con configuraciones guardadas o configurando las opciones directamente en el parámetro `create-environment` de la llamada a `option-settings`.

Para utilizar la API, los SDK o la interfaz de línea de comandos (CLI) de AWS con el objetivo de crear un entorno que admita un estado mejorado, debe hacer lo siguiente:

- Crear un rol de servicio y un perfil de instancia con los [permisos](#) adecuados
- Crear un nuevo entorno con una nueva [versión de la plataforma](#)
- Definir las [opciones de configuración](#) de tipo de sistema de estado, perfil de instancia y rol de servicio

Utilice las siguientes opciones de configuración de los espacios de nombres `aws:elasticbeanstalk:healthreporting:system`, `aws:autoscaling:launchconfiguration` y `aws:elasticbeanstalk:environment` para configurar su entorno para que utilice informes de estado mejorados.

Opciones de configuración de informes de estado mejorados

SystemType

Espacio de nombres: `aws:elasticbeanstalk:healthreporting:system`

Para habilitar los informes de estado mejorados, establezca este parámetro en **enhanced**.

IamInstanceProfile

Espacio de nombres: `aws:autoscaling:launchconfiguration`

Establezca este parámetro en el nombre de un perfil de instancia configurado para su uso con Elastic Beanstalk.

ServiceRole

Espacio de nombres: `aws:elasticbeanstalk:environment`

Establezca este parámetro en un rol de servicio configurado para su uso con Elastic Beanstalk.

ConfigDocument (opcional)

Espacio de nombres: `aws:elasticbeanstalk:healthreporting:system`

Un documento JSON que define las métricas de instancia y del entorno que se publican en CloudWatch. Por ejemplo:

```
{
  "CloudWatchMetrics":
  {
    "Environment":
    {
      "ApplicationLatencyP99.9":60,
      "InstancesSevere":60
    }
    "Instance":
    {
      "ApplicationLatencyP85":60,
      "CPUUser": 60
    }
  }
  "Version":1
}
```

Note

Los documentos config pueden requerir un formato especial, como incluir las comillas en caracteres de escape, en función de cómo los proporcione a Elastic Beanstalk. Para ver ejemplos, consulte [Proporcionar documentos de configuración de métricas personalizadas](#).

Formato de los registros de estado avanzados

AWS Elastic BeanstalkLas plataformas de utilizan un formato de registro de servidor web personalizado para transmitir información de forma eficiente sobre las solicitudes HTTP al sistema de informes de estado mejorado. El sistema analiza los registros, identifica los problemas y determina convenientemente el estado de la instancia y del entorno. Aunque se deshabilite el proxy del servidor web en el entorno y las solicitudes se envíen directamente desde el contenedor web, es posible seguir utilizando informes de estado avanzado. Para ello, configure el servidor de forma que se generen registros en la ubicación y se aplique el formato que utiliza el [agente de estado de Elastic Beanstalk](#).

Note

La información de esta página solo es relevante para las plataformas basadas en Linux. En la plataforma de Windows Server, Elastic Beanstalk recibe información sobre las solicitudes

HTTP directamente del servidor web de IIS. Para obtener más información, consulte [Captura de métricas de servidor web en IIS en Windows Server](#).

Configuración de los registros de servidor web

Las plataformas Elastic Beanstalk están configuradas para generar dos registros con información sobre las solicitudes HTTP. El primero de ellos tiene un formato detallado y proporciona información pormenorizada de la solicitud, como los datos del agente de usuario del solicitante y una marca de tiempo en lenguaje natural.

```
/var/log/nginx/access.log
```

El siguiente ejemplo procede de un proxy nginx que se ejecuta en un entorno de servidor web de Ruby, aunque el formato es similar al de Apache.

```
172.31.24.3 - - [23/Jul/2015:00:21:20 +0000] "GET / HTTP/1.1" 200 11 "-" "curl/7.22.0
(x86_64-pc-linux-gnu) libcurl/7.22.0 OpenSSL/1.0.1 zlib/1.2.3.4 libidn/1.23
librtmp/2.3" "177.72.242.17"
172.31.24.3 - - [23/Jul/2015:00:21:21 +0000] "GET / HTTP/1.1" 200 11 "-" "curl/7.22.0
(x86_64-pc-linux-gnu) libcurl/7.22.0 OpenSSL/1.0.1 zlib/1.2.3.4 libidn/1.23
librtmp/2.3" "177.72.242.17"
172.31.24.3 - - [23/Jul/2015:00:21:22 +0000] "GET / HTTP/1.1" 200 11 "-" "curl/7.22.0
(x86_64-pc-linux-gnu) libcurl/7.22.0 OpenSSL/1.0.1 zlib/1.2.3.4 libidn/1.23
librtmp/2.3" "177.72.242.17"
172.31.24.3 - - [23/Jul/2015:00:21:22 +0000] "GET / HTTP/1.1" 200 11 "-" "curl/7.22.0
(x86_64-pc-linux-gnu) libcurl/7.22.0 OpenSSL/1.0.1 zlib/1.2.3.4 libidn/1.23
librtmp/2.3" "177.72.242.17"
172.31.24.3 - - [23/Jul/2015:00:21:22 +0000] "GET / HTTP/1.1" 200 11 "-" "curl/7.22.0
(x86_64-pc-linux-gnu) libcurl/7.22.0 OpenSSL/1.0.1 zlib/1.2.3.4 libidn/1.23
librtmp/2.3" "177.72.242.17"
```

El segundo log tiene un formato más escueto. Solo contiene información que es pertinente para los informes de estado avanzados. Esta log se genera en una subcarpeta llamada `healthd` y rota cada hora. Los logs antiguos se eliminan inmediatamente después de la rotación.

```
/var/log/nginx/healthd/application.log.2015-07-23-00
```

En el siguiente ejemplo, se muestra un log con el formato de lectura mecánica.

```
1437609879.311"/"200"0.083"0.083"177.72.242.17
```

```
1437609879.874"/"200"0.347"0.347"177.72.242.17
1437609880.006"/bad/path"404"0.001"0.001"177.72.242.17
1437609880.058"/"200"0.530"0.530"177.72.242.17
1437609880.928"/bad/path"404"0.001"0.001"177.72.242.17
```

El formato avanzado del log de estado contiene la siguiente información:

- Hora de la solicitud, en tiempo Unix
- Ruta de la solicitud
- Código de estado HTTP del resultado
- Tiempo de la solicitud
- Tiempo de subida
- Encabezado HTTP X-Forwarded-For

En los proxies de nginx, los valores de tiempo se presentan en segundos de punto flotante, con tres decimales. En Apache, se utilizan microsegundos enteros.

Note

Si aparece una advertencia similar a la siguiente en un archivo de registro (donde DATE-TIME es una fecha y una hora) y utiliza un proxy personalizado (por ejemplo, un entorno de Docker con varios contenedores), debe utilizar `.ebextension` para configurar el entorno de forma que healthd pueda leer los archivos de registro:

```
W, [DATE-TIME #1922] WARN -- : log file "/var/log/nginx/healthd/
application.log.DATE-TIME" does not exist
```

Puede comenzar a usar `.ebextension` en el [ejemplo de Docker Multicontainer](#).

```
/etc/nginx/conf.d/webapp_healthd.conf
```

En el siguiente ejemplo, se muestra la configuración del registro de nginx con el formato del registro de healthd resaltado.

```
upstream my_app {
    server unix:///var/run/puma/my_app.sock;
}
```

```
log_format healthd '$msec"$uri"'
                    '$status"$request_time"$upstream_response_time"'
                    '$http_x_forwarded_for';

server {
    listen 80;
    server_name _ localhost; # need to listen to localhost for worker tier

    if ($time_iso8601 ~ "^(\\d{4})-(\\d{2})-(\\d{2})T(\\d{2})") {
        set $year $1;
        set $month $2;
        set $day $3;
        set $hour $4;
    }

    access_log /var/log/nginx/access.log main;
    access_log /var/log/nginx/healthd/application.log.$year-$month-$day-$hour healthd;

    location / {
        proxy_pass http://my_app; # match the name of upstream directive which is defined
        above
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }

    location /assets {
        alias /var/app/current/public/assets;
        gzip_static on;
        gzip on;
        expires max;
        add_header Cache-Control public;
    }

    location /public {
        alias /var/app/current/public;
        gzip_static on;
        gzip on;
        expires max;
        add_header Cache-Control public;
    }
}
```

/etc/httpd/conf.d/healthd.conf

En el siguiente ejemplo se muestra la configuración de registro de Apache.

```
LogFormat "%{s}t\"%U\"%s\"%D\"%D\"%{X-Forwarded-For}i" healthd
CustomLog "|/usr/sbin/rotatelogs /var/log/httpd/healthd/application.log.%Y-%m-%d-%H
3600" healthd
```

Creación de registros para informes de estado avanzados

Para proporcionar logs al agente de estado, debe hacer lo siguiente:

- Generar logs con el formato correcto, tal y como se explicó en la sección anterior.
- Generar logs en `/var/log/nginx/healthd/`
- Asignar un nombre a los logs que tenga el siguiente formato: `application.log.$year-$month-$day-$hour`
- Rotar los logs una vez por hora.
- No truncar los logs.

Notificaciones y solución de problemas

En esta página, se muestran algunos ejemplos de mensajes sobre problemas comunes y enlaces para obtener más información. Los mensajes de incidencias se muestran en la página [información del entorno](#) de la consola de Elastic Beanstalk y se registran como [eventos](#) cuando los problemas de estado se mantienen durante varias comprobaciones.

Implementaciones

Elastic Beanstalk supervisa el entorno para mantener la coherencia con las implementaciones posteriores. Si se produce un error en una implementación continua, la versión de la aplicación que se ejecuta en las instancias del entorno podría variar. Esto puede producirse si la implementación de uno o varios lotes se realiza correctamente, pero se produce un error antes de completar todos los lotes.

Se encontró una versión incorrecta de la aplicación en 2 de las 5 instancias. Se esperaba la versión "v1" (implementación 1).

Versión incorrecta de la aplicación en algunas instancias del entorno. Se esperaba la versión "v1" (implementación 1).

La versión de la aplicación prevista no se está ejecutando en algunas o en ninguna de las instancias de un entorno.

La versión de la aplicación, "v2" (implementación 2), no es correcta 2). Se esperaba la versión "v1" (implementación 1).

La aplicación implementada en una instancia es diferente a la versión esperada. Si se produce un error en una implementación, la versión esperada se restablece en la versión de la implementación más reciente que se realizó con éxito. En el ejemplo anterior, la primera implementación (versión "v1") se realizó correctamente, pero no la segunda implementación (versión "v2"). Por tanto, se considera que todas las instancias que ejecutan "v2" tienen un estado incorrecto.

Para solucionar este problema, debe iniciar otra implementación. Puede [volver a implementar](#) una versión anterior que sepa que funciona bien o configurar el entorno para que [se omitan las comprobaciones de estado](#) durante la implementación y vuelva a implementarse a la fuerza la nueva versión hasta que se complete.

También puede identificar y terminar las instancias en las que se ejecuta la versión incorrecta de la aplicación. Elastic Beanstalk lanzará instancias con la versión correcta para sustituir a las instancias terminadas. Utilice el [comando "health" de la CLI de EB](#) para identificar en qué instancias se ejecuta una versión incorrecta de la aplicación.

Servidor de aplicaciones

15% El 15 % de las solicitudes experimentan errores HTTP 4xx

20% El 20 % de las solicitudes dirigidas a ELB experimentan errores HTTP 4xx.

Hay un alto porcentaje de solicitudes HTTP dirigidas a una instancia o entorno que están experimentando errores 4xx.

Los códigos de estado de la serie 400 indican que el usuario hizo una solicitud incorrecta; por ejemplo, solicitó una página que no existe (404 Archivo no encontrado) o a la que el usuario no tiene acceso (403 Prohibido). Es normal que se produzcan algunos errores 404. Sin embargo, si el número es alto, podría indicar que hay enlaces internos o externos a páginas que no están disponibles. Estos problemas pueden resolverse corrigiendo los enlaces internos erróneos y redireccionando los enlaces externos que no son correctos.

5% El 5 % de las solicitudes experimentan errores HTTP 5xx

3% El 3 % de las solicitudes dirigidas a ELB experimentan errores HTTP 5xx.

Hay un alto porcentaje de solicitudes HTTP dirigidas a una instancia o entorno que están experimentando errores con códigos de estado de la serie 500.

Un código de estado de la serie 500 indica que el servidor de aplicaciones ha detectado un error interno. Estos errores informan de que hay un error en el código de la aplicación que debe identificarse y resolverse rápidamente.

95% El 95 % de la CPU está en uso

En una instancia, el agente de estado notifica un porcentaje extremadamente alto de uso de la CPU y establece el estado de la instancia en Warning o Degraded.

Escale el entorno para poder aliviar la carga de las instancias.

Instancia de trabajo

20 Hay 20 mensajes en espera en la cola (hace 25 segundos)

Se están agregando solicitudes a la cola del entorno de trabajo más rápido de lo que se pueden procesar. Escale el entorno para aumentar la capacidad.

5 Hay 5 mensajes en la cola de mensajes fallidos (hace 15 segundos)

Se están produciendo errores repetidamente en las solicitudes de trabajo, que se están agregando a [the section called “Colas de mensajes fallidos”](#). Consulte las solicitudes de la cola de mensajes fallidos para ver por qué están produciendo un error.

Otros recursos

4 El número de instancias activas (4) es inferior al tamaño mínimo del grupo de Auto Scaling 5

El número de instancias que se ejecutan en el entorno es inferior al número mínimo configurado para el grupo de Auto Scaling.

Se han eliminado o modificado notificaciones del grupo de Auto Scaling (nombreDelGrupo

Las notificaciones configuradas para el grupo de Auto Scaling se han modificado fuera de Elastic Beanstalk.

Administrar alarmas

Puede crear alarmas para las métricas que está monitorizando a través de la consola de Elastic Beanstalk. Las alarmas ayudan a monitorear los cambios del entorno de AWS Elastic Beanstalk para que pueda identificar y mitigar los problemas fácilmente antes de que se produzcan. Por ejemplo, puede configurar una alarma que le avise cuando la utilización de la CPU de un entorno supere un determinado umbral. De este modo, tendrá la seguridad de que se le informa antes de que se produzca un posible problema. Para obtener más información, consulte [Uso de Elastic Beanstalk con Amazon CloudWatch](#).

Note

Elastic Beanstalk utiliza CloudWatch para el monitoreo y las alarmas, lo que significa que los costos de CloudWatch se aplican a su cuenta de AWS para las alarmas que utilice.

Para obtener más información acerca de las métricas específicas de monitorización, consulte [Informes de estado básicos](#).

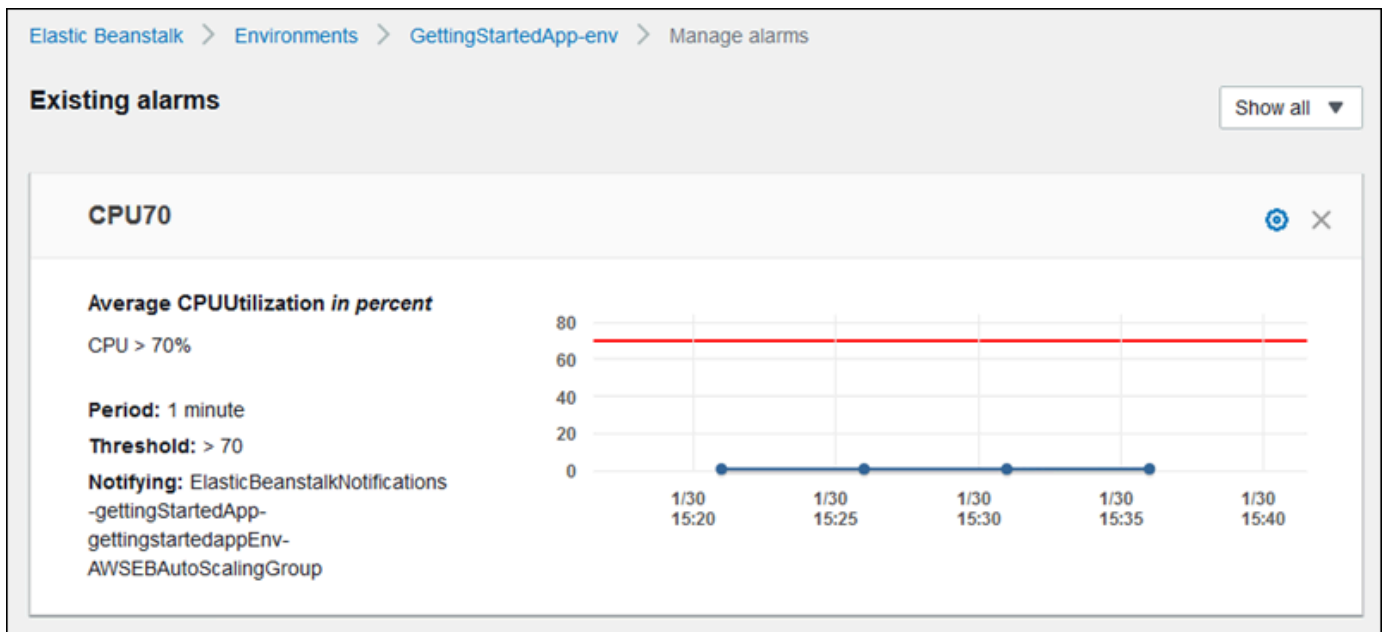
Para comprobar el estado de las alarmas

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Alarms.



La página muestra una lista de alarmas existentes. Si alguna alarma está activa, tendrá la marca



(warning) (aviso).

4. Para filtrar alarmas, elija el menú desplegable y a continuación, seleccione un filtro.
5. Para editar o eliminar una alarma, seleccione



(edit) o




(delete), respectivamente.

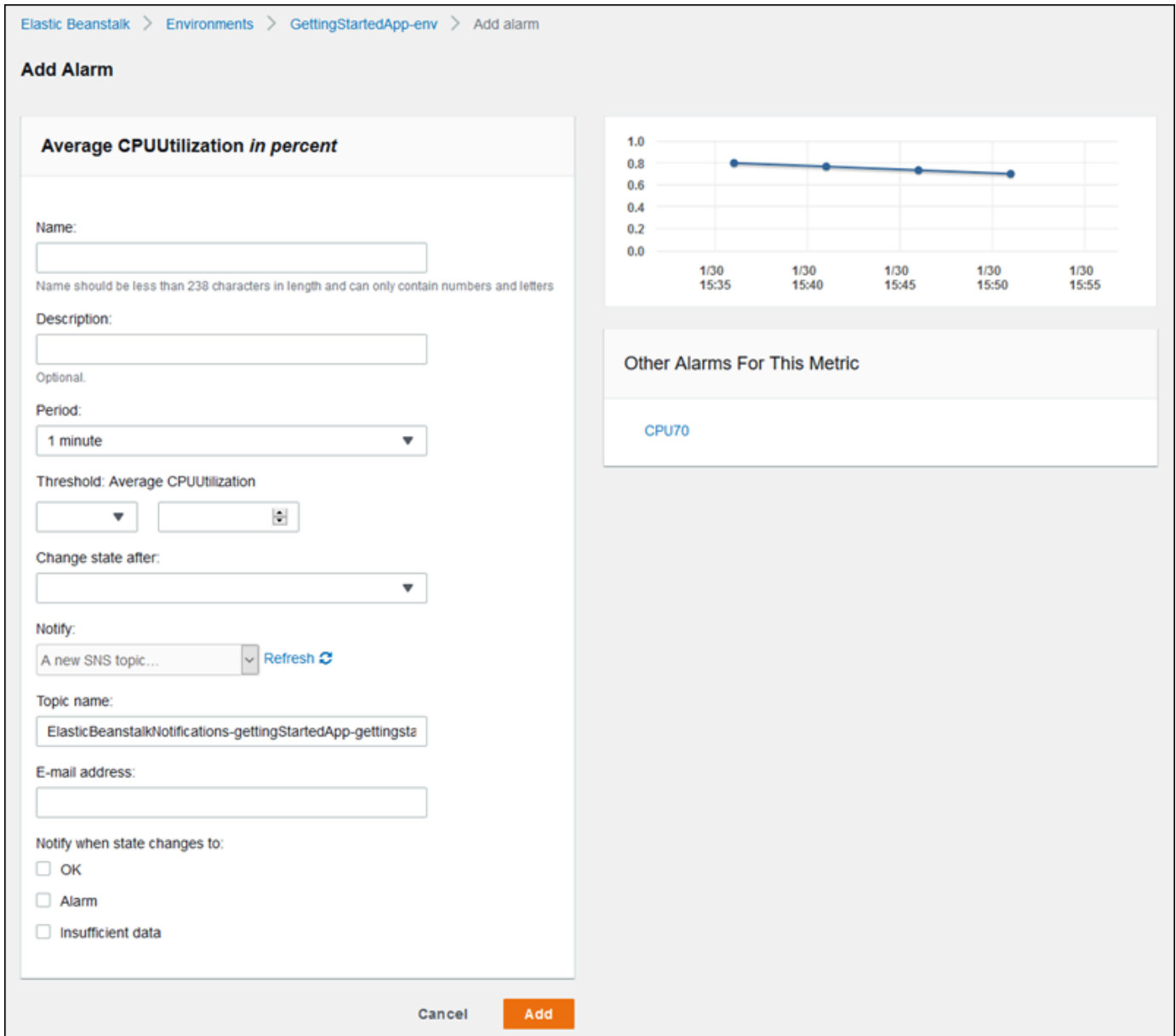
Para crear una alarma

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, seleccione Monitoring (Monitoreo).
4. Busque la métrica para la que desea crear una alarma y a continuación, seleccione  (alarm). Aparecerá la página Add alarm (Agregar alarma) .



The screenshot shows the 'Add Alarm' page in the AWS console. The breadcrumb trail is 'Elastic Beanstalk > Environments > GettingStartedApp-env > Add alarm'. The page title is 'Add Alarm'. The metric selected is 'Average CPUUtilization in percent'. The form fields are as follows:

- Name:** An empty text input field. Below it, a note states: 'Name should be less than 238 characters in length and can only contain numbers and letters'.
- Description:** An empty text input field. Below it, the text 'Optional.' is displayed.
- Period:** A dropdown menu set to '1 minute'.
- Threshold:** A dropdown menu set to 'Average CPUUtilization' and a numeric input field.
- Change state after:** A dropdown menu.
- Notify:** A dropdown menu set to 'A new SNS topic...' with a 'Refresh' button.
- Topic name:** A text input field containing 'ElasticBeanstalkNotifications-gettingStartedApp-gettingsta'.
- E-mail address:** An empty text input field.
- Notify when state changes to:** Three radio buttons: 'OK', 'Alarm', and 'Insufficient data'.

At the bottom of the form are 'Cancel' and 'Add' buttons. To the right of the form is a line graph showing the metric's value over time, with data points at 15:35, 15:40, 15:45, and 15:50. Below the graph is a section titled 'Other Alarms For This Metric' containing a link for 'CPU70'.

5. Especifique los datos de la alarma:
 - Name: nombre de la alarma.
 - Descripción (opcional): breve descripción sobre la finalidad de la alarma.
 - Period: intervalo de tiempo entre lecturas.

- **Threshold:** describe el comportamiento y el valor que la métrica debe superar para activar una alarma.
 - **Change state after:** periodo de tiempo tras el que se activa un cambio de estado de la alarma cuando se ha excedido un umbral.
 - **Notify (Notificar):** el tema de Amazon SNS que se notifica cuando una alarma cambia de estado.
 - **Notify when state changes to:**
 - **OK:** la métrica está dentro del umbral definido.
 - **Alarm:** la métrica ha excedido el umbral definido.
 - **Insufficient data:** la alarma acaba de iniciarse, la métrica no está disponible o no hay suficientes datos disponibles en la métrica para determinar el estado de la alarma.
6. Elija **Add (Agregar)**. El estado del entorno cambia a gris cuando el entorno se actualiza. Puede ver la alarma creada si selecciona **Alrms (Alarmas)** en el panel de navegación.

Visualización del historial de cambios de un entorno de Elastic Beanstalk

Puede utilizar la consola de administración de AWS para ver un historial de cambios de configuración realizados en los entornos de Elastic Beanstalk. Elastic Beanstalk recupera el historial de cambios de los eventos registrados en [AWS CloudTrail](#) y los muestra en una lista que puede explorar y filtrar fácilmente.

El panel **Change History (Historial de cambios)** muestra la siguiente información para los cambios realizados en los entornos:

- La fecha y la hora en que se realizó un cambio
- El usuario de IAM responsable de un cambio realizado
- La herramienta de código abierto (consola o interfaz de línea de comandos de Elastic Beanstalk [CLI de EB]) que se utilizó para realizar el cambio
- El parámetro de configuración y los nuevos valores que se establecieron

Los datos confidenciales que forman parte del cambio, como los nombres de los usuarios de la base de datos afectados por el cambio, no se muestran en el panel.

Para ver el historial de cambios, realice lo siguiente:

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Change history (Historial de cambios).

Date	IAM user	Event source	Environment	Configuration changes
2020-10-16T02:14:15Z	AIDACKCEVSQ6C2EXAMPLE	eb-cli/3.19.0 Python/3.8.5 Windows/10	GettingStartedApp-env	<ul style="list-style-type: none"> Changes made aws:autoscaling:launchconfiguration <ul style="list-style-type: none"> EC2KeyName : example-keyname
2020-10-16T02:10:59Z	AIDACKCEVSQ6C2EXAMPLE2	console.amazonaws.com	GettingStartedApp-env	<ul style="list-style-type: none"> Changes made aws:elasticbeanstalk:healthreporting:system <ul style="list-style-type: none"> ConfigDocument : - aws:elasticbeanstalk:sns:topics <ul style="list-style-type: none"> Notification Endpoint : jane@example.com
2020-10-16T02:02:50Z	AIDACKCEVSQ6C2EXAMPLE	eb-cli/3.19.0 Python/3.8.5 Windows/10	GettingStartedApp-env	-
2020-10-09T21:20:07Z	AIDACKCEVSQ6C2EXAMPLE2	console.amazonaws.com	Ruby-example-dev	<ul style="list-style-type: none"> Changes made aws:elasticbeanstalk:environment:proxy:staticfiles <ul style="list-style-type: none"> /public : Sensitive data removed
2020-10-09T21:17:01Z	AIDACKCEVSQ6C2EXAMPLE3	console.amazonaws.com	Ruby-example-dev	▶ Changes made
2020-10-09T19:05:21Z	AIDACKCEVSQ6C2EXAMPLE3	console.amazonaws.com	Ruby-example-dev	▶ Changes made
2020-10-09T19:03:04Z	AIDACKCEVSQ6C2EXAMPLE3	console.amazonaws.com	Ruby-example-dev	▶ Changes made
2020-10-09T19:00:04Z	AIDACKCEVSQ6C2EXAMPLE3	eb-cli/3.19.0 Python/3.8.5 Windows/10	Ruby-example-dev	-
2020-10-09T18:55:42Z	AIDACKCEVSQ6C2EXAMPLE3	eb-cli/3.19.0 Python/3.8.5 Windows/10	Ruby-example-dev	-

La página Change History (Historial de cambios) contiene una lista detallada con los cambios de configuración realizados en los entornos de Elastic Beanstalk. Puede pasar por la lista al seleccionar < (anterior) o > (siguiente), o un número de página específico. En la columna Configuration changes (Cambios de configuración), seleccione el ícono de flecha para alternar entre expandir y contraer la lista de cambios en el encabezado Changes made (Cambios realizados) . Utilice la barra de búsqueda para filtrar los resultados de la lista del historial de cambios. Puede introducir cualquier cadena para acotar la lista de los cambios que se muestran.

Tenga en cuenta lo siguiente acerca del filtrado de los resultados que se muestran:

- El filtro de búsqueda no distingue entre mayúsculas y minúsculas.
- Puede filtrar los cambios que se muestran en función de la información de la columna Configuration changes (Cambios de configuración), incluso cuando no esté visible debido a que se ha contraído dentro de Changes made (Cambios realizados).
- Solo puede filtrar los resultados que se muestran. Sin embargo, el filtro permanece incluso si selecciona ir a otra página para mostrar más resultados. Los resultados filtrados también se agregarán al conjunto de resultados de la página siguiente.

Los siguientes ejemplos muestran cómo se pueden filtrar los datos que se muestran en la pantalla anterior:

- Ingrese **GettingStartedApp-env** en el cuadro de búsqueda para acotar los resultados e incluir solo los cambios realizados en el entorno denominado GettingStartedApp-env.
- Ingrese **example3** en el cuadro de búsqueda para acotar los resultados e incluir solo los cambios realizados por los usuarios de IAM cuyo nombre de usuario contiene la cadena example3.
- Ingrese **2020-10** en el cuadro de búsqueda para acotar los resultados e incluir solo los cambios realizados durante el mes de octubre de 2020. Cambie el valor de búsqueda **2020-10-16** para filtrar más los resultados que se muestran e incluir solo los cambios realizados el día 16 de octubre de 2020.
- Ingrese **proxy:staticfiles** en el cuadro de búsqueda para acotar los resultados e incluir solo los cambios realizados en el espacio de nombres llamado aws:elasticbeanstalk:environment:proxy:staticfiles. Las filas que se muestran son el resultado del filtro. Esto es cierto incluso para los resultados que se contraen en Changes made (Cambios realizados).

Visualización de la transmisión de eventos de un entorno de Elastic Beanstalk

Puede usar la consola de administración de AWS para obtener acceso a los eventos y notificaciones relacionados con su aplicación.

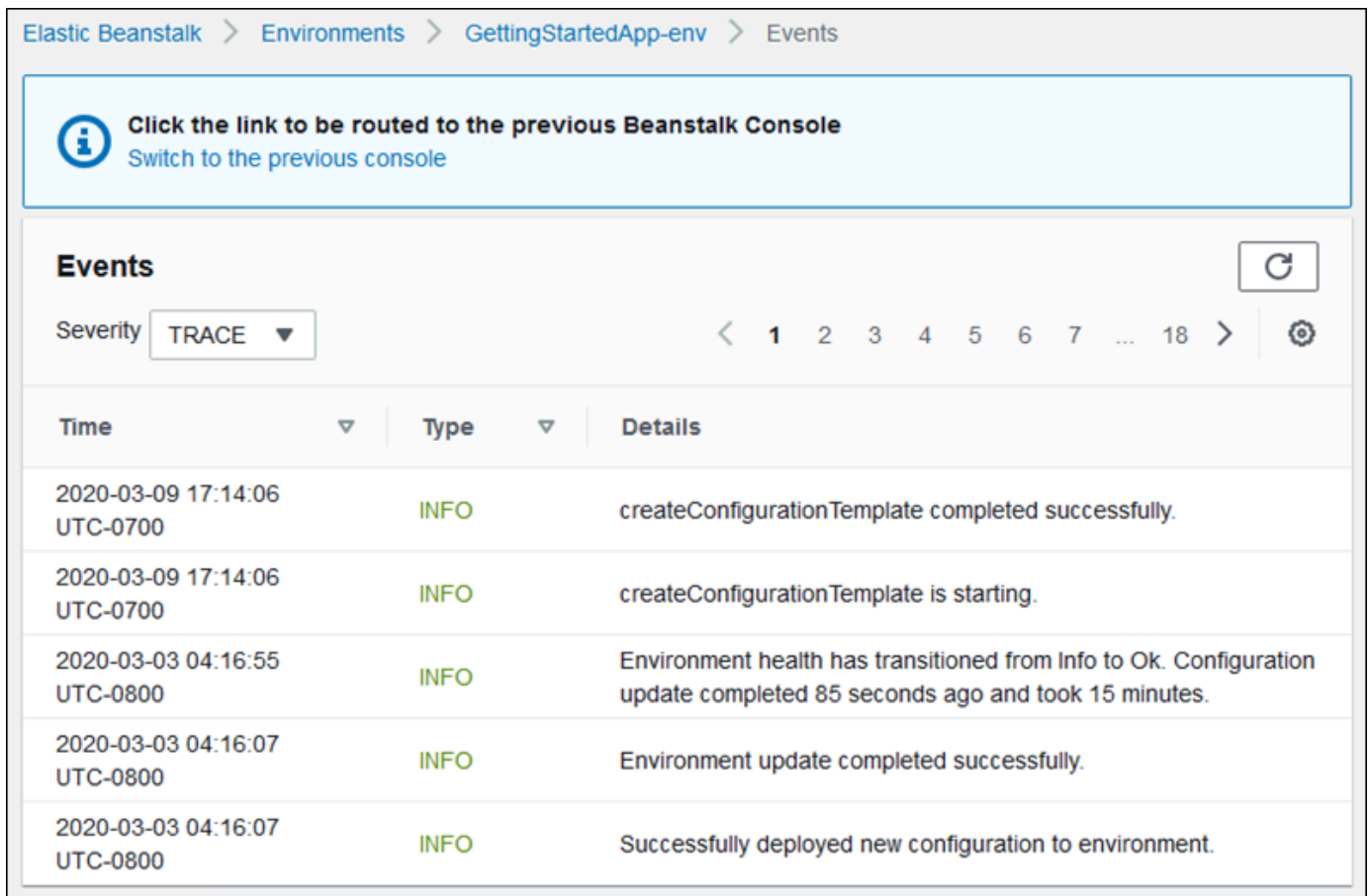
Para ver eventos

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Events (Eventos).



Elastic Beanstalk > Environments > GettingStartedApp-env > Events

Click the link to be routed to the previous Beanstalk Console
Switch to the previous console

Events ↻

Severity < 1 2 3 4 5 6 7 ... 18 > ⚙️

Time	Type	Details
2020-03-09 17:14:06 UTC-0700	INFO	createConfigurationTemplate completed successfully.
2020-03-09 17:14:06 UTC-0700	INFO	createConfigurationTemplate is starting.
2020-03-03 04:16:55 UTC-0800	INFO	Environment health has transitioned from Info to Ok. Configuration update completed 85 seconds ago and took 15 minutes.
2020-03-03 04:16:07 UTC-0800	INFO	Environment update completed successfully.
2020-03-03 04:16:07 UTC-0800	INFO	Successfully deployed new configuration to environment.

La página Events (Eventos) muestra una lista de todos los eventos que se han registrado para el entorno. Puede recorrer la lista eligiendo < (anterior), > (siguiente) o números de página. Puede filtrar por tipo de evento mediante la lista desplegable Severity (Gravedad).

La [CLI de EB](#) y la [AWS CLI](#) proporcionan comandos para recuperar eventos. Si administra su entorno con la CLI de EB, use [eb events](#) para mostrar una lista de eventos. Este comando también tiene una opción `--follow` que continúa mostrando eventos hasta que pulsa Ctrl+C para detener la salida.

Para obtener eventos a través de la AWS CLI, utilice el comando `describe-events` y especifique el entorno por nombre o ID:

```
$ aws elasticbeanstalk describe-events --environment-id e-gbjzqccra3
{
  "Events": [
    {
      "ApplicationName": "elastic-beanstalk-example",
```

```
"EnvironmentName": "elasticBeanstalkExa-env",
"Severity": "INFO",
"RequestId": "a4c7bfd6-2043-11e5-91e2-9114455c358a",
"Message": "Environment update completed successfully.",
"EventDate": "2015-07-01T22:52:12.639Z"
},
...
```

Para obtener más información sobre las herramientas de línea de comandos, consulte [Herramientas](#).

Listado y conexión a instancias de servidor

Puede ver una lista de instancias de Amazon EC2 que ejecutan su entorno de AWS Elastic Beanstalk aplicaciones a través de la consola de Elastic Beanstalk. Puede conectarse a las instancias mediante cualquier cliente SSH. Puede conectarse a las instancias que ejecutan Windows a través de Escritorio remoto.

Notas sobre determinados entornos de desarrollo:

- Para obtener más información sobre cómo enumerar las instancias de servidor y conectarse a ellas mediante el, consulte. AWS Toolkit for Eclipse [Listado y conexión a instancias de servidor](#)
- Para obtener más información sobre cómo enumerar las instancias de servidor y conectarse a ellas mediante el AWS Toolkit for Visual Studio, consulte [Listado y conexión a instancias de servidor](#).

Important

Para poder acceder a las instancias de Amazon EC2 aprovisionadas de Elastic Beanstalk, debe crear un par de claves de Amazon EC2 y configurar las instancias de Amazon EC2 aprovisionadas de Elastic Beanstalk para utilizar el par de claves de Amazon EC2. Para configurar el par de claves de Amazon EC2 puede usar la [consola de administración de AWS](#). Si desea obtener instrucciones acerca de cómo crear un par de claves para Amazon EC2, consulte la Guía de introducción de Amazon EC2. Para obtener más información sobre cómo configurar las instancias de Amazon EC2 para utilizar un par de claves de Amazon EC2, consulte [EC2 key pair](#).

De forma predeterminada, Elastic Beanstalk; no permite conexiones remotas a instancias EC2 en un contenedor de Windows, excepto para las que se encuentran en contenedores de Windows heredados. (Elastic Beanstalk configura las instancias EC2 de contenedores

Windows heredados para que utilicen el puerto 3389 para las conexiones RDP). Puede habilitar las conexiones remotas a las instancias EC2 que ejecutan Windows añadiendo una regla a un grupo de seguridad que autorice el tráfico entrante a dichas instancias. Es absolutamente recomendable que elimine la regla cuando finalice la conexión remota. Puede añadir la regla de nuevo la próxima vez que necesite iniciar sesión de forma remota. Para obtener más información, consulte [Adding a Rule for Inbound RDP Traffic to a Windows Instance](#) y [Connect to Your Windows Instance](#) en la Guía del usuario de Amazon Elastic Compute Cloud para Microsoft Windows.

Para ver las instancias de Amazon EC2 de un entorno y conectarse a ellas

1. Abra la consola de Amazon EC2 en <https://console.aws.amazon.com/ec2/>.
2. En el panel de navegación de la consola, elija Load Balancers (Balanceadores de carga).

The screenshot shows the AWS Management Console interface. On the left, the navigation menu is visible with 'LOAD BALANCING' expanded and 'Load Balancers' selected. The main content area shows the 'Resources' section for the US East (N. Virginia) region, listing various EC2 resources. A 'Create Instance' section is also visible at the bottom.

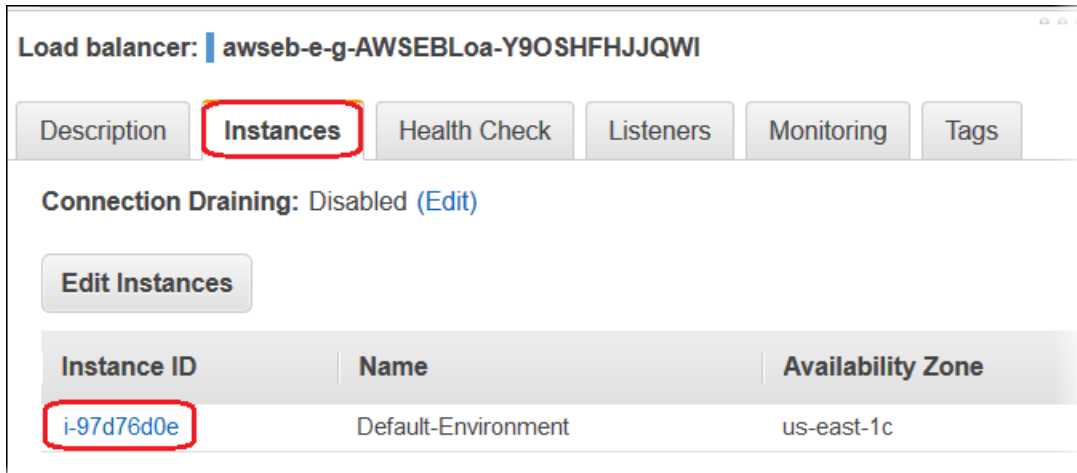
Resources	
You are using the following Amazon EC2 resources in the US East (N. Virginia) region:	
2 Running Instances	0 Elastic IPs
0 Dedicated Hosts	0 Snapshots
2 Volumes	2 Load Balancers
0 Key Pairs	6 Security Groups
0 Placement Groups	

3. Los balanceadores de carga creados por Elastic Beanstalk tienen awseb en el nombre. Busque el balanceador de carga para su entorno y haga clic en él.

The screenshot shows the 'Create Load Balancer' page in the AWS Management Console. A table lists existing load balancers. The first entry, 'awseb-e-g-AWSEBLoa-Y9O...', is highlighted with a red box.

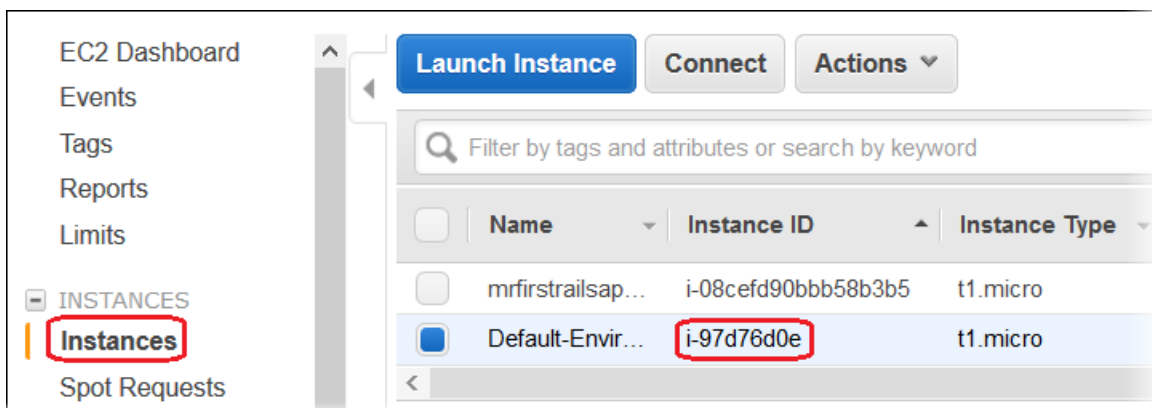
Name	DNS name	State
awseb-e-g-AWSEBLoa-Y9O...	awseb-e-g-AWSEBLoa-Y9O...	
awseb-e-w-AWSEBLoa-2XLI...	awseb-e-w-AWSEBLoa-2XLI...	

- Elija la pestaña Instancias (Instancias) en el panel inferior de la consola.



Se muestra una lista de las instancias que usa el balanceador de carga para su entorno de Elastic Beanstalk. Anote el ID de la instancia a la que desea conectarse.

- En el panel de navegación de la consola de Amazon EC2, elija Instancias (Instancias) y busque el ID de la instancia en la lista.



- Haga clic con el botón derecho en el ID de instancia de la instancia Amazon EC2 que se ejecuta en el balanceador de carga de su entorno y después seleccione Connect (Conectar) en el menú contextual.
- Anote la dirección DNS pública de la instancia que aparece en la pestaña Description.
- Conéctate a una instancia que ejecute Linux mediante el cliente SSH de tu elección y, a continuación, escribe `ssh -i .ec2/mykeypair.pem ec2-user@<public-DNS - >.of-the-instance`

Para obtener más información sobre la conexión a una instancia Linux de Amazon EC2, consulte [Introducción a las instancias de Linux de Amazon EC2 en la Guía del usuario](#) de Amazon EC2.

Si su entorno de Elastic Beanstalk [utiliza la plataforma .NET en Windows Server, consulte Introducción a las instancias Windows de Amazon EC2 en la Guía del usuario de Amazon EC2](#).

Visualización de registros de instancias de Amazon EC2 en su entorno de Elastic Beanstalk

Las instancias EC2 de Amazon en el entorno de Elastic Beanstalk generan registros que se pueden consultar para solucionar problemas relacionados con la aplicación o los archivos de configuración. Los registros creados por el servidor web, el servidor de aplicaciones, los scripts de la plataforma Elastic Beanstalk y AWS CloudFormation se almacenan localmente en instancias individuales. Puede recuperarlos de forma sencilla a través de la [consola de administración del entorno](#) o la CLI de EB. También puede configurar su entorno para transmitir registros a Amazon CloudWatch Logs en tiempo real.

Los registros de finalización son las últimas 100 líneas de los archivos de registro que se utilizan con más frecuencia: los registros operativos de Elastic Beanstalk y los registros del servidor web o del servidor de aplicaciones. Cuando se solicita un registro de finalización utilizando la consola de administración del entorno o eb logs, una instancia del entorno concatena las entradas del registro más frecuentes en un archivo de texto y lo carga a Amazon S3.

Los registros de paquete son registros completos para un mayor conjunto de archivos de registro, como los registros yum y cron o diversos registros de AWS CloudFormation. Cuando solicita los registros de paquete, una instancia del entorno empaqueta todos los archivos de registro en un archivo ZIP y los carga a Amazon S3.

Note

Las plataformas de Elastic Beanstalk Windows Server no admiten registros de paquetes.

Para cargar los registros rotados en Amazon S3, las instancias del entorno deben tener un [perfil de instancia](#) con permiso para escribir en el bucket de Elastic Beanstalk de Amazon S3. Estos permisos están incluidos en el perfil de instancia predeterminado que Elastic Beanstalk le pide que cree cuando lanza un entorno en la consola de Elastic Beanstalk por primera vez.

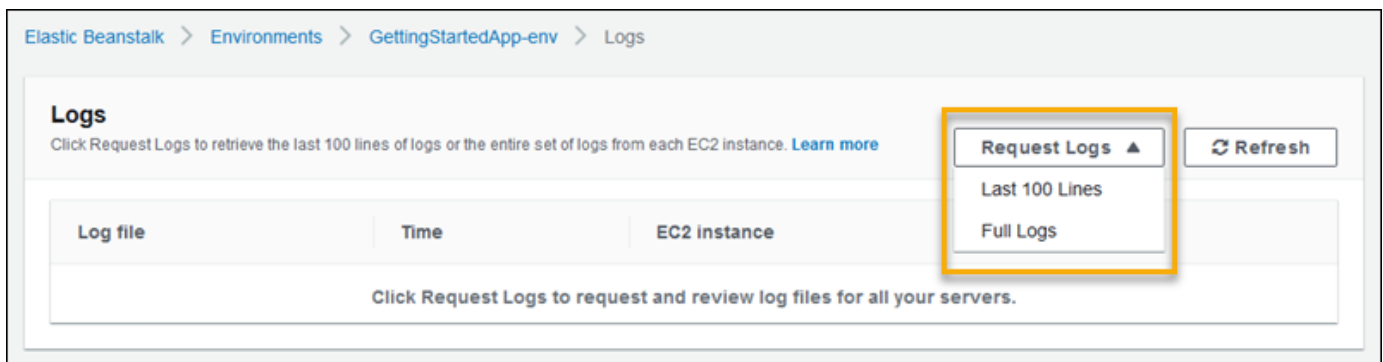
Para recuperar los registros de instancias:

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Logs (Registros).
4. Seleccione Request logs (Solicitar registro), y, a continuación, elija el tipo de registros que desea recuperar. Para obtener los registros de finalización, seleccione Last 100 Lines. Para obtener los registros de paquete, seleccione Full Logs (Registros completos).



5. Cuando Elastic Beanstalk termine de recuperar sus registros, elija Descargar.

Elastic Beanstalk almacena los registros de finalización y de paquetes en un bucket de Amazon S3 y genera una URL de Amazon S3 prefirmada que puede utilizar para acceder a sus registros. Elastic Beanstalk elimina los archivos de Amazon S3 después de una duración de 15 minutos.

Warning

Cualquier persona en posesión de la URL prefirmada de Amazon S3 puede acceder a los archivos antes de que se eliminen. Haga que la URL esté disponible únicamente para partes de confianza.

Note

La política de usuario debe tener el permiso `s3:DeleteObject`. Elastic Beanstalk utiliza sus permisos de usuario para eliminar los registros de Amazon S3.

Si desea conservar los registros, puede configurar el entorno para que los publique en Amazon S3 automáticamente después de rotarlos. Para habilitar la rotación de registros en Amazon S3, siga el procedimiento en [Configuración de la visualización de registros de instancias](#). Las instancias del entorno intentarán cargar los logs que se han rotado una vez cada hora.

Si la aplicación genera registros en una ubicación que no forma parte de la configuración predeterminada de la plataforma del entorno, puede extender la configuración predeterminada con los archivos de configuración ([.ebextensions](#)). Puede agregar los archivos log de la aplicación a los logs de finalización, los logs de paquete o la rotación de logs.

Para transmitir los registros en tiempo real y almacenarlos a largo plazo, configure el entorno para que pueda [transmitir registros a Amazon CloudWatch Logs](#).

Secciones

- [Ubicación de registro en instancias de Amazon EC2](#)
- [Ubicación de registro en Amazon S3](#)
- [Configuración de rotación de registros en Linux](#)
- [Ampliación de la configuración predeterminada de la tarea de registro](#)
- [Transmisión de archivos de registro a Amazon CloudWatch Logs](#)

Ubicación de registro en instancias de Amazon EC2

Los registros se almacenan en ubicaciones estándar de las instancias Amazon EC2 de su entorno. Elastic Beanstalk genera los siguientes registros.

Amazon Linux 2

- `/var/log/eb-engine.log`

Amazon Linux AMI (AL1)

Note

El [18 de julio de 2022](#), Elastic Beanstalk estableció el estado de todas las ramificaciones de la plataforma basadas en la AMI de Amazon Linux (AL1) como retirado. Para obtener más información sobre la migración a una ramificación de la plataforma Amazon Linux 2023 actual y totalmente compatible, consulte [Migración de su aplicación de Linux de Elastic Beanstalk a Amazon Linux 2023 o Amazon Linux 2](#).

- `/var/log/eb-activity.log`
- `/var/log/eb-commandprocessor.log`

Windows Server

- `C:\Program Files\Amazon\ElasticBeanstalk\logs\`
- `C:\cfn\log\cfn-init.log`

Estos registros contienen mensajes sobre las actividades de implementación, como mensajes relacionados con los archivos de configuración ([.ebextensions](#)).

Cada aplicación y cada servidor web guarda los logs en su propia carpeta:

- Apache: `/var/log/httpd/`
- IIS: `C:\inetpub\wwwroot\`
- Node.js: `/var/log/nodejs/`
- nginx: `/var/log/nginx/`
- Passenger: `/var/app/support/logs/`
- Puma: `/var/log/puma/`
- Python: `/opt/python/log/`
- Tomcat: `/var/log/tomcat/`

Ubicación de registro en Amazon S3

Cuando solicita los registros de finalización o de paquete del entorno o cuando las instancias cargan registros rotados, se guardan en el bucket de Elastic Beanstalk de Amazon S3. Elastic Beanstalk

crea un bucket denominado `elasticbeanstalk-region-account-id` para cada región de AWS en la que se crean entornos. En este bucket, los logs se guardan en la ruta `resources/environments/logs/logtype/environment-id/instance-id`.

Por ejemplo, los registros de una instancia `i-0a1fd158` situada en el entorno de Elastic Beanstalk `e-mpcwnwheky` de la región de AWS `us-west-2` de la cuenta `123456789012`, se almacenan en las siguientes ubicaciones:

- Registros de finalización:

```
s3://elasticbeanstalk-us-west-2-123456789012/resources/environments/logs/tail/e-mpcwnwheky/i-0a1fd158
```

- Registros de paquete:

```
s3://elasticbeanstalk-us-west-2-123456789012/resources/environments/logs/bundle/e-mpcwnwheky/i-0a1fd158
```

- Registros rotados:

```
s3://elasticbeanstalk-us-west-2-123456789012/resources/environments/logs/publish/e-mpcwnwheky/i-0a1fd158
```

Note

Encontrará el ID del entorno en la consola de administración del entorno.

Elastic Beanstalk elimina automáticamente los registros de finalización y de paquetes de Amazon S3 15 minutos después de su creación. Los registros rotados se mantienen hasta que se eliminan o se transfieren a S3 Glacier.

Configuración de rotación de registros en Linux

En plataformas Linux, Elastic Beanstalk utiliza `logrotate` para rotar registros periódicamente. Si se ha configurado de este modo, cuando un registro ha rotado localmente, la tarea de rotación de registros lo selecciona y lo carga en Amazon S3. Los registros que han rotado localmente no aparecen de forma predeterminada en los registros de finalización o de paquete.

Puede encontrar los archivos de configuración de Elastic Beanstalk para `logrotate` en `/etc/logrotate.elasticbeanstalk.hourly/`. Esta configuración de rotación es específica de la

plataforma y puede cambiar en futuras versiones de la plataforma. Para obtener más información sobre los ajustes disponibles y configuraciones de ejemplo, ejecute `man logrotate`.

Los archivos de configuración se invocan mediante trabajos cron en `/etc/cron.hourly/`. Para obtener más información acerca de `cron`, ejecute `man cron`.

Ampliación de la configuración predeterminada de la tarea de registro

Elastic Beanstalk utiliza archivos que se encuentran en las subcarpetas de `/opt/elasticbeanstalk/tasks` (Linux) o de `C:\Program Files\Amazon\ElasticBeanstalk\config` (Windows Server) de la instancia de Amazon EC2 para configurar tareas en los registros de finalización, los registros de paquete y la rotación de registros.

En Amazon Linux:

- Registros de finalización:

```
/opt/elasticbeanstalk/tasks/taillogs.d/
```

- Registros de paquete:

```
/opt/elasticbeanstalk/tasks/bundlelogs.d/
```

- Registros rotados:

```
/opt/elasticbeanstalk/tasks/publishlogs.d/
```

En Windows Server:

- Registros de finalización:

```
c:\Program Files\Amazon\ElasticBeanstalk\config\taillogs.d\
```

- Registros rotados:

```
c:\Program Files\Amazon\ElasticBeanstalk\config\publogs.d\
```

Por ejemplo, el archivo `eb-activity.conf` en Linux agrega dos archivos de registro a la tarea de registros de finalización.

`/opt/elasticbeanstalk/tasks/taillogs.d/eb-activity.conf`

```
/var/log/eb-commandprocessor.log
```



```
/var/log/eb-activity.log
```

Puede utilizar archivos de configuración del entorno ([.ebextensions](#)) para agregar sus propios archivos `.conf` a estas carpetas. Un archivo `.conf` muestra los archivos de registro específicos de su aplicación que Elastic Beanstalk agrega a las tareas de archivo de registro.

Utilice la sección [files](#) para agregar archivos de configuración a las tareas que desea modificar. Por ejemplo, el siguiente texto de configuración inserta un archivo de configuración de log en cada instancia del entorno. Este archivo de configuración de log, `cloud-init.conf`, añade `/var/log/cloud-init.log` a los registros de cola.

```
files:
  "/opt/elasticbeanstalk/tasks/taillogs.d/cloud-init.conf" :
    mode: "000755"
    owner: root
    group: root
    content: |
      /var/log/cloud-init.log
```

Agregue este texto a un archivo que tenga la extensión de nombre de archivo `.config` en el paquete de código fuente dentro de una carpeta llamada `.ebextensions`.

```
~/workspace/my-app
|-- .ebextensions
|   |-- tail-logs.config
|-- index.php
`-- styles.css
```

En las plataformas Linux también puede utilizar caracteres comodín en las configuraciones de tareas de registro. Este archivo de configuración agrega todos los archivos con la extensión de nombre de archivo `.log` de la carpeta `log` de la raíz de la aplicación a los logs de paquete.

```
files:
  "/opt/elasticbeanstalk/tasks/bundlelogs.d/applogs.conf" :
    mode: "000755"
    owner: root
    group: root
    content: |
      /var/app/current/log/*.log
```

Las configuraciones de tarea de registro no admiten comodines en las plataformas de Windows.

Note

Para familiarizarse con los procedimientos de personalización de registros, puede implementar una aplicación de ejemplo mediante la [CLI de EB](#). Para ello, la CLI de EB crea un directorio de aplicación local que contiene un subdirectorio de `.ebextensions` con una configuración de ejemplo. También puede utilizar los archivos de registro de la aplicación de ejemplo para explorar la función de recuperación de registros descrita en este tema. Para obtener más información acerca de cómo crear una aplicación de ejemplo con la CLI de EB, consulte los [conceptos básicos de la CLI de EB](#).

Para obtener más información sobre cómo usar archivos de configuración, consulte [Personalización avanzada de entornos con archivos de configuración \(.ebextensions\)](#).

De forma parecida a la ampliación de registros de finalización y los registros de paquete, puede ampliar la rotación de registros con un archivo de configuración. Siempre que Elastic Beanstalk rote sus propios registros y los cargue a Amazon S3, también rota y carga los registros adicionales. La extensión de rotación de registros se comporta de forma distinta en función del sistema operativo de la plataforma. En las siguientes secciones se describen los dos casos.

Ampliación de la rotación de registros en Linux

Como se explica en [Configuración de rotación de registros en Linux](#), Elastic Beanstalk utiliza `logrotate` para rotar registros en plataformas Linux. Al configurar los archivos de registro de su aplicación para la rotación de registros, la aplicación no necesita crear copias de los archivos de registro. Elastic Beanstalk configura `logrotate` para crear una copia de los archivos de registro de la aplicación para cada rotación. Por lo tanto, la aplicación debe mantener archivos de registro desbloqueados cuando no escriba en ellos de forma activa.

Ampliación de la rotación de registros en Windows Server

En Windows Server, al configurar los archivos de registro de su aplicación para la rotación de registros, la aplicación debe rotarlos periódicamente. Elastic Beanstalk busca los archivos con los nombres que comiencen por el patrón que ha configurado y los elige para cargarlos a Amazon S3. Además, se ignoran los puntos en el nombre de archivo y Elastic Beanstalk considera que el nombre hasta el punto será el nombre de archivo de registro base.

Elastic Beanstalk carga todas las versiones de un archivo de registro base excepto la más reciente, puesto que considera que es el archivo de registro de aplicación activo, que podría estar bloqueado. Por lo tanto, la aplicación puede mantener el archivo de registro bloqueado entre las rotaciones.

Ejemplo: su aplicación escribe en un archivo de registro denominado `my_log.log` y especifica este nombre en su archivo `.conf`. La aplicación rota de forma periódica el archivo. Durante el ciclo de rotación de Elastic Beanstalk, encuentra los siguientes archivos en la carpeta del archivo de registro: `my_log.log`, `my_log.0800.log`, `my_log.0830.log`. Elastic Beanstalk los considera a todos como versiones del nombre base `my_log`. El archivo `my_log.log` tiene la hora de modificación más reciente, por lo que Elastic Beanstalk carga solo los otros dos archivos, `my_log.0800.log` y `my_log.0830.log`.

Transmisión de archivos de registro a Amazon CloudWatch Logs

Puede configurar su entorno para transmitir registros a Amazon CloudWatch Logs en la consola de Elastic Beanstalk o mediante [las opciones de configuración](#). Con Amazon CloudWatch Logs, cada instancia del entorno transmite los registros a grupos de registros que se pueden configurar para que se mantengan durante semanas o años, incluso después de que el entorno haya finalizado.

El conjunto de registros transmitidos varía en función del entorno, pero siempre incluye los registros `eb-engine.log` y de acceso del servidor proxy de nginx o Apache que se ejecuta delante de la aplicación.

Puede configurar la transmisión de registros en la consola de Elastic Beanstalk [durante la creación de entorno](#) o [para un entorno existente](#). En el siguiente ejemplo, los registros se guardan durante un máximo de siete días, incluso cuando el entorno ha terminado.

Instance log streaming to CloudWatch Logs

Configure the instances in your environment to stream logs to CloudWatch Logs. You can set the retention to up to ten years and configure Elastic Beanstalk to delete the logs when you terminate your environment.

Log groups
[/aws/elasticbeanstalk/GettingStartedApp-env](#)

Log streaming
(Standard CloudWatch charges apply.)
 Enabled

Retention
7 days

Lifecycle
Keep logs after terminating environment

El siguiente [archivo de configuración](#) habilita la transmisión de registros con 180 días de retención, incluso si el entorno ha terminado.

Example `.ebextensions/log-streaming.config`

```
option_settings:
  aws:elasticbeanstalk:cloudwatch:logs:
    StreamLogs: true
    DeleteOnTerminate: false
    RetentionInDays: 180
```

Uso de Elastic Beanstalk con otros servicios de AWS

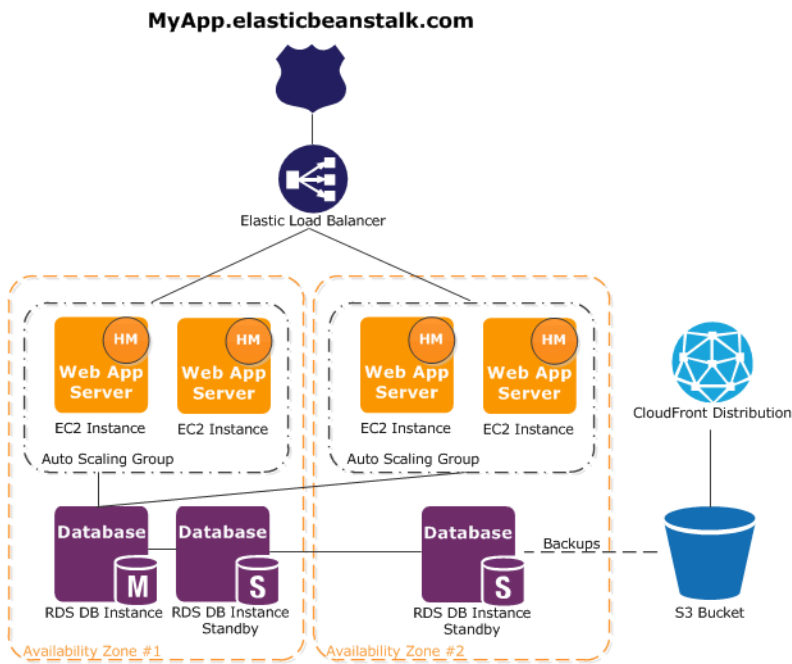
Para implementar los entornos de su aplicación, Elastic Beanstalk administra recursos de otros servicios de AWS o utiliza su funcionalidad. Además, Elastic Beanstalk se integra con otros servicios de AWS aunque no los usa directamente como parte de los entornos. Los temas de esta sección describen numerosas maneras de usar estos servicios adicionales con la aplicación de Elastic Beanstalk.

Temas

- [Información general sobre la arquitectura](#)
- [Usar Elastic Beanstalk con Amazon CloudFront](#)
- [Registro de llamadas a la API de Elastic Beanstalk con AWS CloudTrail](#)
- [Uso de Elastic Beanstalk con Amazon CloudWatch](#)
- [Uso de Elastic Beanstalk con Amazon CloudWatch Logs](#)
- [Usar Elastic Beanstalk con Amazon EventBridge](#)
- [Búsqueda y seguimiento de recursos de Elastic Beanstalk con AWS Config](#)
- [Uso de Elastic Beanstalk con Amazon DynamoDB](#)
- [Uso de Elastic Beanstalk con Amazon ElastiCache](#)
- [Uso de Elastic Beanstalk con Amazon Elastic File System](#)
- [Uso de Elastic Beanstalk con AWS Identity and Access Management](#)
- [Uso de Elastic Beanstalk con Amazon RDS](#)
- [Usar Elastic Beanstalk con Amazon S3](#)
- [Uso de Elastic Beanstalk con Amazon VPC](#)

Información general sobre la arquitectura

El siguiente diagrama muestra un ejemplo de arquitectura de Elastic Beanstalk en varias zonas de disponibilidad que trabajan con otros productos de AWS, tales como Amazon CloudFront, Amazon Simple Storage Service (Amazon S3) y Amazon Relational Database Service (Amazon RDS).



Para planificar la tolerancia a errores, se recomienda tener N+1 instancias de Amazon EC2 y distribuir sus instancias en varias zonas de disponibilidad. En el caso improbable de que una zona de disponibilidad deje de funcionar, el resto de las instancias de Amazon EC2 se ejecutarán en otra zona de disponibilidad. Puede ajustar el Amazon EC2 Auto Scaling para permitir un número mínimo de instancias, así como varias zonas de disponibilidad. Para obtener instrucciones al respecto, consulte [Grupo de Auto Scaling para su entorno de Elastic Beanstalk](#). Para obtener más información sobre la creación de aplicaciones tolerantes a errores, consulte [Creación de aplicaciones tolerantes a errores en AWS](#).

En las siguientes secciones se analiza con más detalle la integración con Amazon CloudFront, Amazon CloudWatch, Amazon DynamoDB, Amazon ElastiCache, Amazon RDS, Amazon Route 53, Amazon Simple Storage Service, Amazon VPC y IAM.

Usar Elastic Beanstalk con Amazon CloudFront

Amazon CloudFront es un servicio web que agiliza la distribución de contenido web estático y dinámico a los usuarios finales, por ejemplo, archivos .html, .css, .php, imágenes y archivos multimedia. CloudFront entrega el contenido a través de una red mundial de ubicaciones de borde. Cuando un usuario final solicita contenido que se distribuye con CloudFront, el usuario se redirige a la ubicación de borde que ofrece la mínima latencia para que el contenido se entregue con el máximo desempeño posible. Si el contenido ya se encuentra en dicha ubicación de borde, CloudFront lo entrega inmediatamente. Si el contenido no se encuentra actualmente en dicha ubicación de borde,

CloudFront lo recupera de un bucket de Amazon S3 o de un servidor HTTP (por ejemplo, un servidor web) que el usuario haya identificado como la fuente para la versión definitiva del contenido.

Después de crear e implementar su aplicación de Elastic Beanstalk, puede registrarse en CloudFront y comenzar a utilizar CloudFront para distribuir su contenido. Obtenga más información sobre CloudFront en la [Guía para desarrolladores de Amazon CloudFront](#).

Registro de llamadas a la API de Elastic Beanstalk con AWS CloudTrail

Elastic Beanstalk está integrado con AWS CloudTrail, un servicio que registra las acciones de usuarios, roles o servicios de AWS en Elastic Beanstalk. CloudTrail captura todas las llamadas de API para Elastic Beanstalk como eventos, incluidas las llamadas desde la consola de Elastic Beanstalk, desde la CLI de EB y desde su código a las API de Elastic Beanstalk. Si crea un registro de seguimiento, puede habilitar la entrega continua de eventos de CloudTrail a un bucket de Amazon S3, incluidos los eventos de AWS IoT Analytics. Si no configura un registro de seguimiento, puede ver los eventos más recientes de la consola de CloudTrail en el Event history (Historial de eventos). Mediante la información recopilada por CloudTrail, puede determinar la solicitud que se realizó a Elastic Beanstalk, la dirección IP desde la que se realizó, quién la realizó y cuándo, etc.

Para obtener más información sobre CloudTrail, consulte la [Guía del usuario de AWS CloudTrail](#).

Información de Elastic Beanstalk en el historial de CloudTrail

CloudTrail se habilita en su cuenta de AWS cuando la crea. Cuando se produce una actividad en Elastic Beanstalk, esta se registra en un evento de CloudTrail junto con los eventos de los demás servicios de AWS y se agrega al historial de eventos. Puede ver, buscar y descargar los últimos eventos de la cuenta de AWS. Para obtener más información, consulte [Ver eventos con el historial de eventos de CloudTrail](#).

Para mantener un registro continuo de los eventos de la cuenta de AWS, incluidos los eventos de Elastic Beanstalk, cree un registro de seguimiento. Un registro de seguimiento permite a CloudTrail enviar archivos de registro a un bucket de Amazon S3. De forma predeterminada, cuando se crea un registro de seguimiento en la consola, el registro de seguimiento se aplica a todas las regiones. El registro de seguimiento registra los eventos de todas las regiones de la partición de AWS y envía los archivos de registro al bucket de Amazon S3 especificado. También es posible configurar otros servicios de AWS para analizar en profundidad y actuar en función de los datos de eventos recopilados en los registros de CloudTrail. Para obtener más información, consulte:

- [Introducción a la creación de registros de seguimiento](#)
- [Consulte Servicios e integraciones compatibles con CloudTrail.](#)
- [Configuración de notificaciones de Amazon SNS para CloudTrail](#)
- [Recibir archivos de registro de CloudTrail de varias regiones](#) y [Recibir archivos de registro de CloudTrail de varias cuentas](#)

CloudTrail registra todas las acciones de Elastic Beanstalk, que se documentan en la [Referencia de la API de AWS Elastic Beanstalk](#). Por ejemplo, las llamadas a las acciones `DescribeApplications`, `UpdateEnvironment` y `ListTagsForResource` generan entradas en los archivos de log de CloudTrail.

Cada entrada de registro o evento contiene información acerca de quién generó la solicitud. La información de identidad del usuario le ayuda a determinar lo siguiente:

- Si la solicitud se realizó con las credenciales raíz o del usuario de IAM.
- Si la solicitud se realizó con credenciales de seguridad temporales de un rol o fue un usuario federado.
- Si la solicitud la realizó otro servicio de AWS.

Para obtener más información, consulte el [Elemento `userIdentity` de CloudTrail](#).

Descripción de las entradas del archivo de registro de Elastic Beanstalk

Un registro de seguimiento es una configuración que permite la entrega de eventos como archivos de registro en un bucket de Amazon S3 que especifique. Los archivos log de CloudTrail pueden contener una o varias entradas de log. Un evento representa una única solicitud de cualquier origen e incluye información acerca de la acción solicitada, la fecha y la hora de la acción, los parámetros de la solicitud, etcétera. Los archivos de registro de CloudTrail no son un rastro de la stack ordenado de las llamadas a las API públicas, por lo que no aparecen en ningún orden específico.

En el ejemplo siguiente se muestra una entrada de registro de CloudTrail que ilustra la acción `UpdateEnvironment` llamada por un usuario de IAM con el nombre `intern` para el entorno `sample-env` en la aplicación `sample-app`.

```
{
  "Records": [{
```



```
"eventVersion": "1.05",
"userIdentity": {
  "type": "IAMUser",
  "principalId": "AIXDAYQEXAMPLEUMLYNGL",
  "arn": "arn:aws:iam::123456789012:user/intern",
  "accountId": "123456789012",
  "accessKeyId": "ASXIAGXEXAMPLEQULKXNV",
  "userName": "intern",
  "sessionContext": {
    "attributes": {
      "mfaAuthenticated": "false",
      "creationDate": "2016-04-22T00:23:24Z"
    }
  },
  "invokedBy": "signin.amazonaws.com"
},
"eventTime": "2016-04-22T00:24:14Z",
"eventSource": "elasticbeanstalk.amazonaws.com",
"eventName": "UpdateEnvironment",
"awsRegion": "us-west-2",
"sourceIPAddress": "255.255.255.54",
"userAgent": "signin.amazonaws.com",
"requestParameters": {
  "applicationName": "sample-app",
  "environmentName": "sample-env",
  "optionSettings": []
},
"responseElements": null,
"requestID": "84ae9ecf-0280-17ce-8612-705c7b132321",
"eventID": "e48b6a08-c6be-4a22-99e1-c53139cbfb18",
"eventType": "AwsApiCall",
"recipientAccountId": "123456789012"
}]
}
```

Uso de Elastic Beanstalk con Amazon CloudWatch

Amazon CloudWatch le permite monitorear, administrar y publicar diversas métricas, así como configurar acciones de alarma en función de los datos de esas métricas. La característica de monitorización de Amazon CloudWatch le permite recopilar, analizar y ver las métricas del sistema y de la aplicación para que pueda tomar decisiones empresariales y operativas de manera más rápida y con mayor confianza.

Puede utilizar Amazon CloudWatch para recopilar métricas sobre sus recursos de Amazon Web Services (AWS), tal como el rendimiento de las instancias de Amazon EC2. También puede publicar sus propias métricas directamente en Amazon CloudWatch. Las alarmas de Amazon CloudWatch le ayudan a aplicar sus decisiones más fácilmente al permitirle enviar notificaciones o realizar cambios automáticamente en los recursos que monitoriza en función de las reglas que defina. Por ejemplo, puede crear alarmas que inicien acciones de Amazon EC2 Auto Scaling y Amazon Simple Notification Service (Amazon SNS) en su nombre.

Elastic Beanstalk utiliza automáticamente Amazon CloudWatch para ayudarle a supervisar el estado de su aplicación y entorno. Puede ir a la consola de Amazon CloudWatch para ver el panel y obtener una visión general de todos los recursos, así como las alarmas. También puede elegir ver más métricas o añadir métricas personalizadas.

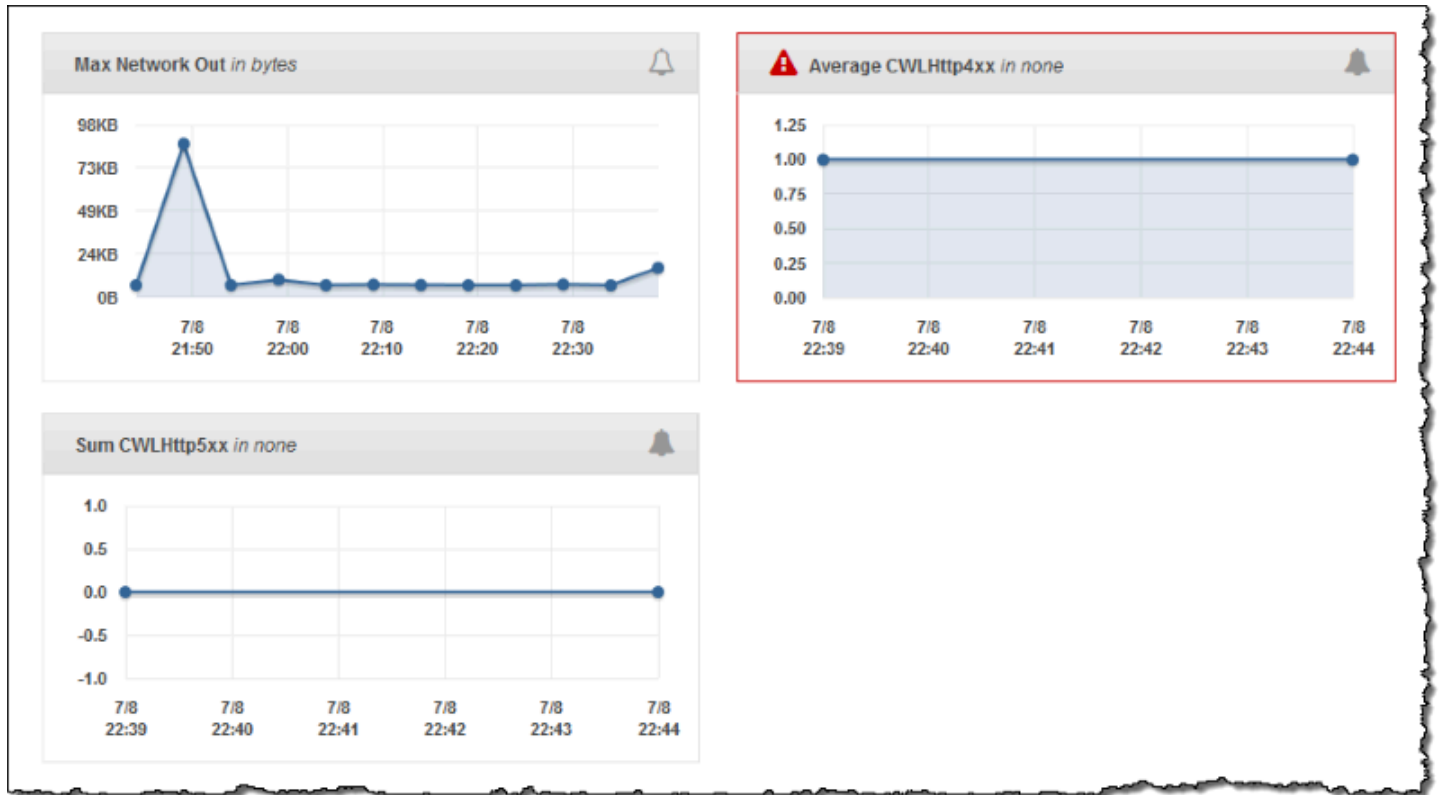
Para obtener más información acerca de Amazon CloudWatch, consulte la [Guía para desarrolladores de Amazon CloudWatch](#). Para obtener un ejemplo de cómo utilizar Amazon CloudWatch con Elastic Beanstalk, consulte [the section called “Ejemplo: usar CloudWatch métricas personalizadas de Amazon”](#).

Uso de Elastic Beanstalk con Amazon CloudWatch Logs

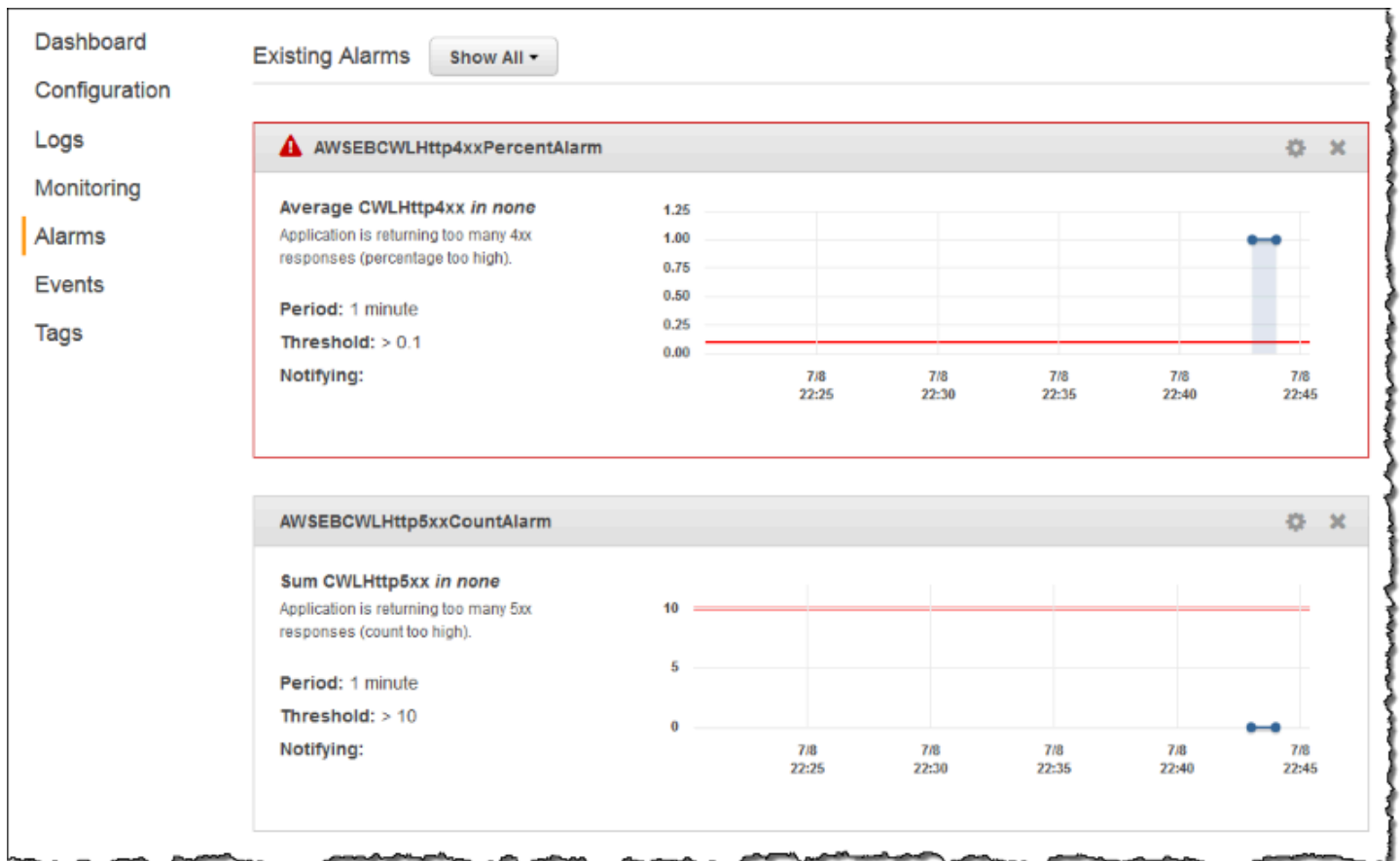
Con CloudWatch Logs, puede supervisar y archivar la aplicación de Elastic Beanstalk, el sistema y los archivos de registro personalizados de instancias de Amazon EC2 de sus entornos. También puede configurar alarmas que le ayuden a reaccionar a determinados eventos de transmisión de logs que extraigan los filtros de las métricas. El agente de CloudWatch Logs instalado en cada instancia de Amazon EC2 del entorno publica puntos de datos de métrica en el servicio de CloudWatch para cada grupo de registro que configure. Cada grupo de registros aplica sus propios patrones de filtro para determinar qué eventos de transmisión de registros se envían a CloudWatch como puntos de datos. Las transmisiones de logs que pertenecen al mismo grupo de logs comparten la misma configuración de retención, monitorización y control de acceso. Puede configurar Elastic Beanstalk para que transmita automáticamente los registros al servicio CloudWatch, como se describe en [Streaming de registros de instancias a CloudWatch Logs](#). Para obtener más información acerca de CloudWatch Logs, incluida la terminología y los conceptos, consulte la [Guía de usuario de Amazon CloudWatch Logs](#).

Además de los registros de instancias, si habilita el [estado mejorado](#) de su entorno, puede configurar dicho entorno para que realice la transmisión de la información de estado a CloudWatch Logs. Consulte [Transmisión de información sobre el estado del entorno Elastic Beanstalk a Amazon CloudWatch Logs](#).

En la figura siguiente se muestran la página de Monitoring (Supervisión) y los gráficos de un entorno que está configurado con la integración de CloudWatch Logs. Las métricas de ejemplo de este entorno se denominan CWLHttp4xx y CWLHttp5xx. Uno de los gráficos muestra que la métrica CWLHttp4xx ha disparado una alarma basada en las condiciones especificadas en los archivos de configuración.



En la figura siguiente se muestran la página Alarmas y los gráficos de las alarmas de ejemplo AWSEBCWLHttp4xxPercentAlarm y AWSEBCWLHttp5xxCountAlarm correspondientes a las métricas CWLHttp4xx y CWLHttp5xx, respectivamente.



Temas

- [Requisitos previos para el streaming de registros de instancias a CloudWatch Logs](#)
- [Cómo Elastic Beanstalk configura CloudWatch Logs](#)
- [Streaming de registros de instancias a CloudWatch Logs](#)
- [Solución de problemas con la integración de CloudWatch Logs](#)
- [Transmisión de información sobre el estado del entorno Elastic Beanstalk a Amazon CloudWatch Logs](#)

Requisitos previos para el streaming de registros de instancias a CloudWatch Logs

Para habilitar el streaming de registros de instancias desde las instancias Amazon EC2 del entorno a CloudWatch Logs, debe cumplir las condiciones que se describen a continuación.

- Plataforma: como esta característica solo está disponible en las versiones de plataforma publicadas en [esta versión](#) o en las posteriores, si utiliza una versión de plataforma anterior, actualice su entorno a uno actual.
- Si no tiene las políticas administradas AWSElasticBeanstalkWebTier o AWSElasticBeanstalkWorkerTier de Elastic Beanstalk en el [perfil de instancia de Elastic Beanstalk](#), debe agregar lo siguiente al perfil para habilitar esta característica.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:PutLogEvents",
        "logs:CreateLogStream"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Cómo Elastic Beanstalk configura CloudWatch Logs

Elastic Beanstalk instala un agente de registro de CloudWatch con la configuración predeterminada en cada instancia que crea. Obtenga más información en la [Referencia de agente de CloudWatch Logs](#).

Cuando habilita el streaming de registros de instancias en CloudWatch Logs, Elastic Beanstalk envía archivos de registro de instancias de su entorno a CloudWatch Logs. Las distintas plataformas transmiten registros diferentes. En la siguiente tabla se muestran los registros por plataforma.

Plataforma / Ramificación de la plataforma	Registros
Docker /	<ul style="list-style-type: none"> • /var/log/eb-engine.log • /var/log/eb-hooks.log

Plataforma / Ramificación de la plataforma	Registros
Ramificación de la plataforma: Docker que se ejecuta en Amazon Linux 2 de 64 bits	<ul style="list-style-type: none"> • /var/log/docker • /var/log/docker-events.log • /var/log/eb-docker/containers/eb-current-app/stdouterr.log • /var/log/nginx/access.log • /var/log/nginx/error.log
Docker / Ramificación de la plataforma: ECS que se ejecuta en Amazon Linux 2 de 64 bits	<ul style="list-style-type: none"> • /var/log/docker-events.log • /var/log/eb-ecs-mgr.log • /var/log/eb-engine.log • /var/log/eb-hooks.log • /var/log/ecs/ecs-agent.log • /var/log/ecs/ecs-init.log
Go .NET Core en Linux Java / Ramificación de la plataforma de Java: Corretto que se ejecuta en Amazon Linux 2 de 64 bits	<ul style="list-style-type: none"> • /var/log/eb-engine.log • /var/log/eb-hooks.log • /var/log/web.stdout.log • /var/log/nginx/access.log • /var/log/nginx/error.log
Node.js Python	<ul style="list-style-type: none"> • /var/log/eb-engine.log • /var/log/eb-hooks.log • /var/log/web.stdout.log • /var/log/httpd/access_log • /var/log/httpd/error_log • /var/log/nginx/access.log • /var/log/nginx/error.log

Plataforma / Ramificación de la plataforma	Registros
Tomcat	<ul style="list-style-type: none"> • /var/log/eb-engine.log
PHP	<ul style="list-style-type: none"> • /var/log/eb-hooks.log • /var/log/httpd/access_log • /var/log/httpd/error_log • /var/log/nginx/access.log • /var/log/nginx/error.log
.NET en Windows Server	<ul style="list-style-type: none"> • C:\inetpub\logs\LogFiles\W3SVC1\u_ex*.log • C:\Program Files\Amazon\ElasticBeanstalk\logs\AWSDeployment.log • C:\Program Files\Amazon\ElasticBeanstalk\logs\Hooks.log
Ruby	<ul style="list-style-type: none"> • /var/log/eb-engine.log • /var/log/eb-hooks.log • /var/log/puma/puma.log • /var/log/web.stdout.log • /var/log/nginx/access.log • /var/log/nginx/error.log

Archivos de registro en plataformas AMI de Amazon Linux

Note

El [18 de julio de 2022](#), Elastic Beanstalk estableció el estado de todas las ramificaciones de la plataforma basadas en la AMI de Amazon Linux (AL1) como retirado. Para obtener más información sobre la migración a una ramificación de la plataforma Amazon Linux 2023 actual y totalmente compatible, consulte [Migración de su aplicación de Linux de Elastic Beanstalk a Amazon Linux 2023 o Amazon Linux 2](#).

En la siguiente tabla se enumeran los archivos de registro transmitidos desde instancias en ramas de plataforma basadas en la AMI de Amazon Linux (anterior a Amazon Linux 2), por plataforma.

Plataforma / Ramificación de la plataforma	Registros
Docker / Ramificación de la plataforma: Docker que se ejecuta en Amazon Linux de 64 bits	<ul style="list-style-type: none"> • /var/log/eb-activity.log • /var/log/nginx/error.log • /var/log/docker-events.log • /var/log/docker • /var/log/nginx/access.log • /var/log/eb-docker/containers/eb-current-app/stdouterr.log
Docker / Ramificación de la plataforma: Multicontainer Docker que se ejecuta en Amazon Linux de 64 bits	<ul style="list-style-type: none"> • /var/log/eb-activity.log • /var/log/ecs/ecs-init.log • /var/log/eb-ecs-mgr.log • /var/log/ecs/ecs-agent.log • /var/log/docker-events.log
Glassfish (Docker preconfigurado)	<ul style="list-style-type: none"> • /var/log/eb-activity.log • /var/log/nginx/error.log • /var/log/docker-events.log • /var/log/docker • /var/log/nginx/access.log
Go	<ul style="list-style-type: none"> • /var/log/eb-activity.log • /var/log/nginx/error.log • /var/log/nginx/access.log
Java / Ramificación de la plataforma: Java 8 que se ejecuta en Amazon Linux de 64 bits Ramificación de la plataforma: Java 7 que se ejecuta en Amazon Linux de 64 bits	<ul style="list-style-type: none"> • /var/log/eb-activity.log • /var/log/nginx/access.log • /var/log/nginx/error.log • /var/log/web-1.error.log • /var/log/web-1.log

Plataforma / Ramificación de la plataforma	Registros
Tomcat	<ul style="list-style-type: none">• /var/log/eb-activity.log• /var/log/httpd/error_log• /var/log/httpd/access_log• /var/log/nginx/error_log• /var/log/nginx/access_log
Node.js	<ul style="list-style-type: none">• /var/log/eb-activity.log• /var/log/nodejs/nodejs.log• /var/log/nginx/error.log• /var/log/nginx/access.log• /var/log/httpd/error.log• /var/log/httpd/access.log
PHP	<ul style="list-style-type: none">• /var/log/eb-activity.log• /var/log/httpd/error_log• /var/log/httpd/access_log
Python	<ul style="list-style-type: none">• /var/log/eb-activity.log• /var/log/httpd/error_log• /var/log/httpd/access_log• /opt/python/log/supervisord.log
Ruby / Ramificación de la plataforma: a: Puma con Ruby que se ejecuta en Amazon Linux de 64 bits	<ul style="list-style-type: none">• /var/log/eb-activity.log• /var/log/nginx/error.log• /var/log/puma/puma.log• /var/log/nginx/access.log

Plataforma / Ramificación de la plataforma	Registros
Ruby / Ramificación de la plataforma: Passenger con Ruby que se ejecuta en Amazon Linux de 64 bits	<ul style="list-style-type: none"> • /var/log/eb-activity.log • /var/app/support/logs/passenger.log • /var/app/support/logs/access.log • /var/app/support/logs/error.log

Elastic Beanstalk configura grupos de registro en CloudWatch Logs para los diversos archivos de registro que transmite. Para recuperar determinados archivos de registro de CloudWatch Logs, debe conocer el nombre del grupo de registro correspondiente. El esquema de nomenclatura del grupo de logs depende del sistema operativo de la plataforma.

En el caso de las plataformas Linux, anteponga `/aws/elasticbeanstalk/environment_name` a la ubicación de archivo de log para obtener el nombre del grupo de logs. Por ejemplo, para recuperar el archivo `/var/log/nginx/error.log`, especifique el nombre del grupo de logs `/aws/elasticbeanstalk/environment_name/var/log/nginx/error.log`.

En el caso de las plataformas Windows, consulte en la siguiente tabla el grupo de registros correspondiente a cada archivo de registro.

Archivo de log en la instancia	Grupo de registros
C:\Program Files\Amazon\ElasticBeanstalk\logs\AWSDeployent.log	/aws/elasticbeanstalk/<environment-name>/EBDeploy-Log
C:\Program Files\Amazon\ElasticBeanstalk\logs\Hooks.log	/aws/elasticbeanstalk/<environment-name>/EBHooks-Log
C:\inetpub\logs\LogFiles (todo el directorio)	/aws/elasticbeanstalk/<environment-name>/IIS-Log

Streaming de registros de instancias a CloudWatch Logs

Puede habilitar el streaming de registros de instancia a CloudWatch Logs mediante la consola de Elastic Beanstalk, la CLI de EB o las opciones de configuración.

Antes de habilitarlo, configure los permisos de IAM para utilizarlos con el agente de CloudWatch Logs. Puede asociar la siguiente política personalizada al [perfil de instancia](#) que asigna a su entorno.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogGroups",
        "logs:DescribeLogStreams"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Streaming de registros de instancia mediante la consola de Elastic Beanstalk

Para transmitir registros de instancia a CloudWatch Logs

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Actualizaciones, supervisión y registro, seleccione Editar.

5. En Instance log streaming to CloudWatch Logs (Streaming de registros de instancias a CloudWatch Logs):
 - Habilite Log streaming (Streaming de registros).
 - Establezca Retention (Retención) en el número de días que se guardarán los registros.
 - Seleccione la opción Lifecycle (Ciclo de vida) que determina si los registros se guardan una vez finalizado el entorno.
6. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

Después de habilitar el streaming de registros, puede volver a la categoría o página de configuración Software y buscar el enlace Log Groups (Grupos de registros). Haga clic en este enlace para ver los registros en la consola de CloudWatch.

Streaming de registros de instancias mediante la CLI de EB

Para habilitar el streaming de registros de instancias a CloudWatch Logs con la CLI de EB, use el comando [eb logs](#).

```
$ eb logs --cloudwatch-logs enable
```

También puede utilizar `eb logs` para recuperar registros de CloudWatch Logs. Puede recuperar todos los registros de instancias del entorno o utilizar las numerosas opciones del comando para especificar los subconjuntos de registros que se recuperarán. Por ejemplo, el siguiente comando recupera el conjunto completo de registros de instancias correspondientes a su entorno y los guarda en un directorio en `.elasticbeanstalk/logs`.

```
$ eb logs --all
```

En concreto, la opción `--log-group` le permite recuperar los registros de instancias de un grupo de registros específicos, correspondientes a un determinado archivo de registro en la instancia. Para ello, debe conocer el nombre del grupo de registros que corresponde al archivo de registro que desea recuperar. Puede encontrar esta información en [Cómo Elastic Beanstalk configura CloudWatch Logs](#).

Streaming de registros de instancias mediante archivos de configuración

Al crear o actualizar un entorno, puede utilizar un archivo de configuración para instalar y configurar el streaming de registros de instancias a CloudWatch Logs. En el siguiente ejemplo, el archivo de

configuración habilita el streaming de logs de instancias predeterminado. Elastic Beanstalk realiza la transmisión del conjunto predeterminado de los archivos de registro de la plataforma del entorno. Para utilizar el ejemplo, copie el texto en un archivo con la extensión `.config` en el directorio `.ebextensions` en el nivel superior del paquete de código fuente de la aplicación.

```
option_settings:
  - namespace: aws:elasticbeanstalk:cloudwatch:logs
    option_name: StreamLogs
    value: true
```

Streaming de archivos de registro personalizados

La integración de Elastic Beanstalk con CloudWatch Logs no admite directamente el streaming de archivos de registro personalizados que genera su aplicación. Para realizar la transmisión de los registros personalizados, utilice un archivo de configuración para instalar directamente el agente de CloudWatch Logs y para configurar los archivos para su envío. Para ver un archivo de configuración de ejemplo, consulte [logs-streamtocloudwatch-linux.config](#).

Note

El ejemplo no funciona en la plataforma Windows.

Para obtener más información acerca de cómo configurar CloudWatch Logs, consulte la [Referencia del agente de CloudWatch Logs](#) en la Guía de usuario de Amazon CloudWatch Logs.

Solución de problemas con la integración de CloudWatch Logs

Si no puede encontrar algunos registros de instancias del entorno que espera en CloudWatch Logs, puede investigar los siguientes problemas comunes:

- Su rol de IAM carece de los permisos de IAM necesarios.
- Usted lanzó su entorno en una Región de AWS que no admite CloudWatch Logs.
- Uno de sus archivos de log personalizados no existe en la ruta que especificó.

Transmisión de información sobre el estado del entorno Elastic Beanstalk a Amazon CloudWatch Logs

Si habilita los informes de [estado mejorado](#) de su entorno, puede configurar dicho entorno para que realice la transmisión de la información de estado a CloudWatch Logs. Esta transmisión es independiente de la transmisión de instancias de registro de Amazon EC2. En este tema se describe el streaming de información de estado de entorno. Para obtener información acerca del streaming de logs de instancias, consulte [Uso de Elastic Beanstalk con Amazon CloudWatch Logs](#).

Al configurar la transmisión de estado del entorno, Elastic Beanstalk crea un grupo de registro de CloudWatch Logs para el estado del entorno. El nombre del grupo de logs es `/aws/elasticbeanstalk/environment-name/environment-health.log`. En este grupo de registros, Elastic Beanstalk crea secuencias de registros denominadas `YYYY-MM-DD#<hash-suffix>` (es posible que haya varias secuencias de registros por fecha).

Cuando cambia el estado del entorno, Elastic Beanstalk agrega un registro a la secuencia de registro de estado. El registro representa la transición del estado: el nuevo estado y una descripción de la causa del cambio. Por ejemplo, el estado de un entorno podría cambiar a grave porque el balanceador de carga tiene un error. Para obtener una descripción de los estados mejorados, consulte [Colores y estados](#).

Requisitos previos de la transmisión del estado del entorno a CloudWatch Logs

Para habilitar la transmisión del estado del entorno a CloudWatch Logs, debe cumplir las siguientes condiciones:

- **Plataforma:** debe utilizar una versión de plataforma que sea compatible con los informes de estado mejorados.
- **Permisos:** debe conceder determinados permisos relacionados con el registro a Elastic Beanstalk, para que pueda actuar en su nombre para realizar la transmisión de la información de estado de su entorno. Si su entorno no usa un rol de servicio que haya creado Elastic Beanstalk, `aws-elasticbeanstalk-service-role`, o el rol vinculado al servicio de la cuenta, `AWSServiceRoleForElasticBeanstalk`, asegúrese de agregar los siguientes permisos a su rol de servicio personalizado.

```
{
  "Effect": "Allow",
  "Action": [
```

```
    "logs:DescribeLogStreams",
    "logs:CreateLogStream",
    "logs:PutLogEvents"
  ],
  "Resource": "arn:aws:logs:*:*:log-group:/aws/elasticbeanstalk/*:log-stream:*"
}
```

Transmitir los registros de estado del entorno a CloudWatch Logs

Puede habilitar la transmisión de estado del entorno a CloudWatch Logs mediante la consola de Elastic Beanstalk, la CLI de EB o las opciones de configuración.

Transmisión de registros de estado del entorno mediante la consola de Elastic Beanstalk

Para transmitir los registros de estado del entorno a CloudWatch Logs

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Monitoring (Monitorización), seleccione Edit (Editar).
5. En Health reporting (Informes de estado), asegúrese de que el informe System (Sistema) está configurado en Enhanced (Mejorado).
6. En Health event streaming to CloudWatch Logs (Streaming de eventos de estado a CloudWatch Logs)
 - Habilite Log streaming (Streaming de registros).
 - Establezca Retention (Retención) en el número de días que se guardarán los registros.
 - Seleccione la opción Lifecycle (Ciclo de vida) que determina si los registros se guardan una vez finalizado el entorno.
7. Para guardar los cambios, elija Aplicar en la parte inferior de la página.

Después de habilitar el streaming de registros, puede volver a la categoría o página de configuración Monitoring (Monitorización) y buscar el enlace Log Group (Grupo de registros). Haga clic en este enlace para ver los registros de estado de entorno en la consola de CloudWatch.

Streaming de logs de estado de entorno mediante la CLI de EB

Para habilitar la transmisión de registros de estado de entorno a CloudWatch Logs; mediante la CLI de EB, use el comando [eb logs](#).

```
$ eb logs --cloudwatch-logs enable --cloudwatch-log-source environment-health
```

También puede utilizar `eb logs` para recuperar registros de CloudWatch Logs. Por ejemplo, el siguiente comando recupera todos los registros de estado correspondientes a su entorno y los guarda en un directorio en `.elasticbeanstalk/logs`.

```
$ eb logs --all --cloudwatch-log-source environment-health
```

Streaming de logs de estado de entorno mediante archivos de configuración

Al crear o actualizar un entorno, puede utilizar un archivo de configuración para instalar y configurar la transmisión de registros de estado de entorno a CloudWatch Logs. Para utilizar el ejemplo siguiente, copie el texto en un archivo con la extensión `.config` en el directorio `.ebextensions` en el nivel superior del paquete de código fuente de la aplicación. El ejemplo configura Elastic Beanstalk para habilitar la transmisión de registros de estado de entorno, mantener los registros después de terminar el entorno y guardarlos durante 30 días.

Example [Archivo de configuración de streaming de estado](#)

```
#####
## Sets up Elastic Beanstalk to stream environment health information
## to Amazon CloudWatch Logs.
## Works only for environments that have enhanced health reporting enabled.
#####

option_settings:
  aws:elasticbeanstalk:cloudwatch:logs:health:
    HealthStreamingEnabled: true
    ### Settings below this line are optional.
    # DeleteOnTerminate: Delete the log group when the environment is
    # terminated. Default is false. If false, the health data is kept
```



```
# RetentionInDays days.  
DeleteOnTerminate: false  
# RetentionInDays: The number of days to keep the archived health data  
# before it expires, if DeleteOnTerminate isn't set. Default is 7 days.  
RetentionInDays: 30
```

Para conocer los valores predeterminados de las opciones y los valores válidos, consulte [aws:elasticbeanstalk:cloudwatch:logs:health](https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/cloudwatch-logs-health.html).

Usar Elastic Beanstalk con Amazon EventBridge

Al utilizar Amazon EventBridge, puede configurar reglas basadas en eventos que monitoreen los recursos de Elastic Beanstalk e inicien acciones de destino que utilicen otros servicios de AWS. Por ejemplo, puede establecer una regla para el envío de notificaciones por correo electrónico al señalar un tema de Amazon SNS siempre que el estado de un entorno de producción cambie al estado Warning (Advertencia). O bien, puede configurar una función de Lambda para que pase una notificación a Slack siempre que el estado del entorno cambie a Degraded (Degradado) o Severe (Grave).

Puede crear reglas en Amazon EventBridge para que actúen en cualquiera de los siguientes eventos de Elastic Beanstalk:

- Cambios de estado para las operaciones del entorno (incluidas las operaciones de creación, actualización y finalización). El evento especifica si el cambio de estado se ha iniciado, se ha realizado correctamente o no se ha podido realizar.
- Cambios de estado para otros recursos. Además de los entornos, otros recursos que se supervisan incluyen balanceadores de carga, grupos de Auto Scaling e instancias.
- Transición de estado para entornos. El evento indica dónde el estado del entorno ha pasado de un estado a otro.
- Cambio de estado para las actualizaciones administradas. El evento especifica si el cambio de estado se ha iniciado, se ha realizado correctamente o no se ha podido realizar.

Para capturar eventos específicos de Elastic Beanstalk que le interesen, defina patrones específicos de eventos que EventBridge puede utilizar para detectar los eventos. Los patrones de eventos tienen la misma estructura que los eventos con los que coinciden. El patrón cita los campos para los que se desea encontrar coincidencias y proporciona los valores que está buscando. Los eventos se emiten en la medida de lo posible. Se entregan desde Elastic Beanstalk a EventBridge casi en tiempo real

en circunstancias operativas normales. Sin embargo, pueden surgir situaciones que retrasen o impidan la entrega de un evento.

Para obtener una lista de los campos incluidos en eventos de Elastic Beanstalk y sus posibles valores de cadena, consulte [Asignación de campos de eventos de Elastic Beanstalk](#). Para obtener información acerca de cómo funcionan las reglas de EventBridge con los patrones de eventos, vea [Eventos y patrones de eventos en EventBridge](#).

Supervisar un recurso de Elastic Beanstalk con EventBridge

Con EventBridge, puede crear reglas que definan las acciones que se deben realizar cuando Elastic Beanstalk emite eventos para sus recursos. Por ejemplo, puede crear una regla que le envíe un mensaje de correo electrónico siempre que cambie el estado de un entorno.


La consola de EventBridge tiene la opción Pre-defined pattern (Patrón predefinido) para crear patrones de eventos de Elastic Beanstalk. Si selecciona esta opción en la consola de EventBridge al crear una regla, puede crear rápidamente un patrón de eventos de Elastic Beanstalk. Solo tiene que seleccionar los campos y los valores del evento. A medida que realiza selecciones, la consola crea y muestra el patrón de eventos. O bien, puede editar manualmente el patrón de eventos que cree y guardarlo como un patrón personalizado. La consola también le brinda la opción de mostrar un evento de ejemplo detallado que puede copiar y pegar en el patrón de eventos que está creando.

Si prefiere escribir o copiar y pegar un patrón de eventos en la consola de EventBridge, puede utilizar la opción Custom pattern (Patrón personalizado) en la consola. Al hacerlo, no es necesario seguir los pasos de selección de campos y valores descritos anteriormente. Este tema ofrece ejemplos de [patrones de coincidencia de eventos](#) y [eventos de Elastic Beanstalk](#) que puede utilizar.

Para crear una regla para un evento de recurso, realice lo siguiente:

1. Inicie sesión en AWS con una cuenta que tenga permisos para usar EventBridge y Elastic Beanstalk.
2. Abra la consola de Amazon EventBridge en <https://console.aws.amazon.com/events/>.
3. En el panel de navegación, seleccione Rules.
4. Elija Create rule (Crear regla).
5. Ingrese un Name (Nombre) para la regla y, opcionalmente, una descripción.
6. En Event bus (Bus de eventos), elija Default (Predeterminado). Cuando un servicio de AWS en su cuenta emite un evento, siempre va al bus de eventos predeterminado de su cuenta.
7. En Rule type (Tipo de regla), elija Rule with an event pattern (Regla con un patrón de evento).

8. Elija Next (Siguiente).
9. En Event source (Origen del evento), elija AWS events or EventBridge partner events (Eventos o eventos de socios de EventBridge).
10. (Opcional) En Sample event (Evento de ejemplo), seleccione AWS events (Eventos de AWS). Ingrese Elastic Beanstalk en el campo de búsqueda. Esto proporcionará una lista de eventos de Elastic Beanstalk de muestra entre los que puede elegir mostrarlos. Este paso simplemente muestra un evento de ejemplo al que puede hacer referencia. No afecta al resultado de la creación de la regla. La sección [Ejemplo de eventos de Elastic Beanstalk](#) , incluida más adelante en este tema, proporciona ejemplos del mismo tipo de eventos.
11. En la sección Event pattern (Patrón de eventos), elija Event pattern form (Formulario de patrón de eventos).

 Note


Si ya tiene texto para un patrón de eventos y no necesita que la consola de EventBridge lo cree por usted, seleccione Custom pattern (JSON editor) (Patrón personalizado [editor JSON]). A continuación, puede introducir manualmente o copiar y pegar el texto en el cuadro Event pattern (Patrón de eventos). Elija Next (Siguiente) y vaya al paso de ingresar un objetivo.

12. En Event source (Origen del evento), elija AWS services (Servicios de AWS).
13. En AWS name (Nombre de AWS), seleccione Elastic Beanstalk.
14. Para Event type (Tipo de evento), seleccione Status Change (Cambio de estado).
15. Este paso describe cómo puede trabajar con los campos de eventos Detail type (Tipo de detalle), Status (Estado) y Severity (Gravedad) de Elastic Beanstalk. A medida que elige estos campos y los valores que desea hacer coincidir, la consola crea y muestra el patrón de eventos.
 - Si selecciona solo un valor en Specific detail type(s) (Tipos de detalles específicos), puede elegir uno o varios valores para el siguiente campo de la jerarquía.
 - Si selecciona más de un valor en Specific detail type(s) (Tipos de detalles específicos), no elija valores específicos para los siguientes campos de la jerarquía. Esto evita una lógica de coincidencia ambigua entre los campos del patrón de eventos.

El campo de evento Environment (Entorno) no se ve afectado por esta jerarquía, por lo que se muestra como se describe en el siguiente paso.

16. Para Environment (Entorno), seleccione Any environment (Cualquier entorno) o Specific environment(s) (Entornos específicos).
 - Si selecciona Specific environment(s) (Entornos específicos), puede elegir uno o varios entornos de la lista desplegable. EventBridge agrega todos los entornos que selecciona en la lista EnvironmentName[], en la sección Detail (Detalle) del patrón de eventos. A continuación, la regla filtra todos los eventos para incluir solo los entornos específicos que elija.
 - Si selecciona Any environment (Cualquier entorno), no se agregarán entornos al patrón de eventos. Debido a esto, la regla no filtra ninguno de los eventos de Elastic Beanstalk según el entorno.
17. Elija Next (Siguiente).
18. En Target types (Tipos de destino), elija AWS service (Servicio de AWS).
19. En Select a target (Seleccionar un destino), elija la acción de destino que debe realizar cuando se reciba un evento de cambio de estado de recurso desde Elastic Beanstalk.

Por ejemplo, puede utilizar un tema de Amazon Simple Notification Service (SNS) para enviar un correo electrónico o un mensaje de texto cuando se produce un evento. Para ello, debe crear un tema de Amazon SNS mediante la consola de Amazon SNS. Para obtener más información, consulte [Uso de Amazon SNS para notificaciones de usuario](#).

 Important

Algunas acciones de destino pueden requerir el uso de otros servicios e incurrir en cargos adicionales, como el servicio Amazon SNS o Lambda. Para obtener más información sobre los precios de AWS, consulte <https://aws.amazon.com/pricing/>.

Algunos servicios forman parte del nivel de uso gratuito de AWS. Si es un cliente nuevo, puede probar estos servicios de forma gratuita. Para obtener más información, consulte <https://aws.amazon.com/free/>.

20. (Opcional) Elija Add another target (Agregar otro destino) para especificar una acción de destino adicional para la regla de eventos.
21. Elija Next (Siguiente).
22. (Opcional) Introduzca una o varias etiquetas para la regla. Para obtener más información, consulte [Amazon EventBridge tags](#) (Etiquetas de Amazon EventBridge) en la Guía del usuario de Amazon EventBridge.
23. Elija Next (Siguiente).

24. Revise los detalles de la regla y elija Create rule (Crear regla).

Ejemplo de patrones de eventos de Elastic Beanstalk

Los patrones de eventos tienen la misma estructura que los eventos con los que coinciden. El patrón cita los campos para los que se desea encontrar coincidencias y proporciona los valores que está buscando.

- Health status change (Cambio del estado) para todos los entornos

```
{
  "source": [
    "aws.elasticbeanstalk"
  ],
  "detail-type": [
    "Health status change"
  ]
}
```

- Health status change (Cambio del estado) para los siguientes entornos: myEnvironment1 y myEnvironment2 Este patrón de eventos filtra para estos dos entornos específicos, mientras que el ejemplo anterior de Health status change (Cambio del estado) que no filtra envía eventos para todos los entornos.

```
{"source": [
  "aws.elasticbeanstalk"
],
"detail-type": [
  "Health status change"
],
"detail": {
  "EnvironmentName": [
    "myEnvironment1",
    "myEnvironment2"
  ]
}
}
```

- Elastic Beanstalk resource status change (Cambio de estado de recursos de Elastic Beanstalk) para todos los entornos

```
{
  "source": [
    "aws.elasticbeanstalk"
  ],
  "detail-type": [
    "Elastic Beanstalk resource status change"
  ]
}
```

- Elastic Beanstalk resource status change (Cambio de estado de recursos de Elastic Beanstalk) con Status Environment update failed (Actualización del entorno fallida) y Severity ERROR para los siguientes entornos: myEnvironment1 y myEnvironment2

```
{"source": [
  "aws.elasticbeanstalk"
],
  "detail-type": [
    "Elastic Beanstalk resource status change"
  ],
  "detail": {
    "Status": [
      "Environment update failed"
    ],
    "Severity": [
      "ERROR"
    ],
    "EnvironmentName": [
      "myEnvironment1",
      "myEnvironment2"
    ]
  }
}
```

- Other resource status change (Otro cambio de estado de recursos) para balanceadores de carga, grupos de Auto Scaling e instancias

```
{
  "source": [
    "aws.elasticbeanstalk"
  ],
  "detail-type": [
    "Other resource status change"
  ]
}
```

```
]
}
```

- Managed update status change (Cambio de estado de actualización administrada) para todos los entornos

```
{
  "source": [
    "aws.elasticbeanstalk"
  ],
  "detail-type": [
    "Managed update status change"
  ]
}
```

- Para capturar todos los eventos de Elastic Beanstalk (excluya la sección detail-type)

```
{
  "source": [
    "aws.elasticbeanstalk"
  ]
}
```

Ejemplo de eventos de Elastic Beanstalk

A continuación se muestra un ejemplo de evento de Elastic Beanstalk para un cambio de estado de recurso:

```
{
  "version":"0",
  "id":"1234a678-1b23-c123-12fd3f456e78",
  "detail-type":"Elastic Beanstalk resource status change",
  "source":"aws.elasticbeanstalk",
  "account":"111122223333",
  "time":"2020-11-03T00:31:54Z",
  "region":"us-east-1",
  "resources":[
    "arn:was:elasticbeanstalk:us-east-1:111122223333:environment/myApplication/myEnvironment"
  ],
  "detail":{
```

```

    "Status":"Environment creation started",
    "EventDate":1604363513951,
    "ApplicationName":"myApplication",
    "Message":"createEnvironment is starting.",
    "EnvironmentName":"myEnvironment",
    "Severity":"INFO"
  }
}

```

A continuación se muestra un ejemplo de evento de Elastic Beanstalk para un cambio del estado:

```

{
  "version":"0",
  "id":"1234a678-1b23-c123-12fd3f456e78",
  "detail-type":"Health status change",
  "source":"aws.elasticbeanstalk",
  "account":"111122223333",
  "time":"2020-11-03T00:34:48Z",
  "region":"us-east-1",
  "resources":[
    "arn:aws:elasticbeanstalk:us-east-1:111122223333:environment/myApplication/myEnvironment"
  ],
  "detail":{
    "Status":"Environment health changed",
    "EventDate":1604363687870,
    "ApplicationName":"myApplication",
    "Message":"Environment health has transitioned from Pending to Ok. Initialization completed 1 second ago and took 2 minutes.",
    "EnvironmentName":"myEnvironment",
    "Severity":"INFO"
  }
}

```

Asignación de campos de eventos de Elastic Beanstalk

La siguiente tabla asigna los campos de evento de Elastic Beanstalk y sus posibles valores de cadena al campo de EventBridge `detail-type`. Para obtener más información sobre cómo funciona EventBridge con los patrones de eventos de un servicio, consulte [Eventos y patrones de eventos en EventBridge](#).

Campo de EventBridge detail-type	Campo de Elastic Beanstalk Status	Campo de Elastic Beanstalk Severity	Campo de Elastic Beanstalk Message
Cambio de estado de recursos de Elastic Beanstalk	La creación del entorno se ha iniciado	INFO	createEnvironment se está iniciando.
	La creación del entorno se ha realizado correctamente	INFO	createEnvironment se ha completado correctamente.
	La creación del entorno se ha realizado correctamente	INFO	Entorno lanzado: <Nombre del entorno>. Sin embargo, hubo problemas durante el lanzamiento. Consulte el registro de sucesos para obtener más detalles.
	Ha ocurrido un error durante la creación del entorno	ERROR	Error al lanzar el entorno.
	La actualización del	INFO	Se está iniciando la actualización del entorno.

Campo de EventBridge detail-type	Campo de Elastic Beanstalk Status	Campo de Elastic Beanstalk Severity	Campo de Elastic Beanstalk Message
	entorno se ha iniciado		
	Actualización del entorno realizada correctamente	INFO	La actualización del entorno se ha completado correctamente.
	Error al actualizar el entorno	ERROR	Error al implementar la configuración.
	La finalización del entorno se ha iniciado	INFO	terminateEnvironment se está iniciando.
	Finalización del entorno realizada correctamente	INFO	terminateEnvironment se ha completado correctamente.
	Error al finalizar el entorno	INFO	Error en el paso de finalización del entorno porque falló al menos uno de los flujos de trabajo de finalización del entorno.

Campo de EventBridge detail-type	Campo de Elastic Beanstalk Status	Campo de Elastic Beanstalk Severity	Campo de Elastic Beanstalk Message
Cambio de estado de otros recursos	Grupo de Auto Scaling creado	INFO	createEnvironment se está iniciando.
	Grupo de Auto Scaling eliminado	INFO	createEnvironment se está iniciando.
	Instancia añadida	INFO	Se ha añadido la instancia [i-123456789a12b1234] a su entorno.
	Instancia eliminada	INFO	Se ha eliminado la instancia [i-123456789a12b1234] de su entorno.
	Balanceador de carga creado	INFO	Balanceador de carga creado denominado: <Nombre de LB>
	Balanceador de carga eliminado	INFO	Balanceador de carga eliminado denominado: <Nombre de LB>
Cambio del estado	El estado se ha cambiado	INFO/ ADVERTENCIA	El estado del entorno ha pasado a <healthStatus>.
	El estado se ha cambiado	INFO/ ADVERTENCIA	El estado del entorno ha pasado de <healthStatus> a <healthStatus>.

Campo de EventBridge detail-type	Campo de Elastic Beanstalk Status	Campo de Elastic Beanstalk Severity	Campo de Elastic Beanstalk Message
Cambio de estado de actualización administrada	Actualización administrada iniciada	INFO	La actualización de la plataforma administrada está en curso.
	Error en la actualización administrada	INFO	Error de actualización administrada, se reintentará en %s minutos.

Búsqueda y seguimiento de recursos de Elastic Beanstalk con AWS Config

[AWS Config](#) proporciona una vista detallada de la configuración de los recursos de AWS de su cuenta de AWS. Puede observar las relaciones entre los recursos, obtener un historial de los cambios de configuración y comprobar cómo cambian las relaciones y configuraciones con el paso del tiempo. Puede usar AWS Config para definir reglas que evalúen la conformidad de los datos de estas configuraciones de recursos.

Varios tipos de recursos de Elastic Beanstalk se integran con AWS Config:

- Aplicaciones
- Versiones de aplicaciones
- Entornos

En la siguiente sección se muestra cómo configurar AWS Config para registrar los recursos de estos tipos.

Para obtener más información sobre AWS Config, consulte la [Guía para desarrolladores de AWS Config](#). Para obtener información acerca de los precios, consulte la [página de información sobre precios de AWS Config](#).

Configuración de AWS Config

Para configurar AWS Config inicialmente, consulte los temas siguientes en la [Guía de desarrollador de AWS Config](#).

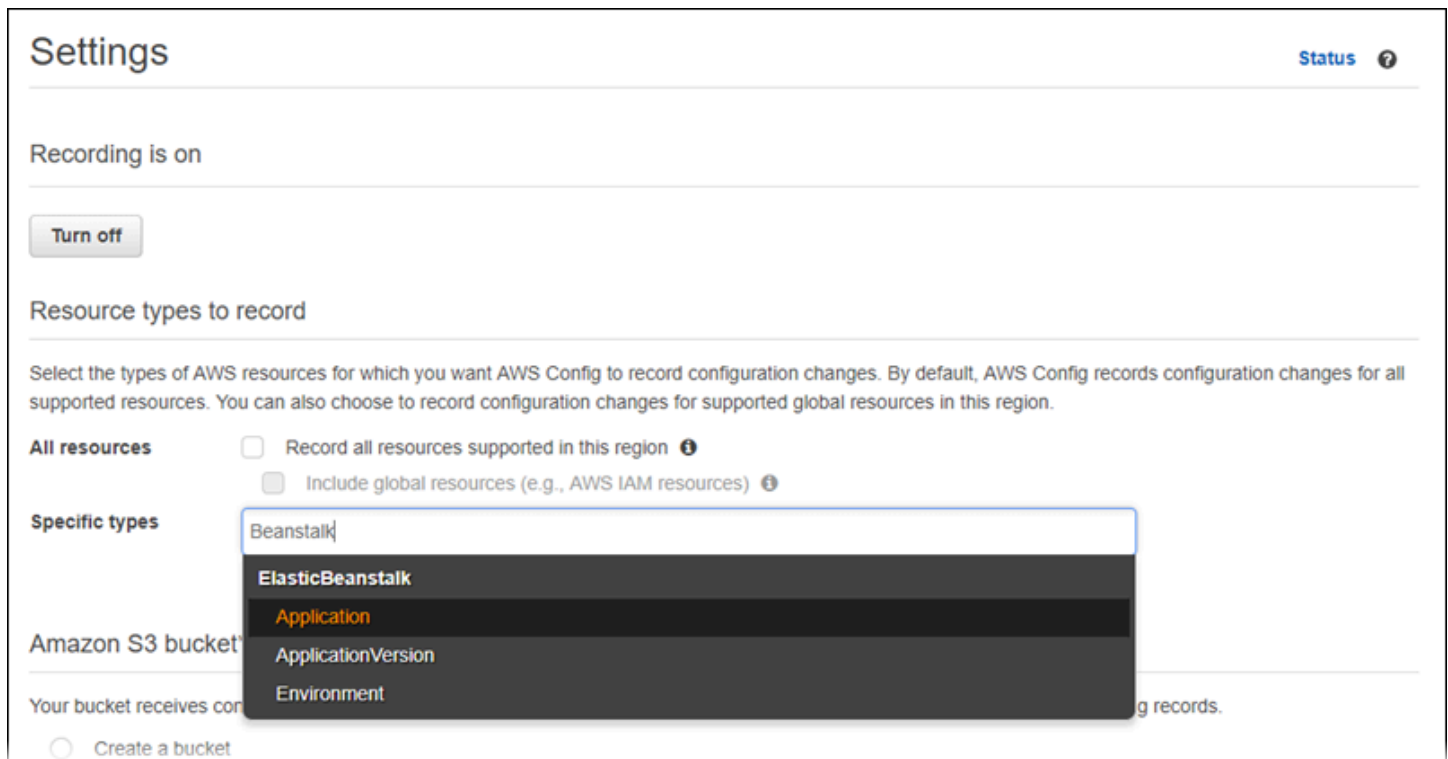
- [Configuración de AWS Config con la consola](#)
- [Configuración de AWS Config con AWS CLI](#)

Configuración de AWS Config para registrar recursos de Elastic Beanstalk

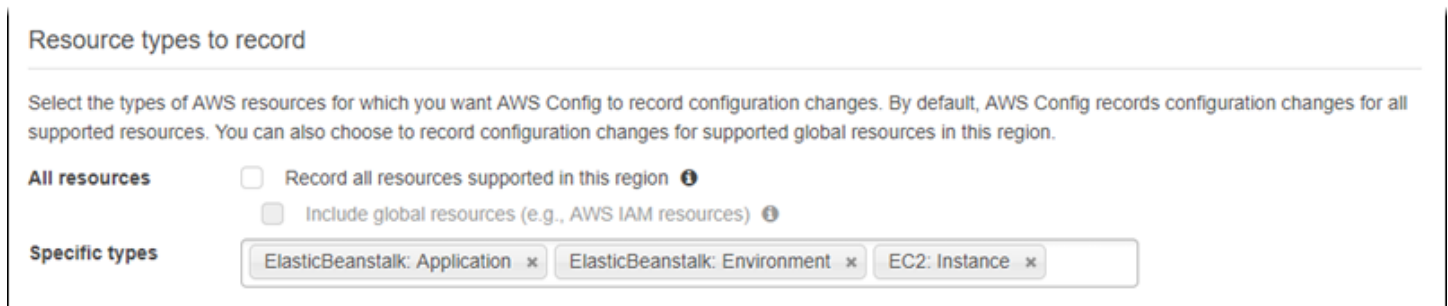
De forma predeterminada, AWS Config registra los cambios de configuración de todos los tipos de recursos regionales admitidos que descubre en la región en la que se ejecuta el entorno. Puede personalizar AWS Config de forma que únicamente registre los cambios de determinados tipos de recursos o los cambios de los recursos globales.

Por ejemplo, puede configurar AWS Config para registrar los cambios en los recursos de Elastic Beanstalk y un subconjunto de otros recursos de AWS que Elastic Beanstalk inicia por usted. Con la [consola de AWS Config](#), puede seleccionar Elastic Beanstalk como recurso en el campo Specific Types (Tipos específicos) de la página Settings (Configuración) de AWS Config. Allí puede elegir registrar cualquiera de los tipos de recursos de Elastic Beanstalk: : Application (Aplicación), ApplicationVersion (Versión de aplicación) y Environment (Entorno).

En la figura siguiente se muestra la página AWS Config Settings (Configuraciones) con los tipos de recursos de Elastic Beanstalk que puede elegir para el registro: Application (Aplicación), ApplicationVersion (Versión de aplicación) y Environment (Entorno).



Después de seleccionar algunos tipos de recursos, la lista Specific types (Tipos específicos) tiene este aspecto.



Para obtener más información acerca de los recursos regionales y globales, así como el procedimiento de personalización completo, consulte [Selección de los recursos que debe registrar AWS Config](#).

Visualización de los detalles de configuración de Elastic Beanstalk en la consola de AWS Config

Puede utilizar la consola de AWS Config para buscar recursos de Elastic Beanstalk y obtener detalles actuales e históricos sobre sus configuraciones. El siguiente ejemplo muestra cómo buscar información sobre un entorno de Elastic Beanstalk.

Para encontrar un entorno de Elastic Beanstalk en la consola de AWS Config

1. Abra la [consola de AWS Config](#).
2. Elija Resources (Recursos).
3. En la página Resource inventory, elija Resources.
4. Abra el menú Resource type (Tipos de recursos), vaya hasta ElasticBeanstalk y elija uno o varios tipos de recursos de Elastic Beanstalk.

Note

Para ver detalles de configuración de otros recursos que Elastic Beanstalk ha creado para su aplicación, elija otros tipos de recursos. Por ejemplo, puede elegir Instance (Instancia) en EC2.

5. Elija Look up (Buscar). Consulte 2 en la figura siguiente.

Config timeline	Compliance	Manage resource
i-0abae959f6fb4b133	Compliant	Manage resource
arn:aws:elasticbeanstalk:us-east-1:270205402845:application/config-demo	--	
e-yaumygtbwr	--	

6. Elija un ID de recurso en la lista de recursos mostrada por AWS Config.

Resource inventory Status ?


Look up existing and deleted resources recorded by AWS Config. View compliance details for each resource or choose the Config timeline icon to see how a particular resource's configuration has changed over time.



Resources Tag Compliance status

EC2: Instance, ElasticBeanstalk: ...

Include deleted resources

Look up

Choose Config timeline  to view a history of configuration details for the resource.

Resource type	Config timeline 	Compliance	Manage resource
▶ EC2 Instance	i-0abae959f6fb4b133	Compliant	
▶ ElasticBeanstalk Application	arn:aws:elasticbeanstalk:us-east-1:270205402845:application/config-demo	--	
▶ ElasticBeanstalk Environment	e-yaumygtbwr	--	

AWS Config muestra los detalles de configuración y otra información sobre el recurso que ha seleccionado.

ElasticBeanstalk Environment e-yaumygtbwr

on February 09, 2018 4:03:54 PM Pacific Standard Time (UTC-08:00)

Manage resource ⓘ

← [] [] [05th February 2018 4:34:35 PM] [06th February 2018 3:43:45 PM] [07th February 2018 11:43:44 PM] →

Now [] []

1 Change 7 Changes

▼ Configuration Details [View Details](#)

Amazon Resource Name am:aws:elasticbeanstalk:us-east-1:270205402845:environment/config-demo/ConfigDemo-env

Resource type AWS::ElasticBeanstalk::Environment

Resource ID e-yaumygtbwr

Resource name ConfigDemo-env

Availability zone Not Applicable

Created on February 05, 2018 3:45:05 PM

Tags (3)

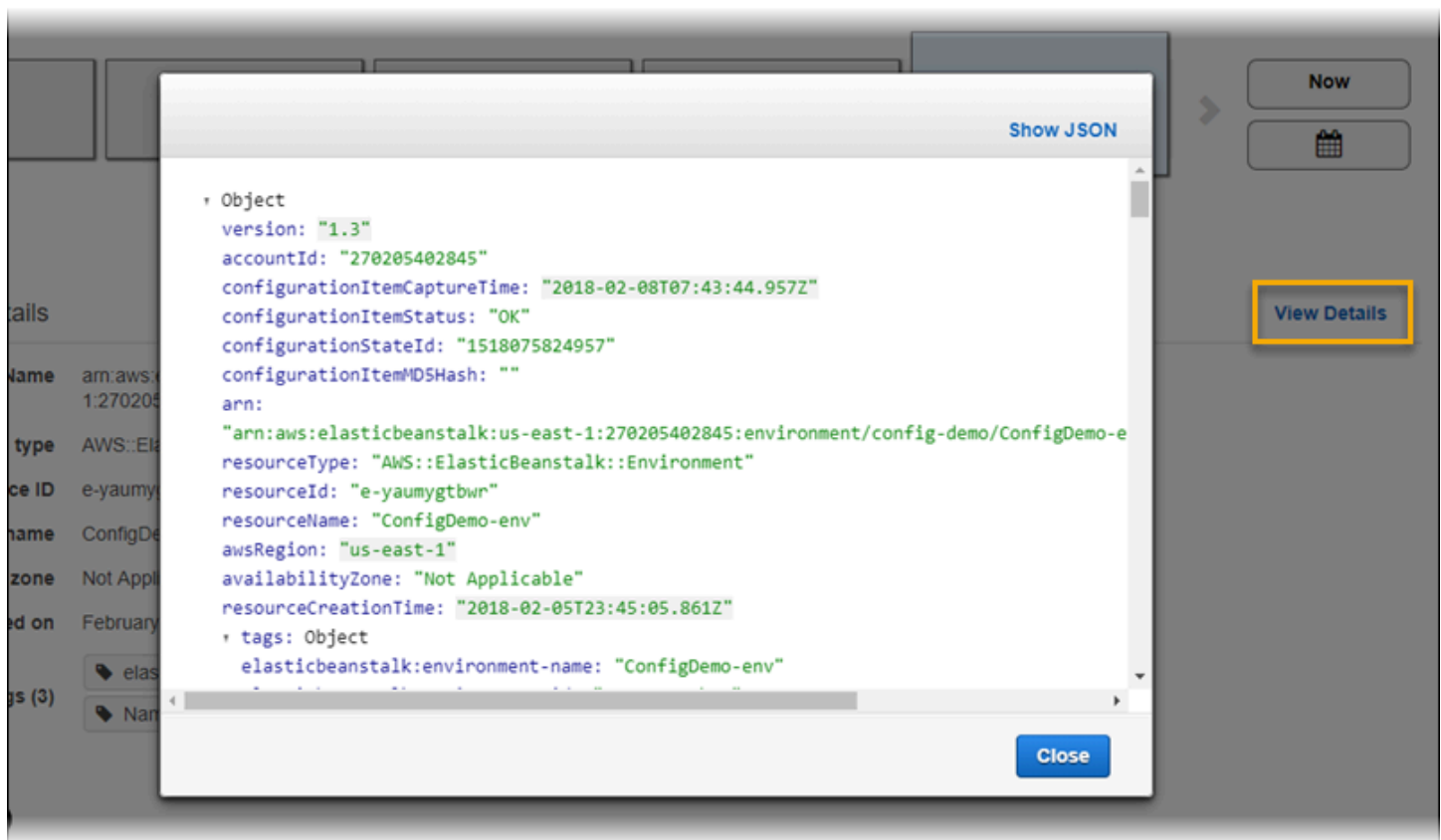
- elasticbeanstalk:envi...
- elasticbeanstalk:envi...
- Name: ConfigDemo-...

► Relationships 5

► Changes 7

► CloudTrail Events 6

Para ver todos los detalles de la configuración registrada, elija View Details (Ver detalles).



Para conocer otras maneras de buscar un recurso y ver información en esta página, consulte [Visualización de historial y configuraciones de recursos de AWS](#) en la Guía del desarrollador de AWS Config.

Evaluación de los recursos de Elastic Beanstalk mediante reglas de AWS Config

Puede crear reglas de AWS Config que representen la configuración ideal para sus recursos de Elastic Beanstalk. Puede utilizar reglas de configuración administradas de AWS predefinidas o definir reglas personalizadas. AWS Config realiza un seguimiento constante de los cambios de configuración de los recursos para determinar si estos cambios infringen algunas de las condiciones de las reglas. La consola de AWS Config muestra el estado de conformidad de sus reglas y recursos.

Si un recurso infringe una regla y está marcado como no conforme, AWS Config puede avisarle mediante un tema de [Amazon Simple Notification Service \(Amazon SNS\)](#). Para consumir mediante programación los datos de estas alertas de AWS Config, utilice una cola de [Amazon Simple Queue Service \(Amazon SQS\)](#) como punto de enlace de notificación para el tema Amazon SNS. Por ejemplo, es posible que desee escribir un código que inicie un flujo de trabajo cuando alguien modifique la configuración de grupo de Auto Scaling del entorno.

Para obtener más información acerca de cómo configurar y utilizar reglas, consulte [Evaluación de recursos con reglas AWS Config](#) en la Guía para desarrolladores de AWS Config.

Uso de Elastic Beanstalk con Amazon DynamoDB

Amazon DynamoDB es un servicio de base de datos NoSQL totalmente administrado que ofrece un rendimiento rápido y predecible, así como una perfecta escalabilidad. Si es un desarrollador, puede utilizar DynamoDB para crear una tabla de base de datos capaz de almacenar y recuperar cualquier cantidad de datos, así como de atender cualquier nivel de tráfico de solicitudes. DynamoDB distribuye automáticamente los datos y el tráfico de la tabla entre un número de servidores suficiente como para administrar la capacidad de solicitudes especificada por el cliente y la cantidad de datos almacenados, al tiempo que mantiene un rendimiento uniforme y rápido. Todos los elementos de datos se almacenan en unidades de estado sólido (SSD) y se replican automáticamente en varias zonas de disponibilidad de una región de AWS a fin de ofrecer las prestaciones integradas de alta disponibilidad y durabilidad de los datos.

Si utiliza [tareas periódicas](#) en un entorno de trabajo, Elastic Beanstalk crea una tabla de DynamoDB y la utiliza para elegir las instancias principales y guardar información sobre la tarea. Cada instancia del entorno intenta escribir en la tabla cada pocos segundos para convertirse en la instancia principal y realiza la tarea de acuerdo con la programación.

Puede utilizar [archivos de configuración](#) si desea crear una tabla de DynamoDB para la aplicación. Consulte [eb-node-express-sample](#) en GitHub para obtener una aplicación de Node.js de ejemplo en la que se crea una tabla con un archivo de configuración y se establece conexión con ella con el AWS SDK for JavaScript en Node.js. Para ver un tutorial de ejemplo en el que se usa DynamoDB con PHP, consulte [Ejemplo: DynamoDB, CloudWatch y SNS](#). Para ver un ejemplo en el que se utiliza AWS SDK for Java, consulte [Administración del estado de la sesión de Tomcat con DynamoDB](#) en la documentación de AWS SDK for Java.

Al crear una tabla de DynamoDB con los archivos de configuración, la tabla no está vinculada con el ciclo de vida de dicho entorno y no se elimina cuando se termina el entorno. Para garantizar que la información personal no se retiene innecesariamente, elimine los registros que ya no necesite o elimine la tabla.

Para obtener más información sobre DynamoDB, consulte la [Guía para desarrolladores de DynamoDB](#).

Uso de Elastic Beanstalk con Amazon ElastiCache

Amazon ElastiCache es un servicio web que permite configurar, administrar y escalar entornos de caché en memoria distribuidos en la nube. Proporciona una capacidad de caché en memoria de alto desempeño, variable en tamaño y económicamente rentable, al tiempo que acaba con la complejidad propia de la implementación y la administración de un entorno de caché distribuido. ElastiCache es un protocolo compatible con Redis y Memcached, por lo que el código, las aplicaciones y las herramientas más conocidas que utiliza hoy en día con los entornos de Redis y Memcached existentes funcionarán a la perfección con el servicio. Para obtener más información acerca de ElastiCache, vaya a la página del producto [Amazon ElastiCache](#).

Para utilizar Elastic Beanstalk con Amazon ElastiCache

1. Cree un clúster de ElastiCache.
 - Para obtener instrucciones sobre cómo crear un clúster de ElastiCache con Redis, vaya a [Introducción a Amazon ElastiCache for Redis](#) en la Guía del usuario de ElastiCache for Redis.
 - Para obtener instrucciones sobre cómo crear un clúster de ElastiCache con Memcached, vaya a [Introducción a Amazon ElastiCache for Memcached](#) en la Guía del usuario de ElastiCache for Memcached.
2. Configure el grupo de seguridad de ElastiCache para permitir el acceso desde el grupo de seguridad de Amazon EC2 que se utiliza en la aplicación de Elastic Beanstalk. Para obtener instrucciones sobre cómo encontrar el nombre del grupo de seguridad de EC2 mediante la consola de administración de AWS, consulte [Grupos de seguridad](#) en la página del documento EC2 Instances (Instancias EC2).
 - Para obtener más información sobre Redis, vaya a [Autorizar acceso](#) en la Guía del usuario de ElastiCache for Redis.
 - Para obtener más información sobre Memcached, vaya a [Autorizar acceso](#) en la Guía del usuario de ElastiCache for Memcached.

Puede utilizar archivos de configuración para personalizar su entorno de Elastic Beanstalk y utilizar ElastiCache. Para obtener archivos de configuración de ejemplo que integran ElastiCache con Elastic Beanstalk, consulte [Ejemplo: ElastiCache](#).

Uso de Elastic Beanstalk con Amazon Elastic File System

Con Amazon Elastic File System (Amazon EFS), puede crear sistemas de archivos de red que las instancias de varias zonas de disponibilidad puedan montar. Un sistema de archivos de Amazon EFS es un recurso de AWS que utiliza grupos de seguridad para controlar el acceso en la red y que se encuentra en su VPC predeterminada o personalizada.

En un entorno de Elastic Beanstalk, puede utilizar Amazon EFS para crear un directorio compartido que almacene los archivos cargados o modificados por los usuarios de la aplicación. Su aplicación puede tratar un volumen de Amazon EFS montado, como el almacenamiento local. De este modo, no tiene que cambiar el código de la aplicación para escalar verticalmente a múltiples instancias.

Para obtener más información acerca de Amazon EFS, consulte la [Guía del usuario de Amazon Elastic File System](#).

Note

Elastic Beanstalk crea un usuario webapp que puede configurar como propietario de los directorios de aplicaciones en las instancias de Amazon EC2. Para obtener más información, consulte [Almacenamiento persistente](#) en el tema Consideraciones sobre el diseño de esta guía.

Secciones

- [Archivos de configuración](#)
- [Sistemas de archivos cifrados](#)
- [Aplicaciones de muestra](#)
- [Limpieza de sistemas de archivos](#)

Archivos de configuración

Elastic Beanstalk proporciona [archivos de configuración](#) que puede utilizar para crear y montar sistemas de archivos de Amazon EFS. Puede crear un volumen de Amazon EFS como parte de su entorno o montar un volumen de Amazon EFS que haya creado de manera independiente de Elastic Beanstalk.

- [storage-efs-createfilesystem.config](#): utiliza la clave `Resources` para crear un nuevo sistema de archivos y puntos de montaje en Amazon EFS. Todas las instancias de su entorno se pueden conectar al mismo sistema de archivos para disfrutar de almacenamiento escalable compartido. Use `storage-efs-mountfilesystem.config` para montar el sistema de archivos en cada instancia.

Recursos internos

Los recursos que crea con archivos de configuración se asocian al ciclo de vida del entorno. Si termina el entorno o elimina el archivo de configuración, estos recursos se pierden.

- [storage-efs-mountfilesystem.config](#): monte un sistema de archivos de Amazon EFS en una ruta local de las instancias de su entorno. Puede crear el volumen como parte del entorno con `storage-efs-createfilesystem.config`. O bien, puede montarlo en su entorno con la consola de Amazon EFS, AWS CLI, o bien AWS SDK.

Para utilizar archivos de configuración, empiece creando el sistema de archivos de Amazon EFS con `storage-efs-createfilesystem.config`. Siga las instrucciones del archivo de configuración y añádale al directorio [.ebextensions](#) del código fuente para crear el sistema de archivos en su VPC.

Implemente su código fuente actualizado en su entorno de Elastic Beanstalk. Esto sirve para confirmar que el sistema de archivos se ha creado correctamente. A continuación, añada `storage-efs-mountfilesystem.config` para montar el sistema de archivos en las instancias de su entorno. Realizar esta operación en dos implementaciones distintas asegura que si la operación de montaje falla, el sistema de archivos se mantiene intacto. Si la realiza en la misma implementación, un problema con algún paso hará que el sistema de archivos termine cuando se produzca un error en la implementación.

Sistemas de archivos cifrados

Amazon EFS admite sistemas de archivos cifrados. El archivo de configuración [storage-efs-createfilesystem.config](#) descrito en este tema define dos opciones personalizadas. Puede utilizar estas opciones para crear un sistema de archivos cifrados de Amazon EFS. Para obtener más información, consulte las instrucciones del archivo de configuración.

Aplicaciones de muestra

Elastic Beanstalk también ofrece aplicaciones de ejemplo que utilizan Amazon EFS para el almacenamiento compartido. Los dos proyectos tienen archivos de configuración que puede utilizar con un instalador estándar de WordPress o Drupal para ejecutar un blog u otro sistema de administración de contenido en un entorno con balanceo de carga. Cuando un usuario carga una foto u otro contenido multimedia, el archivo se almacena en un sistema de archivos de Amazon EFS. Esto evita tener que utilizar la alternativa, que es utilizar un complemento para almacenar los archivos cargados en Amazon S3.

- [WordPress con balanceo de carga](#): incluye los archivos de configuración para instalar WordPress de forma segura y ejecutarlo en un entorno de Elastic Beanstalk con balanceo de carga.
- [Drupal con balanceo de carga](#): incluye los archivos de configuración e instrucciones para instalar Drupal de forma segura y ejecutarlo en un entorno de Elastic Beanstalk con balanceo de carga.

Limpieza de sistemas de archivos

Si creó un sistema de archivos de Amazon EFS que utiliza un archivo de configuración como parte de su entorno de Elastic Beanstalk, Elastic Beanstalk eliminará el sistema de archivos cuando termine el entorno. Para minimizar los costos de almacenamiento de una aplicación en ejecución, elimine periódicamente los archivos que no necesite su aplicación. O bien, asegúrese de que el código de aplicación mantenga el ciclo de vida de los archivos correctamente.

Important

Si creó un sistema de archivos de Amazon EFS fuera de un entorno de Elastic Beanstalk y lo montó en las instancias del entorno, Elastic Beanstalk no eliminará el sistema de archivos cuando termine el entorno. Para asegurarse de que la información personal no se retiene y evitar costos de almacenamiento, elimine los archivos que almacenó su aplicación si ya no los necesita. Si lo prefiere, puede eliminar todo el sistema de archivos.

Uso de Elastic Beanstalk con AWS Identity and Access Management

AWS Identity and Access Management (IAM) le ayuda a controlar de forma segura el acceso a sus AWS recursos. Esta sección contiene materiales de referencia para utilizar las políticas, los perfiles de instancia y los roles de servicio de IAM.

Para obtener información general sobre los permisos, consulte [Roles de servicio, perfiles de instancia y políticas de usuario](#). En la mayoría de los entornos, el rol de servicio y el perfil de instancia que la consola de Elastic Beanstalk le pide que cree cuando inicia el primer entorno tienen todos los permisos necesarios. Del mismo modo, las [políticas administradas](#) que proporciona Elastic Beanstalk para obtener acceso completo y acceso de solo lectura contienen todos los permisos que necesita el usuario para su uso diario.

La [guía del usuario de IAM](#) ofrece información detallada sobre los AWS permisos.

Temas

- [Administración de perfiles de instancia de Elastic Beanstalk](#)
- [Administración de roles de servicio de Elastic Beanstalk](#)
- [Uso de roles vinculados a servicios para Elastic Beanstalk](#)
- [Administración de políticas de usuario de Elastic Beanstalk](#)
- [Formato de nombre de recurso de Amazon para Elastic Beanstalk](#)
- [Recursos y condiciones para las acciones de Elastic Beanstalk](#)
- [Uso de etiquetas para controlar el acceso a los recursos de Elastic Beanstalk](#)
- [Políticas de ejemplo basadas en políticas administradas](#)
- [Políticas de ejemplo basadas en permisos de recursos](#)
- [Impedir el acceso al bucket de Amazon S3 entre entornos](#)

Administración de perfiles de instancia de Elastic Beanstalk

Un perfil de instancia es un contenedor para un rol AWS Identity and Access Management (IAM) que puede usar para pasar información del rol a una instancia de Amazon EC2 cuando se inicia la instancia.

Si su AWS cuenta no tiene un perfil de instancia EC2, debe crear uno mediante el servicio de IAM. Después, puede asignar el perfil de instancia de EC2 a los nuevos entornos que cree. El asistente Crear entorno proporciona información que guía a través del servicio IAM, para que pueda crear un perfil de instancia de EC2 con los permisos necesarios. Después de crear el perfil de instancia, puede volver a la consola para seleccionarlo como perfil de instancia de EC2 y proseguir con los pasos para crear su entorno.

Note

Anteriormente, Elastic Beanstalk creaba un `aws-elasticbeanstalk-ec2-role` perfil de instancia EC2 predeterminado denominado la primera vez AWS que una cuenta creaba un entorno. Este perfil de instancia incluía políticas administradas predeterminadas. Si su cuenta ya tiene este perfil de instancia, seguirá estando disponible para que lo asigne a sus entornos.

Sin embargo, las directrices AWS de seguridad recientes no permiten que un AWS servicio cree funciones automáticamente con políticas de confianza para otros AWS servicios, en este caso EC2. Debido a estas directrices de seguridad, Elastic Beanstalk ya no crea un perfil de instancia `aws-elasticbeanstalk-ec2-role` predeterminado.

Políticas administradas

Elastic Beanstalk proporciona varias políticas administradas para permitir que su entorno satisfaga diferentes casos de uso. Para satisfacer los casos de uso predeterminados de un entorno, estas políticas se deben asociar al rol del perfil de instancia de EC2.

- `AWSElasticBeanstalkWebTier`— Otorga permisos para que la aplicación cargue registros en Amazon S3 y suba información de depuración a AWS X-Ray. Para ver el contenido de las políticas administradas, consulte [AWSElasticBeanstalkWebTier](#) la Guía de referencia de políticas AWS administradas.
- `AWSElasticBeanstalkWorkerTier`— Otorga permisos para la carga de registros, la depuración, la publicación de métricas y las tareas de creación de instancias de trabajo, incluidas la gestión de colas, la elección de líderes y las tareas periódicas. Para ver el contenido de las políticas gestionadas, consulte la Guía [AWSElasticBeanstalkWorkerTier](#) de referencia de políticas AWS gestionadas.
- `AWSElasticBeanstalkMulticontainerDocker`— Otorga permisos para que Amazon Elastic Container Service coordine las tareas del clúster para los entornos de Docker. Para ver el contenido de las

políticas administradas, consulte [AWSElasticBeanstalkMulticontainerDocker](#) la Guía de referencia de políticas AWS administradas.

⚠ Important

Las políticas administradas de Elastic Beanstalk no proporcionan permisos granulares, ya que otorgan todos los permisos potencialmente necesarios para trabajar con aplicaciones de Elastic Beanstalk. En algunos casos, es posible que desee restringir aún más los permisos de nuestras políticas administradas. Para ver un ejemplo de un caso de uso, consulte [Impedir el acceso al bucket de Amazon S3 entre entornos](#).

Nuestras políticas administradas tampoco incluyen los permisos para recursos personalizados que puede agregar a su solución y que no son administrados por Elastic Beanstalk. Para implementar permisos más granulares, permisos mínimos requeridos o permisos personalizados a nivel de recursos, utilice [políticas personalizadas](#).

Política de relación de confianza para EC2

Para permitir que las instancias de EC2 del entorno asuman el rol requerido, el perfil de instancia debe especificar Amazon EC2 como entidad de confianza en la política de relación de confianza de la siguiente manera.

```
{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Para personalizar los permisos, puede añadir políticas al rol asociado al perfil de instancia predeterminado o crear su propio perfil de instancia con un conjunto restringido de permisos.

Secciones

- [Creación de un perfil de instancia](#)
- [Verificación de los permisos asignados al perfil de instancia](#)
- [Actualización de un perfil de instancia out-of-date predeterminado](#)
- [Añadir permisos al perfil de instancia predeterminado](#)

Creación de un perfil de instancia

Un perfil de instancia es un contenedor de un rol de IAM estándar que permite a una instancia EC2 asumir el rol. Puede crear perfiles de instancia adicionales para personalizar los permisos de diferentes aplicaciones. O bien puede crear un perfil de instancia que no conceda permisos a los entornos Docker de capa de trabajo o administrados por ECS, si no utiliza esas características.

Cómo crear un perfil de instancia

1. Abra la [página Roles](#) en la consola de IAM
2. Elija Crear rol.
3. En Tipo de entidad de confianza, seleccione Servicio de AWS .
4. En Caso de uso, elija EC2.
5. Elija Siguiente.
6. Asocie las políticas administradas correspondientes proporcionadas por Elastic Beanstalk y cualquier otra política que proporcione permisos que necesite su aplicación.
7. Elija Siguiente.
8. Escriba un nombre para el rol.
9. (Opcional) Añada etiquetas al rol.
10. Elija Crear rol.

Verificación de los permisos asignados al perfil de instancia

Los permisos asignados al perfil de instancia predeterminado pueden variar en función de cuándo se creó el perfil, la última vez que se lanzó un entorno y el cliente que se utilizó. Puede verificar los permisos del perfil de instancia en la consola de IAM.

Para verificar los permisos del perfil de instancia predeterminado

1. Abra la [página Roles](#) en la consola de IAM

2. Seleccione el rol asignado como perfil de instancia de EC2.
3. Revise la lista de políticas asociadas al rol en la pestaña Permissions (Permisos).
4. Para ver los permisos que concede una política, elija la política.

Actualización de un perfil de instancia out-of-date predeterminado

Si al perfil de instancia predeterminado le faltan los permisos necesarios, puede agregar las políticas administradas al rol asignado como perfil de instancia de EC2 manualmente.

Para agregar políticas administradas al rol adjuntado al perfil de instancia predeterminado

1. Abra la [página Roles](#) en la consola de IAM
2. Seleccione el rol asignado como perfil de instancia de EC2.
3. En la pestaña Permissions (Permisos), elija Attach policies (Asociar políticas).
4. Escriba **AWSElasticBeanstalk** para filtrar las políticas.
5. Seleccione las siguientes políticas y, después, elija Attach policy (Asociar política):
 - AWSElasticBeanstalkWebTier
 - AWSElasticBeanstalkWorkerTier
 - AWSElasticBeanstalkMulticontainerDocker

Añadir permisos al perfil de instancia predeterminado

Si tu aplicación accede a AWS API o recursos para los que no se conceden permisos en el perfil de instancia predeterminado, añada políticas que concedan permisos en la consola de IAM.

Para agregar políticas al rol adjuntado al perfil de instancia predeterminado

1. Abra la [página Roles](#) en la consola de IAM.
2. Seleccione el rol asignado como perfil de instancia de EC2.
3. En la pestaña Permissions (Permisos), elija Attach policies (Adjuntar políticas).
4. Seleccione la política administrada que se aplicará a los demás servicios que utilice la aplicación. Por ejemplo, AmazonS3FullAccess o AmazonDynamoDBFullAccess.
5. Elija Asociar política.

Administración de roles de servicio de Elastic Beanstalk

Para administrar y monitorear su entorno, AWS Elastic Beanstalk realiza acciones en los recursos del entorno en su nombre. Elastic Beanstalk necesita determinados permisos para realizar estas acciones y AWS Identity and Access Management asume funciones de servicio (IAM) para obtener estos permisos.

Elastic Beanstalk debe utilizar las credenciales de seguridad temporales cada vez que asume un rol de servicio. Para obtener estas credenciales, Elastic Beanstalk envía una solicitud a AWS Security Token Service (AWS STS) en un punto de enlace específico de la región. Para obtener más información, consulte [Credenciales de seguridad temporales](#) en la guía del usuario de IAM.

Note

Si el AWS STS punto final de la región en la que se encuentra su entorno está desactivado, Elastic Beanstalk envía la solicitud a un punto final alternativo que no se puede desactivar. Este punto de enlace está asociado a una región diferente. Por lo tanto, la solicitud es una solicitud entre regiones. Para obtener más información, consulte [Activación y desactivación AWS STS en una AWS región en](#) la Guía del usuario de IAM.

Administración de roles de servicio mediante la consola de Elastic Beanstalk y la CLI de EB

Puede usar la consola de Elastic Beanstalk y la CLI de EB a fin de facilitar la configuración de roles de servicio para su entorno con un conjunto suficiente de permisos. Crean un rol de servicio predeterminado y usan políticas administradas en él.

Políticas de rol de servicio administradas

Elastic Beanstalk proporciona una política administrada para el [monitoreo del estado mejorado](#) y otra con los permisos adicionales necesarios para las [actualizaciones de plataforma administradas](#). La consola y la CLI de EB asignan ambas políticas al rol de servicio predeterminado que crean para usted. Solo se deben utilizar estas políticas para este rol de servicio predeterminado. No deben usarse con otros usuarios o roles de sus cuentas.

AWSElasticBeanstalkEnhancedHealth

Esta política concede permisos a Elastic Beanstalk para supervisar el estado de las instancias y del entorno.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticloadbalancing:DescribeInstanceHealth",
        "elasticloadbalancing:DescribeLoadBalancers",
        "elasticloadbalancing:DescribeTargetHealth",
        "ec2:DescribeInstances",
        "ec2:DescribeInstanceStatus",
        "ec2:GetConsoleOutput",
        "ec2:AssociateAddress",
        "ec2:DescribeAddresses",
        "ec2:DescribeSecurityGroups",
        "sqs:GetQueueAttributes",
        "sqs:GetQueueUrl",
        "autoscaling:DescribeAutoScalingGroups",
        "autoscaling:DescribeAutoScalingInstances",
        "autoscaling:DescribeScalingActivities",
        "autoscaling:DescribeNotificationConfigurations",
        "sns:Publish"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}

```

AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy

Esta política concede permisos a Elastic Beanstalk para actualizar los entornos en su nombre para realizar actualizaciones de plataformas administradas.

Agrupaciones de permisos de nivel de servicio

Esta política se agrupa en instrucciones basadas en el conjunto de permisos proporcionados.

- *ElasticBeanstalkPermissions*: este grupo de permisos es para llamar a las acciones del servicio Elastic Beanstalk (API de Elastic Beanstalk).

- *AllowPassRoleToElasticBeanstalkAndDownstreamServices*: este grupo de permisos permite que cualquier rol se transfiera a Elastic Beanstalk y a otros servicios descendentes como AWS CloudFormation.
- *ReadOnlyPermissions*: este grupo de permisos sirve para recopilar información sobre el entorno en ejecución.
- **OperationPermissions*: los grupos con este patrón de nomenclatura son para llamar a las operaciones necesarias a fin de llevar a cabo actualizaciones de plataforma.
- **BroadOperationPermissions*: los grupos con este patrón de nomenclatura son para llamar a las operaciones necesarias a fin de llevar a cabo actualizaciones de plataforma. También incluyen amplios permisos para admitir entornos heredados.
- **TagResource*— Los grupos con este patrón de nomenclatura son para las llamadas que utilizan las tag-on-create API para adjuntar etiquetas a los recursos que se están creando en un entorno de Elastic Beanstalk.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ElasticBeanstalkPermissions",
      "Effect": "Allow",
      "Action": [
        "elasticbeanstalk:*"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowPassRoleToElasticBeanstalkAndDownstreamServices",
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::*:role/*",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": [
            "elasticbeanstalk.amazonaws.com",
            "ec2.amazonaws.com",
            "ec2.amazonaws.com.cn",
            "autoscaling.amazonaws.com",
            "elasticloadbalancing.amazonaws.com",
            "ecs.amazonaws.com",

```

```

        "cloudformation.amazonaws.com"
    ]
}
},
{
  "Sid": "ReadOnlyPermissions",
  "Effect": "Allow",
  "Action": [
    "autoscaling:DescribeAccountLimits",
    "autoscaling:DescribeAutoScalingGroups",
    "autoscaling:DescribeAutoScalingInstances",
    "autoscaling:DescribeLaunchConfigurations",
    "autoscaling:DescribeLoadBalancers",
    "autoscaling:DescribeNotificationConfigurations",
    "autoscaling:DescribeScalingActivities",
    "autoscaling:DescribeScheduledActions",
    "ec2:DescribeAccountAttributes",
    "ec2:DescribeAddresses",
    "ec2:DescribeAvailabilityZones",
    "ec2:DescribeImages",
    "ec2:DescribeInstanceAttribute",
    "ec2:DescribeInstances",
    "ec2:DescribeKeyPairs",
    "ec2:DescribeLaunchTemplates",
    "ec2:DescribeLaunchTemplateVersions",
    "ec2:DescribeSecurityGroups",
    "ec2:DescribeSnapshots",
    "ec2:DescribeSpotInstanceRequests",
    "ec2:DescribeSubnets",
    "ec2:DescribeVpcClassicLink",
    "ec2:DescribeVpcs",
    "elasticloadbalancing:DescribeInstanceHealth",
    "elasticloadbalancing:DescribeLoadBalancers",
    "elasticloadbalancing:DescribeTargetGroups",
    "elasticloadbalancing:DescribeTargetHealth",
    "logs:DescribeLogGroups",
    "rds:DescribeDBEngineVersions",
    "rds:DescribeDBInstances",
    "rds:DescribeOrderableDBInstanceOptions",
    "sns:ListSubscriptionsByTopic"
  ],
  "Resource": [
    "*"
  ]
}
}
}

```



```

    ]
  },
  {
    "Sid": "EC2BroadOperationPermissions",
    "Effect": "Allow",
    "Action": [
      "ec2:AllocateAddress",
      "ec2:AssociateAddress",
      "ec2:AuthorizeSecurityGroupEgress",
      "ec2:AuthorizeSecurityGroupIngress",
      "ec2:CreateLaunchTemplate",
      "ec2:CreateLaunchTemplateVersion",
      "ec2:CreateSecurityGroup",
      "ec2>DeleteLaunchTemplate",
      "ec2>DeleteLaunchTemplateVersions",
      "ec2>DeleteSecurityGroup",
      "ec2:DisassociateAddress",
      "ec2:ReleaseAddress",
      "ec2:RevokeSecurityGroupEgress",
      "ec2:RevokeSecurityGroupIngress"
    ],
    "Resource": "*"
  },
  {
    "Sid": "EC2RunInstancesOperationPermissions",
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": "*",
    "Condition": {
      "ArnLike": {
        "ec2:LaunchTemplate": "arn:aws:ec2:*:*:launch-template/*"
      }
    }
  },
  {
    "Sid": "EC2TerminateInstancesOperationPermissions",
    "Effect": "Allow",
    "Action": [
      "ec2:TerminateInstances"
    ],
    "Resource": "arn:aws:ec2:*:*:instance/*",
    "Condition": {
      "StringLike": {
        "ec2:ResourceTag/aws:cloudformation:stack-id": [

```

```

                "arn:aws:cloudformation:*:*:stack/awseb-e-*",
                "arn:aws:cloudformation:*:*:stack/eb-*"
            ]
        }
    },
    {
        "Sid": "ECSBroadOperationPermissions",
        "Effect": "Allow",
        "Action": [
            "ecs:CreateCluster",
            "ecs:DescribeClusters",
            "ecs:RegisterTaskDefinition"
        ],
        "Resource": "*"
    },
    {
        "Sid": "ECSDeleteClusterOperationPermissions",
        "Effect": "Allow",
        "Action": "ecs:DeleteCluster",
        "Resource": "arn:aws:ecs:*:*:cluster/awseb-*"
    },
    {
        "Sid": "ASGOperationPermissions",
        "Effect": "Allow",
        "Action": [
            "autoscaling:AttachInstances",
            "autoscaling:CreateAutoScalingGroup",
            "autoscaling:CreateLaunchConfiguration",
            "autoscaling:CreateOrUpdateTags",
            "autoscaling>DeleteLaunchConfiguration",
            "autoscaling>DeleteAutoScalingGroup",
            "autoscaling>DeleteScheduledAction",
            "autoscaling:DetachInstances",
            "autoscaling>DeletePolicy",
            "autoscaling:PutScalingPolicy",
            "autoscaling:PutScheduledUpdateGroupAction",
            "autoscaling:PutNotificationConfiguration",
            "autoscaling:ResumeProcesses",
            "autoscaling:SetDesiredCapacity",
            "autoscaling:SuspendProcesses",
            "autoscaling:TerminateInstanceInAutoScalingGroup",
            "autoscaling:UpdateAutoScalingGroup"
        ],
    },

```

```

        "Resource": [
            "arn:aws:autoscaling:*:*:launchConfiguration:*:launchConfigurationName/
awseb-e-*",
            "arn:aws:autoscaling:*:*:launchConfiguration:*:launchConfigurationName/
eb-*",
            "arn:aws:autoscaling:*:*:autoScalingGroup:*:autoScalingGroupName/awseb-
e-*",
            "arn:aws:autoscaling:*:*:autoScalingGroup:*:autoScalingGroupName/eb-*"
        ]
    },
    {
        "Sid": "CFNOperationPermissions",
        "Effect": "Allow",
        "Action": [
            "cloudformation:*"
        ],
        "Resource": [
            "arn:aws:cloudformation:*:*:stack/awseb-*",
            "arn:aws:cloudformation:*:*:stack/eb-*"
        ]
    },
    {
        "Sid": "ELBOperationPermissions",
        "Effect": "Allow",
        "Action": [
            "elasticloadbalancing:AddTags",
            "elasticloadbalancing:ApplySecurityGroupsToLoadBalancer",
            "elasticloadbalancing:ConfigureHealthCheck",
            "elasticloadbalancing>CreateLoadBalancer",
            "elasticloadbalancing>DeleteLoadBalancer",
            "elasticloadbalancing:DeregisterInstancesFromLoadBalancer",
            "elasticloadbalancing:DeregisterTargets",
            "elasticloadbalancing:RegisterInstancesWithLoadBalancer",
            "elasticloadbalancing:RegisterTargets"
        ],
        "Resource": [
            "arn:aws:elasticloadbalancing:*:*:targetgroup/awseb-*",
            "arn:aws:elasticloadbalancing:*:*:targetgroup/eb-*",
            "arn:aws:elasticloadbalancing:*:*:loadbalancer/awseb-*",
            "arn:aws:elasticloadbalancing:*:*:loadbalancer/eb-*",
            "arn:aws:elasticloadbalancing:*:*:loadbalancer/*/awseb-*/*",
            "arn:aws:elasticloadbalancing:*:*:loadbalancer/*/eb-*/*"
        ]
    }
},

```

```
{
  "Sid": "CWLogsOperationPermissions",
  "Effect": "Allow",
  "Action": [
    "logs:CreateLogGroup",
    "logs>DeleteLogGroup",
    "logs:PutRetentionPolicy"
  ],
  "Resource": "arn:aws:logs:*:*:log-group:/aws/elasticbeanstalk/*"
},
{
  "Sid": "S3ObjectOperationPermissions",
  "Effect": "Allow",
  "Action": [
    "s3:DeleteObject",
    "s3:GetObject",
    "s3:GetObjectAcl",
    "s3:GetObjectVersion",
    "s3:GetObjectVersionAcl",
    "s3:PutObject",
    "s3:PutObjectAcl",
    "s3:PutObjectVersionAcl"
  ],
  "Resource": "arn:aws:s3:::elasticbeanstalk-*/*"
},
{
  "Sid": "S3BucketOperationPermissions",
  "Effect": "Allow",
  "Action": [
    "s3:GetBucketLocation",
    "s3:GetBucketPolicy",
    "s3:ListBucket",
    "s3:PutBucketPolicy"
  ],
  "Resource": "arn:aws:s3:::elasticbeanstalk-*"
},
{
  "Sid": "SNSOperationPermissions",
  "Effect": "Allow",
  "Action": [
    "sns:CreateTopic",
    "sns:GetTopicAttributes",
    "sns:SetTopicAttributes",
    "sns:Subscribe"
```

```

    ],
    "Resource": "arn:aws:sns:*:*:ElasticBeanstalkNotifications-*"
  },
  {
    "Sid": "SQSOperationPermissions",
    "Effect": "Allow",
    "Action": [
      "sqs:GetQueueAttributes",
      "sqs:GetQueueUrl"
    ],
    "Resource": [
      "arn:aws:sqs:*:*:awseb-e-*",
      "arn:aws:sqs:*:*:eb-*"
    ]
  },
  {
    "Sid": "CWPutMetricAlarmOperationPermissions",
    "Effect": "Allow",
    "Action": [
      "cloudwatch:PutMetricAlarm"
    ],
    "Resource": [
      "arn:aws:cloudwatch:*:*:alarm:awseb-*",
      "arn:aws:cloudwatch:*:*:alarm:eb-*"
    ]
  },
  {
    "Sid": "AllowECSTagResource",
    "Effect": "Allow",
    "Action": [
      "ecs:TagResource"
    ],
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "ecs:CreateAction": [
          "CreateCluster",
          "RegisterTaskDefinition"
        ]
      }
    }
  }
]

```

```
}
```

Para ver el contenido de una política administrada, también puede utilizar la [página Políticas \(Políticas\)](#) de la consola de IAM.

Important

Las políticas administradas de Elastic Beanstalk no proporcionan permisos granulares, ya que otorgan todos los permisos potencialmente necesarios para trabajar con aplicaciones de Elastic Beanstalk. En algunos casos, es posible que desee restringir aún más los permisos de nuestras políticas administradas. Para ver un ejemplo de un caso de uso, consulte [Impedir el acceso al bucket de Amazon S3 entre entornos](#).

Nuestras políticas administradas tampoco incluyen los permisos para recursos personalizados que puede agregar a su solución y que no son administrados por Elastic Beanstalk. Para implementar permisos más granulares, permisos mínimos requeridos o permisos personalizados a nivel de recursos, utilice [políticas personalizadas](#).

Políticas obsoletas administradas por

En el pasado, Elastic Beanstalk `AWSElasticBeanstalkService` respaldaba la política de funciones de servicio gestionado. Esta política se ha sustituido por `AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy`. Es posible que aún pueda ver y utilizar la política anterior en la consola de IAM.

Para ver el contenido de las políticas administradas, consulte [AWSElasticBeanstalkService](#) la Guía de referencia de políticas AWS administradas.

Sin embargo, le recomendamos que haga la transición al uso de la nueva política administrada (`AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy`). Agregue políticas personalizadas para conceder permisos a recursos personalizados, si tiene alguna.

Usar la consola de Elastic Beanstalk

Cuando lanza un entorno en la consola de Elastic Beanstalk, la consola crea un rol de servicio predeterminado denominado `aws-elasticbeanstalk-service-role` y le asigna políticas administradas con permisos predeterminados.

Para permitir que Elastic Beanstalk asuma el rol `aws-elasticbeanstalk-service-role`, el rol de servicio especifica Elastic Beanstalk como entidad de confianza en la directiva de relación de confianza.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "elasticbeanstalk.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "sts:ExternalId": "elasticbeanstalk"
        }
      }
    }
  ]
}
```

Cuando habilita [actualizaciones de plataforma administradas](#) en su entorno, Elastic Beanstalk asume un rol de servicio de actualizaciones administradas independiente para realizar actualizaciones administradas. De forma predeterminada, la consola de Elastic Beanstalk utiliza el mismo rol de servicio generado, `aws-elasticbeanstalk-service-role`, para el rol de servicio de actualizaciones administradas. Si cambia el rol de servicio predeterminado, la consola establece el rol de servicio de actualizaciones administradas para que utilice el rol vinculado al servicio de actualizaciones administradas, `AWSServiceRoleForElasticBeanstalkManagedUpdates`. Para obtener más información acerca de los roles vinculados a servicios, consulte [the section called “Uso de roles vinculados a servicios”](#).

Note

Debido a problemas de permisos, el servicio Elastic Beanstalk no siempre crea este rol vinculado a un servicio correctamente. Por lo tanto, la consola intenta crearlo de forma explícita. Para asegurarse que su cuenta tiene este rol vinculado al servicio, cree un entorno

como mínimo una vez mediante la consola y configure las actualizaciones administradas para que se habiliten antes de crear el entorno.

Uso de la CLI de EB

Si lanza un entorno mediante el comando [the section called “eb create”](#) de la interfaz de línea de comandos de Elastic Beanstalk (CLI de EB) y no especifica un rol de servicio a través de la opción `--service-role`, Elastic Beanstalk crea el rol de servicio predeterminado `aws-elasticbeanstalk-service-role`. Si el rol de servicio predeterminado ya existe, Elastic Beanstalk lo utiliza para el entorno nuevo. La consola de Elastic Beanstalk también realiza acciones similares en estas situaciones.

A diferencia de la consola, no puede especificar un rol de servicio de actualizaciones administradas cuando usa una opción de comando de la CLI de EB. Si habilita las actualizaciones administradas en el entorno, debe establecer el rol de servicio de actualizaciones administradas mediante las opciones de configuración. En el siguiente ejemplo se habilitan las actualizaciones administradas y se utiliza el rol de servicio predeterminado como rol de servicio de actualizaciones administradas.

Example `.ebextensions/ .config managed-platform-update`

```
option_settings:
  aws:elasticbeanstalk:managedactions:
    ManagedActionsEnabled: true
    PreferredStartTime: "Tue:09:00"
    ServiceRoleForManagedUpdates: "aws-elasticbeanstalk-service-role"
  aws:elasticbeanstalk:managedactions:platformupdate:
    UpdateLevel: patch
    InstanceRefreshEnabled: true
```

Administración de roles de servicio mediante la API de Elastic Beanstalk

Cuando utiliza la acción `CreateEnvironment` de la API de Elastic Beanstalk para crear un entorno, especifique un rol de servicio mediante la opción de configuración `ServiceRole` del espacio de nombres [aws:elasticbeanstalk:environment](#). Para obtener más información sobre el uso del monitoreo mejorado del estado con la API de Elastic Beanstalk, consulte [Uso de informes de estado mejorados con la API de Elastic Beanstalk](#).

Además, si habilita las [actualizaciones de plataformas administradas](#) en su entorno, puede especificar un rol de servicio de actualizaciones administradas

mediante la opción `ServiceRoleForManagedUpdates` del espacio de nombres [aws:elasticbeanstalk:managedactions](#).

Uso de roles vinculados a servicios

Un rol vinculado a un servicio es un tipo único de rol de servicio predefinido por Elastic Beanstalk para incluir todos los permisos que el servicio requiere para llamar a otros servicios en su nombre. AWS El rol vinculado al servicio está asociado a su cuenta. Elastic Beanstalk lo crea una vez y, a continuación, vuelve a utilizarlo al crear entornos adicionales. Para obtener más información sobre el uso de roles vinculados a servicios con entornos de Elastic Beanstalk, consulte [Uso de roles vinculados a servicios para Elastic Beanstalk](#).

Si crea un entorno mediante la API de Elastic Beanstalk y no especifica un rol de servicio, Elastic Beanstalk crea un [rol vinculado al servicio de monitoreo](#) para su cuenta, si todavía no existe. Elastic Beanstalk utiliza este rol para el entorno nuevo. También puede utilizar IAM para crear un rol vinculado al servicio de monitoreo de la cuenta con antelación. Una vez que su cuenta tenga este rol, puede usarla para crear un entorno mediante la API de Elastic Beanstalk, la consola de Elastic Beanstalk o la CLI de EB.

Si habilita las [actualizaciones de plataforma administradas](#) para el entorno y especifica `AWSServiceRoleForElasticBeanstalkManagedUpdates` como valor para la opción `ServiceRoleForManagedUpdates` del espacio de nombres [aws:elasticbeanstalk:managedactions](#), Elastic Beanstalk crea un [rol vinculado a servicios de actualizaciones administradas](#) para su cuenta, si todavía no existe. Elastic Beanstalk utiliza el a fin de realizar actualizaciones administradas para el entorno nuevo.

Note

Cuando Elastic Beanstalk intenta crear roles vinculados al servicio de actualizaciones administradas y de monitorización para su cuenta al crear un entorno, debe tener el permiso `iam:CreateServiceLinkedRole`. Si no tiene este permiso, se produce un error en la creación del entorno y aparece un mensaje que explica el problema.

Como alternativa, otro usuario con permiso para crear roles vinculados a servicios puede utilizar IAM para crear el rol vinculado al servicio con antelación. Con este método, no necesita el permiso `iam:CreateServiceLinkedRole` para crear su entorno.

Verificación de los permisos del rol de servicio predeterminado

Los permisos concedidos por el rol de servicio predeterminado pueden variar en función de cuándo se crearon, la última vez que se lanzó un entorno y el cliente que se utilizó. En la consola de IAM, puede verificar los permisos concedidos por el rol de servicio predeterminado.

Para verificar los permisos del rol de servicio predeterminado

1. En la consola de IAM, abra la [página Roles](#).
2. Elija `aws-elasticbeanstalk-service-role`
3. Revise la lista de políticas asociadas al rol en la pestaña Permissions (Permisos).
4. Para ver los permisos que concede una política, elija la política.

Actualizar un rol de servicio out-of-date predeterminado

Si al rol de servicio predeterminado le faltan los permisos necesarios, puede actualizarlo [creando un nuevo entorno](#) en la consola de administración del entorno de Elastic Beanstalk.

También puede agregar las políticas administradas al rol de servicio predeterminado de forma manual.

Para añadir políticas administradas al rol de servicio predeterminado

1. En la consola de IAM, abra la [página Roles](#).
2. Elija `aws-elasticbeanstalk-service-role`.
3. En la pestaña Permissions (Permisos), elija Attach policies (Adjuntar políticas).
4. Escriba **AWSElasticBeanstalk** para filtrar las políticas.
5. Seleccione las siguientes políticas y, después, elija Attach policy (Asociar política):
 - `AWSElasticBeanstalkEnhancedHealth`
 - `AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy`

Añadir permisos al rol de servicio predeterminado

Si la aplicación incluye archivos de configuración que hacen referencia a AWS recursos para los que los permisos no están incluidos en el rol de servicio predeterminado, es posible que Elastic Beanstalk necesite permisos adicionales. Estos permisos adicionales son necesarios para resolver estas

referencias cuando procesa los archivos de configuración durante una actualización administrada. Si faltan permisos, la actualización produce un error y Elastic Beanstalk devuelve un mensaje que indica el permiso que se necesita. Siga estos pasos a fin de agregar permisos para servicios adicionales al rol de servicio predeterminado en la consola de IAM.

Para añadir políticas adicionales al rol de servicio predeterminado

1. En la consola de IAM, abra la [página Roles](#).
2. Elija `aws-elasticbeanstalk-service-role`
3. En la pestaña Permissions (Permisos), elija Attach policies (Adjuntar políticas).
4. Seleccione la política administrada que se aplicará a los demás servicios que utilice la aplicación. Por ejemplo, `AmazonAPIGatewayAdministrator` o `AmazonElasticFileSystemFullAccess`.
5. Elija Asociar política.

Creación de un rol de servicio

Si no puede utilizar el rol de servicio predeterminado, cree un rol de servicio.

Para crear un rol de servicio

1. En la consola de IAM, abra la [página Roles](#).
2. Elija Crear rol.
3. En AWS service (Servicio de AWS), elija AWS Elastic Beanstalk y, a continuación, seleccione su caso de uso.
4. Elija Siguiente: permisos.
5. Asocie las políticas administradas `AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy` y `AWSElasticBeanstalkEnhancedHealth` y cualquier otra política que proporcione permisos que necesite su aplicación.
6. Elija Siguiente: etiquetas.
7. (Opcional) Añada etiquetas al rol.
8. Elija Siguiente: Revisar.
9. Escriba un nombre para el rol.
10. Elija Create role (Crear rol).

Aplique el rol de servicio personalizado cuando cree un entorno con el [asistente de creación del entorno](#) o la opción `--service-role` del comando `eb create`.

Uso de roles vinculados a servicios para Elastic Beanstalk

AWS Elastic Beanstalk [utiliza roles vinculados al servicio AWS Identity and Access Management \(IAM\)](#). Un rol vinculado a un servicio es un tipo único de rol de IAM que está vinculado directamente a Elastic Beanstalk. Elastic Beanstalk predefine las funciones vinculadas al servicio e incluyen todos los permisos que el servicio requiere para llamar a otros servicios en su nombre. AWS

Elastic Beanstalk define algunos tipos de roles vinculados a servicios:

- Rol vinculado al servicio de monitorización : permite a Elastic Beanstalk supervisar el estado de los entornos en ejecución y publicar notificaciones de eventos de monitorización.
- Rol vinculado al servicio de mantenimiento : permite a Elastic Beanstalk realizar actividades de mantenimiento regulares para sus entornos en funcionamiento.
- Rol vinculado al servicio de actualizaciones administradas: permite que Elastic Beanstalk realice actualizaciones programadas de plataforma de los entornos en ejecución.

Temas

- [Rol de monitorización vinculado al servicio](#)
- [Rol vinculado al servicio de mantenimiento](#)
- [Rol vinculado al servicio de actualizaciones administradas](#)

Rol de monitorización vinculado al servicio

AWS Elastic Beanstalk [utiliza roles vinculados al servicio AWS Identity and Access Management \(IAM\)](#). Un rol vinculado a un servicio es un tipo único de rol de IAM que está vinculado directamente a Elastic Beanstalk. Elastic Beanstalk predefine las funciones vinculadas al servicio e incluyen todos los permisos que el servicio requiere para llamar a otros servicios en su nombre. AWS

Un rol vinculado a un servicio simplifica la configuración de Elastic Beanstalk porque ya no tendrá que agregar manualmente los permisos necesarios. Elastic Beanstalk define los permisos de sus roles vinculados a servicio y, a menos que se defina lo contrario, solo Elastic Beanstalk puede asumir sus roles. Los permisos definidos incluyen las políticas de confianza y de permisos, y que la política de permisos no se pueda adjuntar a ninguna otra entidad de IAM.

Solo es posible eliminar un rol vinculado a un servicio después de eliminar sus recursos relacionados. De esta forma se protegen los recursos de Elastic Beanstalk, ya que evita que se puedan eliminar accidentalmente permisos de acceso a los recursos.

Para obtener información acerca de otros servicios que son compatibles con roles vinculados a servicios, consulte [Servicios de AWS que funcionan con IAM](#) y busque los servicios que muestran Yes (Sí) en la columna Service-Linked Role (Rol vinculado a servicios). Elija una opción Sí con un enlace para ver la documentación acerca del rol vinculado a servicios en cuestión.

Permisos de rol vinculado a servicios para Elastic Beanstalk

Elastic Beanstalk usa el rol vinculado al servicio AWSServiceRoleForElasticBeanstalk denominado: Permite que Elastic Beanstalk supervise el estado de los entornos en ejecución y publique notificaciones de eventos de estado.

El rol AWSServiceRoleForElasticBeanstalk vinculado al servicio confía en los siguientes servicios para asumir el rol:

- `elasticbeanstalk.amazonaws.com`

La política de permisos del rol AWSServiceRoleForElasticBeanstalk vinculado al servicio contiene todos los permisos que Elastic Beanstalk necesita para realizar acciones en su nombre:

AllowCloudformationReadOperationsOnElasticBeanstalkStacks

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCloudformationReadOperationsOnElasticBeanstalkStacks",
      "Effect": "Allow",
      "Action": [
        "cloudformation:DescribeStackResource",
        "cloudformation:DescribeStackResources",
        "cloudformation:DescribeStacks"
      ],
      "Resource": [
        "arn:aws:cloudformation:*:*:stack/awseb-*",
        "arn:aws:cloudformation:*:*:stack/eb-*"
      ]
    }
  ],
}
```

```
{
  "Sid": "AllowOperations",
  "Effect": "Allow",
  "Action": [
    "autoscaling:DescribeAutoScalingGroups",
    "autoscaling:DescribeAutoScalingInstances",
    "autoscaling:DescribeNotificationConfigurations",
    "autoscaling:DescribeScalingActivities",
    "autoscaling:PutNotificationConfiguration",
    "ec2:DescribeInstanceStatus",
    "ec2:AssociateAddress",
    "ec2:DescribeAddresses",
    "ec2:DescribeInstances",
    "ec2:DescribeSecurityGroups",
    "elasticloadbalancing:DescribeInstanceHealth",
    "elasticloadbalancing:DescribeLoadBalancers",
    "elasticloadbalancing:DescribeTargetHealth",
    "elasticloadbalancing:DescribeTargetGroups",
    "sqs:GetQueueAttributes",
    "sqs:GetQueueUrl",
    "sns:Publish"
  ],
  "Resource": [
    "*"
  ]
}
```

Debe configurar permisos para permitir a una entidad de IAM (como un usuario, grupo o rol) crear, editar o eliminar un rol vinculado a servicios. Para obtener más información, consulte [Permisos de roles vinculados a servicios](#) en la Guía del usuario de IAM.

Como alternativa, puede usar una política AWS administrada para [proporcionar acceso total](#) a Elastic Beanstalk.

Creación de un rol vinculado a servicios para Elastic Beanstalk

No necesita crear manualmente un rol vinculado a servicios. Cuando crea un entorno de Elastic Beanstalk mediante la API de Elastic Beanstalk y no especifica un rol de servicio, Elastic Beanstalk crea el rol vinculado a servicios.

⚠ Important

Si utilizaba el servicio Elastic Beanstalk antes del 27 de septiembre de 2017, cuando comenzó a `AWSServiceRoleForElasticBeanstalk` admitir el rol vinculado al servicio, y su cuenta lo necesitaba, Elastic Beanstalk creó el rol en su cuenta. `AWSServiceRoleForElasticBeanstalk` Para obtener más información, consulte [Un nuevo rol ha aparecido en mi cuenta de IAM](#).

Cuando Elastic Beanstalk intente `AWSServiceRoleForElasticBeanstalk` crear el rol vinculado al servicio para su cuenta al crear un entorno, debe tener el permiso.

`iam:CreateServiceLinkedRole` Si no tiene este permiso, se produce un error en la creación del entorno y verá un mensaje que explica el problema.

Como alternativa, otro usuario con permiso para crear roles vinculados a servicios puede utilizar IAM para crear previamente el rol vinculado al servicio con antelación. A continuación, puede crear su entorno incluso sin tener el permiso `iam:CreateServiceLinkedRole`.

Usted (u otro usuario) pueden utilizar la consola de IAM para crear un rol vinculado a servicios con el caso de uso de Elastic Beanstalk. En la CLI de IAM o la API de IAM, cree un rol vinculado a servicio con el nombre de servicio `elasticbeanstalk.amazonaws.com`. Para obtener más información, consulte [Crear un rol vinculado a un servicio](#) en la Guía del usuario de IAM. Si elimina este rol vinculado al servicio, puede utilizar este mismo proceso para volver a crear el rol.

Si elimina este rol vinculado a servicios y necesita crearlo de nuevo, puede utilizar el mismo proceso para volver a crear el rol en su cuenta. Cuando crea un entorno de Elastic Beanstalk utilizando la API de Elastic Beanstalk y no especifica una función de servicio, Elastic Beanstalk crea de nuevo el rol vinculado a servicios.

Edición de un rol vinculado a servicios para Elastic Beanstalk

Elastic Beanstalk no permite editar el rol vinculado al servicio. `AWSServiceRoleForElasticBeanstalk` Después de crear un rol vinculado al servicio, no podrá cambiar el nombre del rol, ya que varias entidades podrían hacer referencia al rol. Sin embargo, sí puede editar la descripción del rol con IAM. Para obtener más información, consulte [Editar un rol vinculado a servicios](#) en la Guía del usuario de IAM.

Eliminación de un rol vinculado a servicios para Elastic Beanstalk

Si ya no necesita usar una característica o servicio que requieran un rol vinculado a un servicio, le recomendamos que elimine dicho rol. Así no tendrá una entidad no utilizada que no se monitorice ni mantenga de forma activa. Sin embargo, debe limpiar los recursos de su rol vinculado al servicio antes de eliminarlo manualmente.

Saneamiento de un rol vinculado a servicios

Antes de poder utilizar IAM para eliminar un rol vinculado a servicios, primero debe asegurarse de que todos los entornos de Elastic Beanstalk están utilizando un rol de servicio diferente o de que están terminados.

Note

Si el servicio Elastic Beanstalk está utilizando el rol vinculado al servicio cuando intenta terminar los entornos, podría producirse un error en la finalización. En tal caso, espere unos minutos e intente de nuevo la operación.

Para terminar un entorno de Elastic Beanstalk que usa la (consola) `AWSServiceRoleForElasticBeanstalk`

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. Elija Actions (Acciones) y, a continuación, Terminate Environment (Terminar el entorno).
4. Utilice el cuadro de diálogo en pantalla para confirmar la terminación del entorno.

Consulte [eb terminate](#) para obtener más información sobre cómo terminar un entorno de Elastic Beanstalk mediante la CLI de EB.

Consulte [TerminateEnvironment](#) para obtener más información sobre la finalización de un entorno de Elastic Beanstalk mediante la API.

Eliminación manual de un rol vinculado a servicios

Utilice la consola de IAM, la CLI de IAM o la API de IAM para eliminar el rol vinculado al `AWSServiceRoleForElasticBeanstalk` servicio. Para más información, consulte [Eliminación de un rol vinculado a servicios](#) en la Guía del usuario de IAM.

Regiones admitidas para roles vinculados a servicios de Elastic Beanstalk

Elastic Beanstalk admite el uso de roles vinculados a servicios en todas las regiones en las que el servicio está disponible. Para obtener más información, consulte [Puntos de conexión y cuotas de AWS Elastic Beanstalk](#).

Rol vinculado al servicio de mantenimiento

AWS Elastic Beanstalk [utiliza roles vinculados al servicio AWS Identity and Access Management \(IAM\)](#). Un rol vinculado a un servicio es un tipo único de rol de IAM que está vinculado directamente a Elastic Beanstalk. Elastic Beanstalk predefine las funciones vinculadas al servicio e incluyen todos los permisos que el servicio requiere para llamar a otros servicios en su nombre. AWS

Un rol vinculado a un servicio simplifica la configuración de Elastic Beanstalk porque ya no tendrá que agregar manualmente los permisos necesarios. Elastic Beanstalk define los permisos de sus roles vinculados a servicio y, a menos que se defina lo contrario, solo Elastic Beanstalk puede asumir sus roles. Los permisos definidos incluyen las políticas de confianza y de permisos, y que la política de permisos no se pueda adjuntar a ninguna otra entidad de IAM.

Solo es posible eliminar un rol vinculado a un servicio después de eliminar sus recursos relacionados. De esta forma se protegen los recursos de Elastic Beanstalk, ya que evita que se puedan eliminar accidentalmente permisos de acceso a los recursos.

Para obtener información acerca de otros servicios que son compatibles con roles vinculados a servicios, consulte [Servicios de AWS que funcionan con IAM](#) y busque los servicios que muestran Yes (Sí) en la columna Service-Linked Role (Rol vinculado a servicios). Elija una opción Sí con un enlace para ver la documentación acerca del rol vinculado a servicios en cuestión.

Permisos de rol vinculado a servicios para Elastic Beanstalk

Elastic Beanstalk usa el rol vinculado al servicio

`AWSServiceRoleForElasticBeanstalkMaintenance` denominado: Permite a Elastic Beanstalk realizar actividades de mantenimiento periódicas para los entornos en ejecución.

El rol `AWSServiceRoleForElasticBeanstalkMaintenance` vinculado al servicio confía en los siguientes servicios para asumir el rol:

- `maintenance.elasticbeanstalk.amazonaws.com`

La política de permisos del rol `AWSServiceRoleForElasticBeanstalkMaintenance` vinculado al servicio contiene todos los permisos que Elastic Beanstalk necesita para realizar acciones en su nombre:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCloudformationChangeSetOperationsOnElasticBeanstalkStacks",
      "Effect": "Allow",
      "Action": [
        "cloudformation:CreateChangeSet",
        "cloudformation:DescribeChangeSet",
        "cloudformation:ExecuteChangeSet",
        "cloudformation>DeleteChangeSet",
        "cloudformation:ListChangeSets",
        "cloudformation:DescribeStacks"
      ],
      "Resource": [
        "arn:aws:cloudformation:*:*:stack/awseb-*",
        "arn:aws:cloudformation:*:*:stack/eb-*"
      ]
    }
  ]
}
```

Debe configurar permisos para permitir a una entidad de IAM (como un usuario, grupo o rol) crear, editar o eliminar un rol vinculado a servicios. Para obtener más información, consulte [Permisos de roles vinculados a servicios](#) en la Guía del usuario de IAM.

Como alternativa, puede usar una política AWS administrada para [proporcionar acceso total](#) a Elastic Beanstalk.

Creación de un rol vinculado a servicios para Elastic Beanstalk

No necesita crear manualmente un rol vinculado a servicios. Cuando crea un entorno de Elastic Beanstalk mediante la API de Elastic Beanstalk y no especifica un perfil de instancia, Elastic Beanstalk crea el rol vinculado a servicios.

⚠ Important

Este rol vinculado a servicios puede aparecer en su cuenta si se ha completado una acción en otro servicio que utilice las características compatibles con este rol. Si utilizaba el servicio Elastic Beanstalk antes del 18 de abril de 2019, cuando comenzó a `AWSServiceRoleForElasticBeanstalkMaintenance` admitir el rol vinculado al servicio, y su cuenta lo necesitaba, Elastic Beanstalk creó el rol en su cuenta. `AWSServiceRoleForElasticBeanstalkMaintenance` Para obtener más información, consulte [Un nuevo rol ha aparecido en mi cuenta de IAM](#).

Si elimina este rol vinculado a servicios y necesita crearlo de nuevo, puede utilizar el mismo proceso para volver a crear el rol en su cuenta. Cuando crea un entorno de Elastic Beanstalk utilizando la API de Elastic Beanstalk y no especifica un perfil de instancia, Elastic Beanstalk crea de nuevo el rol vinculado a servicios.

Edición de un rol vinculado a servicios para Elastic Beanstalk

Elastic Beanstalk no permite editar el rol vinculado al servicio.

`AWSServiceRoleForElasticBeanstalkMaintenance` Después de crear un rol vinculado al servicio, no podrá cambiar el nombre del rol, ya que varias entidades podrían hacer referencia al rol. Sin embargo, sí puede editar la descripción del rol con IAM. Para obtener más información, consulte [Editar un rol vinculado a servicios](#) en la Guía del usuario de IAM.

Eliminación de un rol vinculado a servicios para Elastic Beanstalk

Si ya no necesita usar una característica o servicio que requieran un rol vinculado a un servicio, le recomendamos que elimine dicho rol. Así no tendrá una entidad no utilizada que no se monitorice ni mantenga de forma activa. Sin embargo, debe limpiar los recursos de su rol vinculado al servicio antes de eliminarlo manualmente.

Saneamiento de un rol vinculado a servicios

Para poder utilizar IAM para eliminar un rol vinculado al servicio, primero debe terminar los entornos de Elastic Beanstalk que utiliza el rol.

Note

Si el servicio Elastic Beanstalk está utilizando el rol vinculado al servicio cuando intenta terminar los entornos, podría producirse un error en la finalización. En tal caso, espere unos minutos e intente de nuevo la operación.

Para terminar un entorno de Elastic Beanstalk que usa la (consola)
AWSServiceRoleForElasticBeanstalkMaintenance

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. Elija Actions (Acciones) y, a continuación, Terminate Environment (Terminar el entorno).
4. Utilice el cuadro de diálogo en pantalla para confirmar la terminación del entorno.

Consulte [eb terminate](#) para obtener más información sobre cómo terminar un entorno de Elastic Beanstalk mediante la CLI de EB.

Consulte [TerminateEnvironment](#) para obtener más información sobre la finalización de un entorno de Elastic Beanstalk mediante la API.

Eliminación manual de un rol vinculado a servicios

Utilice la consola de IAM, la CLI de IAM o la API de IAM para eliminar el rol vinculado al AWSServiceRoleForElasticBeanstalkMaintenance servicio. Para más información, consulte [Eliminación de un rol vinculado a servicios](#) en la Guía del usuario de IAM.

Regiones admitidas para roles vinculados a servicios de Elastic Beanstalk

Elastic Beanstalk admite el uso de roles vinculados a servicios en todas las regiones en las que el servicio está disponible. Para obtener más información, consulte [Puntos de conexión y cuotas de AWS Elastic Beanstalk](#).

Rol vinculado al servicio de actualizaciones administradas

AWS Elastic Beanstalk [utiliza roles vinculados al servicio AWS Identity and Access Management \(IAM\)](#). Un rol vinculado a un servicio es un tipo único de rol de IAM que está vinculado directamente a Elastic Beanstalk. Elastic Beanstalk predefine las funciones vinculadas al servicio e incluyen todos los permisos que el servicio requiere para llamar a otros servicios en su nombre. AWS

Un rol vinculado a un servicio simplifica la configuración de Elastic Beanstalk porque ya no tendrá que agregar manualmente los permisos necesarios. Elastic Beanstalk define los permisos de sus roles vinculados a servicio y, a menos que se defina lo contrario, solo Elastic Beanstalk puede asumir sus roles. Los permisos definidos incluyen las políticas de confianza y de permisos, y que la política de permisos no se pueda adjuntar a ninguna otra entidad de IAM.

Solo es posible eliminar un rol vinculado a un servicio después de eliminar sus recursos relacionados. De esta forma se protegen los recursos de Elastic Beanstalk, ya que evita que se puedan eliminar accidentalmente permisos de acceso a los recursos.

Para obtener información acerca de otros servicios que son compatibles con roles vinculados a servicios, consulte [Servicios de AWS que funcionan con IAM](#) y busque los servicios que muestran Yes (Sí) en la columna Service-Linked Role (Rol vinculado a servicios). Elija una opción Sí con un enlace para ver la documentación acerca del rol vinculado a servicios en cuestión.

Permisos de rol vinculado a servicios para Elastic Beanstalk

Elastic Beanstalk usa el rol vinculado al servicio

`AWSServiceRoleForElasticBeanstalkManagedUpdates` denominado: Permite a Elastic Beanstalk realizar actualizaciones de plataforma programadas de los entornos en ejecución.

El rol `AWSServiceRoleForElasticBeanstalkManagedUpdates` vinculado al servicio confía en los siguientes servicios para asumir el rol:

- `managedupdates.elasticbeanstalk.amazonaws.com`

La política administrada `AWSElasticBeanstalkManagedUpdatesServiceRolePolicy` otorga al rol `AWSServiceRoleForElasticBeanstalkManagedUpdates` vinculado al servicio todos los permisos que Elastic Beanstalk necesita para completar las acciones de actualización administrada en su nombre. Para ver el contenido de las políticas administradas, consulte la [AWSElasticBeanstalkManagedUpdatesServiceRolePolicy](#) página de la Guía de referencia de políticas administradas.AWS

Debe configurar permisos para permitir a una entidad de IAM (como un usuario, grupo o rol) crear, editar o eliminar un rol vinculado a servicios. Para obtener más información, consulte [Permisos de roles vinculados a servicios](#) en la Guía del usuario de IAM.

Como alternativa, puede usar una política AWS administrada para [proporcionar acceso total](#) a Elastic Beanstalk.

Creación de un rol vinculado a servicios para Elastic Beanstalk

No necesita crear manualmente un rol vinculado a servicios. Cuando crea un entorno de Elastic Beanstalk mediante la API de Elastic Beanstalk, habilita las actualizaciones administradas y especifica `AWSServiceRoleForElasticBeanstalkManagedUpdates` como valor para la opción `ServiceRoleForManagedUpdates` del espacio de nombres [aws:elasticbeanstalk:managedactions](#), Elastic Beanstalk crea el rol vinculado a servicios.

Cuando Elastic Beanstalk intente `AWSServiceRoleForElasticBeanstalkManagedUpdates` crear el rol vinculado al servicio para su cuenta al crear un entorno, debe tener el permiso `iam:CreateServiceLinkedRole`. Si no tiene este permiso, se produce un error en la creación del entorno y verá un mensaje que explica el problema.

Como alternativa, otro usuario con permiso para crear roles vinculados a servicios puede utilizar IAM para crear previamente el rol vinculado al servicio con antelación. A continuación, puede crear su entorno incluso sin tener el permiso `iam:CreateServiceLinkedRole`.

Usted (u otro usuario) pueden utilizar la consola de IAM para crear un rol vinculado a servicios con el caso de uso de Elastic Beanstalk Managed Updates (Actualizaciones administradas de Elastic Beanstalk). En la CLI de IAM o la API de IAM, cree un rol vinculado a servicio con el nombre de servicio `managedupdates.elasticbeanstalk.amazonaws.com`. Para obtener más información, consulte [Crear un rol vinculado a un servicio](#) en la Guía del usuario de IAM. Si elimina este rol vinculado al servicio, puede utilizar este mismo proceso para volver a crear el rol.

Si elimina este rol vinculado a servicios y necesita crearlo de nuevo, puede utilizar el mismo proceso para volver a crear el rol en su cuenta. Cuando crea un entorno de Elastic Beanstalk mediante la API de Elastic Beanstalk, habilita las actualizaciones administradas y especifica `AWSServiceRoleForElasticBeanstalkManagedUpdates` como valor para la opción `ServiceRoleForManagedUpdates` del espacio de nombres [aws:elasticbeanstalk:managedactions](#), Elastic Beanstalk crea la función vinculada a servicios de nuevo.

Edición de un rol vinculado a servicios para Elastic Beanstalk

Elastic Beanstalk no permite editar el rol vinculado al servicio.

`AWSServiceRoleForElasticBeanstalkManagedUpdates` Después de crear un rol vinculado al servicio, no podrá cambiar el nombre del rol, ya que varias entidades podrían hacer referencia al rol. Sin embargo, sí puede editar la descripción del rol con IAM. Para obtener más información, consulte [Editar un rol vinculado a servicios](#) en la Guía del usuario de IAM.

Eliminación de un rol vinculado a servicios para Elastic Beanstalk

Si ya no necesita usar una característica o servicio que requieran un rol vinculado a un servicio, le recomendamos que elimine dicho rol. Así no tendrá una entidad no utilizada que no se monitorice ni mantenga de forma activa. Sin embargo, debe limpiar los recursos de su rol vinculado al servicio antes de eliminarlo manualmente.

Saneamiento de un rol vinculado a servicios

Antes de poder utilizar IAM para eliminar un rol vinculado a servicios, primero debe asegurarse de que los entornos de Elastic Beanstalk con actualizaciones administradas habilitadas utilizan un rol de servicio diferente o están terminados.

Note

Si el servicio Elastic Beanstalk está utilizando el rol vinculado al servicio cuando intenta terminar los entornos, podría producirse un error en la finalización. En tal caso, espere unos minutos e intente de nuevo la operación.

Para terminar un entorno de Elastic Beanstalk que usa la (consola)

`AWSServiceRoleForElasticBeanstalkManagedUpdates`

1. Abra la consola de [Elastic Beanstalk](#) y, en la lista Regiones, seleccione su. Región de AWS
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. Elija **Actions** (Acciones) y, a continuación, **Terminate Environment** (Terminar entorno).
4. Utilice el cuadro de diálogo en pantalla para confirmar la terminación del entorno.

Consulte [eb terminate](#) para obtener más información sobre cómo terminar un entorno de Elastic Beanstalk mediante la CLI de EB.

Consulte [TerminateEnvironment](#) para obtener más información sobre la finalización de un entorno de Elastic Beanstalk mediante la API.

Eliminación manual de un rol vinculado a servicios

Utilice la consola de IAM, la CLI de IAM o la API de IAM para eliminar el rol vinculado al `AWSServiceRoleForElasticBeanstalkManagedUpdates` servicio. Para más información, consulte [Eliminación de un rol vinculado a servicios](#) en la Guía del usuario de IAM.

Regiones admitidas para roles vinculados a servicios de Elastic Beanstalk

Elastic Beanstalk admite el uso de roles vinculados a servicios en todas las regiones en las que el servicio está disponible. Para obtener más información, consulte [Puntos de conexión y cuotas de AWS Elastic Beanstalk](#).

Administración de políticas de usuario de Elastic Beanstalk

AWS Elastic Beanstalk proporciona dos políticas administradas que le permiten asignar acceso total o acceso de solo lectura a todos los recursos que administra Elastic Beanstalk. Puede adjuntar las políticas a los usuarios o grupos AWS Identity and Access Management (de IAM) o a las funciones que asumen sus usuarios.

Políticas de usuario administradas

- **AdministratorAccess- AWSElasticBeanstalk** — Otorga al usuario todos los permisos administrativos para crear, modificar y eliminar las aplicaciones, las versiones de las aplicaciones, los ajustes de configuración, los entornos y sus recursos subyacentes de Elastic Beanstalk. Para ver el contenido de las políticas administradas, consulte la `AWSElasticBeanstalk` página [AdministratorAccess-](#) de la Guía de referencia de políticas AWS administradas.
- **AWSElasticBeanstalkReadOnly**— Permite al usuario ver las aplicaciones y los entornos, pero no realizar operaciones que los modifiquen. Proporciona acceso de solo lectura a todos los recursos de Elastic Beanstalk y a otros recursos AWS que recupera la consola de Elastic Beanstalk. Tenga en cuenta que el acceso de solo lectura no permite realizar acciones como descargar

archivos registro de Elastic Beanstalk para poder consultarlos. Esto se debe a que los registros se almacenan en el bucket de Amazon S3, donde Elastic Beanstalk requeriría permiso de escritura. Consulte el ejemplo al final de este tema para obtener información sobre cómo habilitar el acceso a los logs de Elastic Beanstalk. Para ver el contenido de las políticas administradas, consulte la página de la Guía de referencia de políticas administradas. [AWSElasticBeanstalkReadOnlyAWS](#)

Important

Las políticas administradas de Elastic Beanstalk no proporcionan permisos granulares, ya que otorgan todos los permisos potencialmente necesarios para trabajar con aplicaciones de Elastic Beanstalk. En algunos casos, es posible que desee restringir aún más los permisos de nuestras políticas administradas. Para ver un ejemplo de un caso de uso, consulte [Impedir el acceso al bucket de Amazon S3 entre entornos](#).

Nuestras políticas administradas tampoco incluyen los permisos para recursos personalizados que puede agregar a su solución y que no son administrados por Elastic Beanstalk. Para implementar permisos más granulares, permisos mínimos requeridos o permisos personalizados a nivel de recursos, utilice [políticas personalizadas](#).

Políticas obsoletas administradas por

Anteriormente, Elastic Beanstalk admitía otras dos políticas de usuarios administrados y. `AWSElasticBeanstalkFullAccess` `AWSElasticBeanstalkReadOnlyAccess` Planeamos retirar estas políticas anteriores. Es posible que aún pueda verlas y utilizarlas en la consola de IAM. No obstante, se recomienda que realice la transición al uso de las nuevas políticas de usuario administradas y agregue políticas personalizadas para conceder permisos a recursos personalizados, si tiene alguno.

Políticas para la integración con otros servicios

También ofrecemos políticas más detalladas que le permiten integrar su entorno con otros servicios, si prefiere utilizarlos.

- `AWSElasticBeanstalkRoleCWL`— Permite que un entorno gestione los grupos de CloudWatch registros de Amazon Logs.
- `AWSElasticBeanstalkRoleRDS`— Permite que un entorno integre una instancia de Amazon RDS.
- `AWSElasticBeanstalkRoleWorkerTier`— Permite a un nivel de entorno de trabajo crear una tabla de Amazon DynamoDB y una cola de Amazon SQS.

- **AWSElasticBeanstalkRoleECS**— Permite que un entorno Docker multicontenedor gestione los clústeres de Amazon ECS.
- **AWSElasticBeanstalkRoleCore**— Permite las operaciones principales de un entorno de servicios web.
- **AWSElasticBeanstalkRoleSNS**— Permite que un entorno permita la integración de temas de Amazon SNS.

Para ver la fuente JSON de una política administrada específica, consulte la [Guía de referencia de políticas AWS administradas](#).

Control del acceso con políticas administradas

Puede usar políticas administradas para conceder acceso completo o acceso de solo lectura a Elastic Beanstalk. Elastic Beanstalk actualiza estas políticas automáticamente cuando se necesitan permisos adicionales para tener acceso a nuevas características.

Para aplicar una política administrada a usuarios o grupos de IAM

1. Abra la [página Políticas \(Políticas\)](#) en la consola de IAM.
2. Escriba **AWSElasticBeanstalk** en el cuadro de búsqueda para filtrar las políticas.
3. En la lista de políticas, selecciona la casilla de verificación situada junto a **AWSElasticBeanstalkReadOnly AdministratorAccess- AWSElasticBeanstalk**.
4. Seleccione Policy actions (Acciones de la política) y, a continuación, Attach (Adjuntar).
5. Seleccione uno o varios usuarios y grupos a los que asociar la política. Puede utilizar el menú Filter (Filtro) y el cuadro de búsqueda para filtrar la lista entidades principales.
6. Elija Asociar política.

Creación de una política de usuario personalizada

Puede crear su propia política de IAM para permitir o denegar acciones específicas de la API de Elastic Beanstalk en recursos específicos de Elastic Beanstalk y controlar el acceso a recursos personalizados que no son administrados por Elastic Beanstalk. Para obtener más información acerca de cómo asociar una política a un usuario o grupo, consulte [Trabajar con políticas](#) en la Guía del usuario de IAM. Para obtener más información acerca de cómo crear una política personalizada, consulte [Creación de políticas de IAM](#) en la Guía del usuario de IAM.

Note

Aunque puede restringir la forma en que un usuario interactúa con las API de Elastic Beanstalk, actualmente no existe ninguna manera eficaz de impedir que los usuarios que tienen permiso para crear los recursos subyacentes necesarios creen otros recursos en Amazon EC2 y otros servicios.

Piense en estas políticas como un medio eficaz de distribuir las responsabilidades de Elastic Beanstalk, no como una forma de proteger todos los recursos subyacentes.

En noviembre de 2019, Elastic Beanstalk lanzó soporte para [plantillas de lanzamiento de Amazon EC2](#). Se trata de un nuevo tipo de recurso que el grupo de Auto Scaling del entorno puede utilizar para lanzar instancias Amazon EC2 y que requiere nuevos permisos. La mayoría de los clientes no deberían verse afectados, ya que los entornos pueden seguir utilizando las configuraciones de lanzamiento del recurso heredado si la política del usuario carece de los permisos necesarios. Sin embargo, si intenta utilizar una nueva característica que requiera plantillas de lanzamiento de Amazon EC2 y tiene una política personalizada, es posible que se produzca un error al crear o actualizar el entorno. En este caso, asegúrese de que la política personalizada tenga los siguientes permisos.


Permisos necesarios para las plantillas de inicio de Amazon EC2

- EC2:CreateLaunchTemplate
- EC2:CreateLaunchTemplateVersions
- EC2>DeleteLaunchTemplate
- EC2>DeleteLaunchTemplateVersions
- EC2:DescribeLaunchTemplate
- EC2:DescribeLaunchTemplateVersions

Una política de IAM contiene instrucciones de política que describen los permisos que desea conceder. Cuando cree una instrucción de política de Elastic Beanstalk, debe saber cómo usar las cuatro partes siguientes de una instrucción de política:

- El efecto especifica si se permiten o deniegan las acciones de la instrucción.
- La acción especifica las [operaciones de la API](#) que desea controlar. Por ejemplo, utilice `elasticbeanstalk:CreateEnvironment` para especificar la operación

CreateEnvironment. Algunas operaciones, como la creación de un entorno, requieren permisos adicionales. Para obtener más información, consulte [Recursos y condiciones para las acciones de Elastic Beanstalk](#).

 Note

Para utilizar la operación API [UpdateTagsForResource](#), especifique una de las siguientes dos acciones virtuales (o ambas) en lugar del nombre de operación API:

`elasticbeanstalk:AddTags`

Controla los permisos para llamar a UpdateTagsForResource y pasar una lista de etiquetas para añadir en el parámetro TagsToAdd.

`elasticbeanstalk:RemoveTags`

Controla el permiso para llamar a UpdateTagsForResource y pasar una lista de claves de etiquetas para eliminar del parámetro TagsToRemove.

- El recurso especifica los recursos cuyo acceso desea controlar. Para especificar recursos de Elastic Beanstalk, enumere el [nombre de recurso de Amazon](#) (ARN) de cada recurso.
- (opcional) La condición especifica las restricciones sobre los permisos concedidos en la instrucción. Para obtener más información, consulte [Recursos y condiciones para las acciones de Elastic Beanstalk](#).

En las siguientes secciones se muestran algunos casos en los que puede considerar el uso de una política de usuario personalizada.

Habilitar la creación del entorno limitado de Elastic Beanstalk

La política del siguiente ejemplo permite al usuario llamar a la acción CreateEnvironment para crear un entorno cuyo nombre comience por **Test** con la aplicación y versión de la aplicación especificadas.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CreateEnvironmentPerm",
      "Action": [
        "elasticbeanstalk:CreateEnvironment"
      ]
    }
  ]
}
```

```

    ],
    "Effect": "Allow",
    "Resource": [
        "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My First Elastic
        Beanstalk Application/Test*"
    ],
    "Condition": {
        "StringEquals": {
            "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-
            east-2:123456789012:application/My First Elastic Beanstalk Application"],
            "elasticbeanstalk:FromApplicationVersion": ["arn:aws:elasticbeanstalk:us-
            east-2:123456789012:applicationversion/My First Elastic Beanstalk Application/First
            Release"]
        }
    }
},
{
    "Sid": "AllNonResourceCalls",
    "Action": [
        "elasticbeanstalk:CheckDNSAvailability",
        "elasticbeanstalk:CreateStorageLocation"
    ],
    "Effect": "Allow",
    "Resource": [
        "*"
    ]
}
]
}

```

La política anterior muestra cómo conceder acceso limitado a las operaciones de Elastic Beanstalk. Para poder lanzar realmente un entorno, el usuario debe tener permiso para crear también los AWS recursos que alimentan el entorno. Por ejemplo, la siguiente política concede acceso al conjunto predeterminado de recursos para un entorno de servidor web:

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:*",
                "ecs:*",

```

```

    "elasticloadbalancing:*",
    "autoscaling:*",
    "cloudwatch:*",
    "s3:*",
    "sns:*",
    "cloudformation:*",
    "sqs:*"
  ],
  "Resource": "*"
}
]
}

```

Habilitar el acceso a los registros de Elastic Beanstalk almacenados en Amazon S3

La política del siguiente ejemplo permite a un usuario extraer registros de Elastic Beanstalk, almacenarlos en un escenario en Amazon S3 y recuperarlos.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:DeleteObject",
        "s3:GetObjectAcl",
        "s3:PutObjectAcl"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::elasticbeanstalk-*"
    }
  ]
}

```

Note

Para restringir estos permisos únicamente a la ruta de registros, utilice el siguiente formato de recursos.

```

"arn:aws:s3:::elasticbeanstalk-us-east-2-123456789012/resources/environments/
logs/*"

```

Habilitación de la administración de una aplicación específica de Elastic Beanstalk

La política del siguiente ejemplo permite a un usuario administrar entornos y otros recursos en una aplicación de Elastic Beanstalk específica. La política niega las acciones de Elastic Beanstalk en recursos de otras aplicaciones, y también niega la creación y eliminación de aplicaciones de Elastic Beanstalk.

Note

La política no deniega el acceso a los recursos a través de otros servicios. Demuestra una forma eficaz de distribuir responsabilidades para administrar las aplicaciones de Elastic Beanstalk entre diferentes usuarios, no como una forma de proteger los recursos subyacentes.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "elasticbeanstalk:CreateApplication",
        "elasticbeanstalk>DeleteApplication"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Deny",
      "Action": [
        "elasticbeanstalk:CreateApplicationVersion",
        "elasticbeanstalk:CreateConfigurationTemplate",
        "elasticbeanstalk:CreateEnvironment",
        "elasticbeanstalk>DeleteApplicationVersion",
        "elasticbeanstalk>DeleteConfigurationTemplate",
        "elasticbeanstalk>DeleteEnvironmentConfiguration",
        "elasticbeanstalk:DescribeApplicationVersions",
        "elasticbeanstalk:DescribeConfigurationOptions",
        "elasticbeanstalk:DescribeConfigurationSettings",
        "elasticbeanstalk:DescribeEnvironmentResources",
        "elasticbeanstalk:DescribeEnvironments",
```

```

    "elasticbeanstalk:DescribeEvents",
    "elasticbeanstalk>DeleteEnvironmentConfiguration",
    "elasticbeanstalk:RebuildEnvironment",
    "elasticbeanstalk:RequestEnvironmentInfo",
    "elasticbeanstalk:RestartAppServer",
    "elasticbeanstalk:RetrieveEnvironmentInfo",
    "elasticbeanstalk:SwapEnvironmentCNAMEs",
    "elasticbeanstalk:TerminateEnvironment",
    "elasticbeanstalk:UpdateApplicationVersion",
    "elasticbeanstalk:UpdateConfigurationTemplate",
    "elasticbeanstalk:UpdateEnvironment",
    "elasticbeanstalk:RetrieveEnvironmentInfo",
    "elasticbeanstalk:ValidateConfigurationSettings"
  ],
  "Resource": [
    "*"
  ],
  "Condition": {
    "StringNotEquals": {
      "elasticbeanstalk:InApplication": [
        "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/myapplication"
      ]
    }
  }
}
]
}

```

Formato de nombre de recurso de Amazon para Elastic Beanstalk

Especifique un recurso para una política de IAM que use el Nombre de recurso de Amazon (ARN) de ese recurso. Para Elastic Beanstalk, el ARN tiene el siguiente formato.

```
arn:aws:elasticbeanstalk:region:account-id:resource-type/resource-path
```

Donde:

- *region* es la región en la que se encuentra el recurso (por ejemplo, **us-west-2**).
- *account-id* es el ID de cuenta de AWS, sin guiones (por ejemplo, **123456789012**).

- *resource-type* identifica el tipo del recurso Elastic Beanstalk, por ejemplo, `environment`. Consulte la siguiente tabla para obtener una lista de todos los tipos de recursos de Elastic Beanstalk.
- *resource-path* es la parte que identifica el recurso específico. Un recurso Elastic Beanstalk tiene una ruta que identifica de forma única ese recurso. Consulte la tabla que aparece a continuación para ver el formato de la ruta de recurso de cada tipo de recurso. Por ejemplo, un entorno siempre está asociado a una aplicación. La ruta de recurso del entorno **myEnvironment** en la aplicación **myApp** sería la siguiente:

```
myApp/myEnvironment
```

Elastic Beanstalk tiene varios tipos de recursos que puede especificar en una política. La siguiente tabla muestra el formato de ARN de cada tipo de recurso y un ejemplo.

Tipo de recurso	Formato de ARN
application	<pre>arn:aws:elasticbeanstalk: <i>region</i>:<i>account-id</i> :application/<i>application-name</i></pre> <p>Ejemplo: arn:aws:elasticbeanstalk:us-east-2:1234567890:12:application/My App</p>
applicationversion	<pre>arn:aws:elasticbeanstalk: <i>region</i>:<i>account-id</i> :applicationversion/<i>application-name</i> /<i>version-label</i></pre> <p>Ejemplo: arn:aws:elasticbeanstalk:us-east-2:1234567890:12:applicationversion/My App/My Version</p>
configurationtemplate	<pre>arn:aws:elasticbeanstalk: <i>region</i>:<i>account-id</i> :configurationtemplate/<i>application-name</i> /<i>template-name</i></pre> <p>Ejemplo: arn:aws:elasticbeanstalk:us-east-2:1234567890:12:configurationtemplate/My App/My Template</p>
environment	<pre>arn:aws:elasticbeanstalk: <i>region</i>:<i>account-id</i> :environment/<i>application-name</i> /<i>environment-name</i></pre>

Tipo de recurso	Formato de ARN
platform	<p>Ejemplo: arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/MyEnvironment</p> <p>arn:aws:elasticbeanstalk: <i>region:account-id</i> :platform / <i>platform-name</i> /<i>platform-version</i></p> <p>Ejemplo: arn:aws:elasticbeanstalk:us-east-2:123456789012:platform/MyPlatform/1.0</p>
solutions tack	<p>arn:aws:elasticbeanstalk: <i>region::solutionstack/ solutions tack-name</i></p> <p>Ejemplo: arn:aws:elasticbeanstalk:us-east-2::solutions tack/32bit Amazon Linux running Tomcat 7</p>

Un entorno, versión de la aplicación y plantilla de configuración siempre están incluidos en una aplicación específica. Observará que estos recursos tienen un nombre de aplicación en su ruta de recurso para que se identifiquen inequívocamente por su nombre de recurso y la aplicación que contienen. Aunque las plantillas de configuración y los entornos utilizan pilas de soluciones, estas pilas no son específicas de una aplicación o cuenta de AWS y no tienen la aplicación o cuenta de AWS en su ARN.

Recursos y condiciones para las acciones de Elastic Beanstalk

En esta sección se describen los recursos y condiciones que puede usar en instrucciones de política para conceder permisos que permitan realizar acciones de Elastic Beanstalk específicas en recursos de Elastic Beanstalk.

Las condiciones le permiten especificar permisos a los recursos que necesita la acción para completarse. Por ejemplo, cuando llama a la acción `CreateEnvironment`, debe especificar también la versión de la aplicación que se va a implementar, así como la aplicación que contiene ese nombre de aplicación. Cuando configura permisos para la acción `CreateEnvironment`, especifica la aplicación y la versión de la aplicación en las que desee que actúe la acción con las condiciones `InApplication` y `FromApplicationVersion`.

Además, puede especificar la configuración del entorno con una pila de soluciones (FromSolutionStack) o una plantilla de configuración (FromConfigurationTemplate). La siguiente instrucción de política permite que la acción CreateEnvironment cree un entorno con el nombre **myenv** (especificado por Resource) en la aplicación **My App** (especificada por la condición InApplication) que utilice la versión de la aplicación **My Version** (FromApplicationVersion) con una configuración de **32bit Amazon Linux running Tomcat 7** (FromSolutionStack):

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "elasticbeanstalk:CreateEnvironment"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"
      ],
      "Condition": {
        "StringEquals": {
          "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"],
          "elasticbeanstalk:FromApplicationVersion": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/My App/My Version"],
          "elasticbeanstalk:FromSolutionStack": ["arn:aws:elasticbeanstalk:us-east-2::solutionstack/32bit Amazon Linux running Tomcat 7"]
        }
      }
    }
  ]
}
```

Note

La mayoría de las claves de condición mencionadas en este tema son específicas de Elastic Beanstalk y sus nombres contienen el prefijo `elasticbeanstalk:`. Por cuestiones de brevedad, omitimos este prefijo de los nombres de clave de condición cuando los mencionamos en las siguientes secciones. Por ejemplo, mencionamos `InApplication` en lugar de su nombre completo `elasticbeanstalk:InApplication`.

Por el contrario, mencionamos algunas claves de condición utilizadas en los servicios de AWS e incluimos su prefijo `aws:` para destacar la excepción. Los ejemplos de política siempre muestran los nombres de claves de condición completos, incluido el prefijo.

Secciones

- [Información sobre la política de las acciones de Elastic Beanstalk](#)
- [Claves de condición para acciones de Elastic Beanstalk](#)

Información sobre la política de las acciones de Elastic Beanstalk

En la siguiente tabla se muestran todas las acciones de Elastic Beanstalk, el recurso en el que actúa cada acción e información contextual adicional que se puede proporcionar utilizando condiciones.

Información de políticas de Elastic Beanstalk, incluidos recursos, condiciones, ejemplos y dependencias

Recurso	Condiciones	Instrucción de ejemplo
Acción: AbortEnvironmentUpdate		
application environment	aws:ResourceTag/ <i>key-name</i> (Optional) aws:TagKeys (Optional)	<p>La siguiente política permite a un usuario cancelar las operaciones de actualización de entornos en una aplicación denominada My App.</p> <pre>{ "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:AbortEnvironmentUpdate"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"] }] }</pre>

Recurso	Condiciones	Instrucción de ejemplo
		<pre data-bbox="743 212 797 281">] }</pre>

Acción: [CheckDNSAvailability](#)

"*"	N/A	<pre data-bbox="743 449 1308 957">{ "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:CheckDNS vailability"], "Effect": "Allow", "Resource": "*" }] }</pre>
-----	-----	---

Acción: [ComposeEnvironments](#)

Recurso	Condiciones	Instrucción de ejemplo
application	<code>aws:ResourceTag/ <i>key-name</i></code> (Optional) <code>aws:TagKeys</code> (Optional)	<p>La siguiente política permite a un usuario crear entornos que pertenezcan a una aplicación denominada My App.</p> <pre>{ "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:ComposeEnvironments"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App"] }] }</pre>

Acción: [CreateApplication](#)

Recurso	Condiciones	Instrucción de ejemplo
application	<code>aws:RequestTag/ <i>key-name</i></code> (Optional) <code>aws:TagKeys</code> (Optional)	<p>Este ejemplo permite que la acción <code>CreateApplication</code> cree aplicaciones cuyos nombres comienzan por DivA:</p> <pre>{ "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:CreateApplication"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/DivA*"] }] }</pre>

Acción: [CreateApplicationVersion](#)

Recurso	Condiciones	Instrucción de ejemplo
applicationversion	InApplication aws:RequestTag/ <i>key-name</i> (Optional) aws:TagKeys (Optional)	<p>Este ejemplo permite a la acción <code>CreateApplicationVersion</code> crear versiones de la aplicación con el nombre (*) en la aplicación My App:</p> <pre> { "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:CreateApplicationVersion"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/MyApp/*"], "Condition": { "StringEquals": { "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"] } } }] } </pre>

Acción: [CreateConfigurationTemplate](#)

Recurso	Condiciones	Instrucción de ejemplo
configurationtemplate	InApplication FromApplication FromApplicationVersion FromConfigurationTemplate FromEnvironment FromSolutionStack aws:RequestTag/ <i>key-name</i> (Optional) aws:TagKeys (Optional)	<p>La siguiente política permite que la acción <code>CreateConfigurationTemplate</code> cree plantillas de configuración cuyos nombres comienzan por My Template (<code>My Template*</code>) en la aplicación My App.</p> <pre> { "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:CreateConfigurationTemplate"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:configurationtemplate/My App/My Template*"], "Condition": { "StringEquals": { "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App", "elasticbeanstalk:FromSolutionStack": ["arn:aws:elasticbeanstalk:us-east-2::solutionstack/32bit Amazon Linux running Tomcat 7"]] } } }] } </pre>

Acción: [CreateEnvironment](#)

Recurso	Condiciones	Instrucción de ejemplo
environment	InApplication FromApplicationVersion FromConfigurationTemplate FromSolutionStack aws:RequestTag/ <i>key-name</i> (Optional) aws:TagKeys (Optional)	<p>La siguiente política permite que la acción <code>CreateEnvironment</code> cree un entorno cuyo nombre es myenv en la aplicación My App que utilice la pila de soluciones 32bit Amazon Linux running Tomcat 7:</p> <pre> { "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:CreateEnvironment"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"], "Condition": { "StringEquals": { "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App", "elasticbeanstalk:FromApplicationVersion": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/My App/My Version", "elasticbeanstalk:FromSolutionStack": ["arn:aws:elasticbeanstalk:us-east-2::solutionstack/32bit Amazon Linux running Tomcat 7"]]] } } }] } </pre>

Acción: [CreatePlatformVersion](#)

Recurso	Condiciones	Instrucción de ejemplo
platform	aws:RequestTag/ <i>key-name</i> (Optional) aws:TagKeys (Optional)	<p>En este ejemplo, se permite que la acción CreatePlatformVersion cree las versiones de la plataforma que se dirigen a la región us-east-2, cuyos nombres comienzan por us-east-2_:</p> <pre>{ "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:CreatePlatformVersion"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:platform/us-east-2_*"] }] }</pre>

Acción: [CreateStorageLocation](#)

"*"	N/A	<pre>{ "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:CreateStorageLocation"], "Effect": "Allow", "Resource": "*" }] }</pre>
-----	-----	---

Recurso	Condiciones	Instrucción de ejemplo
Acción: DeleteApplication		
application	aws:ResourceTag/ <i>key-name</i> (Optional) aws:TagKeys (Optional)	<p>La siguiente política permite que la acción DeleteApplication elimine la aplicación My App:</p> <pre data-bbox="730 430 1507 1102"> { "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:DeleteApplication"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"] }] } </pre>
Acción: DeleteApplicationVersion		

Recurso	Condiciones	Instrucción de ejemplo
applicati onversion	InApplica tion aws:Resou rceTag/ <i>key- name</i> (Optional) aws:TagKeys (Optional)	<p>La siguiente política permite que la acción DeleteApp licationVersion elimine una versión de la aplicación cuyo nombre sea My Version en la aplicación My App:</p> <pre data-bbox="732 443 1507 1436"> { "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:DeleteApp licationVersion"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us- east-2:123456789012:applicationversion/My App/My Version"], "Condition": { "StringEquals": { "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2 :123456789012:application/My App"] } } }] } </pre>

Acción: [DeleteConfigurationTemplate](#)

Recurso	Condiciones	Instrucción de ejemplo
configura tiontempl ate	InApplica tion (Optional) aws:Resou rceTag/ <i>key-</i> <i>name</i> (Optional) aws:TagKeys (Optional)	<p>La siguiente política permite que la acción <code>DeleteConfigurationTemplate</code> elimine una plantilla de configuración cuyo nombre sea My Template en la aplicación My App. La especificación del nombre de aplicación como condición es opcional.</p> <pre data-bbox="732 491 1507 1205"> { "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:DeleteCon figurationTemplate"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us- east-2:123456789012:configurationtemplate/ My App/My Template"] }] } </pre>

Acción: [DeleteEnvironmentConfiguration](#)

Recurso	Condiciones	Instrucción de ejemplo
environment	InApplication (Optional)	<p>La siguiente política permite que la acción <code>DeleteEnvironmentConfiguration</code> elimine una configuración temporal del entorno myenv en la aplicación My App. La especificación del nombre de aplicación como condición es opcional.</p> <pre>{ "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:DeleteEnvironmentConfiguration"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"] }] }</pre>

Acción: [DeletePlatformVersion](#)

Recurso	Condiciones	Instrucción de ejemplo
platform	aws:ResourceTag/ <i>key-name</i> (Optional) aws:TagKeys (Optional)	<p>La siguiente política permite que la acción DeletePlatformVersion elimine las versiones de la plataforma que se dirigen a la región us-east-2 , cuyos nombres comienzan por us-east-2_ :</p> <pre data-bbox="732 443 1507 1115"> { "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:DeletePlatformVersion"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:platform/us-east-2_*"] }] }</pre>

Acción: [DescribeApplications](#)

Recurso	Condiciones	Instrucción de ejemplo
application	aws:ResourceTag/ <i>key-name</i> (Optional) aws:TagKeys (Optional)	<p>La siguiente política permite que la acción DescribeApplications describa la aplicación My App.</p> <pre data-bbox="730 346 1502 1018">{ "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:DescribeApplications"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"] }] }</pre>

Acción: [DescribeApplicationVersions](#)

Recurso	Condiciones	Instrucción de ejemplo
applicati onversion	InApplica tion (Optional) aws:Resou rceTag/ <i>key- name</i> (Optional) aws:TagKeys (Optional)	<p>La siguiente política permite que la acción DescribeApplicationVersions describa una versión de la aplicación cuyo nombre sea My Version en la aplicación My App. La especificación del nombre de aplicación como condición es opcional.</p> <pre data-bbox="732 491 1507 1205"> { "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:DescribeA pplicationVersions"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us- east-2:123456789012:applicationversion/My App/My Version"] }] } </pre>

Acción: [DescribeConfigurationOptions](#)

Recurso	Condiciones	Instrucción de ejemplo
environment configurationtemplate solutions tack	InApplication (Optional) aws:ResourceTag/ <i>key-name</i> (Optional) aws:TagKeys (Optional)	<p>La siguiente política permite que la acción DescribeConfigurationOptions describa las opciones de configuración del entorno myenv en la aplicación My App. La especificación del nombre de aplicación como condición es opcional.</p> <pre data-bbox="732 491 1507 1125"> { "Version": "2012-10-17", "Statement": [{ "Action": "elasticbeanstalk:DescribeConfigurationOptions", "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"] }] } </pre>

Acción: [DescribeConfigurationSettings](#)

Recurso	Condiciones	Instrucción de ejemplo
environment configurationtemplate	InApplication (Optional) aws:ResourceTag/ <i>key-name</i> (Optional) aws:TagKeys (Optional)	<p>La siguiente política permite que la acción DescribeConfigurationSettings describa los ajustes de configuración del entorno myenv en la aplicación My App. La especificación del nombre de aplicación como condición es opcional.</p> <pre data-bbox="732 491 1507 1125"> { "Version": "2012-10-17", "Statement": [{ "Action": "elasticbeanstalk:DescribeConfigurationSettings", "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"] }] } </pre>

Acción: [DescribeEnvironmentHealth](#)

Recurso	Condiciones	Instrucción de ejemplo
environment	<code>aws:ResourceTag/ <i>key-name</i></code> (Optional) <code>aws:TagKeys</code> (Optional)	<p>La siguiente política permite el uso de <code>DescribeEnvironmentHealth</code> para recuperar información de estado de un entorno llamado myenv.</p> <pre>{ "Version": "2012-10-17", "Statement": [{ "Action": "elasticbeanstalk:DescribeEnvironmentHealth", "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"] }] }</pre>

Acción: [DescribeEnvironmentResources](#)

Recurso	Condiciones	Instrucción de ejemplo
environment	InApplication (Optional) aws:ResourceTag/ <i>key-name</i> (Optional) aws:TagKeys (Optional)	<p>La siguiente política permite que la acción DescribeEnvironmentResources devuelva la lista de recursos de AWS del entorno myenv en la aplicación My App. La especificación del nombre de aplicación como condición es opcional.</p> <pre>{ "Version": "2012-10-17", "Statement": [{ "Action": "elasticbeanstalk:DescribeEnvironmentResources", "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"] }] }</pre>

Acción: [DescribeEnvironments](#)

Recurso	Condiciones	Instrucción de ejemplo
environment	InApplication (Optional) aws:ResourceTag/ <i>key-name</i> (Optional) aws:TagKeys (Optional)	<p>La siguiente política permite que la acción DescribeEnvironments describa los entornos myenv y myotherenv en la aplicación My App. La especificación del nombre de aplicación como condición es opcional.</p> <pre data-bbox="730 493 1507 1249"> { "Version": "2012-10-17", "Statement": [{ "Action": "elasticbeanstalk:DescribeEnvironments", "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv", "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App2/myotherenv"] }] } </pre>

Acción: [DescribeEvents](#)

Recurso	Condiciones	Instrucción de ejemplo
application applicationversion configurationtemplate environment	InApplication aws:ResourceTag/ <i>key-name</i> (Optional) aws:TagKeys (Optional)	<p>La siguiente política permite que la acción DescribeEvents muestre las descripciones de eventos del entorno myenv y la versión de la aplicación My Version en la aplicación My App.</p> <pre> { "Version": "2012-10-17", "Statement": [{ "Action": "elasticbeanstalk:DescribeEvents", "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv", "arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/My App/My Version"], "Condition": { "StringEquals": { "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"] } } }] } </pre>

Acción: [DescribeInstancesHealth](#)

Recurso	Condiciones	Instrucción de ejemplo
environment	N/A	<p>La siguiente política permite el uso de <code>DescribeInstancesHealth</code> para recuperar información de estado de las instancias de un entorno llamado myenv.</p> <pre data-bbox="730 441 1507 1081">{ "Version": "2012-10-17", "Statement": [{ "Action": "elasticbeanstalk:DescribeInstancesHealth", "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"] }] }</pre>

Acción: [DescribePlatformVersion](#)

Recurso	Condiciones	Instrucción de ejemplo
platform	aws:ResourceTag/ <i>key-name</i> (Optional) aws:TagKeys (Optional)	<p>La siguiente política permite que la acción DescribePlatformVersion describa las versiones de la plataforma que se dirigen a la región us-east-2, cuyos nombres comienzan por us-east-2_:</p> <pre>{ "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:DescribePlatformVersion"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:platform/us-east-2_*"] }] }</pre>

Acción: [ListAvailableSolutionStacks](#)

Recurso	Condiciones	Instrucción de ejemplo
solutions tack	N/A	<p>La siguiente política permite que la acción <code>ListAvailableSolutionStacks</code> devuelva únicamente la pila de soluciones 32bit Amazon Linux running Tomcat 7.</p> <pre data-bbox="730 441 1507 1071">{ "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:ListAvailableSolutionStacks"], "Effect": "Allow", "Resource": "arn:aws:elasticbeanstalk:us-east-2::solutionstack/32bit Amazon Linux running Tomcat 7" }] }</pre>

Acción: [ListPlatformVersions](#)

Recurso	Condiciones	Instrucción de ejemplo
platform	<code>aws:RequestTag/ <i>key-name</i></code> (Optional) <code>aws:TagKeys</code> (Optional)	<p>En este ejemplo, se permite que la acción <code>CreatePlatformVersion</code> cree las versiones de la plataforma que se dirigen a la región <code>us-east-2</code>, cuyos nombres comienzan por <code>us-east-2_</code>:</p> <pre>{ "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:ListPlatformVersions"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:platform/us-east-2_*"] }] }</pre>

Acción: [ListTagsForResource](#)

Recurso	Condiciones	Instrucción de ejemplo
application applicationversion configurationtemplate environment platform	aws:ResourceTag/ <i>key-name</i> (Optional) aws:TagKeys (Optional)	<p>La siguiente política permite que la acción <code>ListTagsForResource</code> obtenga una lista de etiquetas de recursos existentes solo si tienen una etiqueta denominada <code>stage</code> con el valor <code>test</code>:</p> <pre> { "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:ListTagsForResource"], "Effect": "Allow", "Resource": "*", "Condition": { "StringEquals": { "aws:ResourceTag/stage": ["test"] } } }] } </pre>

Acción: [RebuildEnvironment](#)

Recurso	Condiciones	Instrucción de ejemplo
environment	InApplication aws:ResourceTag/ <i>key-name</i> (Optional) aws:TagKeys (Optional)	<p>La siguiente política permite que la acción <code>RebuildEnvironment</code> vuelva a crear el entorno <code>myenv</code> en la aplicación <code>My App</code>.</p> <pre data-bbox="732 394 1507 1388"> { "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:RebuildEnvironment"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"], "Condition": { "StringEquals": { "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"] } } }] } </pre>

Acción: [RequestEnvironmentInfo](#)

Recurso	Condiciones	Instrucción de ejemplo
environment	InApplication aws:ResourceTag/ <i>key-name</i> (Optional) aws:TagKeys (Optional)	<p>La siguiente política permite que la acción RequestEnvironmentInfo recopile información del entorno myenv en la aplicación My App.</p> <pre data-bbox="732 394 1507 1388"> { "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:RequestEnvironmentInfo"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"], "Condition": { "StringEquals": { "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"] } } }] } </pre>

Acción: [RestartAppServer](#)

Recurso	Condiciones	Instrucción de ejemplo
environment	InApplication	<p>La siguiente política permite que la acción <code>RestartAppServer</code> reinicie el servidor contenedor de la aplicación del entorno <code>myenv</code> en la aplicación <code>My App</code>.</p> <pre data-bbox="732 394 1507 1346">{ "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:RestartAppServer"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"], "Condition": { "StringEquals": { "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"] } } }] }</pre>

Acción: [RetrieveEnvironmentInfo](#)

Recurso	Condiciones	Instrucción de ejemplo
environment	InApplication aws:ResourceTag/ <i>key-name</i> (Optional) aws:TagKeys (Optional)	<p>La siguiente política permite que la acción <code>RetrieveEnvironmentInfo</code> recupere la información recopilada del entorno myenv en la aplicación My App.</p> <pre data-bbox="732 443 1507 1436"> { "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:RetrieveEnvironmentInfo"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"], "Condition": { "StringEquals": { "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"] } } }] } </pre>

Acción: [SwapEnvironmentCNAMEs](#)

Recurso	Condiciones	Instrucción de ejemplo
environment	InApplication (Optional) FromEnvironment (Optional)	<p>La siguiente política permite que la acción <code>SwapEnvironmentCNAMEs</code> intercambie los registros CNAME de los entornos <code>mysrcenv</code> y <code>mydestenv</code>.</p> <pre data-bbox="730 394 1507 1228"> { "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:SwapEnvironmentCNAMEs"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/mysrcenv", "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/mydestenv"] }] } </pre>

Acción: [TerminateEnvironment](#)

Recurso	Condiciones	Instrucción de ejemplo
environment	<p>InApplication</p> <p>aws:ResourceTag/ <i>key-name</i> (Optional)</p> <p>aws:TagKeys (Optional)</p>	<p>La siguiente política permite que la acción <code>Terminate Environment</code> termine el entorno myenv en la aplicación My App.</p> <pre>{ "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:Terminate Environment"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us- east-2:123456789012:environment/My App/ myenv"], "Condition": { "StringEquals": { "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2 :123456789012:application/My App"] } } }] }</pre>

Acción: [UpdateApplication](#)

Recurso	Condiciones	Instrucción de ejemplo
application	<p>aws:ResourceTag/ <i>key-name</i> (Optional)</p> <p>aws:TagKeys (Optional)</p>	<p>La siguiente política permite que la acción <code>UpdateApplication</code> actualice las propiedades de la aplicación My App.</p> <pre>{ "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:UpdateApplication"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"] }] }</pre>

Acción: [UpdateApplicationResourceLifecycle](#)

Recurso	Condiciones	Instrucción de ejemplo
application	aws:ResourceTag/ <i>key-name</i> (Optional) aws:TagKeys (Optional)	<p>La siguiente política permite que la acción UpdateApplicationResourceLifecycle actualice la configuración del ciclo de vida de la aplicación My App.</p> <pre>{ "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:UpdateApplicationResourceLifecycle"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"] }] }</pre>

Acción: [UpdateApplicationVersion](#)

Recurso	Condiciones	Instrucción de ejemplo
applicati onversion	InApplica tion aws:Resou rceTag/ <i>key- name</i> (Optional) aws:TagKeys (Optional)	<p>La siguiente política permite que la acción <code>UpdateApplicationVersion</code> actualice las propiedades de la versión de la aplicación My Version en la aplicación My App.</p> <pre data-bbox="732 443 1507 1436"> { "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:UpdateApp licationVersion"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us- east-2:123456789012:applicationversion/My App/My Version"], "Condition": { "StringEquals": { "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2 :123456789012:application/My App"] } } }] } </pre>

Acción: [UpdateConfigurationTemplate](#)

Recurso	Condiciones	Instrucción de ejemplo
configura tiontempl ate	InApplica tion aws:Resou rceTag/ <i>key- name</i> (Optional) aws:TagKeys (Optional)	<p>La siguiente política permite que la acción UpdateConfigurationTemplate actualice las propiedades u opciones de la plantilla de configuración My Template en la aplicación My App.</p> <pre data-bbox="732 443 1507 1436"> { "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:UpdateConfigurationTemplate"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:configurationtemplate/My App/My Template"], "Condition": { "StringEquals": { "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"] } } }] } </pre>

Acción: [UpdateEnvironment](#)

Recurso	Condiciones	Instrucción de ejemplo
environment	<p>InApplication</p> <p>FromApplicationVersion</p> <p>FromConfigurationTemplate</p> <p>aws:ResourceTag/ <i>key-name</i> (Optional)</p> <p>aws:TagKeys (Optional)</p>	<p>La siguiente política permite que la acción <code>UpdateEnvironment</code> actualice el entorno <code>myenv</code> en la aplicación <code>My App</code> implementando la versión de la aplicación <code>My Version</code>.</p> <pre> { "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:UpdateEnvironment"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"], "Condition": { "StringEquals": { "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App", "elasticbeanstalk:FromApplicationVersion": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/My App/My Version"]] } } }] } </pre>

Acción: [UpdateTagsForResource](#) – AddTags

Recurso	Condiciones	Instrucción de ejemplo
application applicationversion configurationtemplate environment platform	aws:ResourceTag/ <i>key-name</i> (Optional) aws:RequestTag/ <i>key-name</i> (Optional) aws:TagKeys (Optional)	<p>La acción AddTags es una de dos acciones virtuales asociada a la API UpdateTagsForResource .</p> <p>La siguiente política permite que la acción AddTags modifique etiquetas de entornos existentes solo si tienen una etiqueta denominada stage con el valor test:</p> <pre> { "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:AddTags"], "Effect": "Allow", "Resource": "*", "Condition": { "StringEquals": { "aws:ResourceTag/stage": ["test"] } } }] } </pre>

Acción: [UpdateTagsForResource](#) – RemoveTags

Recurso	Condiciones	Instrucción de ejemplo
application applicationversion configurationtemplate environment platform	aws:ResourceTag/ <i>key-name</i> (Optional) aws:TagKeys (Optional)	<p>La acción <code>RemoveTags</code> es una de dos acciones virtuales asociada a la API UpdateTagsForResource.</p> <p>La siguiente política deniega la acción <code>RemoveTags</code> para solicitar la eliminación de una etiqueta denominada a stage de recursos existentes:</p> <pre> { "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:RemoveTags"], "Effect": "Deny", "Resource": "*", "Condition": { "ForAnyValue:StringEquals": { "aws:TagKeys": ["stage"] } } }] } </pre>

Acción: [ValidateConfigurationSettings](#)

Recurso	Condiciones	Instrucción de ejemplo
template environment	InApplication aws:ResourceTag/ <i>key-name</i> (Optional) aws:TagKeys (Optional)	<p>La siguiente política permite que la acción <code>ValidateConfigurationSettings</code> valide los valores de configuración con el entorno myenv en la aplicación My App.</p> <pre> { "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:ValidateConfigurationSettings"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"], "Condition": { "StringEquals": { "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"] } } }] } </pre>

Claves de condición para acciones de Elastic Beanstalk

Las claves le permiten especificar condiciones que expresen dependencias, restrinjan los permisos o especifiquen restricciones en los parámetros de entrada de una acción. Elastic Beanstalk admite las siguientes claves.

InApplication

Especifica la aplicación que contiene el recurso en el que opera la acción.

El siguiente ejemplo permite que la acción `UpdateApplicationVersion` actualice las propiedades de la versión de la aplicación **My Version**. La condición `InApplication` especifica **My App** como el contenedor de **My Version**.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "elasticbeanstalk:UpdateApplicationVersion"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/My App/My Version"
      ],
      "Condition": {
        "StringEquals": {
          "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"]
        }
      }
    }
  ]
}
```

FromApplicationVersion

Especifica una versión de la aplicación como una dependencia o una restricción en un parámetro de entrada.

La siguiente ejemplo permite que la acción `UpdateEnvironment` actualice el entorno **myenv** en la aplicación **My App**. La condición `FromApplicationVersion` restringe el parámetro `VersionLabel` para permitir únicamente la versión de la aplicación **My Version** para actualizar el entorno.

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```

{
  "Action": [
    "elasticbeanstalk:UpdateEnvironment"
  ],
  "Effect": "Allow",
  "Resource": [
    "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"
  ],
  "Condition": {
    "StringEquals": {
      "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"],
      "elasticbeanstalk:FromApplicationVersion": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/My App/My Version"]
    }
  }
}

```

FromConfigurationTemplate

Especifica una plantilla de configuración como una dependencia o una restricción en un parámetro de entrada.

La siguiente ejemplo permite que la acción `UpdateEnvironment` actualice el entorno **myenv** en la aplicación **My App**. La condición `FromConfigurationTemplate` restringe el parámetro `TemplateName` para permitir únicamente la plantilla de configuración **My Template** para actualizar el entorno.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "elasticbeanstalk:UpdateEnvironment"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"
      ],
      "Condition": {
        "StringEquals": {

```

```

    "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-
east-2:123456789012:application/My App"],
    "elasticbeanstalk:FromConfigurationTemplate":
["arn:aws:elasticbeanstalk:us-east-2:123456789012:configurationtemplate/My App/My
Template"]
    }
  }
}
]
}

```

FromEnvironment

Especifica un entorno como una dependencia o una restricción en un parámetro de entrada.

El siguiente ejemplo permite que la acción `SwapEnvironmentCNAMEs` intercambie los registros CNAME en **My App** para todos los entornos cuyos nombres comienzan por **mysrcenv** y **mydestenv**, pero no para los entornos cuyos nombres comienzan por **mysrcenvPROD*** y **mydestenvPROD***.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "elasticbeanstalk:SwapEnvironmentCNAMEs"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/
mysrcenv*",
        "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/
mydestenv*"
      ],
      "Condition": {
        "StringNotLike": {
          "elasticbeanstalk:FromEnvironment": [
            "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/
mysrcenvPROD*",
            "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/
mydestenvPROD*"
          ]
        }
      }
    }
  ]
}

```

```

    }
  }
]
}
```

FromSolutionStack

Especifica una pila de soluciones como una dependencia o una restricción en un parámetro de entrada.

La siguiente política permite que la acción `CreateConfigurationTemplate` cree plantillas de configuración cuyos nombres comienzan por **My Template** (`My Template*`) en la aplicación **My App**. La condición `FromSolutionStack` restringe el parámetro `solutionstack` para permitir únicamente la pila de soluciones **32bit Amazon Linux running Tomcat 7** como el valor de entrada de ese parámetro.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "elasticbeanstalk:CreateConfigurationTemplate"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:elasticbeanstalk:us-east-2:123456789012:configurationtemplate/My
App/My Template*"
      ],
      "Condition": {
        "StringEquals": {
          "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-
east-2:123456789012:application/My App"],
          "elasticbeanstalk:FromSolutionStack": ["arn:aws:elasticbeanstalk:us-
east-2::solutionstack/32bit Amazon Linux running Tomcat 7"]
        }
      }
    }
  ]
}
```

`aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, `aws:TagKeys`

Especifique las condiciones basados en etiquetas. Para obtener información, consulte [Uso de etiquetas para controlar el acceso a los recursos de Elastic Beanstalk](#).

Uso de etiquetas para controlar el acceso a los recursos de Elastic Beanstalk

Las condiciones en las declaraciones de política de usuarios de AWS Identity and Access Management (IAM) forman parte de la sintaxis que se utiliza para especificar permisos a los recursos que necesitan las acciones de Elastic Beanstalk para completarse. Para obtener más información acerca de la especificación de las condiciones de declaración de política, consulte [Recursos y condiciones para las acciones de Elastic Beanstalk](#). El uso de etiquetas en las condiciones es una manera de controlar el acceso a los recursos y las solicitudes. Para obtener información sobre cómo etiquetar recursos de Elastic Beanstalk, consulte [Etiquetar recursos de la aplicación Elastic Beanstalk](#). En este tema, se explica el control de acceso basado en etiquetas.

Al diseñar políticas de IAM, es posible que se establezcan permisos detallados mediante la concesión de acceso a recursos específicos. A medida que crezca la cantidad de recursos que administra, esta tarea será más complicada. El etiquetado de recursos y uso de etiquetas en las condiciones de declaración de política pueden facilitar esta tarea. Puede conceder acceso de forma masiva a cualquier recurso con una determinada etiqueta. A continuación, aplique repetidamente esta etiqueta a los recursos pertinentes durante la creación o después.

Las etiquetas se pueden asociar al recurso o pasarse dentro de la solicitud a los servicios que admiten etiquetado. En Elastic Beanstalk, los recursos pueden tener etiquetas y algunas acciones pueden incluir etiquetas. Al crear una política de IAM, puede utilizar las claves de condición de etiqueta para controlar:

- Los usuarios que pueden realizar acciones en un entorno, en función de las etiquetas que ya tiene.
- Las etiquetas que se pueden pasar en la solicitud de una acción.
- Si se pueden utilizar claves de etiqueta específicas en una solicitud.

Para la sintaxis y semántica completas de las claves de condición de etiquetas, consulte [Control del acceso mediante etiquetas](#) en la Guía del usuario de IAM.

Los siguientes ejemplos muestran cómo especificar condiciones de etiquetas en las políticas para los usuarios de Elastic Beanstalk.

Example 1: Limitar acciones en función de etiquetas en la solicitud

La política de usuario administrada de Elastic Beanstalk `AdministratorAccess-AWSElasticBeanstalk` proporciona a los usuarios permisos ilimitados para realizar cualquier acción de Elastic Beanstalk en cualquier recurso administrado por Elastic Beanstalk.

La siguiente política limita esta posibilidad y deniega el permiso a usuarios no autorizados para crear entornos de producción de Elastic Beanstalk. Para ello, deniega la acción `CreateEnvironment` si la solicitud especifica una etiqueta denominada `stage` con uno de los valores `gamma` o `prod`. Además, la política impide que estos usuarios no autorizados manipulen la fase de los entornos de producción no permitiendo acciones de modificación de etiquetas para incluir estos mismos valores de etiqueta o para eliminar la etiqueta `stage` completamente. El administrador de un cliente debe asociar esta política de IAM a los usuarios de IAM no autorizados, además de la política de usuario administrada.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "elasticbeanstalk:CreateEnvironment",
        "elasticbeanstalk:AddTags"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/stage": ["gamma", "prod"]
        }
      }
    },
    {
      "Effect": "Deny",
      "Action": [
        "elasticbeanstalk:RemoveTags"
      ],
      "Resource": "*",
      "Condition": {
        "ForAnyValue:StringEquals": {
```

```
        "aws:TagKeys": ["stage"]
    }
}
]
```

Example 2: Limitar acciones en función de etiquetas de recursos

La política de usuario administrada de Elastic Beanstalk `AdministratorAccess-AWSElasticBeanstalk` proporciona a los usuarios permisos ilimitados para realizar cualquier acción de Elastic Beanstalk en cualquier recurso administrado por Elastic Beanstalk.

La siguiente política deniega permiso a usuarios no autorizados para realizar acciones en entornos de producción de Elastic Beanstalk. Para ello, deniega acciones específicas si el entorno tiene una etiqueta denominada `stage` con uno de los valores `gamma` o `prod`. El administrador de un cliente debe asociar esta política de IAM a los usuarios de IAM no autorizados, además de la política de usuario administrada.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "elasticbeanstalk:AddTags",
        "elasticbeanstalk:RemoveTags",
        "elasticbeanstalk:DescribeEnvironments",
        "elasticbeanstalk:TerminateEnvironment",
        "elasticbeanstalk:UpdateEnvironment",
        "elasticbeanstalk:ListTagsForResource"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/stage": ["gamma", "prod"]
        }
      }
    }
  ]
}
```

Example 3: Permitir acciones en función de etiquetas en la solicitud

La política siguiente concede a los usuarios permiso para crear aplicaciones de desarrollo de Elastic Beanstalk.

Para ello, permite las acciones `CreateApplication` y `AddTags` si la solicitud especifica una etiqueta denominada `stage` con el valor `development`. La condición `aws:TagKeys` garantiza que el usuario no pueda añadir otras claves de etiqueta. En particular, garantiza la distinción entre mayúsculas y minúsculas de la clave de etiqueta `stage`. Tenga en cuenta que esta política es útil para los usuarios de IAM que no tienen la política de usuario administrada de Elastic Beanstalk `AdministratorAccess-AWSElasticBeanstalk`. La política administrada proporciona a los usuarios permisos ilimitados para realizar cualquier acción de Elastic Beanstalk en cualquier recurso administrado por Elastic Beanstalk.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticbeanstalk:CreateApplication",
        "elasticbeanstalk:AddTags"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/stage": "development"
        },
        "ForAllValues:StringEquals": {
          "aws:TagKeys": ["stage"]
        }
      }
    }
  ]
}
```

Example 4: Permitir acciones en función de etiquetas de recursos

La siguiente política concede a los usuarios permiso para realizar acciones sobre aplicaciones de desarrollo de Elastic Beanstalk y obtener información sobre ellas.

Para ello, permite las acciones específicas si la aplicación especifica una etiqueta denominada `stage` con el valor `development`. La condición `aws:TagKeys` garantiza que el usuario no pueda añadir otras claves de etiqueta. En particular, garantiza la distinción entre mayúsculas y minúsculas de la clave de etiqueta `stage`. Tenga en cuenta que esta política es útil para los usuarios de IAM que no tienen la política de usuario administrada de Elastic Beanstalk `AdministratorAccess-AWSElasticBeanstalk`. La política administrada proporciona a los usuarios permisos ilimitados para realizar cualquier acción de Elastic Beanstalk en cualquier recurso administrado por Elastic Beanstalk.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticbeanstalk:UpdateApplication",
        "elasticbeanstalk>DeleteApplication",
        "elasticbeanstalk:DescribeApplications"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/stage": "development"
        },
        "ForAllValues:StringEquals": {
          "aws:TagKeys": ["stage"]
        }
      }
    }
  ]
}
```

Políticas de ejemplo basadas en políticas administradas

En esta sección se muestra cómo controlar el acceso de los usuarios a AWS Elastic Beanstalk e incluye políticas de ejemplo que proporcionan el acceso necesario para situaciones comunes. Estas políticas se derivan de las políticas administradas de Elastic Beanstalk. Para obtener información acerca de cómo asociar políticas administradas a usuarios y grupos, consulte [Administración de políticas de usuario de Elastic Beanstalk](#).

En este escenario, Ejemplo Corp. es una empresa de software con tres equipos responsables del sitio web de la empresa: los administradores que se encargan de la infraestructura, los desarrolladores que crean el software para el sitio web y un equipo de control de calidad que comprueba el sitio web. Para ayudarle a administrar los permisos a sus recursos de Elastic Beanstalk, Ejemplo Corp. crea tres grupos a los que pertenecen los miembros de cada equipo respectivo: Administradores, Desarrolladores y Evaluadores. Ejemplo Corp. quiere que el grupo Administradores tenga acceso completo a todas las aplicaciones, entornos y sus recursos subyacentes, para poder crear y eliminar todos los recursos de Elastic Beanstalk, además de solucionar los problemas con estos recursos. Los desarrolladores requieren permisos para ver todos los recursos de Elastic Beanstalk y para crear e implementar versiones de la aplicación. Los desarrolladores no deben poder crear nuevas aplicaciones o entornos ni terminar entornos en ejecución. Los evaluadores deben ver todos los recursos de Elastic Beanstalk para monitorear y probar aplicaciones. Los evaluadores no deberían poder realizar cambios en ningún recurso de Elastic Beanstalk.

Las siguientes políticas de ejemplo proporcionan los permisos necesarios para cada grupo.

Ejemplo 1: Grupo Administradores: todas las API de Elastic Beanstalk y servicios relacionados

La siguiente política concede a los usuarios permisos para todas las acciones necesarias para utilizar Elastic Beanstalk. Esta política también permite a Elastic Beanstalk aprovisionar y administrar recursos en su nombre en los siguientes servicios. Elastic Beanstalk utiliza estos servicios adicionales para aprovisionar los recursos subyacentes cuando se crea un entorno.

- Amazon Elastic Compute Cloud
- Elastic Load Balancing
- Auto Scaling
- Amazon CloudWatch
- Amazon Simple Storage Service
- Amazon Simple Notification Service
- Amazon Relational Database Service
- AWS CloudFormation

Tenga en cuenta que esta política es un ejemplo. Concede un amplio conjunto de permisos a los servicios de AWS que Elastic Beanstalk utiliza para administrar aplicaciones y entornos. Por ejemplo,

`ec2:*` permite a un usuario de AWS Identity and Access Management (IAM) realizar cualquier acción en cualquier recurso de Amazon EC2 en la cuenta de AWS. Estos permisos no se limitan a los recursos que utiliza con Elastic Beanstalk. Como práctica recomendada, debe conceder a las personas únicamente los permisos necesarios para llevar a cabo sus tareas.

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "elasticbeanstalk:*",
        "ec2:*",
        "elasticloadbalancing:*",
        "autoscaling:*",
        "cloudwatch:*",
        "s3:*",
        "sns:*",
        "rds:*",
        "cloudformation:*"
      ],
      "Resource" : "*"
    }
  ]
}
```

Ejemplo 2: Grupo de desarrolladores: todas excepto las operaciones con privilegios elevados

La siguiente política deniega permisos para crear aplicaciones y entornos, y permite todas las demás acciones de Elastic Beanstalk.

Tenga en cuenta que esta política es un ejemplo. Concede un amplio conjunto de permisos a los productos de AWS que Elastic Beanstalk utiliza para administrar aplicaciones y entornos. Por ejemplo, `ec2:*` permite a un usuario de IAM realizar cualquier acción en cualquier recurso de Amazon EC2 en la cuenta de AWS. Estos permisos no se limitan a los recursos que utiliza con Elastic Beanstalk. Como práctica recomendada, debe conceder a las personas únicamente los permisos necesarios para llevar a cabo sus tareas.

```
{
  "Version" : "2012-10-17",
```

```

"Statement" : [
  {
    "Action" : [
      "elasticbeanstalk:CreateApplication",
      "elasticbeanstalk:CreateEnvironment",
      "elasticbeanstalk>DeleteApplication",
      "elasticbeanstalk:RebuildEnvironment",
      "elasticbeanstalk:SwapEnvironmentCNAMEs",
      "elasticbeanstalk:TerminateEnvironment"],
    "Effect" : "Deny",
    "Resource" : "*"
  },
  {
    "Action" : [
      "elasticbeanstalk:*",
      "ec2:*",
      "elasticloadbalancing:*",
      "autoscaling:*",
      "cloudwatch:*",
      "s3:*",
      "sns:*",
      "rds:*",
      "cloudformation:*"],
    "Effect" : "Allow",
    "Resource" : "*"
  }
]
}

```

Ejemplo 3: Evaluadores: solo consulta

La siguiente política permite acceso de solo lectura a todas las aplicaciones, versiones de la aplicación, eventos y entornos. No permite realizar ninguna acción.

```

{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "elasticbeanstalk:Check*",
        "elasticbeanstalk:Describe*",
        "elasticbeanstalk:List*",

```

```

    "elasticbeanstalk:RequestEnvironmentInfo",
    "elasticbeanstalk:RetrieveEnvironmentInfo",
    "ec2:Describe*",
    "elasticloadbalancing:Describe*",
    "autoscaling:Describe*",
    "cloudwatch:Describe*",
    "cloudwatch:List*",
    "cloudwatch:Get*",
    "s3:Get*",
    "s3:List*",
    "sns:Get*",
    "sns:List*",
    "rds:Describe*",
    "cloudformation:Describe*",
    "cloudformation:Get*",
    "cloudformation:List*",
    "cloudformation:Validate*",
    "cloudformation:Estimate*"
  ],
  "Resource" : "*"
}
]
}

```

Políticas de ejemplo basadas en permisos de recursos

En esta sección se describe un caso de uso para controlar los permisos de usuario para acciones de Elastic Beanstalk que acceden a recursos específicos de Elastic Beanstalk. Examinaremos las políticas de ejemplo que permiten este caso de uso. Para obtener más información sobre las directivas sobre los recursos de Elastic Beanstalk, consulte [Creación de una política de usuario personalizada](#). Para obtener información sobre cómo asociar políticas a usuarios y grupos, consulte [Administración de políticas de IAM](#) en Uso de AWS Identity and Access Management.

En nuestro caso de uso, Example Corp. es una pequeña empresa de consultoría que desarrolla aplicaciones para dos clientes diferentes. John es el director de desarrollo que supervisa el desarrollo de las dos aplicaciones de Elastic Beanstalk: app1 y app2. John realiza tareas de desarrollo y pruebas en las dos aplicaciones y solo él puede actualizar el entorno de producción de las dos aplicaciones. Estos son los permisos que necesita para app1 y app2:

- Ver aplicaciones, versiones de aplicaciones, entornos y plantillas de configuración
- Crear versiones de la aplicación e implementarlas en el entorno de ensayo

- Actualizar el entorno de producción
- Crear y terminar entornos

Jill es un evaluador que necesita tener acceso para ver los siguientes recursos con el fin de poder monitorear y probar las dos aplicaciones: aplicaciones, versiones de la aplicación, entornos y plantillas de configuración. Sin embargo, no debería poder hacer cambios en ningún recurso de Elastic Beanstalk.

Jack es el desarrollador de app1 que necesita acceso para ver todos los recursos de app1, y también debe crear versiones de la aplicación para app1 e implementarlas en el entorno de ensayo.

Judy es la administradora de la cuenta de AWS de Example Corp. Ha creado usuarios de IAM para John, Jill y Jack, y ha asociado las siguientes políticas a esos usuarios para conceder los permisos correspondientes a las aplicaciones app1 y app2.

Ejemplo 1: John: jefe de desarrollo de app1, app2

Hemos dividido la política de John en tres políticas distintas para que sean más fáciles de leer y administrar. Juntos, conceden a John los permisos que necesita para realizar acciones de desarrollo, pruebas e implementación en las dos aplicaciones.

La primera política especifica acciones para Auto Scaling, Amazon S3, Amazon EC2, CloudWatch, Amazon SNS, Elastic Load Balancing, Amazon RDS y AWS CloudFormation. Elastic Beanstalk utiliza estos servicios adicionales para aprovisionar los recursos subyacentes cuando se crea un entorno.

Tenga en cuenta que esta política es un ejemplo. Concede un amplio conjunto de permisos a los productos de AWS que Elastic Beanstalk utiliza para administrar aplicaciones y entornos. Por ejemplo, `ec2:*` permite a un usuario de IAM realizar cualquier acción en cualquier recurso de Amazon EC2 en la cuenta de AWS. Estos permisos no se limitan a los recursos que utiliza con Elastic Beanstalk. Como práctica recomendada, debe conceder a las personas únicamente los permisos necesarios para llevar a cabo sus tareas.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:*",
        "ecs:*",
```

```

        "ecr:*",
        "elasticloadbalancing:*",
        "autoscaling:*",
        "cloudwatch:*",
        "s3:*",
        "sns:*",
        "cloudformation:*",
        "dynamodb:*",
        "rds:*",
        "sqs:*",
        "logs:*",
        "iam:GetPolicyVersion",
        "iam:GetRole",
        "iam:PassRole",
        "iam:ListRolePolicies",
        "iam:ListAttachedRolePolicies",
        "iam:ListInstanceProfiles",
        "iam:ListRoles",
        "iam:ListServerCertificates",
        "acm:DescribeCertificate",
        "acm:ListCertificates",
        "codebuild:CreateProject",
        "codebuild>DeleteProject",
        "codebuild:BatchGetBuilds",
        "codebuild:StartBuild"
    ],
    "Resource": "*"
}
]
}

```

La segunda política especifica las acciones de Elastic Beanstalk que John puede realizar en los recursos de app1 y app2. La instrucción `AllCallsInApplications` permite que se realicen todas las acciones de Elastic Beanstalk (`"elasticbeanstalk:*"`) en todos los recursos de app1 y app2 (por ejemplo, `elasticbeanstalk:CreateEnvironment`). La instrucción `AllCallsOnApplications` permite que se realicen todas las acciones de Elastic Beanstalk (`"elasticbeanstalk:*"`) en todos los recursos de las aplicaciones app1 y app2 (por ejemplo, `elasticbeanstalk:DescribeApplications`, `elasticbeanstalk:UpdateApplication`, etc.). La instrucción `AllCallsOnSolutionStacks` permite todas las acciones de Elastic Beanstalk (`"elasticbeanstalk:*"`) para los recursos de la pila de soluciones (por ejemplo, `elasticbeanstalk:ListAvailableSolutionStacks`).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllCallsInApplications",
      "Action": [
        "elasticbeanstalk:*"
      ],
      "Effect": "Allow",
      "Resource": [
        "*"
      ],
      "Condition": {
        "StringEquals": {
          "elasticbeanstalk:InApplication": [
            "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/app1",
            "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/app2"
          ]
        }
      }
    },
    {
      "Sid": "AllCallsOnApplications",
      "Action": [
        "elasticbeanstalk:*"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/app1",
        "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/app2"
      ]
    },
    {
      "Sid": "AllCallsOnSolutionStacks",
      "Action": [
        "elasticbeanstalk:*"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:elasticbeanstalk:us-east-2::solutionstack/*"
      ]
    }
  ]
}
```

```
}
```

La tercera directiva especifica las acciones de Elastic Beanstalk en las que la segunda política necesita permisos para completar esas acciones de Elastic Beanstalk. La instrucción `AllNonResourceCalls` permite la acción `elasticbeanstalk:CheckDNSAvailability`, que es necesaria para llamar a `elasticbeanstalk:CreateEnvironment` y para otras acciones. También permite la acción `elasticbeanstalk:CreateStorageLocation`, que es necesaria para `elasticbeanstalk:CreateApplication`, `elasticbeanstalk:CreateEnvironment` y para otras acciones.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllNonResourceCalls",
      "Action": [
        "elasticbeanstalk:CheckDNSAvailability",
        "elasticbeanstalk:CreateStorageLocation"
      ],
      "Effect": "Allow",
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Ejemplo 2: Jill: evaluadora de app1, app2

Hemos dividido la política de Jill en tres políticas distintas para que sean más fáciles de leer y administrar. Juntos, conceden a Jill los permisos que necesita para las acciones de pruebas y monitorización en las dos aplicaciones.

La primera política especifica las acciones `Describe*`, `List*` y `Get*` en Auto Scaling, Amazon S3, Amazon EC2, CloudWatch, Amazon SNS, Elastic Load Balancing, Amazon RDS y AWS CloudFormation (para tipos de contenedores no heredados), lo que permite que las acciones de Elastic Beanstalk puedan recuperar la información pertinente sobre los recursos subyacentes de las aplicaciones `app1` y `app2`.

```
{
  "Version": "2012-10-17",
```

```

"Statement":[
  {
    "Effect":"Allow",
    "Action":[
      "ec2:Describe*",
      "elasticloadbalancing:Describe*",
      "autoscaling:Describe*",
      "cloudwatch:Describe*",
      "cloudwatch:List*",
      "cloudwatch:Get*",
      "s3:Get*",
      "s3:List*",
      "sns:Get*",
      "sns:List*",
      "rds:Describe*",
      "cloudformation:Describe*",
      "cloudformation:Get*",
      "cloudformation:List*",
      "cloudformation:Validate*",
      "cloudformation:Estimate*"
    ],
    "Resource":"*"
  }
]
}

```

La segunda política especifica las acciones de Elastic Beanstalk que Jill puede realizar en los recursos de app1 y app2. La instrucción `AllReadCallsInApplications` le permite llamar a las acciones `Describe*` y a las acciones de información del entorno. La instrucción `AllReadCallsOnApplications` le permite llamar a las acciones `DescribeApplications` y `DescribeEvents` en los recursos de las aplicaciones app1 y app2. La instrucción `AllReadCallsOnSolutionStacks` le permite ver las acciones en las que participan los recursos de la pila de soluciones (`ListAvailableSolutionStacks`, `DescribeConfigurationOptions` y `ValidateConfigurationSettings`).

```

{
  "Version": "2012-10-17",
  "Statement":[
    {
      "Sid":"AllReadCallsInApplications",
      "Action":[
        "elasticbeanstalk:Describe*",

```

```

        "elasticbeanstalk:RequestEnvironmentInfo",
        "elasticbeanstalk:RetrieveEnvironmentInfo"
    ],
    "Effect": "Allow",
    "Resource": [
        "*"
    ],
    "Condition": {
        "StringEquals": {
            "elasticbeanstalk:InApplication": [
                "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/app1",
                "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/app2"
            ]
        }
    }
},
{
    "Sid": "AllReadCallsOnApplications",
    "Action": [
        "elasticbeanstalk:DescribeApplications",
        "elasticbeanstalk:DescribeEvents"
    ],
    "Effect": "Allow",
    "Resource": [
        "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/app1",
        "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/app2"
    ]
},
{
    "Sid": "AllReadCallsOnSolutionStacks",
    "Action": [
        "elasticbeanstalk:ListAvailableSolutionStacks",
        "elasticbeanstalk:DescribeConfigurationOptions",
        "elasticbeanstalk:ValidateConfigurationSettings"
    ],
    "Effect": "Allow",
    "Resource": [
        "arn:aws:elasticbeanstalk:us-east-2::solutionstack/*"
    ]
}
]
}

```

La tercera directiva especifica las acciones de Elastic Beanstalk en las que la segunda política necesita permisos para completar esas acciones de Elastic Beanstalk. La instrucción `AllNonResourceCalls` permite la acción `elasticbeanstalk:CheckDNSAvailability`, que es necesaria para algunas acciones de consulta.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllNonResourceCalls",
      "Action": [
        "elasticbeanstalk:CheckDNSAvailability"
      ],
      "Effect": "Allow",
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Ejemplo 3: Jack: desarrollador de app1

Hemos dividido la política de Jack en tres políticas distintas para que sean más fáciles de leer y administrar. Juntos, conceden a Jack los permisos que necesita para realizar acciones de pruebas, monitorización e implementación en el recurso app1.

La primera política especifica las acciones de Auto Scaling, Amazon S3, Amazon EC2, CloudWatch, Amazon SNS, Elastic Load Balancing, Amazon RDS y AWS CloudFormation (para tipos de contenedores no heredados), lo que permite que las acciones de Elastic Beanstalk puedan recuperar la información pertinente sobre los recursos subyacentes de las aplicaciones app1. Para ver una lista de los tipos de contenedores no heredados admitidos, consulte [the section called “¿Por qué algunas versiones de la plataforma están marcadas como heredadas?”](#)

Tenga en cuenta que esta política es un ejemplo. Concede un amplio conjunto de permisos a los productos de AWS que Elastic Beanstalk utiliza para administrar aplicaciones y entornos. Por ejemplo, `ec2:*` permite a un usuario de IAM realizar cualquier acción en cualquier recurso de Amazon EC2 en la cuenta de AWS. Estos permisos no se limitan a los recursos que utiliza con Elastic Beanstalk. Como práctica recomendada, debe conceder a las personas únicamente los permisos necesarios para llevar a cabo sus tareas.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:*",
        "elasticloadbalancing:*",
        "autoscaling:*",
        "cloudwatch:*",
        "s3:*",
        "sns:*",
        "rds:*",
        "cloudformation:*"
      ],
      "Resource": "*"
    }
  ]
}
```

La segunda política especifica las acciones de Elastic Beanstalk que Jack puede realizar en el recurso de app1.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllReadCallsAndAllVersionCallsInApplications",
      "Action": [
        "elasticbeanstalk:Describe*",
        "elasticbeanstalk:RequestEnvironmentInfo",
        "elasticbeanstalk:RetrieveEnvironmentInfo",
        "elasticbeanstalk:CreateApplicationVersion",
        "elasticbeanstalk>DeleteApplicationVersion",
        "elasticbeanstalk:UpdateApplicationVersion"
      ],
      "Effect": "Allow",
      "Resource": [
        "*"
      ],
      "Condition": {
        "StringEquals": {
          "elasticbeanstalk:InApplication": [

```



```

        "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/app1"
    ]
  }
},
{
  "Sid": "AllReadCallsOnApplications",
  "Action": [
    "elasticbeanstalk:DescribeApplications",
    "elasticbeanstalk:DescribeEvents"
  ],
  "Effect": "Allow",
  "Resource": [
    "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/app1"
  ]
},
{
  "Sid": "UpdateEnvironmentInApplications",
  "Action": [
    "elasticbeanstalk:UpdateEnvironment"
  ],
  "Effect": "Allow",
  "Resource": [
    "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/app1/app1-
staging*"
  ],
  "Condition": {
    "StringEquals": {
      "elasticbeanstalk:InApplication": [
        "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/app1"
      ]
    },
    "StringLike": {
      "elasticbeanstalk:FromApplicationVersion": [
        "arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/
app1/*"
      ]
    }
  }
},
{
  "Sid": "AllReadCallsOnSolutionStacks",
  "Action": [
    "elasticbeanstalk:ListAvailableSolutionStacks",

```

```

        "elasticbeanstalk:DescribeConfigurationOptions",
        "elasticbeanstalk:ValidateConfigurationSettings"
    ],
    "Effect": "Allow",
    "Resource": [
        "arn:aws:elasticbeanstalk:us-east-2::solutionstack/*"
    ]
}
]
}

```

La tercera directiva especifica las acciones de Elastic Beanstalk en las que la segunda política necesita permisos para completar esas acciones de Elastic Beanstalk. La instrucción `AllNonResourceCalls` permite la acción `elasticbeanstalk:CheckDNSAvailability`, que es necesaria para llamar a `elasticbeanstalk:CreateEnvironment` y para otras acciones. También permite la acción `elasticbeanstalk:CreateStorageLocation`, que es necesaria para `elasticbeanstalk:CreateEnvironment` y para otras acciones.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllNonResourceCalls",
      "Action": [
        "elasticbeanstalk:CheckDNSAvailability",
        "elasticbeanstalk:CreateStorageLocation"
      ],
      "Effect": "Allow",
      "Resource": [
        "*"
      ]
    }
  ]
}

```

Impedir el acceso al bucket de Amazon S3 entre entornos

Elastic Beanstalk proporciona políticas administradas para AWS gestionar los recursos que requieren los entornos de Elastic Beanstalk de su cuenta. AWS Los permisos proporcionados de forma predeterminada a una aplicación de su AWS cuenta tienen acceso a los recursos de S3 que pertenecen a otras aplicaciones de la misma cuenta. AWS

Si su AWS cuenta ejecuta varias aplicaciones de Beanstalk, puede reducir la seguridad de sus políticas creando su propia [política personalizada](#) para adjuntarla a su propio [rol de servicio](#) o [perfil de instancia](#) para cada entorno. A continuación, puede limitar los permisos de S3 de su política personalizada a un entorno específico.

Note

Ten en cuenta que eres responsable de mantener tu política personalizada. Si cambia una política gestionada por Elastic Beanstalk en la que se basa la política personalizada, tendrá que modificar la política personalizada con los cambios correspondientes en la política base. Para obtener un historial de cambios de las políticas administradas de Elastic Beanstalk, consulte. [Actualizaciones de Elastic AWS Beanstalk a las políticas administradas](#)

Ejemplo de permisos restringidos

El siguiente ejemplo se basa en la política [AWSElasticBeanstalkWebTier](#) administrada.

La política predeterminada incluye las siguientes líneas para los permisos de los buckets de S3. Esta política predeterminada no limita las acciones del bucket de S3 a entornos o aplicaciones específicos.

```
{
  "Sid" : "BucketAccess",
  "Action" : [
    "s3:Get*",
    "s3:List*",
    "s3:PutObject"
  ],
  "Effect" : "Allow",
  "Resource" : [
    "arn:aws:s3:::elasticbeanstalk-*",
    "arn:aws:s3:::elasticbeanstalk-*/*"
  ]
}
```

Puede limitar el acceso calificando recursos específicos para un rol de servicio especificado como `Principal`. En el siguiente ejemplo, se proporcionan los `aws-elasticbeanstalk-ec2-role-my-example-env` permisos del rol de servicio personalizado para los buckets de S3 del entorno con `idmy-example-env-ID`.

Example Otorgue permisos solo a los buckets de S3 de un entorno específico

```
{
  "Sid": "BucketAccess",
  "Action": [
    "s3:Get*",
    "s3:List*",
    "s3:PutObject"
  ],
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::...:role/aws-elasticbeanstalk-ec2-role-my-example-env"
  },
  "Resource": [
    "arn:aws:s3:::elasticbeanstalk-my-region-account-id-12345",
    "arn:aws:s3:::elasticbeanstalk-my-region-account-id-12345/resources/environments/my-example-env-ID/*"
  ]
}
```

Note

El ARN del recurso debe incluir el ID del entorno de Elastic Beanstalk (no el nombre del entorno). [Puede obtener el identificador del entorno en la consola de Elastic Beanstalk en la página de información general del entorno.](#) También puede usar el comando AWS CLI [describe-environments](#) para obtener esta información.

Para obtener más información que le ayude a actualizar los permisos de los buckets de S3 para sus entornos de Elastic Beanstalk, consulte los siguientes recursos:

- [Usar Elastic Beanstalk con Amazon S3](#) en esta guía
- [Tipos de recursos definidos por Amazon S3](#) en la guía de referencia de autorización de servicios
- [Formato ARN en la Guía](#) del usuario de IAM

Uso de Elastic Beanstalk con Amazon RDS

Puede utilizar Elastic Beanstalk con Amazon Relational Database Service (Amazon RDS) para configurar, operar y escalar una base de datos relacional. Existen las siguientes dos opciones para comenzar.

- Crear una nueva base de datos en Amazon RDS.
- Comenzar con una base de datos que ya ha sido [creada por Elastic Beanstalk](#) y que posteriormente ha sido [desacoplada](#) de un entorno de Beanstalk. Para obtener más información, consulte [the section called “Base de datos”](#).

Puede utilizar cualquiera de las dos opciones para ejecutar una instancia de base de datos en Amazon RDS y configurar que la aplicación se conecte a ella durante el lanzamiento. Puede conectar varios entornos a una base de datos y también realizar actualizaciones sin problemas con implementaciones azul-verde.

Note

Si no ha usado una instancia de base de datos con su aplicación anteriormente, recomendamos agregar primero una a un entorno de prueba con la consola de Elastic Beanstalk. De este modo, puede verificar que la aplicación puede leer las propiedades del entorno, crear una cadena de conexión y conectarse a una instancia de base de datos, sin el trabajo de configuración adicional necesario para una base de datos independiente. Para obtener más información, consulte [Añadir una base de datos al entorno de Elastic Beanstalk](#).

Para permitir que las instancias de Amazon EC2 del entorno se conecten a una base de datos externa, configure el grupo de grupo de seguridad para el grupo de Auto Scaling que está asociado a su entorno. Puede adjuntar el mismo grupo de seguridad que está asociado a su instancia de base de datos. O bien, puede utilizar un grupo de seguridad separado. Si adjunta otro grupo de seguridad, debe configurar el grupo de seguridad que está adjunto a la base de datos para que permita el acceso entrante desde este grupo de seguridad.

Note

Puede conectar el entorno a una base de datos agregando una regla al grupo de seguridad asociado a la base de datos. Esta regla debe permitir el acceso entrante desde el grupo

de seguridad generado automáticamente que Elastic Beanstalk adjunta al grupo de Auto Scaling de su entorno. Sin embargo, tenga en cuenta que, al crear esta regla, también crea una dependencia entre los dos grupos de seguridad. Posteriormente, cuando intente terminar el entorno, Elastic Beanstalk no podrá eliminar el grupo de seguridad del entorno ya que el grupo de seguridad de la base de datos depende de él.

Después de que lance la instancia de base de datos y configure los grupos de seguridad, puede pasar la información de conexión como el punto de enlace y la contraseña a su aplicación utilizando las propiedades del entorno. Es el mismo mecanismo que usa Elastic Beanstalk en segundo plano cuando ejecuta una instancia de base de datos en su entorno.

Para agregar una capa adicional de seguridad, puede almacenar la información de la conexión en Amazon S3 y configurar Elastic Beanstalk para que la recupere durante la implementación. Con [archivos de configuración \(.ebextensions\)](#), puede configurar las instancias de su entorno para recuperar de forma segura archivos de Amazon S3 cuando implemente la aplicación.

Temas

- [Lanzamiento y conexión a una instancia de Amazon RDS externa en una VPC predeterminada](#)
- [Lanzamiento y conexión a una instancia de Amazon RDS externa en EC2 Classic](#)
- [Almacenamiento de las credenciales de Amazon RDS en AWS Secrets Manager](#)
- [Limpieza de una instancia externa de Amazon RDS](#)

Lanzamiento y conexión a una instancia de Amazon RDS externa en una VPC predeterminada

Para utilizar una base de datos externa con una aplicación que se ejecuta en Elastic Beanstalk tiene dos opciones. Puede lanzar una instancia de base de datos con Amazon RDS. Cualquier instancia que lance con Amazon RDS es completamente independiente de Elastic Beanstalk y de sus entornos de Elastic Beanstalk. Esto significa que puede utilizar cualquier motor de base de datos y tipo de instancia admitidos por Amazon RDS, incluso los que no son utilizados por Elastic Beanstalk.

O, como alternativa al lanzamiento de una nueva instancia de base de datos, puede comenzar con una base de datos ya [creada por Elastic Beanstalk](#) y posteriormente [desacoplada](#) de un entorno de Beanstalk. Para obtener más información, consulte [the section called “Base de datos”](#). Con esta

opción, no necesita completar el procedimiento para lanzar una nueva base de datos. Sin embargo, sí necesita completar los procedimientos posteriores que se describen en este tema.

Los siguientes procedimientos describen el proceso para una [VPC predeterminada](#). El proceso es el mismo si utiliza una VPC personalizada. Los únicos requisitos adicionales son que su entorno y la instancia de la base de datos se encuentren en la misma subred o en subredes que puedan comunicarse entre sí. Para obtener más información sobre cómo configurar una VPC personalizada para su uso con Elastic Beanstalk, consulte [Uso de Elastic Beanstalk con Amazon VPC](#).

Note

- Si comienza con una base de datos creada por Elastic Beanstalk y posteriormente desacoplada de un entorno de Beanstalk, puede omitir el primer grupo de pasos y continuar con los pasos agrupados en Modificar las reglas de entrada del grupo de seguridad de la instancia de RDS.
- Si planea utilizar la base de datos que desacopla para un entorno de producción, verifique que el tipo de almacenamiento que utiliza la base de datos es el adecuado para su carga de trabajo. Para obtener más información, consulte [Almacenamiento de instancias de base de datos](#) y [Modificación de una instancia de base de datos](#) en la Guía del usuario de Amazon RDS.

Para lanzar una instancia de base de datos de RDS en una VPC predeterminada

1. Abra la [consola de RDS](#).
2. En el panel de navegación, seleccione Databases (Bases de datos).
3. Elija Create database (Crear base de datos).
4. Elija Standard Create (Creación estándar).

Important

No elija la opción Easy Create (Creación sencilla). Si lo elige, no puede configurar los ajustes necesarios para lanzar esta base de datos de RDS.

5. En Additional configuration (Configuración adicional), escriba **ebdb** en Initial database name (Nombre de base de datos inicial).

6. Revise la configuración predeterminada y ajústela de acuerdo con sus requisitos específicos. Preste atención a las siguientes opciones:
 - DB instance class (Clase de instancia de base de datos): elija un tamaño de instancia que cuente con la memoria y la potencia de CPU apropiadas para su carga de trabajo.
 - Multi-AZ deployment (Implementación Multi-AZ): para obtener alta disponibilidad, establezca esta opción en Create an Aurora Replica/Reader node in a different AZ (Crear una réplica de Aurora/nodo de lector en otra zona de disponibilidad).
 - Master username (Nombre de usuario maestro) y Master password (Contraseña maestra): nombre de usuario y contraseña de la base de datos. Anote esta configuración, ya que la necesitará posteriormente.
7. Compruebe los valores predeterminados de las demás opciones y, a continuación, elija Create database (Crear base de datos).

A continuación, modifique el grupo de seguridad adjunto a su instancia de base de datos para permitir el tráfico entrante en el puerto correspondiente. Este es el mismo grupo de seguridad que adjuntará a su entorno de Elastic Beanstalk más adelante. Como resultado, la regla que agregue concederá permiso de acceso entrante a otros recursos del mismo grupo de seguridad.

Modificar las reglas de entrada del grupo de seguridad adjuntas a la instancia de RDS

1. Abra la [consola de Amazon RDS](#).
2. Seleccione Databases (Bases de datos).
3. Elija el nombre de la instancia de base de datos para ver sus detalles.
4. En la sección Connectivity (Conectividad), tome nota de los valores de Subnets (Subredes), Security groups (Grupos de seguridad) y Endpoint (Punto de conexión) que aparecen en esta página. Esto es para que pueda utilizar esta información más adelante.
5. En Security (Seguridad) se muestra el grupo de seguridad que está asociado a la instancia de base de datos. Abra el enlace para ver el grupo de seguridad en la consola de Amazon EC2.

The screenshot shows the 'Connectivity' tab in the AWS Management Console. It is divided into three columns: 'Endpoint & port', 'Networking', and 'Security'. The 'Security' column is highlighted with a yellow box, showing 'VPC security groups' with 'rds-launch-wizard-4 (sg-07ecc7ed5b2c0c099)' listed as '(active)'. Other details include 'Public accessibility: Yes', 'Certificate authority: rds-ca-2015', and 'Certificate authority date: Mar 5th, 2020'.

6. En los detalles del grupo de seguridad, elija la pestaña Inbound (Entrada).
7. Elija Edit (Editar).
8. Seleccione Add Rule (Agregar regla).
9. En Type (Tipo), elija el motor de base de datos que utiliza la aplicación.
10. En Source (Origen), escriba **sg-** para ver una lista de grupos de seguridad disponibles. Elija el grupo de seguridad que está asociado al grupo de Auto Scaling que se utiliza con su entorno de Elastic Beanstalk. Esto es para que las instancias de Amazon EC2 en el entorno puedan acceder a la base de datos.


The screenshot shows the 'Edit inbound rules' dialog box. It has a table with columns: 'Type', 'Protocol', 'Port Range', 'Source', and 'Description'. Two rules are listed, both for 'MySQL/Aurora' on 'TCP' port '3306'. The 'Source' field of the second rule is highlighted with a yellow box, showing a dropdown menu with 'sg-' selected and a list of security groups, including 'sg-07ecc7ed5b2c0c099 - rds-launch-wizard-4'. Below the table is an 'Add Rule' button and a note: 'NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.' At the bottom right are 'Cancel' and 'Save' buttons.

11. Seleccione Save (Guardar).

A continuación, agregue el grupo de seguridad de la instancia de base de datos al entorno en ejecución. En este procedimiento Elastic Beanstalk vuelva a aprovisionar todas las instancias de su entorno con el grupo de seguridad adicional adjunto.

Para agregar un grupo de seguridad al entorno

- Haga una de las siguientes acciones:
 - Para añadir un grupo de seguridad mediante la consola de Elastic Beanstalk
 - a. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
 - b. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

 Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

- c. En el panel de navegación, elija Configuration (Configuración).
 - d. En la categoría de configuración Instances (Instancias), elija Edit (Editar).
 - e. En EC2 security groups (Grupos de seguridad de EC2), elija el grupo de seguridad que desea asignar a las instancias, además del grupo de seguridad de instancia que crea Elastic Beanstalk.
 - f. Para guardar los cambios, elija Aplicar en la parte inferior de la página.
 - g. Lea la advertencia y, a continuación, elija Confirm (Confirmar).
- Para agregar un grupo de seguridad con un [archivo de configuración](#), utilice el archivo de ejemplo [securitygroup-addexisting.config](#).

A continuación, pase la información de la conexión a su entorno mediante propiedades de entorno. Cuando [agrega una instancia de base de datos a su entorno](#) con la consola de Elastic Beanstalk, Elastic Beanstalk utiliza propiedades del entorno tales como RDS_HOSTNAME para pasar la información de conexión a su aplicación. Puede utilizar las mismas propiedades. De este modo, utiliza el mismo código de aplicación tanto con las instancias de base de datos integradas como con las instancias de base de datos externas. O, como alternativa, puede elegir sus propios nombres de propiedad.

Para configurar las propiedades de entorno de una instancia de base de datos de Amazon RDS

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Actualizaciones, supervisión y registro, seleccione Editar.
5. En la sección Environment properties (Propiedades del entorno), defina las variables que lee la aplicación para crear una cadena de conexión. Para conseguir compatibilidad con los entornos que tienen una instancia de base de datos de RDS, utilice los siguientes nombres y valores. Encontrará todos los valores, excepto la contraseña, en la [consola de RDS](#).

Nombre de la propiedad	Descripción	Valor de la propiedad
RDS_HOSTNAME	El nombre de host de la instancia de base de datos.	En la pestaña Connectivity & security (Conectividad y seguridad) de la consola de Amazon RDS: Endpoint (Punto de enlace).
RDS_PORT	El puerto en donde la instancia de base de datos acepta las conexiones. El valor predeterminado varía dependiendo del motor de base de datos.	En la pestaña Connectivity & security (Conectividad y seguridad) de la consola de Amazon RDS: Port (Puerto).
RDS_DB_NAME	El nombre de la base de dato, ebdb .	En la pestaña Configuration (Configuración) de la consola de Amazon RDS:

Nombre de la propiedad	Descripción	Valor de la propiedad
		DB Name (Nombre de la base de datos).
RDS_USERNAME	El nombre de usuario que ha configurado para la base de datos.	En la pestaña Configuración (Configuración) de la consola de Amazon RDS: Master username (Nombre de usuario maestro).
RDS_PASSWORD	La contraseña que ha configurado para la base de datos.	No está disponible como referencia en la consola de Amazon RDS.

Environment Properties

The following properties are passed into the application as environment variables. [Learn more.](#)

Property Name	Property Value
RDS_DB_NAME	<input type="text" value="ebdb"/> ✕
RDS_HOSTNAME	<input type="text" value="webapp-db.jxzb5mpaniu.us-wes"/> ✕
RDS_PORT	<input type="text" value="5432"/> ✕
RDS_USERNAME	<input type="text" value="webapp-admin"/> ✕
<input type="text" value="RDS_PASSWORD"/>	<input type="text" value="kUj5uKxmWDMYc403"/> +

6. Para guardar los cambios, elija Aplicar en la parte inferior de la página.


Si aún no programó la aplicación para leer las propiedades del entorno y crear una cadena de conexión, consulte los siguientes temas específicos del lenguaje para obtener instrucciones:

- Java SE: [Conexión a una base de datos \(plataformas Java SE\)](#)
- Java con Tomcat: [Conexión a una base de datos \(plataformas Tomcat\)](#)
- Node.js: [Conexión a una base de datos](#)
- .NET: [Conexión a una base de datos](#)
- PHP: [Conexión a una base de datos con PDO o MySQLi](#)
- Python: [Conexión a una base de datos](#)
- Ruby: [Conexión a una base de datos](#)

Por último, en función de cuándo lea la aplicación las variables de entorno, es posible que necesite reiniciar el servidor de la aplicación en las instancias del entorno.

Para reiniciar los servidores de aplicaciones de su entorno


1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

 Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. Elija Actions (Acciones) y, a continuación, elija Restart app server(s) (Reiniciar los servidores de las aplicaciones).

Lanzamiento y conexión a una instancia de Amazon RDS externa en EC2 Classic

 Important

La compatibilidad estándar de Amazon EC2-Classic llegará a su fin el 15 de agosto de 2022. Para evitar interrupciones en las cargas de trabajo, le recomendamos que migre de Amazon EC2-Classic a una VPC antes de esa fecha. También se solicita que no se lance ningún recurso de AWS en Amazon EC2-Classic de ahora en más y, en su lugar, se utilice Amazon

VPC. Para obtener más información, consulte [Migración de EC2-Classic a una VPC](#) y la entrada del blog [EC2-Classic Networking se retira: cómo prepararse](#).

Si utiliza EC2 Classic (sin VPC) con AWS Elastic Beanstalk, el procedimiento cambia ligeramente debido a las diferencias en la forma en que funcionan los grupos de seguridad. En EC2 Classic, las instancias de base de datos no pueden utilizar grupos de seguridad de EC2, por lo que obtienen un grupo de seguridad de base de datos que solo funciona con Amazon RDS.

Puede agregar reglas a un grupo de seguridad de base de datos que permitan el acceso entrante desde grupos de seguridad de EC2. Sin embargo, no puede adjuntar un grupo de seguridad de base de datos al grupo de Auto Scaling asociado a su entorno. Para no tener que crear una dependencia entre el grupo de seguridad de base de datos y su entorno, debe crear un tercer grupo de seguridad en Amazon EC2. Luego, debe agregar una regla en el grupo de seguridad de base de datos para darle acceso entrante al nuevo grupo de seguridad. Por último, debe asignarlo al grupo de Auto Scaling en el entorno de Elastic Beanstalk.

Note

- Si está comenzando con una base de datos que fue creada por Elastic Beanstalk y posteriormente desacoplada de un entorno de Beanstalk, puede omitir el primer grupo de pasos y continuar con los pasos agrupados en Crear un grupo de seguridad puente.
- Si planea utilizar la base de datos que desacopla para un entorno de producción, verifique que el tipo de almacenamiento que utiliza la base de datos es el adecuado para su carga de trabajo. Para obtener más información, consulte [Almacenamiento de instancias de base de datos](#) y [Modificación de una instancia de base de datos](#) en la Guía del usuario de Amazon RDS.

Para lanzar una instancia de RDS en EC2 Classic (sin VPC)

1. Abra la [consola de administración de RDS](#).
2. Elija Create database (Crear base de datos).
3. Avance por el asistente. Anote los valores que introduzca para las siguientes opciones:
 - Master Username (Nombre de usuario maestro)
 - Master Password (Contraseña maestra)

4. Cuando llegue a Configure advanced settings (Configurar opciones avanzadas), elija lo siguiente para Network and Security (Red y seguridad):
 - VPC: **Not in VPC**. Si esta opción no está disponible, es posible que su cuenta no sea compatible con [EC2-Classic](#) o que haya elegido un [tipo de instancia que solo está disponible en VPC](#).
 - Zona de disponibilidad – **No Preference**
 - Grupos de seguridad de base de datos – **Create new Security Group**
5. Configure las demás opciones y elija Create database (Crear base de datos). Anote los valores que introduzca para las siguientes opciones:
 - Database Name (Nombre de base de datos)
 - Database Port (Puerto de base de datos)

En EC2-Classic, la instancia de base de datos tiene un grupo de seguridad de base de datos en lugar de un grupo de seguridad de VPC. No puede adjuntar un grupo de seguridad de base de datos a su entorno de Elastic Beanstalk. En cambio, debe crear un nuevo grupo de seguridad que pueda autorizar a acceder a la instancia de base de datos y adjuntarse a su entorno. Denominaremos a este grupo de seguridad puente y le asignaremos el nombre **webapp-bridge**.

Para crear un grupo de seguridad puente

1. Abra la [consola de Amazon EC2](#).
2. Elija Security Groups (Grupos de seguridad) en Network & Security (Red y seguridad) en la barra lateral de navegación.
3. Elija Create Security Group.
4. En Security group name (Nombre del grupo de seguridad), escriba **webapp-bridge**.
5. En Description (Descripción), escriba **Provide access to DB instance from Elastic Beanstalk environment instances**.
6. En VPC, deje el valor predeterminado seleccionado.
7. Elija Create (Crear).

A continuación, modifique el grupo de seguridad asociado a la instancia de base de datos para permitir el tráfico entrante desde el grupo de seguridad puente.

Modificar las reglas de ingreso del grupo de seguridad de su instancia de RDS

1. Abra la [consola de Amazon RDS](#).
2. Seleccione Databases (Bases de datos).
3. Elija el nombre de la instancia de base de datos para ver sus detalles.
4. En la sección Connectivity (Conectividad), en Security (Seguridad), se muestra el grupo de seguridad que está asociado a la instancia de base de datos. Abra el enlace para ver el grupo de seguridad en la consola de Amazon EC2.
5. En el grupo de seguridad, establezca Connection Type (Tipo de conexión) en EC2 Security Group (Grupo de seguridad de EC2).
6. Establezca EC2 Security Group Name (Nombre de grupo de seguridad de EC2) en el nombre del grupo de seguridad puente que ha creado.
7. Elija Authorize (Autorizar).

A continuación, añada el grupo de seguridad puente a su entorno en ejecución. Este procedimiento requiere que todas las instancias del entorno se aprovisionen con el grupo de seguridad adicional asociado.

Para agregar un grupo de seguridad al entorno

- Haga una de las siguientes acciones:
 - Para añadir un grupo de seguridad mediante la consola de Elastic Beanstalk
 - a. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
 - b. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.


- c. En el panel de navegación, elija Configuration (Configuración).
- d. En la categoría de configuración Instances (Instancias), elija Edit (Editar).

- e. En EC2 security groups (Grupos de seguridad de EC2), elija el grupo de seguridad que desea asignar a las instancias, además del grupo de seguridad de instancia que crea Elastic Beanstalk.
 - f. Para guardar los cambios, elija Aplicar en la parte inferior de la página.
 - g. Lea la advertencia y, a continuación, elija Confirm (Confirmar).
- Para agregar un grupo de seguridad con un [archivo de configuración](#), utilice el archivo de ejemplo [securitygroup-addexisting.config](#).

A continuación, pase la información de la conexión a su entorno mediante propiedades de entorno. Cuando [agrega una instancia de base de datos a su entorno](#) con la consola de Elastic Beanstalk, Elastic Beanstalk utiliza propiedades del entorno tales como RDS_HOSTNAME para pasar la información de conexión a su aplicación. Puede utilizar las mismas propiedades para usar el mismo código de aplicación tanto con instancias de base de datos integradas como con instancias de base de datos externas. O, como alternativa, puede elegir sus propios nombres de propiedad.

Para configurar propiedades de entorno

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

 Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. En el panel de navegación, elija Configuration (Configuración).
4. En la categoría de configuración Actualizaciones, supervisión y registro, seleccione Editar.
5. En la sección Environment Properties (Propiedades del entorno), defina las variables que lee la aplicación para crear una cadena de conexión. Para conseguir compatibilidad con los entornos que tienen una instancia de RDS integrada, utilice lo siguiente:
 - RDS_DB_NAME: el nombre de base de datos que está en la consola de Amazon RDS.
 - RDS_USERNAME: el nombre de usuario maestro que introduce al añadir la base de datos al entorno.
 - RDS_PASSWORD: la contraseña maestra que introduce al añadir la base de datos al entorno.

- RDS_HOSTNAME: el punto de enlace de la instancia de base de datos que está en la consola de Amazon RDS.
- RDS_PORT: el puerto que está en la consola de Amazon RDS.

Environment Properties

The following properties are passed into the application as environment variables. [Learn more.](#)

Property Name	Property Value
RDS_DB_NAME	<input type="text" value="ebdb"/> ✕
RDS_HOSTNAME	<input type="text" value="webapp-db.jxzc b5mpaniu.us-wes"/> ✕
RDS_PORT	<input type="text" value="5432"/> ✕
RDS_USERNAME	<input type="text" value="webapp-admin"/> ✕
<input type="text" value="RDS_PASSWORD"/>	<input type="text" value="kUj5uKxmWDMYc403"/> +

[Cancel](#) [Apply](#)

6. Seleccione Apply (Aplicar)


Si todavía no programó la aplicación para leer las propiedades del entorno y crear una cadena de conexión, consulte los siguientes temas específicos del lenguaje para obtener instrucciones:

- Java SE: [Conexión a una base de datos \(plataformas Java SE\)](#)
- Java con Tomcat: [Conexión a una base de datos \(plataformas Tomcat\)](#)
- Node.js: [Conexión a una base de datos](#)
- .NET: [Conexión a una base de datos](#)
- PHP: [Conexión a una base de datos con PDO o MySQLi](#)
- Python: [Conexión a una base de datos](#)
- Ruby: [Conexión a una base de datos](#)

Por último, en función de cuándo lea la aplicación las variables de entorno, es posible que necesite reiniciar el servidor de la aplicación en las instancias del entorno.

Reiniciar los servidores de aplicaciones para su entorno

1. Abra la [consola de Elastic Beanstalk](#) y, en la lista Regions (Regiones), seleccione su Región de AWS.
2. En el panel de navegación, elija Environments (Entornos) y, a continuación, elija el nombre del entorno en la lista.

 Note

Si tiene muchos entornos, utilice la barra de búsqueda para filtrar la lista de entornos.

3. Elija Actions (Acciones) y, a continuación, elija Restart app server(s) (Reiniciar los servidores de las aplicaciones).

Almacenamiento de las credenciales de Amazon RDS en AWS Secrets Manager

AWS Secrets Manager lo ayuda a mejorar su posición de seguridad, ya que le permite almacenar y recuperar credenciales cifradas. El almacenamiento de las credenciales en Secrets Manager ayuda a evitar una posible concesión por parte de cualquier persona que pueda inspeccionar la aplicación o los componentes que se relacionen con ella. Su código puede realizar una llamada en tiempo de ejecución al servicio Secrets Manager para recuperar las credenciales de forma dinámica. Secrets Manager también ofrece características como los componentes de almacenamiento en caché de secretos del cliente para lenguajes de tiempos de ejecución, que incluyen Python, Go y Java.

Para obtener más información, consulte los siguientes temas en la Guía del usuario de AWS Secrets Manager.

- [Cómo Amazon RDS usa AWS Secrets Manager](#)
- [Creación de un secreto de base de datos de AWS Secrets Manager](#)
- [Retrieve secrets from AWS Secrets Manager](#)

Limpieza de una instancia externa de Amazon RDS

Cuando conecta una instancia de Amazon RDS externa a su entorno de Elastic Beanstalk, la instancia de base de datos no depende del ciclo de vida del entorno, por lo tanto, no se elimina cuando termina el entorno. Para garantizar que información personal que podría haber almacenado en la instancia de base de datos no se conserva innecesariamente, elimine los registros que ya no necesite. De forma alternativa, elimine la instancia de base de datos.

Usar Elastic Beanstalk con Amazon S3

Amazon Simple Storage Service (Amazon S3) proporciona un almacenamiento de datos de larga duración y con tolerancia a fallos.

Elastic Beanstalk crea un bucket de Amazon S3 denominado `elasticbeanstalk-region-account-id` para cada región en la que se crean entornos. Elastic Beanstalk utiliza este bucket para almacenar objetos, por ejemplo, archivos de configuración temporales, necesarios para el correcto funcionamiento de la aplicación.

Elastic Beanstalk no activa el cifrado predeterminado para el bucket de Amazon S3 que crea. Esto significa que, de forma predeterminada, los objetos se almacenan en el bucket sin cifrar (y solo pueden acceder los usuarios autorizados). Algunas aplicaciones requieren que se cifren todos los objetos cuando están almacenados en un disco duro, en una base de datos, etc. (también conocido como cifrado en reposo). Si tiene este requisito, puede configurar los buckets de su cuenta para el cifrado predeterminado. Para obtener más información, consulte [Cifrado predeterminado de Amazon S3 para los buckets de S3](#) en la Guía del usuario de Amazon Simple Storage Service.

Contenido del bucket de Elastic Beanstalk Amazon S3

En la tabla siguiente se enumeran algunos objetos que Elastic Beanstalk almacena en el `elasticbeanstalk-*` bucket de Amazon S3. La tabla también muestra los objetos que se tienen que eliminar manualmente. Para evitar costos de almacenamiento innecesarios y para garantizar que no se retiene información personal, asegúrese de eliminar manualmente estos objetos cuando ya no les necesita.

Objeto	¿Cuándo se almacenan?	¿Cuándo se eliminan?
Versiones de la aplicación	Al crear un entorno o implementar el código de la aplicación en un entorno existente, Elastic Beanstalk almacena una versión de la aplicación en Amazon S3 y la asocia con el entorno.	Durante la eliminación de la aplicación y de acuerdo con Ciclo de vida de la versión .
Paquetes de código fuente	Cuando carga una nueva versión de aplicación mediante la consola de Elastic Beanstalk o la CLI de EB, Elastic Beanstalk almacena una copia de la misma en Amazon S3 y la establece como paquete de origen de su entorno.	Manualmente. Al eliminar una versión de la aplicación, puede elegir Delete versions from Amazon S3 (Eliminar versiones de Amazon S3) para también eliminar el paquete de código fuente relacionado. Para obtener más información, consulte Administración de versiones de la aplicación .
Plataformas personalizadas	Cuando crea una plataforma personalizada, Elastic Beanstalk almacena temporalmente datos relacionados en Amazon S3.	Una vez completada correctamente la creación de la plataforma personalizada.
Archivos de registro	Puede solicitar a Elastic Beanstalk que recupere archivos de registro de instancias (registros de cola o de paquetes) y los almacene en Amazon S3. También puede habilitar la rotación de los logs y configurar el entorno para que los publique en Amazon S3 automáticamente después de rotarlos.	Registros de finalización y de paquete: 15 minutos después de haberse creado. Registros rotados: manualmente.
Configuraciones guardadas	Manualmente.	Manualmente.

Eliminar objetos en el bucket de Elastic Beanstalk Amazon S3

Cuando finaliza un entorno o elimina una aplicación, Elastic Beanstalk elimina la mayoría de los objetos relacionados de Amazon S3. Para minimizar los costos de almacenamiento de una aplicación en ejecución, elimine periódicamente los objetos que no necesite su aplicación. Además, preste atención a los objetos que tiene que eliminar manualmente, como se indica en [Contenido del bucket de Elastic Beanstalk Amazon S3](#). Para garantizar que la información confidencial no se retiene innecesariamente, elimine estos objetos cuando no les necesita.

- Elimine las versiones de aplicación que ya no prevea utilizar en su aplicación. Al eliminar una versión de la aplicación, puede seleccionar Delete versions from Amazon S3 (Eliminar versiones de Amazon S3) para eliminar también el paquete de código fuente relacionado, es decir, una copia del código fuente y los archivos de configuración de su aplicación, que Elastic Beanstalk cargó en Amazon S3 al implementar una aplicación o cargar una versión de la aplicación. Para obtener información sobre cómo eliminar una versión de la aplicación, consulte [Administración de versiones de la aplicación](#).
- Elimine los registros rotados que no necesite. También puede descargarlos o moverlos a Amazon S3 Glacier para analizarlos con más detalle.
- Elimine las configuraciones guardadas que ya no vaya a usar en ningún entorno.

Eliminar el bucket de Elastic Beanstalk Amazon S3

Cuando Elastic Beanstalk crea un bucket, también crea una política de bucket que se aplica al nuevo bucket. Esta política tiene dos propósitos:

- Permitir que los entornos escriban en el bucket.
- Evitar la eliminación accidental del bucket.

Debido a la política que Elastic Beanstalk aplica a los buckets que crea para sus entornos, no podrá eliminarlos, a menos que primero elimine la política del bucket de forma deliberada. Puede eliminar la política de bucket en la sección Permisos de las propiedades del bucket en la consola de Amazon S3.

Advertencia

Si elimina un bucket que Elastic Beanstalk creó en su cuenta y todavía tiene aplicaciones existentes y entornos en ejecución en la región correspondiente, es posible que sus aplicaciones dejen de funcionar correctamente. Por ejemplo:

- Cuando se realiza un escalado ascendente de un entorno, Elastic Beanstalk debe poder encontrar la versión de la aplicación del entorno en el bucket de Amazon S3 y utilizarla para iniciar nuevas instancias de Amazon EC2.
- Al crear una plataforma personalizada, Elastic Beanstalk utiliza almacenamiento temporal de Amazon S3 durante el proceso de creación.

Le recomendamos que elimine los objetos innecesarios específicos del bucket de Amazon S3 de Elastic Beanstalk, en lugar de eliminar todo el bucket.

Para eliminar un bucket de almacenamiento de Elastic Beanstalk (consola)

El procedimiento general para eliminar un bucket de S3 también se describe en [Eliminar un bucket de S3](#) en la Guía del usuario de Amazon S3. Dado que vamos a eliminar un bucket creado por Elastic Beanstalk mediante el siguiente procedimiento, incluimos pasos adicionales para eliminar primero la política de bucket.

1. Abra la [consola de Amazon S3](#).
2. Abra la página del bucket de almacenamiento de Elastic Beanstalk seleccionando el nombre del bucket.
3. Elija la pestaña Permissions (Permisos).
4. Elija Bucket Policy (Política de bucket).
5. Elija Eliminar (Delete).
6. Vuelva a la página principal de la consola de Amazon S3 y, a continuación, seleccione el bucket de almacenamiento de Elastic Beanstalk.
7. Seleccione Delete Bucket (Eliminar bucket).
8. Confirme que desea eliminar el bucket introduciendo el nombre del bucket en el campo de texto y, a continuación, elija Eliminar bucket.

Uso de Elastic Beanstalk con Amazon VPC

Puede utilizar una [Amazon Virtual Private Cloud](#) (Amazon VPC) a fin de crear una red segura para la aplicación Elastic Beanstalk y los recursos relacionados de AWS. Al crear su entorno, puede elegir qué VPC, subredes y grupos de seguridad se utilizan para sus instancias de aplicación y el balanceador de carga. Puede utilizar cualquier configuración de VPC que desee siempre que cumpla los requisitos siguientes.

Requisitos de la VPC

- **Acceso a internet:** las instancias deben tener acceso a Internet a través de uno de los métodos siguientes:
 - **Subred pública:** las instancias tienen una dirección IP pública y usan un gateway de Internet para obtener acceso a Internet.
 - **Subred privada:** las instancias usan un dispositivo NAT para obtener acceso a Internet.

Note

Si configura los [Puntos de enlace de la VPC](#) en su VPC para conectarse a los servicios `elasticbeanstalk` y `elasticbeanstalk-health`, el acceso a internet es opcional y solo se requiere si la aplicación lo necesita específicamente. Sin puntos finales de la VPC, su VPC debe tener acceso a internet.

La VPC predeterminada que Elastic Beanstalk configura para usted proporciona acceso a Internet.

Elastic Beanstalk no admite configuraciones de proxy como `HTTPS_PROXY` para la configuración de un proxy web.

- **NTP:** las instancias del entorno de Elastic Beanstalk usan Network Time Protocol (NTP, protocolo de hora de red) para sincronizar el reloj del sistema. Si las instancias no pueden comunicarse en el puerto UDP 123, es posible que el reloj no esté sincronizado y surjan problemas con los informes de estado de Elastic Beanstalk. Asegúrese de que los grupos de seguridad de la VPC y las listas de control de acceso de red permiten el tráfico UDP de entrada y de salida en el puerto 123 para evitar estos problemas.

El repositorio [elastic-beanstalk-samples](#) proporciona plantillas de AWS CloudFormation que puede utilizar con el fin de crear una VPC para utilizarla con los entornos de Elastic Beanstalk.

Para crear recursos con una plantilla de AWS CloudFormation

1. Clone el repositorio de ejemplos o descargue una plantilla con los enlaces en [README](#).
2. Abra la [consola de AWS CloudFormation](#).
3. Elija Create stack (Crear pila).
4. Elija Upload a template to Amazon S3 (Cargar una plantilla en Amazon S3).
5. Elija Upload file (Cargar archivo) y cargue el archivo de plantilla desde su máquina local.
6. Elija Next (Siguiendo) y siga las instrucciones para crear una pila con los recursos de la plantilla.

Cuando termina la creación de la pila, observe la pestaña Outputs (Salidas) para ver el ID de VPC y los ID de subred. Utilícelos para configurar la VPC en la [categoría de configuración de red](#) del asistente para crear un nuevo entorno.

Temas

- [VPC pública](#)
- [VPC pública/privada](#)
- [VPC privada](#)
- [Ejemplo: Lanzamiento de una aplicación Elastic Beanstalk en una VPC con hosts bastión](#)
- [Ejemplo: Lanzamiento de Elastic Beanstalk en una VPC con Amazon RDS](#)
- [Uso de Elastic Beanstalk con puntos de enlace de la VPC](#)

VPC pública

Plantilla de AWS CloudFormation: [vpc-public.yaml](#)

Configuración (carga balanceada)

- Visibilidad de balanceador de carga : público
- Subredes de balanceador de carga: ambas subredes públicas
- IP pública de instancia: habilitada
- Subredes de instancia: ambas subredes públicas
- Grupos de seguridad de instancia: añade el grupo de seguridad predeterminado

Configuración (instancia individual)

- Subredes de instancia: una de las subredes públicas
- Grupos de seguridad de instancia: añade el grupo de seguridad predeterminado

Un diseño de VPC básico solo público incluye una o varias subredes públicas, un gateway de internet y un grupo de seguridad predeterminado que permite el tráfico entre los recursos de la VPC. Cuando crea un entorno en la VPC, Elastic Beanstalk crea recursos adicionales que varían en función del tipo de entorno.

Recursos de la VPC

- Instancia individual: Elastic Beanstalk crea un grupo de seguridad para la instancia de aplicación que permite el tráfico en el puerto 80 desde Internet y asigna a la instancia una dirección IP elástica para que tenga una dirección IP pública. El nombre de dominio del entorno se resuelve como la dirección IP pública de la instancia.
- Con balanceo de carga: Elastic Beanstalk crea un grupo de seguridad para el balanceador de carga que permite el tráfico en el puerto 80 desde Internet y un grupo de seguridad para las instancias de aplicación que permite el tráfico desde el grupo de seguridad del balanceador de carga. El nombre de dominio del entorno se resuelve como la dirección IP pública del balanceador de carga.

Esto es similar a la forma en que Elastic Beanstalk gestiona la red cuando se utiliza la VPC predeterminada. La seguridad en una subred pública depende del balanceador de carga y de los grupos de seguridad de las instancias creados por Elastic Beanstalk. Es la configuración más económica, ya que no requiere un gateway NAT.

VPC pública/privada

Plantilla deAWS CloudFormation: [vpc-privatepublic.yaml](#)

Configuración (carga balanceada)

- Visibilidad de balanceador de carga : público
- Subredes de balanceador de carga: ambas subredes públicas
- IP pública de instancia: deshabilitada
- Subredes de instancia: ambas subredes privadas

- Grupos de seguridad de instancia: añade el grupo de seguridad predeterminado

Para aumentar la seguridad, añade subredes privadas a la VPC para crear un diseño público-privado. Este diseño requiere un balanceador de carga y un gateway NAT en las subredes públicas, y le permite ejecutar las instancias de aplicación, bases de datos y otros recursos en subredes privadas. Las instancias de subredes privadas solo puede comunicarse con internet a través del balanceador de carga y el gateway NAT.

VPC privada

Plantilla de AWS CloudFormation: [vpc-private.yaml](#)

Configuración (carga balanceada)

- Visibilidad de balanceador de carga: privada
- Subredes de balanceador de carga: ambas subredes privadas
- IP pública de instancia: deshabilitada
- Subredes de instancia: ambas subredes privadas
- Grupos de seguridad de instancia: añade el grupo de seguridad predeterminado

En aplicaciones internas que no deban ser accesibles desde internet, puede ejecutarlo todo en subredes privadas y configurar el balanceador de carga para que se dirija hacia el interior (cambiar Load balancer visibility (Visibilidad del balanceador de carga) a Internal (Interna)). Esta plantilla crea una VPC sin subredes públicas y sin gateway de internet. Utilice este diseño para las aplicaciones que solo deban ser accesibles desde la misma VPC o desde una VPN adjunta.

Ejecución de un entorno Elastic Beanstalk en una VPC privada

Cuando crea su entorno de Elastic Beanstalk en una VPC privada, el entorno no tiene acceso a Internet. Es posible que su aplicación necesite acceso al servicio Elastic Beanstalk u otros servicios. Su entorno podría utilizar informes de estado mejorados. En este caso, las instancias del entorno envían información de estado al servicio de salud mejorado. Y el código de Elastic Beanstalk en las instancias del entorno envía parte del tráfico a otros servicios de AWS y parte del tráfico a los puntos de enlace que no son de AWS (por ejemplo, con el fin de descargar paquetes de dependencias para su aplicación). Estos son algunos pasos que puede que deba seguir en este caso para asegurarse de que el entorno funciona correctamente.

- Configure los puntos de enlace de la VPC para Elastic Beanstalk: Elastic Beanstalk y su servicio de mantenimiento mejorado admiten puntos de enlace de la VPC que garantizan que el tráfico a estos servicios permanezca dentro de la red de Amazon y no requiera acceso a Internet. Para obtener más información, consulte [. the section called “Puntos de conexión de la VPC”](#).
- Configure los puntos de enlace de la VPC para servicios adicionales: las instancias de Elastic Beanstalk envían tráfico a varios servicios más de AWS en su nombre: Amazon Simple Storage Service (Amazon S3), Amazon Simple Queue Service (Amazon SQS), AWS CloudFormation y Amazon CloudWatch Logs. También debe configurar los puntos de enlace de la VPC para estos servicios. Para obtener información detallada acerca de los puntos de enlace de la VPC, incluidos los vínculos por servicio, consulte [Puntos de enlace de la VPC](#) en la Guía de usuario de Amazon VPC.

Note

Algunos servicios de AWS, como Elastic Beanstalk, admiten puntos de enlace de la VPC en un número limitado de regiones de AWS. Cuando diseñe la solución de VPC privada, compruebe que Elastic Beanstalk y los demás servicios dependientes mencionados aquí admitan puntos de enlace de la VPC en la región de AWS que elija.

- Proporcionar una imagen de Docker privada: en un entorno de [Docker](#) el código de las instancias del entorno podría intentar extraer la imagen de Docker configurada de internet durante la creación del entorno y fallar. Para evitar este error, [cree una imagen de Docker personalizada](#) en su entorno o utilice una imagen de Docker almacenada en [Amazon Elastic Container Registry](#) (Amazon ECR) y [configure un extremo de VPC para el servicio Amazon ECR](#).
- Habilitar los nombres DNS: el código de Elastic Beanstalk en las instancias del entorno envía tráfico a todos los servicios de AWS mediante sus puntos de enlace públicos. Para asegurarse de que este tráfico pasa, elija la opción Enable DNS name (Habilitar nombre del DNS) cuando configure todos los puntos de enlace de la VPC de tipo interfaz. Esto añade una entrada DNS en la VPC que asigna el punto de enlace del servicio público al punto de enlace de la VPC de tipo interfaz.

Important

Si la VPC no es privada y tiene acceso público a internet, y si Enable DNS name (Habilitar nombre del DNS) está deshabilitado para cualquier punto de enlace de la VPC, el tráfico hacia su servicio respectivo se desplaza a través del internet público. Probablemente no sea lo que pretendía. Es fácil detectar este problema con una VPC privada, ya que evita

que este tráfico pase y recibe errores. Sin embargo, con una VPC pública, no se obtiene ninguna indicación.

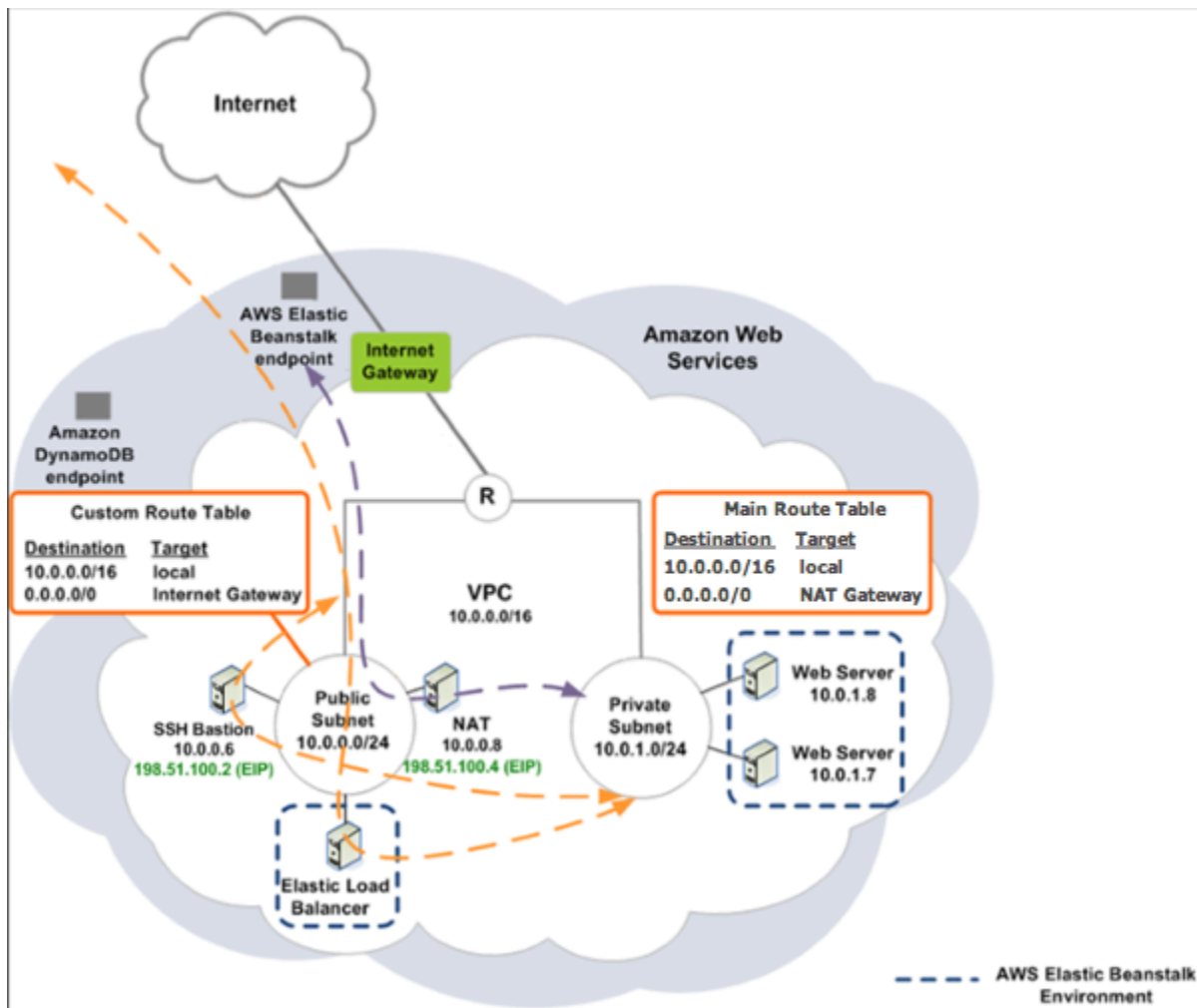
- Include application dependencies (Incluir dependencias de aplicaciones): si la aplicación tiene dependencias como paquetes de tiempo de ejecución de idiomas, podría intentar descargarlas e instalarlas desde internet durante la creación del entorno y fallar. Para evitar este error, incluya todos los paquetes de dependencias en el código fuente de la aplicación.
- Utilizar una versión de plataforma actual: asegúrese de que su entorno utiliza la versión de la plataforma publicada el 24 de febrero de 2020 o una versión más reciente. Utilice específicamente una versión de plataforma que se publicó en o después de una de estas dos actualizaciones: [Linux Update 2020-02-28](#), [Windows Update 2020-02-24](#).

Note

La razón por la que se necesita una versión de plataforma actualizada es que las versiones anteriores tenían un problema que impedía que las entradas DNS creadas por la opción Enable DNS name (Habilitar nombre del DNS) funcionaran correctamente para Amazon SQS.

Ejemplo: Lanzamiento de una aplicación Elastic Beanstalk en una VPC con hosts bastión

Si las instancias de Amazon EC2 están en una subred privada, no podrá conectarse a ellas de forma remota. Para conectarse a las instancias, puede configurar servidores bastión en la subred pública para que actúen como servidores proxy. Por ejemplo, puede configurar forwarders del puerto SSH o gateways de RDP en la subred pública para que actúen como servidores proxy con el tráfico dirigido a los servidores de la base de datos que procede de la propia red. Esta sección contiene un ejemplo acerca de cómo crear una VPC con subredes públicas y privadas. Las instancias se encuentran en la subred privada, mientras que el host bastión, el gateway NAT y el balanceador de carga se encuentran en la subred pública. Su infraestructura será similar al diagrama siguiente.



Para implementar una aplicación de Elastic Beanstalk dentro de una VPC mediante un host bastión, complete los pasos que se describen en los apartados siguientes.

Pasos

- [Creación de una VPC con subredes públicas y privadas](#)
- [Creación y configuración del grupo de seguridad del host bastión](#)
- [Actualización del grupo de seguridad de instancias](#)
- [Creación de un host bastión](#)

Creación de una VPC con subredes públicas y privadas

Complete todos los procedimientos de [VPC pública/privada](#). Cuando implemente la aplicación, debe especificar un par de claves de Amazon EC2 en las instancias para que pueda conectarse a ellas

de forma remota. Para obtener más información sobre cómo especificar el par de claves en las instancias, consulte [Instancias de Amazon EC2 para el entorno de Elastic Beanstalk](#).

Creación y configuración del grupo de seguridad del host bastión

Cree un grupo de seguridad para el host bastión y agregue reglas que permitan el tráfico SSH procedente de Internet y el tráfico SSH saliente dirigido a la subred privada que contiene las instancias Amazon EC2.

Para crear el grupo de seguridad del host bastión

1. Abra la consola de Amazon VPC en <https://console.aws.amazon.com/vpc/>.
2. En el panel de navegación, elija Security Groups (Grupos de seguridad).
3. Elija Create Security Group (Crear grupo de seguridad).
4. En el cuadro de diálogo Create Security Group (Crear grupo de seguridad), escriba lo siguiente y elija Yes, Create (Sí, crear).

Name tag (Etiqueta de nombre) (opcional)

Introduzca una etiqueta de nombre para el grupo de seguridad.

Nombre del grupo

Introduzca el nombre del grupo de seguridad.

Descripción

Introduzca una descripción para el grupo de seguridad.

VPC

Seleccione la VPC.

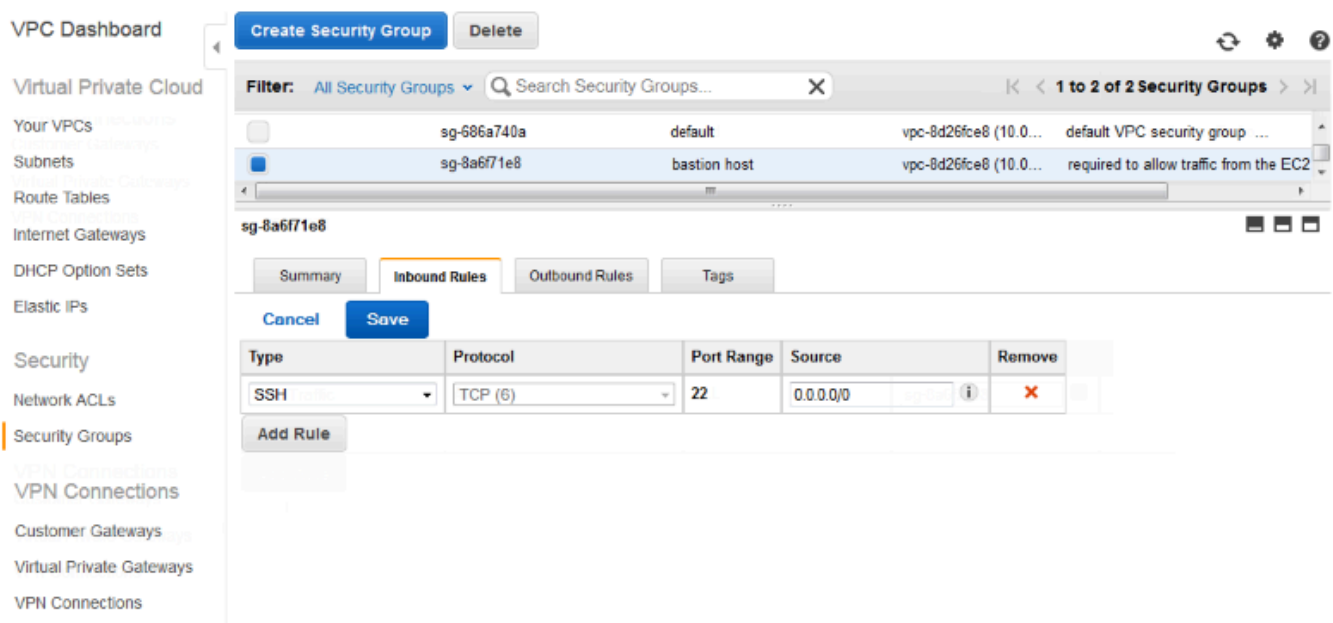
Se crea el grupo de seguridad y aparece en la página Security Groups (Grupos de seguridad). Observe que tiene un ID (p. ej., sg-xxxxxxxx). Puede que deba activar la columna Group ID (ID de grupo); para ello, haga clic en Show/Hide (Mostrar/Ocultar) en la esquina superior derecha de la página.

Para configurar el grupo de seguridad del host bastión

1. En la lista de grupo de seguridad, active la casilla del grupo de seguridad que acaba de crear para el host bastión.
2. En la pestaña Inbound Rules, elija Edit.
3. Si es necesario, elija Add another rule (Añadir otra regla).
4. Si el host bastión es una instancia de Linux, en Type (Tipo), seleccione SSH.

Si el host bastión es una instancia de Windows, en Type (Tipo), seleccione RDP.

5. Escriba el rango de CIDR de origen que desee en el campo Source (Origen) y haga clic en Save (Guardar).



6. En la pestaña Outbound Rules (Reglas de salida), elija Edit (Editar).
7. Si es necesario, elija Add another rule (Añadir otra regla).
8. En Type (Tipo), seleccione el tipo que ha especificado para la regla de entrada.
9. En el campo Source (Origen), escriba el rango de CIDR de la subred de los hosts en la subred privada de la VPC.

Para encontrarla:

- a. Abra la consola de Amazon VPC en <https://console.aws.amazon.com/vpc/>.
- b. En el panel de navegación, elija Subnets.

- c. Anote el valor en IPv4 CIDR para cada Availability Zone (Zona de disponibilidad) en la que tenga hosts con los que desee que se conecte el host bastión.

Note

Si tiene hosts en varias zonas de disponibilidad, cree una regla de salida para cada una de ellas.

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR	Availability Zone
	subnet-9de72a91	available	vpc-7523e317	172.31.0.0/20	4091		us-east-1c
	subnet-04575af9	available	vpc-7523e317	172.31.16.0/20	4090		us-east-1a
	subnet-3d33734c	available	vpc-7523e317	172.31.32.0/20	4091		us-east-1b
	subnet-caa74a67	available	vpc-7523e317	172.31.48.0/20	4091		us-east-1e
	subnet-9dfcb9f5	available	vpc-7523e317	172.31.64.0/20	4091		us-east-1f

10. Seleccione Save.

Actualización del grupo de seguridad de instancias

De forma predeterminada, el grupo de seguridad que creó para las instancias no permite el tráfico entrante. Aunque Elastic Beanstalk va a modificar el grupo predeterminado para que las instancias permitan el tráfico SSH, debe modificar el grupo de seguridad de instancias predeterminado para que permita el tráfico RDP si las instancias son de Windows.

Si desea actualizar el grupo de seguridad de instancias para RDP

1. En la lista de grupos de seguridad, active la casilla del grupo de seguridad de instancias.
2. En la pestaña Inbound (Entrada), seleccione Edit (Editar).
3. Si es necesario, elija Add another rule (Añadir otra regla).
4. Especifique los valores siguientes y haga clic en Save (Guardar).

Tipo

RDP

Protocolo

TCP

Rango de puertos

3389

Source

Especifique el ID del grupo de seguridad del host bastión (por ejemplo, sg-8a6f71e8) y seleccione Save (Guardar).

Creación de un host bastión

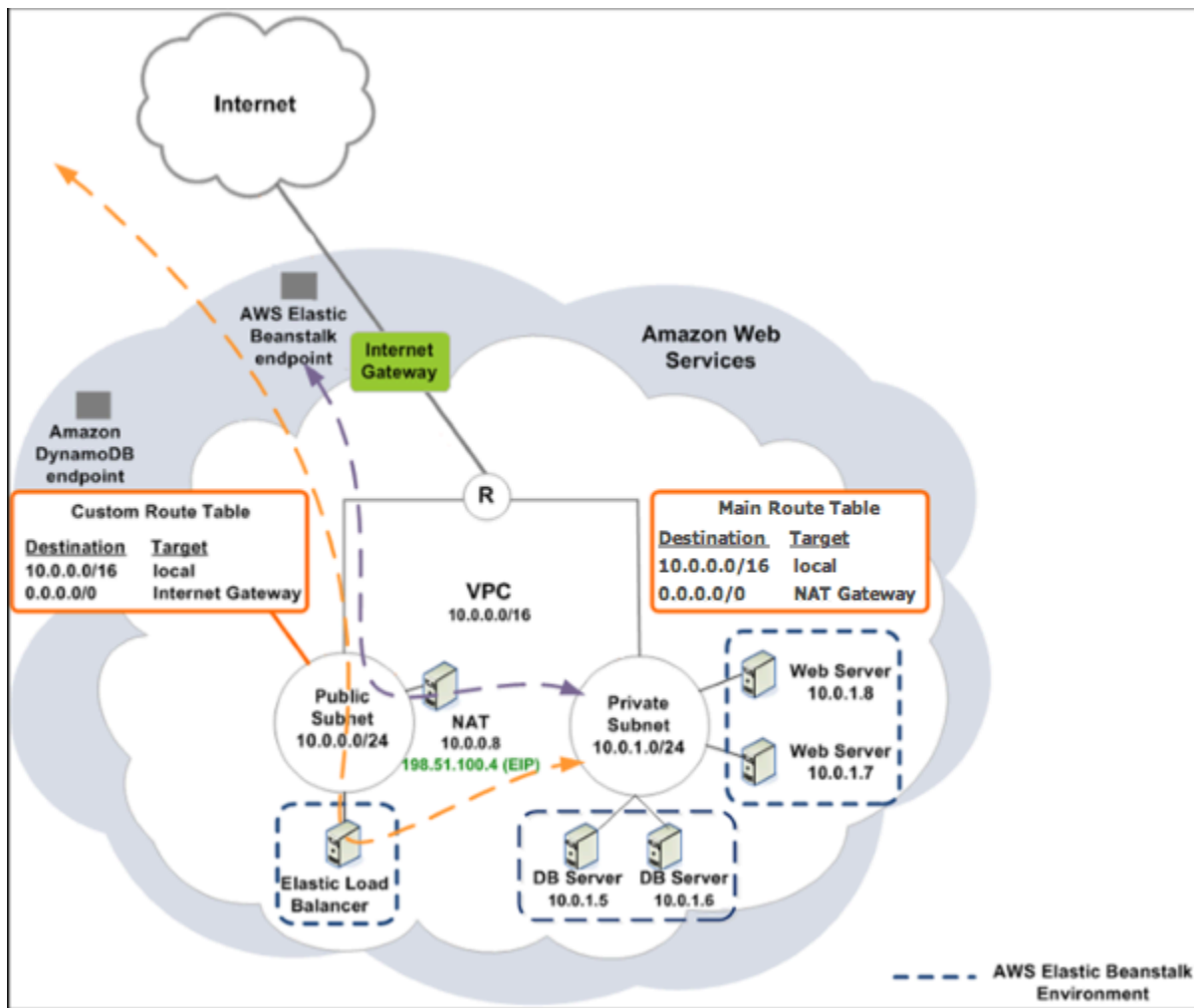
Para crear un host bastión, lance una instancia Amazon EC2 en una subred pública que actuará como host bastión.

Para obtener más información sobre la configuración de un host bastión para instancias Windows en la subred privada, consulte [Controlling Network Access to EC2 Instances Using a Bastion Server \(Controlar el acceso a redes para instancias EC2 utilizando un servidor bastión\)](#).

Para obtener más información sobre la configuración de un host bastión para instancias Linux en la subred privada, consulte [Securely Connect to Linux Instances Running in a Private Amazon VPC](#).

Ejemplo: Lanzamiento de Elastic Beanstalk en una VPC con Amazon RDS

En esta sección, encontrará las tareas para implementar una aplicación de Elastic Beanstalk con Amazon RDS en una VPC utilizando una gateway de NAT. Su infraestructura será similar al diagrama siguiente.



Note

Si nunca antes ha utilizado una instancia de base de datos con la aplicación, intente [agregar una a un entorno de prueba](#) y [establecer conexión con una instancia de base de datos externa](#) antes de agregar una configuración de VPC al conjunto.

Creación de una VPC con subredes públicas y privadas

Puede utilizar la [consola de Amazon VPC](#) para crear una VPC.

Para crear una VPC

1. Inicie sesión en la [consola de Amazon VPC](#).

- En el panel de navegación, elija VPC Dashboard (Panel de VPC). A continuación, elija Create VPC (Crear VPC).
- Elija VPC with Public and Private Subnets (VPC con subredes privadas y públicas) y, a continuación, Select (Seleccionar).

Step 1: Select a VPC Configuration

VPC with a Single Public Subnet

VPC with Public and Private Subnets

VPC with Public and Private Subnets and Hardware VPN Access

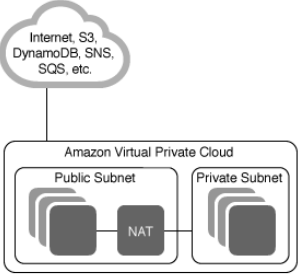
VPC with a Private Subnet Only and Hardware VPN Access

In addition to containing a public subnet, this configuration adds a private subnet whose instances are not addressable from the Internet. Instances in the private subnet can establish outbound connections to the Internet via the public subnet using Network Address Translation (NAT).

Creates:

A /16 network with two /24 subnets. Public subnet instances use Elastic IPs to access the Internet. Private subnet instances access the Internet via Network Address Translation (NAT). (Hourly charges for NAT devices apply.)

[Select](#)



[Cancel and Exit](#)

- El balanceador de carga Elastic Load Balancing y las instancias Amazon EC2 deben estar en la misma zona de disponibilidad para comunicarse entre sí. Elija la misma zona de disponibilidad en cada lista de zona de disponibilidad.

Step 2: VPC with Public and Private Subnets

IPv4 CIDR block: (65531 IP addresses available)

IPv6 CIDR block: No IPv6 CIDR Block
 Amazon provided IPv6 CIDR block

VPC name:

Public subnet's IPv4 CIDR: (251 IP addresses available)

Availability Zone: No Preference

Public subnet name:

Private subnet's IPv4 CIDR: (251 IP addresses available)

Availability Zone: No Preference

Private subnet name:

You can add more subnets after AWS creates the VPC.

Specify the details of your NAT gateway ([NAT gateway rates apply](#)). Use a NAT instance instead

Elastic IP Allocation ID:

Service endpoints [Add Endpoint](#)

Enable DNS hostnames: Yes No

Hardware tenancy:

Enable ClassicLink: Yes No

[Cancel and Exit](#) [Back](#) [Create VPC](#)

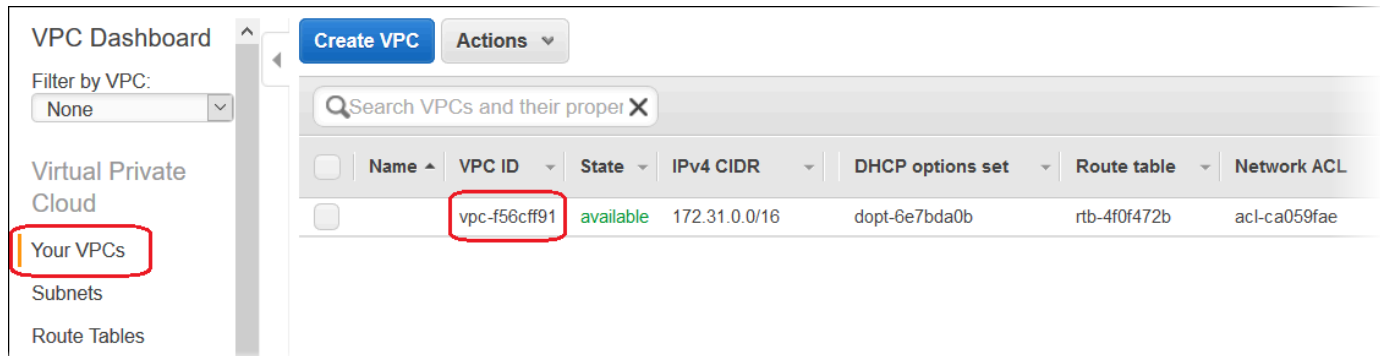
- Elija una dirección IP elástica para gateway NAT.
- Seleccione Create VPC.

El asistente comienza a crear la VPC, las subredes y la gateway de Internet. También actualiza la tabla de ruteo principal y crea una personalizada. Por último, el asistente crea una gateway NAT en la subred pública.

Note

Puede elegir lanzar una instancia NAT en la subred pública en lugar de una gateway NAT. Para obtener más información, consulte [Situación 2: VPC con subredes privadas y públicas \(NAT\)](#) en la Guía de usuario de Amazon VPC.

- Una vez creada la VPC correctamente, obtendrá un ID de VPC. Necesitará este valor para el siguiente paso. Para ver el ID de la VPC, seleccione Your VPCs (Sus VPC) en el panel izquierdo de la [consola de Amazon VPC](#).



Creación de un grupo de subredes de base de datos

Un grupo de subredes de base de datos de una VPC es una colección de subredes (normalmente privadas) que se pueden designar para las instancias de base de datos de RDS de los backend. Cada grupo de subredes de base de datos debe tener al menos una subred por cada una de las zonas de disponibilidad de una región de AWS determinada. Para obtener más información, consulte [Creación de una subred en la VPC](#).

Creación de un grupo de subredes de base de datos

- Abra la [consola de Amazon RDS](#).
- En el panel de navegación, elija Subnet groups.
- Elija Create DB Subnet Group.
- Elija Name (Nombre) y escriba el nombre del grupo de subredes de base de datos.

5. Elija Description (Descripción) y describa el grupo de subredes de base de datos.
6. Para VPC, elija el ID de la VPC que ha creado.
7. En Add subnets (Añadir subredes), elija Add all the subnets related to this VPC (Añadir todas las subredes relacionadas con esta VPC).

Add subnets
Add subnet(s) to this subnet group. You may add subnets one at a time below or add all the subnets related to this VPC. You may make additions/edits after this group is created. A minimum of 2 subnets is required.

Add all the subnets related to this VPC

Availability zone
Choose an availability zone ▼

Subnet
Choose a subnet ▼ **Add subnet**

Subnets in this subnet group (4)

Availability zone	Subnet ID	CIDR block	Action
us-east-2c	subnet-da3408ae	10.0.1.0/24	Remove
us-east-2c	subnet-db3408af	10.0.0.0/24	Remove
us-east-2b	subnet-4f195024	10.0.2.0/24	Remove
us-east-2a	subnet-fe064f95	10.0.3.0/24	Remove

Cancel **Create**

8. Cuando haya terminado, seleccione Create (Crear).

El nuevo grupo de subredes de base de datos aparece en la lista de grupos de subredes de la consola de Amazon RDS. En el panel de detalles que se encuentra en la parte inferior de la página, puede elegir en esa lista para ver los detalles (por ejemplo, todas las subredes asociadas al grupo).

Implementar en Elastic Beanstalk

Una vez configurada la VPC, puede crear el entorno dentro de ella e implementar la aplicación en Elastic Beanstalk. Puede hacerlo con la consola de Elastic Beanstalk o con los conjuntos de herramientas de AWS, AWS CLI, la CLI de EB o la API de Elastic Beanstalk. Si utiliza la consola de Elastic Beanstalk, solo tiene que cargar el archivo `.war` o `.zip` y seleccionar la configuración VPC dentro del asistente. Después, Elastic Beanstalk crea el entorno dentro de la VPC e implementa la aplicación. Por otra parte, puede utilizar los conjuntos de herramientas de AWS, AWS CLI, la CLI

de EB o la API de Elastic Beanstalk para implementar la aplicación. Para ello, tiene que definir los valores de configuración de la VPC en un archivo de configuración e implementarlo con el paquete de código fuente. En este tema se ofrecen instrucciones para ambos métodos.

Implementación con la consola de Elastic Beanstalk

La consola de Elastic Beanstalk le guía en el proceso de creación de un nuevo entorno dentro de la VPC. Tiene que proporcionar un archivo `.war` (para aplicaciones Java) o un archivo `.zip` (para el resto de aplicaciones). En la página VPC Configuration (Configuración de VPC) del asistente de entorno de Elastic Beanstalk, debe hacer las selecciones siguientes:

VPC

Seleccione la VPC.

VPC security group (Grupo de seguridad de VPC)

Seleccione el grupo de seguridad de la instancia que ha creado antes.

ELB visibility (Visibilidad de ELB)

Seleccione `External` si el balanceador de carga debe estar disponible públicamente o seleccione `Internal` si el balanceador de carga solo debe estar disponible dentro de la VPC.

Seleccione las subredes para el balanceador de carga y las instancias EC2. Asegúrese de que selecciona la subred pública para el balanceador de carga y la subred privada para las instancias Amazon EC2. De manera predeterminada, el asistente para la creación de VPC crea la subred pública en `10.0.0.0/24` y la subred privada en `10.0.1.0/24`.

Puede ver el ID de la subred eligiendo Subnets (Subredes) en la [consola de Amazon VPC](#).

The screenshot shows the AWS VPC Dashboard. On the left, the 'Subnets' link is highlighted with a red box. The main area displays a table of subnets:

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	Availability Zone
	subnet-3ba3c75e	available	vpc-f56cff91	172.31.64.0/20	4091	us-east-1a
	subnet-ec18feb4	available	vpc-f56cff91	172.31.16.0/20	4089	us-east-1d

Below the table, the details for 'subnet-ec18feb4' are shown under the 'Summary' tab:

Subnet ID:	subnet-ec18feb4	Availability Zone:	us-east-1d
IPv4 CIDR:	172.31.16.0/20	Route table:	rtb-4f0f472b
IPv6 CIDR:		Network ACL:	acl-ca059fae
State:	available	Default subnet:	yes
VPC:	vpc-f56cff91	Auto-assign Public IP:	yes
Available IPs:	4089	Auto-assign IPv6 address:	no

Implementar con los conjuntos de herramientas de AWS, la CLI de EB, la AWS CLI o la API

Cuando se implementa la aplicación en Elastic Beanstalk con los conjuntos de herramientas de AWS, la CLI de EB, la AWS CLI o la API, puede especificar la configuración de las opciones de la VPC en un archivo e implementarlo con el paquete de código fuente. Para obtener más información, consulte [Personalización avanzada de entornos con archivos de configuración \(.ebextensions\)](#).

Cuando actualice las opciones de configuración, tendrá que especificar lo siguiente:

- VPCId: contiene el ID de la VPC.
- Subnets: contiene el ID de la subred del grupo de Auto Scaling. En este ejemplo, es el ID de la subred privada.
- ELBSubnets: contiene el ID de la subred del balanceador de carga. En este ejemplo, es el ID de la subred pública.
- SecurityGroups: contiene el ID de los grupos de seguridad.
- DBSubnets: contiene el ID de las subredes de base de datos.

Note

Si utiliza subredes de base de datos, deberá crear subredes adicionales en la VPC para cubrir todas las zonas de disponibilidad de la región de AWS.

Si lo desea, también puede especificar los siguientes datos:

- **ELBScheme**: especifique `internal` si desea crear un balanceador de carga interno en la VPC para que no se pueda obtener acceso a la aplicación de Elastic Beanstalk desde el exterior de la VPC.

A continuación, se incluye un ejemplo de la configuración que se utiliza al implementar la aplicación de Elastic Beanstalk en una VPC. Para obtener más información sobre la configuración de VPC (con ejemplos acerca de cómo definir las opciones, los valores predeterminados y los valores válidos), consulte el espacio de nombres `aws:ec2:vpc` en [Opciones de configuración](#).

```
option_settings:
  - namespace: aws:autoscaling:launchconfiguration
    option_name: EC2KeyName
    value: ec2keypair

  - namespace: aws:ec2:vpc
    option_name: VPCId
    value: vpc-170647c

  - namespace: aws:ec2:vpc
    option_name: Subnets
    value: subnet-4f195024

  - namespace: aws:ec2:vpc
    option_name: ELBSubnets
    value: subnet-fe064f95

  - namespace: aws:ec2:vpc
    option_name: DBSubnets
    value: subnet-fg148g78

  - namespace: aws:autoscaling:launchconfiguration
    option_name: InstanceType
    value: m1.small

  - namespace: aws:autoscaling:launchconfiguration
    option_name: SecurityGroups
    value: sg-7f1ef110
```

Note

Si utiliza subredes de bases de datos, asegúrese de que tiene subredes en la VPC para cubrir todas las zonas de disponibilidad de la región de AWS.

Uso de Elastic Beanstalk con puntos de enlace de la VPC

Un punto de enlace de la VPC le permite conectar de forma privada su VPC a los servicios de AWS compatibles y a servicios de punto de enlace de la VPC basados en AWS PrivateLink sin necesidad de una gateway de internet, un dispositivo NAT, una conexión de VPN o una conexión de AWS Direct Connect.

Las instancias de su VPC no necesitan direcciones IP públicas para comunicarse con los recursos del servicio. El tráfico entre su VPC y el servicio no sale de la red de Amazon. Para obtener información detallada acerca de los puntos de enlace de la VPC, consulte la sección sobre [puntos de enlace de la VPC](#) en la Guía de usuario de Amazon VPC.

AWS Elastic Beanstalk es compatible con AWS PrivateLink, que proporciona conectividad privada al servicio de Elastic Beanstalk y elimina la exposición del tráfico a la Internet pública. Para habilitar la aplicación con el fin de que envíe solicitudes a Elastic Beanstalk mediante AWS PrivateLink, configure un tipo de punto de enlace de la VPC conocido como punto de enlace de la VPC de la interfaz (punto de enlace de la interfaz). Para obtener más información, consulte [Puntos de enlace de la VPC de la interfaz \(AWS PrivateLink\)](#) en la Guía del usuario de Amazon VPC.

Note

Elastic Beanstalk es compatible con AWS PrivateLink y los puntos de enlace de la VPC de la interfaz en un número limitado de regiones de AWS. Trabajamos para ampliar la compatibilidad a más regiones de AWS en un futuro próximo.

Configuración de un punto de enlace de la VPC para Elastic Beanstalk

Para crear el punto de enlace de la VPC de la interfaz para el servicio de Elastic Beanstalk en la VPC, siga el procedimiento [Crear un punto de enlace de la interfaz](#). En Service name (Nombre del servicio), seleccione `com.amazonaws.región.elasticbeanstalk`.

Si la VPC está configurada con acceso público a Internet, la aplicación podrá acceder a Elastic Beanstalk a través de Internet mediante el punto de enlace público `elasticbeanstalk.region.amazonaws.com`. Puede evitarlo asegurándose de que `Enable DNS name` (Habilitar nombre de la DNS) esté habilitado durante la creación del punto de enlace (está establecido como “true” de forma predeterminada). Esto añade una entrada DNS en la VPC que asigna el punto de enlace del servicio público al punto de enlace de la VPC de tipo interfaz.

Configuración de un punto de enlace de la VPC para mejorar el estado

Si ha habilitado los [informes de estado mejorados](#) de su entorno, puede configurar la información de estado mejorada para que también se envíe a través de AWS PrivateLink. El daemon `healthd`, un componente de Elastic Beanstalk en las instancias de su entorno, envía información de estado mejorada a un servicio de salud mejorado de Elastic Beanstalk independiente. Para crear un punto de enlace de la VPC de tipo interfaz para este servicio en la VPC, siga el procedimiento [Crear un punto de enlace de la interfaz](#). En `Service name` (Nombre del servicio), seleccione `com.amazonaws.region.elasticbeanstalk-health`.

Important

El daemon `healthd` envía información de estado mejorada al punto de enlace público, `elasticbeanstalk-health.region.amazonaws.com`. Si la VPC está configurada con acceso público a internet y `Enable DNS name` (Habilitar nombre de DNS) está deshabilitado en el punto de enlace de la VPC, la información de estado mejorada viaja a través de internet público. Probablemente esta no sea su intención cuando configure un punto de enlace de la VPC de estado mejorado. Asegúrese de que `Enable DNS name` (Habilitar nombre de DNS) esté habilitado (está establecido como “true” de forma predeterminada).

Uso de puntos de enlace de la VPC en una VPC privada

Una VPC privada, o una subred privada de una VPC, no tiene acceso público a internet. Es posible que desee ejecutar su entorno de Elastic Beanstalk en una [VPC privada](#) y configurar los puntos de enlace de la VPC de tipo interfaz para mejorar la seguridad. En este caso, tenga en cuenta que su entorno podría intentar conectarse a Internet por otros motivos además de para ponerse en contacto con el servicio de Elastic Beanstalk. Para obtener más información sobre cómo ejecutar un entorno en una VPC privada, consulte [the section called “Ejecución de un entorno Elastic Beanstalk en una VPC privada”](#).

Uso de políticas de punto de enlace para controlar el acceso con puntos de enlace de la VPC

De forma predeterminada, un punto de enlace de la VPC permite el acceso completo al servicio al que está asociado. Al crear o modificar un punto de enlace, puede adjuntar una política de punto de enlace.

Una política de punto de enlace es una política de recursos de AWS Identity and Access Management (IAM) que controla el acceso desde el punto de enlace al servicio especificado. La política de punto final es específica del punto final. Es independiente de cualquier política de usuario o instancia de IAM que pueda tener su entorno y no las reemplaza ni las sobrescribe. Para obtener información detallada sobre la creación y el uso de políticas de punto de enlace de la VPC, consulte [Control de acceso a los servicios con puntos de enlace de la VPC](#) en la Guía de usuario de Amazon VPC.

En el ejemplo siguiente se deniega a todos los usuarios el permiso para terminar un entorno a través del punto final de la VPC y se permite el acceso completo a todas las demás acciones.

```
{
  "Statement": [
    {
      "Action": "*",
      "Effect": "Allow",
      "Resource": "*",
      "Principal": "*"
    },
    {
      "Action": "elasticbeanstalk:TerminateEnvironment",
      "Effect": "Deny",
      "Resource": "*",
      "Principal": "*"
    }
  ]
}
```

Note

En este momento, solo el servicio principal de Elastic Beanstalk permite adjuntar una política de punto de enlace a su punto de enlace de la VPC. El servicio de mantenimiento mejorado no admite políticas de punto final.

Configuración de su máquina de desarrollo para su uso con Elastic Beanstalk

Esta página muestra cómo configurar el equipo local para el desarrollo de una aplicación de AWS Elastic Beanstalk. Explica la estructura de carpetas, el control de código fuente y las herramientas de la CLI.

Temas

- [de una carpeta del proyecto](#)
- [Configurar el control de la fuente](#)
- [Configurar un repositorio remoto](#)
- [Instalación de la CLI de EB](#)
- [Instalación de AWS CLI](#)

de una carpeta del proyecto

Cree una carpeta para el proyecto. Puede almacenar la carpeta en cualquier lugar de su disco local siempre y cuando tenga permiso para leer y escribir en ella. Puede crear una carpeta en la carpeta del usuario. Si tiene previsto trabajar en varias aplicaciones, cree las carpetas del proyecto dentro de otra carpeta con un nombre similar a `workspace` o `projects` para mantenerlo todo organizado:

```
workspace/  
|-- my-first-app  
`-- my-second-app
```

El contenido de la carpeta del proyecto variará en función del contenedor o la plataforma web que utilice la aplicación.

Note

Evite usar carpetas y rutas con caracteres de comillas simples (') o comillas dobles (") en el nombre de carpeta o cualquier elemento de la ruta. Algunos comandos de Elastic Beanstalk producen un error cuando se ejecutan en una carpeta con uno de estos caracteres en el nombre.

Configurar el control de la fuente

Configure el control de código fuente para evitar eliminar por error archivos o código del proyecto y para disponer de un medio de revertir los cambios que hacen que su proyecto deje de funcionar.

Si no dispone de un sistema de control de código fuente, considere la posibilidad de usar Git, una opción gratuita y fácil de usar que se integra bien con la interfaz de línea de comandos (CLI) de Elastic Beanstalk. Visite la [página de inicio de Git](#) para instalar Git.

Siga las instrucciones que se detallan en el sitio web de Git para instalar y configurar Git y, a continuación, ejecute `git init` en la carpeta del proyecto para configurar un repositorio local:

```
~/workspace/my-first-app$ git init
Initialized empty Git repository in /home/local/username/workspace/my-first-app/.git/
```

Cuando añada contenido a la carpeta del proyecto y lo actualice, valide los cambios en su repositorio Git:

```
~/workspace/my-first-app$ git add default.jsp
~/workspace/my-first-app$ git commit -m "add default JSP"
```

Cada vez que valida el contenido, crea una snapshot de su proyecto que puede restaurar más adelante si algo va mal. Para obtener más información sobre los comandos y los flujos de trabajo de Git, consulte la [documentación de Git](#).

Configurar un repositorio remoto

¿Qué ocurre si el disco duro se bloquea o desea trabajar en su proyecto en otro equipo? Para realizar un backup del código fuente online y tener acceso a él desde cualquier equipo, configure un repositorio remoto en el que pueda enviar sus validaciones.

AWS CodeCommit permite crear un repositorio privado en la nube de AWS. CodeCommit es gratuito en el [nivel gratuito de AWS](#) para un máximo de cinco usuarios de AWS Identity and Access Management (IAM) en la cuenta. Para obtener información sobre precios, consulte la página [Precios de CodeCommitAWS](#).

Consulte la [Guía del usuario de AWS CodeCommit](#) para obtener instrucciones de configuración.

GitHub es otra opción popular para almacenar el código de su proyecto online. Le permite crear un repositorio público online de forma gratuita y admite también repositorios privados por una cuota mensual. Inscríbese en GitHub en github.com.

Después de crear un repositorio remoto para el proyecto, asócielo a su repositorio local con `git remote add`:

```
~/workspace/my-first-app$ git remote add origin ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/my-repo
```

Instalación de la CLI de EB

Utilice la [CLI de EB](#) para administrar sus entornos de Elastic Beanstalk y monitorizar el estado desde la línea de comandos. Consulte [Instalación de la CLI de EB](#) para obtener instrucciones de instalación.

De forma predeterminada, la CLI de EB empaqueta toda la carpeta del proyecto y la carga en Elastic Beanstalk como un paquete de código fuente. Cuando utiliza Git y la CLI de EB a la vez, puede impedir que los archivos de clases compiladas se validen en el código fuente con `.gitignore` y evitar que los archivos de código fuente se implementen con `.ebignore`.

También puede [configurar la CLI de EB para implementar un artefacto de compilación](#) (un archivo WAR o ZIP) en lugar del contenido de la carpeta del proyecto.

Instalación de AWS CLI

La AWS Command Line Interface (AWS CLI) es un cliente unificado para los servicios de AWS que proporciona comandos para todas las operaciones de las API públicas. Estos comandos son de un nivel inferior a los que proporciona la CLI de EB, por lo que a menudo se necesitan más comandos para realizar una operación con la AWS CLI. Sin embargo, la AWS Command Line Interface le permite trabajar con cualquier aplicación o entorno que se ejecute en su cuenta sin tener que configurar un repositorio en su equipo local. Utilice la AWS CLI para crear scripts que simplifiquen o automaticen las tareas operativas.

Para obtener más información sobre los servicios compatibles y descargar la AWS Command Line Interface, consulte [AWS Command Line Interface](#).

Uso de la interfaz de línea de comandos de Elastic Beanstalk (CLI de EB)

La CLI de EB es una interfaz de línea de comandos AWS Elastic Beanstalk que proporciona comandos interactivos que simplifican la creación, actualización y supervisión de entornos desde un repositorio local. Utilice la CLI de EB en los ciclos de pruebas y desarrollo cotidianos como alternativa a la consola de Elastic Beanstalk .

Note

La versión actual de la CLI de EB tiene un conjunto básico de comandos distinto al de las versiones anteriores a la 3.0. Si utiliza una versión anterior, consulte [Migración a EB CLI 3 y CodeCommit](#) para obtener información de migración.

Después de [instalar la CLI de EB](#) y configurar un directorio de proyecto, puede crear entornos con un solo comando:

```
~/my-app$ eb create my-env
```

El código fuente de la CLI de EB es un proyecto de código abierto. Reside en el [aws/aws-elastic-beanstalk-cli](#) GitHub repositorio. Puede participar informando de problemas, haciendo sugerencias y enviando solicitudes de extracción. ¡Valoramos sus aportaciones! Para un entorno en el que solo desee utilizar la CLI de EB tal como está, le recomendamos que la instale utilizando uno de los scripts de instalación de la CLI de EB, como se detalla en [the section called “Instalación de la CLI de EB con scripts de configuración”](#).

Anteriormente, Elastic Beanstalk utilizaba una CLI distinta que proporcionaba acceso directo a las operaciones de la API y que se llamaba [CLI de la API de Elastic Beanstalk](#). Se ha sustituido por el [AWS CLI](#), que proporciona la misma funcionalidad pero para las API de todos los AWS servicios.

Con él, AWS CLI tiene acceso directo a la API de Elastic Beanstalk. AWS CLI Es ideal para la creación de scripts, pero no es tan fácil de usar desde la línea de comandos debido a la cantidad de comandos que hay que ejecutar y a la cantidad de parámetros de cada comando. Por ejemplo, para crear un entorno, se necesita una importante cantidad de comandos:

```
~$ aws elasticbeanstalk check-dns-availability --cname-prefix my-cname
```



```
~$ aws elasticbeanstalk create-application-version --application-name my-application --  
version-label v1 --source-bundle S3Bucket=DOC-EXAMPLE-BUCKET,S3Key=php-proxy-sample.zip  
~$ aws elasticbeanstalk create-environment --cname-prefix my-cname --application-name  
my-app --version-label v1 --environment-name my-env --solution-stack-name "64bit  
Amazon Linux 2015.03 v2.0.0 running Ruby 2.2 (Passenger Standalone)"
```

Para obtener información sobre la instalación de la CLI de EB, la configuración de un repositorio y el uso de entornos, consulte los siguientes temas.

Temas

- [Instalación de la CLI de EB](#)
- [Configuración de la CLI de EB](#)
- [Administración de entornos de Elastic Beanstalk con la CLI de EB](#)
- [Uso de la CLI de EB con AWS CodeBuild](#)
- [Uso de la CLI de EB con Git](#)
- [Uso de la CLI de EB con AWS CodeCommit](#)
- [Uso de la CLI de EB para monitorear el estado del entorno](#)
- [Administración de múltiples entornos Elastic Beanstalk como un grupo con la CLI de EB](#)
- [Solución de problemas con la CLI de EB](#)
- [Referencia de los comandos de la CLI de EB](#)
- [CLI de EB 2.6 \(retirado\)](#)
- [Interfaz de línea de comandos de Elastic Beanstalk API \(retirada\)](#)

Instalación de la CLI de EB

La interfaz de línea de comandos de AWS Elastic Beanstalk (CLI de EB) es un cliente de línea de comandos que puede utilizar para crear, configurar y administrar entornos de Elastic Beanstalk. Para obtener más información acerca de la CLI de EB, consulte [CLI DE EB](#).

Temas

- [Instalación de la CLI de EB con scripts de configuración](#)
- [Instalación manual de la CLI de EB](#)

Instalación de la CLI de EB con scripts de configuración

La forma más fácil y recomendada de instalar la CLI de EB es utilizar los [scripts de configuración de la CLI de EB](#) disponibles en GitHub. Utilice los scripts para instalar la CLI de EB en Linux, macOS o Windows. Los scripts instalan la CLI de EB y sus dependencias, incluidas Python y pip. Los scripts también crean un entorno virtual para la CLI de EB. Para obtener instrucciones de instalación, consulte el repositorio [aws/aws-elastic-beanstalk-cli-setup](#) en GitHub.

Instalación manual de la CLI de EB

Para instalar la CLI de EB, recomendamos utilizar los [scripts de configuración de la CLI de EB](#). Si los scripts de configuración no son compatibles con su entorno de desarrollo, instale manualmente la CLI de EB.

El método de distribución principal para la CLI de EB en Linux, macOS y Windows es pip. Se trata de un administrador de paquetes para Python que proporciona una manera sencilla de instalar, actualizar y eliminar paquetes de Python y sus dependencias. Para macOS, también puede obtener la última versión de la CLI de EB con Homebrew.

Notas de compatibilidad

La CLI de EB se desarrolló en Python y requiere la versión 3.11 de Python o una posterior.

Recomendamos utilizar los [scripts de configuración de la CLI de EB](#) para instalar la CLI de EB y sus dependencias. Si instala manualmente la CLI de EB, puede resultar difícil administrar los conflictos de dependencia en su entorno de desarrollo.

La CLI de EB y [AWS Command Line Interface](#) (AWS CLI) comparten una dependencia en el paquete Python [botocore](#). Debido a un cambio brusco en botocore, diferentes versiones de estas dos herramientas de interfaz de línea de comandos (CLI) dependen de diferentes versiones de botocore.

Las versiones más recientes de las dos CLI son compatibles. Si necesita utilizar una versión anterior, consulte en la siguiente tabla la versión compatible que se ha de utilizar.

Versión de CLI de EB	AWS CLIVersión compatible
3.14.5 o anterior	1.16.9 o anterior

Versión de CLI de EB	AWS CLIVersión compatible
3.14.6 o posterior	1.16.11 o posterior

Instalación de la CLI de EB

Si ya tiene `pip` y una versión compatible de Python, utilice el siguiente procedimiento para instalar la CLI de EB:

Si no dispone de Python ni `pip`, utilice el procedimiento para el sistema operativo que utiliza.

- [Instalación de Python, pip y la CLI de EB en Linux](#)
- [Instalación de la CLI de EB en macOS](#)
- [Instalación de Python, pip y la CLI de EB en Windows](#)

Para instalar la CLI de EB

1. Ejecute el siguiente comando.

```
$ pip install awsebcli --upgrade --user
```

La opción `--upgrade` indica a `pip` que actualice los requisitos ya instalados. La opción `--user` indica a `pip` que instale el programa en un subdirectorio de su directorio de usuarios para no modificar las bibliotecas que usa su sistema operativo.

Note

Si tiene problemas al intentar instalar la CLI de EB con `pip`, puede [instalar la CLI de EB en un entorno virtual](#) para aislar la herramienta y sus dependencias, o utilizar una versión de Python diferente de la que usa habitualmente.

2. Añada la ruta al archivo ejecutable a su variable `PATH`:

- En Linux y macOS:

Linux: `~/.local/bin`

macOS: `~/Library/Python/3.7/bin`

Para modificar la variable PATH (Linux, Unix o macOS):

- a. Busque el script de perfil de su shell en su carpeta de usuario. Si no está seguro de cuál es el shell que tiene, ejecute `echo $SHELL`.

```
$ ls -a ~
.  ..  .bash_logout  .bash_profile  .bashrc  Desktop  Documents  Downloads
```

- Bash: `.bash_profile`, `.profile` o `.bash_login`.
- Zsh: `.zshrc`
- Tcsh: `.tcshrc`, `.cshrc` o `.login`.

- b. Añada un comando de exportación al script de su perfil. El siguiente ejemplo añade la ruta representada por `LOCAL_PATH` a la variable PATH actual.

```
export PATH=LOCAL_PATH:$PATH
```

- c. Cargue el script del perfil descrito en el primer paso en la sesión actual. El siguiente ejemplo carga el script del perfil representado por `PROFILE_SCRIPT`.

```
$ source ~/PROFILE_SCRIPT
```

- En Windows:

Python 3.7: `%USERPROFILE%\AppData\Roaming\Python\Python37\Scripts`

Versiones anteriores de Python: `%USERPROFILE%\AppData\Roaming\Python\Scripts`

Para modificar la variable PATH (Windows):

- a. Pulse la tecla de Windows y, a continuación, escriba **environment variables**.
- b. Elija Edit environment variables for your account (Editar las variables de entorno de esta cuenta).
- c. Elija PATH y, a continuación, Edit (Editar).
- d. Añada rutas al campo Variable value (Valor de la variable), separadas por punto y coma. Por ejemplo: `C:\item1\path;C:\item2\path`
- e. Elija OK (Aceptar) dos veces para aplicar la nueva configuración.

- f. Cierre las ventanas del símbolo del sistema en ejecución y, a continuación, vuelva a abrir una ventana del símbolo del sistema.
3. Compruebe que la CLI de EB se ha instalado correctamente ejecutando `eb --version`.

```
$ eb --version
EB CLI 3.14.8 (Python 3.7)
```

La CLI de EB se actualiza periódicamente para añadir funciones que admitan [las últimas características de Elastic Beanstalk](#). Para actualizar a la última versión de la CLI de EB, ejecute el comando de instalación de nuevo.

```
$ pip install awsebcli --upgrade --user
```

Si necesita desinstalar la CLI de EB, utilice `pip uninstall`.

```
$ pip uninstall awsebcli
```

Instalación de Python, pip y la CLI de EB en Linux

La CLI de EB requiere Python 2.7, 3.4 o posterior. Si la distribución no incluye Python, o incluye una versión anterior, instale Python antes de instalar `pip` y la CLI de EB.

Para instalar Python 3.7 en Linux

1. Determine si Python ya está instalado.

```
$ python --version
```

Note

Si la distribución de Linux incluye Python, es posible que tenga que instalar el paquete de desarrollador de Python para obtener los encabezados y las bibliotecas necesarios para compilar extensiones e instalar la CLI de EB. Utilice el administrador de paquetes para instalar el paquete de desarrollador (normalmente denominado `python-dev` o `python-devel`).

2. Si no está instalado Python 2.7 o una versión posterior, instale Python 3.7 con el administrador de paquetes de su distribución. El comando y el nombre del paquete varían:

- En derivados de Debian, como por ejemplo Ubuntu, use APT.

```
$ sudo apt-get install python3.7
```

- En Red Hat y sus derivados, use yum.

```
$ sudo yum install python37
```

- En SUSE y sus derivados, use zypper.

```
$ sudo zypper install python3-3.7
```

3. Para verificar que Python está instalado correctamente, abra un terminal o shell y ejecute el siguiente comando.

```
$ python3 --version  
Python 3.7.3
```

Instale pip con el script proporcionado por Python Packaging Authority y después instale la CLI de EB.

Para instalar **pip** y la CLI de EB

1. Descargue el script de instalación desde pypa.io.

```
$ curl -O https://bootstrap.pypa.io/get-pip.py
```

El script descarga e instala la última versión de pip y otro paquete necesario denominado `setuptools`.

2. Ejecute el script con Python.

```
$ python3 get-pip.py --user  
Collecting pip  
  Downloading pip-8.1.2-py2.py3-none-any.whl (1.2MB)  
Collecting setuptools  
  Downloading setuptools-26.1.1-py2.py3-none-any.whl (464kB)
```

```
Collecting wheel
  Downloading wheel-0.29.0-py2.py3-none-any.whl (66kB)
Installing collected packages: pip, setuptools, wheel
Successfully installed pip setuptools wheel
```

Si invoca la versión de Python 3 directamente utilizando el comando `python3` en lugar de `python`, se asegurará de que `pip` se instale en la ubicación adecuada, incluso si hay una versión anterior de Python en el sistema.

3. Añada la ruta del ejecutable, `~/.local/bin`, a la variable `PATH`.

Para modificar la variable `PATH` (Linux, Unix o macOS):

- a. Busque el script de perfil de su shell en su carpeta de usuario. Si no está seguro de cuál es el shell que tiene, ejecute `echo $SHELL`.

```
$ ls -a ~
.  ..  .bash_logout  .bash_profile  .bashrc  Desktop  Documents  Downloads
```

- Bash: `.bash_profile`, `.profile` o `.bash_login`.
 - Zsh: `.zshrc`
 - Tcsh: `.tcshrc`, `.cshrc` o `.login`.
- b. Añada un comando de exportación al script de su perfil. El siguiente ejemplo añade la ruta representada por `LOCAL_PATH` a la variable `PATH` actual.

```
export PATH=LOCAL_PATH:$PATH
```

- c. Cargue el script del perfil descrito en el primer paso en la sesión actual. El siguiente ejemplo carga el script del perfil representado por `PROFILE_SCRIPT`.

```
$ source ~/.PROFILE_SCRIPT
```

4. Compruebe que `pip` está instalado correctamente.

```
$ pip --version
pip 8.1.2 from ~/.local/lib/python3.7/site-packages (python 3.7)
```

5. Use `pip` para instalar la CLI de EB.

```
$ pip install awsebcli --upgrade --user
```

6. Compruebe que la CLI de EB esté instalada correctamente.

```
$ eb --version  
EB CLI 3.14.8 (Python 3.7)
```

Para actualizar a la versión más reciente, ejecute el comando de instalación de nuevo.

```
$ pip install awsebcli --upgrade --user
```

Instalación de la CLI de EB en macOS

Si utiliza el administrador de paquetes brew, puede instalar la CLI de EB con el comando Homebrew. También puede instalar Python y pip, y después utilizar pip para instalar la CLI de EB.

Instalación de la CLI de EB con Homebrew

La última versión de la CLI de EB suele estar disponible en Homebrew un par de días después de que aparezca en pip.

Para instalar la CLI de EB con **Homebrew**

1. Asegúrese de que tiene la última versión de Homebrew.

```
$ brew update
```

2. Ejecute `brew install awsebcli`.

```
$ brew install awsebcli
```

3. Compruebe que la CLI de EB esté instalada correctamente.

```
$ eb --version  
EB CLI 3.14.8 (Python 3.7)
```


Instalación de Python, pip y la CLI de EB en macOS

Puede instalar la última versión de Python y pip y, a continuación, utilizarlos para instalar la CLI de EB.

Para instalar la CLI de EB en macOS

1. Descargue e instale Python de la [página de descargas](#) de [Python.org](#). Utilizamos la versión 3.7 para la demostración.

Note

La CLI de EB requiere Python 2 versión 2.7 o Python 3 en las versiones de 3.4 a 3.7.

2. Instale pip con el script proporcionado por Python Packaging Authority.

```
$ curl -O https://bootstrap.pypa.io/get-pip.py
$ python3 get-pip.py --user
```

3. Use pip para instalar la CLI de EB.

```
$ pip3 install awsebcli --upgrade --user
```

4. Añada la ruta del ejecutable, ~/Library/Python/3.7/bin, a la variable PATH.

Para modificar la variable PATH (Linux, Unix o macOS):

- a. Busque el script de perfil de su shell en su carpeta de usuario. Si no está seguro de cuál es el shell que tiene, ejecute echo \$SHELL.

```
$ ls -a ~
.  ..  .bash_logout  .bash_profile  .bashrc  Desktop  Documents  Downloads
```

- Bash: .bash_profile, .profile o .bash_login.
- Zsh: .zshrc
- Tcsh: .tcshrc, .cshrc o .login.

- b. Añada un comando de exportación al script de su perfil. El siguiente ejemplo añade la ruta representada por *LOCAL_PATH* a la variable PATH actual.

```
export PATH=LOCAL_PATH:$PATH
```

- c. Cargue el script del perfil descrito en el primer paso en la sesión actual. El siguiente ejemplo carga el script del perfil representado por *PROFILE_SCRIPT*.

```
$ source ~/PROFILE_SCRIPT
```

5. Compruebe que la CLI de EB esté instalada correctamente.

```
$ eb --version  
EB CLI 3.14.8 (Python 3.7)
```

Para actualizar a la versión más reciente, ejecute el comando de instalación de nuevo.

```
$ pip3 install awsebcli --upgrade --user
```

Instalación de Python, pip y la CLI de EB en Windows

La Python Software Foundation ofrece instaladores para Windows que incluyen pip.

Para instalar Python 3.6 y **pip** (Windows)

1. Descargue el archivo ejecutable de instalación de la versión más reciente de Python para Windows x86-64 desde la [página de descargas](#) de [Python.org](#).
2. Ejecute el archivo ejecutable de instalación de Python que descargó en el paso anterior.

Seleccione las siguientes opciones en la ventana del instalador de Python para configurar los pasos de instalación de la CLI de EB que se indican a continuación.

- a. Elija agregar el archivo ejecutable de Python a su ruta.
- b. Seleccione Install Now (Instalar ahora).

Note

Para obtener más información sobre las opciones de instalación, consulte la página [Uso de Python en Windows](#) del sitio web de Python.

El sitio web de documentación ofrece un menú desplegable en la parte superior de la página donde puede seleccionar la versión de Python para la documentación.

El instalador instala Python en su carpeta de usuario y añade sus directorios de archivos ejecutables a su ruta de usuario.

Para instalar la AWS CLI con **pip** (Windows)

1. En el menú Inicio, abra una ventana del símbolo del sistema.
2. Compruebe que tanto Python como pip estén instalados correctamente con los siguientes comandos.

```
C:\Users\myname> python --version
Python 3.11.4
C:\Users\myname> pip --version
pip 23.1.2 from C:\Users\myname\AppData\Local\Programs\Python\Python311\Lib\site-packages\pip (python 3.11)
```

3. Instalación de la CLI de EB con pip.

```
C:\Users\myname> pip install awsebcli --upgrade --user
```

4. Agregue la siguiente ruta ejecutable a la variable Path del entorno en su cuenta de usuario de Windows. La ubicación puede ser diferente en función de si instala Python para un usuario o para todos los usuarios.

```
%USERPROFILE%\AppData\Roaming\Python\Python311\Scripts
```

5. Reinicie una shell de comandos nueva para que la variable Path nueva surta efecto.
6. Compruebe que la CLI de EB esté instalada correctamente.

```
C:\Users\myname> eb --version
EB CLI 3.14.8 (Python 3.11)
```

Para actualizar a la versión más reciente, ejecute el comando de instalación de nuevo.

```
C:\Users\myname> pip install awsebcli --upgrade --user
```

Instalación de la CLI de EB en un entorno virtual

Puede evitar conflictos de versiones con otros paquetes `pip` instalando la CLI de EB en un entorno virtual.

Para instalar la CLI de EB en un entorno virtual

1. Instale `virtualenv` con `pip`.

```
$ pip install --user virtualenv
```

2. Cree un entorno virtual.

```
$ virtualenv ~/eb-ve
```

Para utilizar un Python ejecutable diferente al predeterminado, utilice la opción `-p`.

```
$ virtualenv -p /usr/bin/python3.7 ~/eb-ve
```

3. Active el entorno virtual.

Linux, Unix o macOS

```
$ source ~/eb-ve/bin/activate
```

Windows

```
$ %USERPROFILE%\eb-ve\Scripts\activate
```

4. Instale la CLI de EB.

```
(eb-ve)~$ pip install awsebcli --upgrade
```

5. Compruebe que la CLI de EB esté instalada correctamente.

```
$ eb --version  
EB CLI 3.14.8 (Python 3.7)
```

Puede utilizar el comando `deactivate` para salir del entorno virtual. Siempre que inicie una nueva sesión, vuelva a ejecutar el comando de activación.

Para actualizar a la versión más reciente, ejecute el comando de instalación de nuevo.

```
(eb-ve)~$ pip install awsebcli --upgrade
```

Configuración de la CLI de EB

Después de [instalar la CLI de EB](#), puede configurar el directorio del proyecto y la CLI de EB ejecutando `eb init`.

En el siguiente ejemplo, se muestran los pasos de configuración cuando se ejecuta `eb init` por primera vez en una carpeta de proyecto llamada `eb`.

Para inicializar un proyecto de la CLI de EB

1. En primer lugar, la CLI de EB le pide que seleccione una región. Escriba el número correspondiente a la región que desea utilizar y, a continuación, pulse Intro.

```
~/eb $ eb init
Select a default region
1) us-east-1 : US East (N. Virginia)
2) us-west-1 : US West (N. California)
3) us-west-2 : US West (Oregon)
4) eu-west-1 : Europe (Ireland)
5) eu-central-1 : Europe (Frankfurt)
6) ap-south-1 : Asia Pacific (Mumbai)
7) ap-southeast-1 : Asia Pacific (Singapore)
...
(default is 3): 3
```

2. A continuación, proporcione la clave de acceso y la clave secreta para que la CLI de EB pueda administrar los recursos por usted. Las claves de acceso se crean en la consola de AWS Identity and Access Management. Si no tiene claves, consulte [¿Cómo obtengo credenciales de seguridad?](#) en la Referencia general de Amazon Web Services.

```
You have not yet set up your credentials or your credentials are incorrect.
You must provide your credentials.
(aws-access-id): AKIAJOUAASEXAMPLE
```

```
(aws-secret-key): 5ZRIrtTM4ciIAvd4EXAMPLEDtm+PiPSzpoK
```

- Una aplicación de Elastic Beanstalk es un recurso que contiene un conjunto de versiones de la aplicación (código fuente), entornos y configuraciones guardadas y que está asociado con una única aplicación web. Cada vez que implementa el código fuente en Elastic Beanstalk con la CLI de EB, se crea una nueva versión de la aplicación y se agrega a la lista.

```
Select an application to use
1) [ Create new Application ]
(default is 1): 1
```

- El nombre de aplicación predeterminada es el nombre de la carpeta en la que se ejecuta `eb init`. Especifique un nombre que describa el proyecto.

```
Enter Application Name
(default is "eb"): eb
Application eb has been created.
```

- Seleccione una plataforma que coincida con el lenguaje o el marco en el que se desarrolló la aplicación web. Si todavía no ha empezado a desarrollar la aplicación, seleccione la plataforma que le interese. Pronto aprenderá a lanzar una aplicación de ejemplo y siempre puede cambiar este ajuste con posterioridad.

```
Select a platform.
1) Node.js
2) PHP
3) Python
4) Ruby
5) Tomcat
6) IIS
7) Docker
8) Multi-container Docker
9) GlassFish
10) Go
11) Java
(default is 1): 1
```

- Elija **yes** (sí) para asignar un par de claves SSH a las instancias del entorno de Elastic Beanstalk. Esto le permite conectarse directamente a ellas para la solución de problemas.

```
Do you want to set up SSH for your instances?
```

```
(y/n): y
```

7. Elija un par de claves existente o cree uno nuevo. Si desea utilizar `eb init` para crear un nuevo par de claves, `ssh-keygen` debe estar instalado en la máquina local y disponible desde la línea de comando. La CLI de EB registra automáticamente el nuevo par de claves en Amazon EC2 y almacena la clave privada localmente en una carpeta denominada `.ssh` en el directorio del usuario.

```
Select a keypair.  
1) [ Create new KeyPair ]  
(default is 1): 1
```

La instalación de la CLI de EB ya está configurada y lista para utilizarse. Consulte [Administración de entornos de Elastic Beanstalk con la CLI de EB](#) para obtener instrucciones sobre cómo crear y trabajar con un entorno de Elastic Beanstalk.

Configuración avanzada

- [Ignorar archivos usando .ebignore](#)
- [Usar perfiles con nombre](#)
- [Implementar un artefacto en lugar de la carpeta del proyecto](#)
- [Ajustes de configuración y precedencia](#)
- [Metadatos de instancia](#)

Ignorar archivos usando .ebignore

Puede indicar a la CLI de EB que omita determinados archivos del directorio del proyecto añadiendo el archivo `.ebignore` al directorio. Este archivo funciona como un archivo `.gitignore`. Cuando se implementa el directorio del proyecto en Elastic Beanstalk y se crea una nueva versión de la aplicación, la CLI de EB no incluye los archivos especificados por `.ebignore` en el paquete de código fuente que crea.

Si no existe `.ebignore`, pero sí existe `.gitignore`, la CLI de EB omite los archivos especificados en `.gitignore`. Si existe `.ebignore`, la CLI de EB no lee `.gitignore`.

Cuando `.ebignore` está presente, la CLI de EB no utiliza comandos `git` para crear su paquete de código fuente. Esto significa que la CLI de EB omite los archivos especificados en `.ebignore` e incluye todos los demás archivos. En concreto, incluye archivos de código fuente sin validar.

Note

En Windows, si se agrega un archivo `.ebignore`, la CLI de EB sigue enlaces simbólicos e incluye el archivo vinculado al crear un paquete de código fuente. Es un error conocido que se resolverá en una futura actualización.

Usar perfiles con nombre

Si guarda las credenciales como un perfil con nombre en un archivo `credentials` o `config`, puede utilizar la opción `--profile` para especificar explícitamente un perfil. Por ejemplo, el comando siguiente crea una nueva aplicación con el perfil `user2`.

```
$ eb init --profile user2
```

También puede cambiar el perfil predeterminado configurando la variable de entorno `AWS_EB_PROFILE`. Cuando esta variable está configurada, la CLI de EB lee las credenciales del perfil especificado en lugar de `default` o `eb-cli`.

Linux, macOS o Unix

```
$ export AWS_EB_PROFILE=user2
```

Windows

```
> set AWS_EB_PROFILE=user2
```

Implementar un artefacto en lugar de la carpeta del proyecto

Puede indicar a la CLI de EB que implemente un archivo ZIP o WAR que usted genere como parte de un proceso de compilación diferente añadiendo las siguientes líneas a `.elasticbeanstalk/config.yml` en la carpeta del proyecto.

```
deploy:  
  artifact: path/to/buildartifact.zip
```

Si configura la CLI de EB en el [repositorio de Git](#) y no confirma el artefacto en el código fuente, use la opción `--staged` para implementar la última compilación.


```
~/eb$ eb deploy --staged
```

Ajustes de configuración y precedencia

La CLI de EB usa una cadena de provisión para buscar las credenciales de AWS en diversos lugares; por ejemplo, en las variables de entorno del sistema o del usuario y en los archivos de configuración de AWS.

La CLI de EB busca las credenciales y las opciones de configuración en el siguiente orden:

1. Opciones de la línea de comandos: especifique un perfil con nombre mediante `--profile` para anular la configuración predeterminada.
2. Variables de entorno: `AWS_ACCESS_KEY_ID` y `AWS_SECRET_ACCESS_KEY`.
3. Archivo de credenciales de AWS: está situado en `~/.aws/credentials` en los sistemas Linux y OS X, o en `C:\Users\USERNAME\.aws\credentials` en los sistemas Windows. Este archivo puede contener varios perfiles con nombre, además de un perfil predeterminado.
4. El [archivo de configuración de AWS CLI](#): está situado en `~/.aws/config` en los sistemas Linux y OS X, o en `C:\Users\USERNAME\.aws\config` en los sistemas Windows. Este archivo puede contener un perfil predeterminado, [perfiles con nombre](#) y parámetros de configuración específicos de AWS CLI para cada uno.
5. Archivo de configuración heredado de la CLI de EB: está situado en `~/.elasticbeanstalk/config` en los sistemas Linux y OS X, o en `C:\Users\USERNAME\.elasticbeanstalk\config` en los sistemas Windows.
6. Credenciales de perfil de instancia: estas credenciales pueden utilizarse en instancias Amazon EC2 con un rol de instancia asignado y se entregan a través del servicio de metadatos de Amazon EC2. El [perfil de instancia](#) debe tener permiso para usar Elastic Beanstalk.

Si el archivo de credenciales contiene un perfil con el nombre "eb-cli", la CLI de EB preferirá ese perfil frente al predeterminado. Si no se encuentran perfiles, o se encuentra un perfil pero no tiene permiso para utilizar Elastic Beanstalk, la CLI de EB le pedirá uno cuando especifique las claves.

Metadatos de instancia

Para utilizar la CLI de EB desde una instancia de Amazon EC2, cree un rol que tenga acceso a los recursos necesarios y asigne ese rol a la instancia cuando se lance. Lance la instancia e instale la CLI de EB mediante `pip`.

```
~$ sudo pip install awsebcli
```

pip viene preinstalado en Amazon Linux.

La CLI de EB lee las credenciales desde los metadatos de la instancia. Para obtener más información, consulte [Permitir el acceso a los recursos de AWS a aplicaciones que se ejecutan en instancias de Amazon EC2](#) en la Guía del usuario de IAM.

Administración de entornos de Elastic Beanstalk con la CLI de EB

Después de [instalar la CLI de EB](#) y [configurar el directorio del proyecto](#), estará listo para crear un entorno de Elastic Beanstalk mediante la CLI de EB, para implementar el código fuente y las actualizaciones de configuración y para obtener registros y eventos.

Note

Para crear entornos con la CLI de EB, se requiere una [función de servicio](#). Puede crear una función de servicio creando un entorno en la consola de Elastic Beanstalk. Si no cuenta con ninguna función de servicio, el EB CLI intentará crear uno cuando ejecute `eb create`.

La CLI de EB devuelve un código de salida cero (0) para todos los comandos correctos y un código de salida distinto de cero cuando encuentra un error.

Los siguientes ejemplos utilizan una carpeta del proyecto vacía llamada `eb` que se ha inicializado con la CLI de EB para usarla con una aplicación Docker de ejemplo.

Comandos básicos

- [Eb create](#)
- [Eb status](#)
- [Eb health](#)
- [Eb events](#)
- [Eb logs](#)
- [Eb open](#)
- [Eb deploy](#)

- [Eb config](#)
- [Eb terminate](#)

Eb create

Para crear su primer entorno, ejecute [eb create](#) y siga las instrucciones. Si el directorio del proyecto contiene código fuente, la CLI de EB lo empaquetará e implementará en su entorno. De lo contrario, se usará una aplicación de ejemplo.

```
~/eb$ eb create
Enter Environment Name
(default is eb-dev): eb-dev
Enter DNS CNAME prefix
(default is eb-dev): eb-dev
WARNING: The current directory does not contain any source code. Elastic Beanstalk is
launching the sample application instead.
Environment details for: elasticBeanstalkExa-env
  Application name: elastic-beanstalk-example
  Region: us-west-2
  Deployed Version: Sample Application
  Environment ID: e-j3pmc8tscn
  Platform: 64bit Amazon Linux 2015.03 v1.4.3 running Docker 1.6.2
  Tier: WebServer-Standard
  CNAME: eb-dev.elasticbeanstalk.com
  Updated: 2015-06-27 01:02:24.813000+00:00
Printing Status:
INFO: createEnvironment is starting.
-- Events -- (safe to Ctrl+C) Use "eb abort" to cancel the command.
```

Pueden transcurrir varios minutos hasta que su entorno esté preparado. Presione Ctrl+C para volver a la línea de comandos mientras se crea el entorno.

Eb status

Ejecute `eb status` para ver el estado actual de su entorno. Cuando el estado es `ready`, la aplicación de ejemplo está disponible en `elasticbeanstalk.com` y el entorno está listo para su actualización.

```
~/eb$ eb status
Environment details for: elasticBeanstalkExa-env
  Application name: elastic-beanstalk-example
```

```

Region: us-west-2
Deployed Version: Sample Application
Environment ID: e-gbzqc3jcra
Platform: 64bit Amazon Linux 2015.03 v1.4.3 running Docker 1.6.2
Tier: WebServer-Standard
CNAME: elasticbeanstalkexa-env.elasticbeanstalk.com
Updated: 2015-06-30 01:47:45.589000+00:00
Status: Ready
Health: Green

```

Eb health

Utilice el comando `eb health` para ver [información de estado](#) de las instancias de su entorno y el estado general del entorno. Utilice la opción `--refresh` para ver el estado en una vista interactiva que se actualiza cada 10 segundos.

```

~/eb$ eb health
api                               Ok                               2016-09-15 18:39:04
WebServer                          Java 8
total      ok      warning  degraded  severe  info  pending  unknown
  3         3         0         0         0         0         0         0

instance-id      status      cause      health
Overall          Ok
i-0ef05ec54918bf567  Ok
i-001880c1187493460  Ok
i-04703409d90d7c353  Ok

instance-id      r/sec      %2xx      %3xx      %4xx      %5xx      p99      p90      p75
p50      p10
Overall          8.6      100.0      0.0      0.0      0.0      0.083*  0.065  0.053
0.040  0.019
i-0ef05ec54918bf567  2.9      29      0      0      0      0.069*  0.066  0.057
0.050  0.023
i-001880c1187493460  2.9      29      0      0      0      0.087*  0.069  0.056
0.050  0.034
i-04703409d90d7c353  2.8      28      0      0      0      0.051*  0.027  0.024
0.021  0.015

instance-id      type      az      running      load 1  load 5      user%  nice%
system%  idle%  iowait%
i-0ef05ec54918bf567  t2.micro  1c      23 mins      0.19  0.05      3.0  0.0
0.3  96.7  0.0

```

```

i-001880c1187493460  t2.micro  1a  23 mins  0.0  0.0  3.2  0.0
0.3  96.5  0.0
i-04703409d90d7c353  t2.micro  1b  1 day  0.0  0.0  3.6  0.0
0.2  96.2  0.0

instance-id      status      id  version      ago
deployments
i-0ef05ec54918bf567  Deployed  28  app-bc1b-160915_181041  20 mins
i-001880c1187493460  Deployed  28  app-bc1b-160915_181041  20 mins
i-04703409d90d7c353  Deployed  28  app-bc1b-160915_181041  27 mins

```

Eb events

Utilice `eb events` para ver una lista de eventos generados por Elastic Beanstalk.

```

~/eb$ eb events
2015-06-29 23:21:09 INFO createEnvironment is starting.
2015-06-29 23:21:10 INFO Using elasticbeanstalk-us-east-2-EXAMPLE as Amazon S3
storage bucket for environment data.
2015-06-29 23:21:23 INFO Created load balancer named: awseb-e-g-AWSEBLoa-EXAMPLE
2015-06-29 23:21:42 INFO Created security group named: awseb-e-gbzqc3jcra-stack-
AWSEBSecurityGroup-EXAMPLE
...

```

Eb logs

Utilice `eb logs` para obtener los registros de una instancia de su entorno. De forma predeterminada, `eb logs` obtiene los registros de la primera instancia lanzada y los muestra en la salida estándar. Puede especificar un ID de instancia con la opción `--instance` para obtener los registros de una instancia específica.

La opción `--all` obtiene los registros de todas las instancias y los guarda en subdirectorios bajo `.elasticbeanstalk/logs`.

```

~/eb$ eb logs --all
Retrieving logs...
Logs were saved to /home/local/ANT/mwunderl/ebcli/environments/test/.elasticbeanstalk/
logs/150630_201410
Updated symlink at /home/local/ANT/mwunderl/ebcli/environments/test/.elasticbeanstalk/
logs/latest

```

Eb open

Para abrir el sitio web de su entorno en un navegador, utilice `eb open`:

```
~/eb$ eb open
```

En un entorno con ventanas, su navegador predeterminado se abrirá en una nueva ventana. En un entorno de terminal, se usará un explorador de línea de comandos (como `w3m`) si está disponible.

Eb deploy

Una vez que el entorno esté listo, puede actualizarlo mediante `eb deploy`.

Este comando funciona mejor con código fuente que se vaya a empaquetar e implementar, así que para este ejemplo hemos creado un `Dockerfile` en el directorio del proyecto con el siguiente contenido:

```
~/eb/Dockerfile
```

```
FROM ubuntu:12.04

RUN apt-get update
RUN apt-get install -y nginx zip curl

RUN echo "daemon off;" >> /etc/nginx/nginx.conf
RUN curl -o /usr/share/nginx/www/master.zip -L https://codeload.github.com/gabrielecirulli/2048/zip/master
RUN cd /usr/share/nginx/www/ && unzip master.zip && mv 2048-master/* . && rm -rf 2048-master master.zip

EXPOSE 80

CMD ["/usr/sbin/nginx", "-c", "/etc/nginx/nginx.conf"]
```

Este `Dockerfile` implementa una imagen de Ubuntu 12.04 e instala el juego 2048. Ejecute `eb deploy` para cargar la aplicación en su entorno:

```
~/eb$ eb deploy
Creating application version archive "app-150630_014338".
Uploading elastic-beanstalk-example/app-150630_014338.zip to S3. This may take a while.
Upload Complete.
```

```
INFO: Environment update is starting.  
-- Events -- (safe to Ctrl+C) Use "eb abort" to cancel the command.
```

Cuando ejecuta `eb deploy`, la CLI de EB empaqueta el contenido del directorio del proyecto y lo implementa en su entorno.

Note

Si ha inicializado un repositorio Git en la carpeta del proyecto, la CLI de EB siempre implementará la última validación, aunque tenga cambios pendientes. Valide los cambios antes de ejecutar `eb deploy` para implementarlos en su entorno.

Eb config

Eche un vistazo a las opciones de configuración disponibles para su entorno en ejecución con el comando `eb config`:

```
~/eb$ eb config  
ApplicationName: elastic-beanstalk-example  
DateUpdated: 2015-06-30 02:12:03+00:00  
EnvironmentName: elasticBeanstalkExa-env  
SolutionStackName: 64bit Amazon Linux 2015.03 v1.4.3 running Docker 1.6.2  
settings:  
  AWSEBAutoScalingScaleDownPolicy.aws:autoscaling:trigger:  
    LowerBreachScaleIncrement: '-1'  
  AWSEBAutoScalingScaleUpPolicy.aws:autoscaling:trigger:  
    UpperBreachScaleIncrement: '1'  
  AWSEBCloudwatchAlarmHigh.aws:autoscaling:trigger:  
    UpperThreshold: '6000000'  
  ...
```

Este comando rellena una lista de opciones de configuración disponibles en un editor de texto. Muchas de las opciones que se muestran tienen un valor `null`; estas opciones no están configuradas de forma predeterminada, pero pueden modificarse para actualizar los recursos de su entorno. Para obtener más información sobre estas opciones, consulte [Opciones de configuración](#).

Eb terminate

Si ha terminado de usar el entorno de momento, use `eb terminate` para terminarlo.

```
~/eb$ eb terminate
```

```
The environment "eb-dev" and all associated instances will be terminated.
```

```
To confirm, type the environment name: eb-dev
```

```
INFO: terminateEnvironment is starting.
```

```
INFO: Deleted CloudWatch alarm named: awseb-e-jc8t3pmscn-stack-
```

```
AWSEBCloudwatchAlarmHigh-1XLMU7DNCBV6Y
```

```
INFO: Deleted CloudWatch alarm named: awseb-e-jc8t3pmscn-stack-
```

```
AWSEBCloudwatchAlarmLow-8IVIO4W2SCXS
```

```
INFO: Deleted Auto Scaling group policy named: arn:aws:autoscaling:us-
```

```
east-2:123456789012:scalingPolicy:1753d43e-ae87-4df6-
```

```
a405-11d31f4c8f97:autoScalingGroupName/awseb-e-jc8t3pmscn-stack-
```

```
AWSEBAutoScalingGroup-90TTS2ZL4MXV:policyName/awseb-e-jc8t3pmscn-stack-
```

```
AWSEBAutoScalingScaleUpPolicy-A070H1BMUQAJ
```

```
INFO: Deleted Auto Scaling group policy named: arn:aws:autoscaling:us-
```

```
east-2:123456789012:scalingPolicy:1fd24ea4-3d6f-4373-
```

```
affc-4912012092ba:autoScalingGroupName/awseb-e-jc8t3pmscn-stack-
```

```
AWSEBAutoScalingGroup-90TTS2ZL4MXV:policyName/awseb-e-jc8t3pmscn-stack-
```

```
AWSEBAutoScalingScaleDownPolicy-LSWFUMZ46H1V
```

```
INFO: Waiting for EC2 instances to terminate. This may take a few minutes.
```

```
-- Events -- (safe to Ctrl+C)
```

Para obtener una lista completa de comandos de la CLI de EB disponibles, consulte la [Referencia de los comandos de la CLI de EB](#).

Important

Si termina un entorno, también debe eliminar todas las asignaciones de CNAME que ha creado, ya que otros clientes pueden reutilizar un nombre de host disponible. Asegúrese de eliminar los registros de DNS que apuntan a su entorno terminado para evitar que se cree una entrada de DNS colgante. Una entrada de DNS colgante puede exponer el tráfico de Internet destinado a su dominio a vulnerabilidades de seguridad. También puede presentar otros riesgos.

Para más información, consulte [Protección contra registros de delegación colgantes en Route 53](#) en la Guía para desarrolladores de Amazon Route 53. También puede obtener más información sobre las entradas de DNS colgantes en [Protecciones de dominio mejoradas para solicitudes de Amazon CloudFront](#) en el Blog de seguridad de AWS.

Uso de la CLI de EB con AWS CodeBuild

[AWS CodeBuild](#) compila el código fuente, ejecuta pruebas unitarias y produce artefactos listos para implementarse. Puede utilizar CodeBuild junto con la CLI de EB para automatizar la creación de la aplicación a partir del código fuente. La creación del entorno y cada implementación posterior comienzan por un paso de creación y la posterior implementación de la aplicación resultante.

Note

Algunas regiones no ofrecen CodeBuild. La integración entre Elastic Beanstalk y CodeBuild no funciona en estas regiones.

Para obtener información sobre los servicios de AWS que se ofrecen en cada región, consulte la [Tabla de regiones](#).

Creación de una aplicación

Para crear una aplicación de Elastic Beanstalk que utilice CodeBuild

1. Incluya un archivo de especificación de compilación de CodeBuild, [buildspec.yml](#), en la carpeta de la aplicación.
2. Añada una entrada `eb_codebuild_settings` con opciones específicas de Elastic Beanstalk al archivo.
3. Ejecute [eb init](#) en la carpeta.

Note

No utilice los caracteres de punto (.) ni espacio () en Application name (Nombre de la aplicación) cuando utilice la CLI de EB con CodeBuild.

Elastic Beanstalk amplía el [formato de archivo de especificación de compilación de CodeBuild](#) para que incluya las siguientes opciones de configuración adicionales:

```
eb_codebuild_settings:
  CodeBuildServiceRole: role-name
  ComputeType: size
  Image: image
```

Timeout: *minutes*

CodeBuildServiceRole

El ARN o nombre del rol de servicio de AWS Identity and Access Management (IAM) que CodeBuild puede utilizar para interactuar con los servicios de AWS dependientes en su nombre. Este valor es obligatorio. Si lo omite, cualquier comando `eb create` o `eb deploy` posterior produce un error.

Para obtener más información sobre cómo crear un rol de servicio para CodeBuild, consulte [Crear un rol de servicio de CodeBuild](#) en la AWS CodeBuild Guía del usuario.

Note

También necesita permisos para realizar acciones en el propio CodeBuild. La política de usuario administrada de Elastic Beanstalk `AdministratorAccess-AWSElasticBeanstalk` incluye todos los permisos de acción de CodeBuild necesarios. Si no utiliza la política administrada, asegúrese de conceder los siguientes permisos en su política de usuario.

```
"codebuild:CreateProject",  
"codebuild>DeleteProject",  
"codebuild:BatchGetBuilds",  
"codebuild:StartBuild"
```

Para obtener más información, consulte [Administración de políticas de usuario de Elastic Beanstalk](#).

ComputeType

La cantidad de recursos que utiliza el contenedor Docker en el entorno de compilación de CodeBuild. Los valores válidos son `BUILD_GENERAL1_SMALL`, `BUILD_GENERAL1_MEDIUM` y `BUILD_GENERAL1_LARGE`.

Image

Nombre de la imagen de Docker Hub o de Amazon ECR que CodeBuild utiliza para el entorno de compilación. Esta imagen de Docker debe contener todas las herramientas y bibliotecas de tiempo de ejecución necesarias para construir su código y debe coincidir con la plataforma de destino de su aplicación. CodeBuild administra y mantiene un conjunto de imágenes

específicamente destinadas utilizarse con Elastic Beanstalk. Se recomienda utilizar una de ellas. Para obtener más información, consulte [Imágenes de Docker proporcionadas por CodeBuild](#) en la AWS CodeBuild Guía del usuario.

El valor `Image` es opcional. Si lo omite, el comando `eb init` intenta elegir la imagen que mejor se adapte a su plataforma de destino. Además, si ejecuta `eb init` en modo interactivo y no puede elegir la imagen automáticamente, le solicita que seleccione una. Al final de una inicialización correcta, `eb init` escribe la imagen elegida en el archivo `buildspec.yml`.

Timeout

Periodo en minutos durante el que la compilación de CodeBuild se ejecuta antes de agotar el tiempo de espera. Este valor es opcional. Para obtener más información acerca de los valores válidos y predeterminados, consulte [Create a Build Project in CodeBuild \(Creación de un proyecto de compilación en CodeBuild\)](#).

Note

Este tiempo de espera controla la duración máxima de una ejecución de CodeBuild y la CLI de EB lo respeta también como parte de este primer paso para crear una versión de la aplicación. Es distinto del valor que puede especificar con la opción `--timeout` de los comandos [eb create](#) o [eb deploy](#). El último valor controla la duración máxima que la CLI de EB debe esperar para la creación o la actualización del entorno.

Creación e implementación del código de la aplicación

Cada vez que es necesario implementar el código de la aplicación, la CLI de EB utiliza CodeBuild para ejecutar una compilación y, después, implementa los artefactos de la compilación resultantes en el entorno. Esto ocurre cuando crea un entorno de Elastic Beanstalk para su aplicación con el comando [eb create](#) y cada vez que implemente cambios de código en el entorno posteriormente con el comando [eb deploy](#).

Si se produce un error en el paso de CodeBuild, no se inicia la creación o la implementación del entorno.

Uso de la CLI de EB con Git

La CLI de EB proporciona integración con Git. En esta sección se proporciona información general sobre cómo utilizar Git con la CLI de EB.

Para instalar Git e inicializar el repositorio Git

1. Descargue la versión más reciente de Git en <http://git-scm.com>
2. Inicialice el repositorio Git escribiendo lo siguiente:

```
~/eb$ git init
```

La CLI de EB reconocerá ahora que su aplicación está configurada con Git.

3. Si aún no ha ejecutado `eb init`, hágalo ahora:

```
~/eb$ eb init
```

Asociación de entornos Elastic Beanstalk con ramas de Git

Puede asociar un entorno diferente a cada ramificación de su código. Al desproteger una ramificación, los cambios se implementan en el entorno asociado. Por ejemplo, puede escribir lo siguiente para asociar su entorno de producción con su ramificación principal y un entorno de desarrollo independiente con su ramificación de desarrollo:

```
~/eb$ git checkout mainline
~/eb$ eb use prod
~/eb$ git checkout develop
~/eb$ eb use dev
```

Implementación de cambios

De forma predeterminada, la CLI de EB implementa la última validación de la ramificación actual, usando el ID y el mensaje de validación como la etiqueta y la descripción de la versión de compilación, respectivamente. Si desea implementar los cambios en su entorno sin validación, puede utilizar la opción `--staged` para implementar los cambios que se han añadido al área de ensayo.

Para implementar los cambios sin validación

1. Añada archivos nuevos y modificados al área de ensayo:

```
~/eb$ git add .
```

2. Implemente los cambios almacenados con `eb deploy`:

```
~/eb$ eb deploy --staged
```

Si ha configurado la CLI de EB para [implementar un artefacto](#) y no valida el artefacto en su repositorio Git, use la opción `--staged` para implementar la última compilación.

Uso de submódulos de Git

Algunos proyectos de código se benefician de tener submódulos de Git, es decir, repositorios en el repositorio de nivel superior. Al implementar el código mediante `eb create` o `eb deploy`, la CLI de EB puede incluir submódulos en el archivo zip de la versión de la aplicación y cargarlos con el resto del código.

Puede controlar la inclusión de submódulos mediante la `include_git_submodules` opción en la sección `global` de la CLI del archivo de configuración de la CLI de EB, `.elasticbeanstalk/config.yml` en la carpeta de su proyecto.

Para incluir submódulos, configure esta opción en `true`:

```
global:
  include_git_submodules: true
```

Si falta la opción `include_git_submodules` o se ha configurado en `false`, la CLI de EB no incluye submódulos en el archivo zip cargado.

Consulte [Herramientas de Git: submódulos](#) para obtener más información acerca de los submódulos de Git.

Comportamiento predeterminado

Al ejecutar `eb init` para configurar el proyecto, la CLI de EB agrega la opción `include_git_submodules` y la configura en `true`. Esto garantiza que los submódulos que tenga en el proyecto se incluyen en las implementaciones.

La CLI de EB CLI no siempre admite la inclusión de submódulos. Para evitar un cambio accidental e indeseable en los proyectos que hubieran existido antes de que agregáramos la compatibilidad con submódulos, la CLI de EB no incluye submódulos si falta la opción `include_git_submodules`. Si tiene uno de estos proyectos existentes y desea incluir submódulos en las implementaciones, agregue la opción y configúrela en `true`, tal como se explica en esta sección.

Comportamiento de CodeCommit

La integración de Elastic Beanstalk en [CodeCommit](#) no admite submódulos en este momento. Si ha habilitado su entorno para que se integre con CodeCommit, los submódulos no se incluyen en las implementaciones.

Asignación de etiquetas de Git a la versión de la aplicación

Puede utilizar una etiqueta de Git como su etiqueta de versión para identificar qué versión de la aplicación se ejecuta en su entorno. Por ejemplo, escriba lo siguiente:

```
~/eb$ git tag -a v1.0 -m "My version 1.0"
```

Uso de la CLI de EB con AWS CodeCommit

Puede utilizar la CLI de EB para implementar la aplicación directamente desde el repositorio de AWS CodeCommit. Con CodeCommit, puede cargar únicamente los cambios del repositorio en la implementación, en lugar de cargar todo el proyecto. De este modo, ahorrará tiempo y ancho de banda si tiene un proyecto de gran tamaño o una conectividad a Internet limitada. La CLI de EB inserta las confirmaciones locales y las utiliza para crear versiones de la aplicación cuando se utiliza `eb appversion`, `eb create` o `eb deploy`.

Para poder implementar los cambios, la integración con CodeCommit requiere que confirme primero los cambios. Sin embargo, a medida que realice el desarrollo o la depuración, es posible que no quiera insertar los cambios que aún no ha confirmado. Puede evitar confirmar los cambios almacenándolos provisionalmente y utilizando `eb deploy --staged` (que realiza una implementación estándar). O puede confirmar los cambios primero en una ramificación de desarrollo o prueba

y fusionarlos después en una ramificación principal cuando el código esté listo. Con `eb use`, puede configurar la CLI de EB para realizar la implementación en un entorno de la ramificación de desarrollo y en otro de la ramificación principal.

Note

Algunas regiones no ofrecen CodeCommit. La integración entre Elastic Beanstalk y CodeCommit no funciona en estas regiones.

Para obtener información sobre los servicios de AWS que se ofrecen en cada región, consulte la [Tabla de regiones](#).

Secciones

- [Requisitos previos](#)
- [Crear un repositorio de CodeCommit con la CLI de EB](#)
- [Implementación desde su repositorio CodeCommit](#)
- [Configuración de entornos y ramificaciones adicionales](#)
- [Usar un repositorio de CodeCommit existente](#)

Requisitos previos

Para utilizar CodeCommit con AWS Elastic Beanstalk, necesita un repositorio Git local (ya sea uno existente o uno nuevo que haya creado) con al menos una confirmación, [permiso para usar CodeCommit](#), y un entorno de Elastic Beanstalk en una región compatible con CodeCommit. El entorno y el repositorio deben estar en la misma región.

Para inicializar el repositorio de Git

1. Ejecute `git init` en la carpeta del proyecto.

```
~/my-app$ git init
```

2. Almacene provisionalmente los archivos del proyecto con `git add`.

```
~/my-app$ git add .
```

3. Confirme los cambios con `git commit`.

```
~/my-app$ git commit -m "Elastic Beanstalk application"
```

Crear un repositorio de CodeCommit con la CLI de EB

Para comenzar a utilizar CodeCommit, ejecute [eb init](#). Durante la configuración del repositorio, la CLI de EB le pide que utilice CodeCommit para almacenar el código y acelerar las implementaciones. Aunque anteriormente configurara el proyecto con `eb init`, puede ejecutarlo de nuevo para configurar CodeCommit.

Para crear un repositorio CodeCommit con la CLI de EB

1. Ejecute `eb init` en la carpeta del proyecto. Durante la configuración, la CLI de EB le pregunta si desea usar CodeCommit para almacenar el código y acelerar las implementaciones. Aunque anteriormente configurara el proyecto con `eb init`, puede volver a ejecutarlo de nuevo para configurar CodeCommit. Escriba **y** en el indicador para configurar CodeCommit.

```
~/my-app$ eb init
Note: Elastic Beanstalk now supports AWS CodeCommit; a fully-managed source control
  service. To learn more, see Docs: https://aws.amazon.com/codecommit/
Do you wish to continue with CodeCommit? (y/n)(default is n): y
```

2. Seleccione `Create new Repository` (Crear nuevo repositorio).

```
Select a repository
1) my-repo
2) [ Create new Repository ]
(default is 2): 2
```

3. Escriba un nombre de repositorio o pulse Intro para aceptar el nombre predeterminado.

```
Enter Repository Name
(default is "codecommit-origin"): my-app
Successfully created repository: my-app
```

4. Seleccione una ramificación existente para las confirmaciones o utilice la CLI de EB para crear una nueva ramificación.

```
Enter Branch Name
***** Must have at least one commit to create a new branch with CodeCommit *****
```



```
(default is "mainline"): ENTER  
Successfully created branch: mainline
```

Implementación desde su repositorio CodeCommit

Cuando configura CodeCommit con el repositorio de la CLI de EB, la CLI de EB utiliza el contenido del repositorio para crear paquetes de código fuente. Cuando ejecuta `eb deploy` o `eb create`, la CLI de EB inserta nuevas confirmaciones y utiliza la revisión HEAD de la ramificación para crear el archivo que se implementa en las instancias EC2 del entorno.

Para utilizar la integración de CodeCommit con la CLI de EB

1. Cree un nuevo entorno con `eb create`.

```
~/my-app$ eb create my-app-env  
Starting environment deployment via CodeCommit  
--- Waiting for application versions to be pre-processed ---  
Finished processing application version app-ac1ea-161010_201918  
Setting up default branch  
Environment details for: my-app-env  
  Application name: my-app  
  Region: us-east-2  
  Deployed Version: app-ac1ea-161010_201918  
  Environment ID: e-pm5mvvkfnd  
  Platform: 64bit Amazon Linux 2016.03 v2.1.6 running Java 8  
  Tier: WebServer-Standard  
  CNAME: UNKNOWN  
  Updated: 2016-10-10 20:20:29.725000+00:00  
Printing Status:  
INFO: createEnvironment is starting.  
...
```

La CLI de EB utiliza la última confirmación de la ramificación monitorizada para crear la versión de la aplicación que se va a implementar en el entorno.

2. Si hay nuevas confirmaciones locales, utilice `eb deploy` para insertar las confirmaciones e implementarlas en el entorno.

```
~/my-app$ eb deploy  
Starting environment deployment via CodeCommit  
INFO: Environment update is starting.
```

```
INFO: Deploying new version to instance(s).
INFO: New application version was deployed to running EC2 instances.
INFO: Environment update completed successfully.
```

3. Si desea probar los cambios antes de confirmarlo, utilice la opción `--staged` para implementar los cambios que agregó en el área de ensayo con `git add`.

```
~/my-app$ git add new-file
~/my-app$ eb deploy --staged
```

Si utiliza la opción `--staged`, se realiza una implementación estándar y se omite CodeCommit.

Configuración de entornos y ramificaciones adicionales

La configuración de CodeCommit se aplica a una única ramificación. Puede utilizar `eb use` y `eb codesource` para configurar otras ramificaciones o modificar la configuración de la ramificación actual.

Para configurar la integración de CodeCommit con la CLI de EB

1. Para cambiar la ramificación remota, utilice la opción [--source del comando eb use](#).

```
~/my-app$ eb use test-env --source my-app/test
```

2. Para crear una nueva ramificación y el entorno, extraiga una nueva ramificación, insértela en CodeCommit, cree el entorno y utilice `eb use` para conectar la ramificación local, la ramificación remota y el entorno.

```
~/my-app$ git checkout -b production
~/my-app$ git push --set-upstream production
~/my-app$ eb create production-env
~/my-app$ eb use --source my-app/production production-env
```

3. Para configurar CodeCommit de forma interactiva, utilice [eb codesource codecommit](#).

```
~/my-app$ eb codesource codecommit
Current CodeCommit setup:
  Repository: my-app
  Branch: test
Do you wish to continue (y/n): y
```

```

Select a repository
1) my-repo
2) my-app
3) [ Create new Repository ]
(default is 2): 2

Select a branch
1) mainline
2) test
3) [ Create new Branch with local HEAD ]
(default is 1): 1

```

4. Para deshabilitar la integración de CodeCommit, utilice [eb codesource local](#).

```

~/my-app$ eb codesource local
Current CodeCommit setup:
  Repository: my-app
  Branch: mainline
Default set to use local sources

```

Usar un repositorio de CodeCommit existente

Si ya tiene un repositorio de CodeCommit y desea usarlo con Elastic Beanstalk, ejecute `eb init` en la raíz del repositorio de Git local.

Para utilizar un repositorio de CodeCommit existente con la CLI de EB

1. Clone su repositorio de CodeCommit.

```

~$ git clone ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/my-app

```

2. Eche un vistazo e inserte una rama para usarla para su entorno Elastic Beanstalk.

```

~/my-app$ git checkout -b dev-env
~/my-app$ git push --set-upstream origin dev-env

```

3. Ejecute `eb init`. Elija el mismo nombre de ramificación, repositorio y región que está utilizando actualmente.

```

~/my-app$ eb init

```

```
Select a default region
1) us-east-1 : US East (N. Virginia)
2) us-west-1 : US West (N. California)
3) us-west-2 : US West (Oregon)
4) eu-west-1 : Europe (Ireland)
5) eu-central-1 : Europe (Frankfurt)
6) ap-south-1 : Asia Pacific (Mumbai)
7) ap-southeast-1 : Asia Pacific (Singapore)
...
(default is 3): 1
...
Note: Elastic Beanstalk now supports AWS CodeCommit; a fully-managed source control
  service. To learn more, see Docs: https://aws.amazon.com/codecommit/
Do you wish to continue with CodeCommit? (y/n)(default is n): y

Select a repository
1) my-app
2) [ Create new Repository ]
(default is 1): 1

Select a branch
1) mainline
2) dev-env
3) [ Create new Branch with local HEAD ]
(default is 2): 2
```

Para obtener más información acerca del uso de `eb init`, consulte [Configuración de la CLI de EB](#)

Uso de la CLI de EB para monitorear el estado del entorno

La [interfaz de línea de comandos de Elastic Beanstalk](#) (CLI de EB) es una herramienta de línea de comando que permite administrar los entornos de AWS Elastic Beanstalk. También puede utilizar la CLI de EB para monitorizar el estado del entorno en tiempo real y con más detalle que lo que permite actualmente la consola de Elastic Beanstalk.

Después de [instalar](#) y [configurar](#) la CLI de EB, puede [lanzar un nuevo entorno](#) e implementar el código en él con el comando `eb create`. Si ya tiene un entorno creado en la consola de Elastic Beanstalk, puede asociarlo a la CLI de EB ejecutando `eb init` en una carpeta del proyecto y siguiendo después las instrucciones (la carpeta del proyecto puede estar vacía).

⚠ Important

Asegúrese de que está utilizando la última versión de la CLI de EB ejecutando `pip install` con la opción `--upgrade`:

```
$ sudo pip install --upgrade awsebcli
```

Para obtener instrucciones sobre la instalación de la CLI de EB, consulte [Instalación de la CLI de EB](#).

Si desea utilizar la CLI de EB para monitorear el estado del entorno, primero debe configurar una carpeta de proyecto local ejecutando `eb init` y siguiendo las instrucciones. Para obtener instrucciones completas, consulte [Configuración de la CLI de EB](#).

Si ya tiene un entorno que se ejecuta en Elastic Beanstalk y desea utilizar la CLI de EB para monitorizar su estado, siga estos pasos para adjuntarlo al entorno existente.

Para adjuntar la CLI de EB a un entorno existente

1. Abra un terminal de línea de comando y desplácese hasta la carpeta del usuario.
2. Cree y abra una nueva carpeta en el entorno.
3. Ejecute el comando `eb init` y seleccione la aplicación y el entorno cuyo estado desea monitorear. Si solo hay un entorno que ejecute la aplicación seleccionada, la CLI de EB lo seleccionará automáticamente y no será necesario que lo elija usted, tal y como se muestra en el ejemplo siguiente.

```
~/project$ eb init
Select an application to use
1) elastic-beanstalk-example
2) [ Create new Application ]
(default is 2): 1
Select the default environment.
You can change this later by typing "eb use [environment_name]".
1) elasticBeanstalkEx2-env
2) elasticBeanstalkExa-env
(default is 1): 1
```

Para monitorear el estado a través de la CLI de EB

1. Abra una línea de comando y desplácese hasta la carpeta del proyecto.
2. Ejecute el comando `eb health` para mostrar el estado de las instancias del entorno. En este ejemplo, existen cinco instancias que se ejecutan en un entorno de Linux.

```
~/project $ eb health
elasticBeanstalkExa-env                               Ok
2015-07-08 23:13:20
WebServer
  Ruby 2.1 (Puma)
total          ok      warning  degraded  severe    info    pending  unknown
5             5         0         0         0         0       0         0

instance-id   status      cause
health

Overall       Ok
i-d581497d    Ok
i-d481497c    Ok
i-136e00c0    Ok
i-126e00c1    Ok
i-8b2cf575    Ok

instance-id   r/sec      %2xx      %3xx      %4xx      %5xx      p99      p90      p75
p50          p10
requests
Overall       671.8      100.0     0.0       0.0       0.0       0.003    0.002    0.001
0.001        0.000
i-d581497d    143.0      1430      0          0          0          0.003    0.002    0.001
0.001        0.000
i-d481497c    128.8      1288      0          0          0          0.003    0.002    0.001
0.001        0.000
i-136e00c0    125.4      1254      0          0          0          0.004    0.002    0.001
0.001        0.000
i-126e00c1    133.4      1334      0          0          0          0.003    0.002    0.001
0.001        0.000
i-8b2cf575    141.2      1412      0          0          0          0.003    0.002    0.001
0.001        0.000

instance-id   type        az    running    load 1    load 5      user%    nice%
system% idle% iowait%
i-d581497d    t2.micro    1a    12 mins    0.0       0.04       6.2     0.0
1.0  92.5     0.1
```

i-d481497c	t2.micro	1a	12 mins	0.01	0.09	5.9	0.0
1.6	92.4	0.1					
i-136e00c0	t2.micro	1b	12 mins	0.15	0.07	5.5	0.0
0.9	93.2	0.0					
i-126e00c1	t2.micro	1b	12 mins	0.17	0.14	5.7	0.0
1.4	92.7	0.1					
i-8b2cf575	t2.micro	1c	1 hour	0.19	0.08	6.5	0.0
1.2	92.1	0.1					

instance-id	status	id	version	ago
			deployments	
i-d581497d	Deployed	1	Sample Application	12 mins
i-d481497c	Deployed	1	Sample Application	12 mins
i-136e00c0	Deployed	1	Sample Application	12 mins
i-126e00c1	Deployed	1	Sample Application	12 mins
i-8b2cf575	Deployed	1	Sample Application	1 hour

En este ejemplo, hay una sola instancia que se ejecuta en un entorno de Windows.

```
~/project $ eb health
WindowsSampleApp-env                               Ok
  2018-05-22 17:33:19
WebServer                                           IIS 10.0 running on 64bit
Windows Server 2016/2.2.0
total      ok      warning  degraded  severe    info     pending  unknown
  1         1         0         0         0         0        0        0

instance-id      status      cause
Overall          Ok
i-065716fba0e08a351  Ok

instance-id      r/sec      %2xx      %3xx      %4xx      %5xx      p99      p90
p75      p50      p10
Overall          13.7      100.0     0.0      0.0      0.0      1.403    0.970
0.710    0.413    0.079
i-065716fba0e08a351  2.4      100.0     0.0      0.0      0.0      1.102*   0.865
0.601    0.413    0.091

instance-id      type      az      running      % user time      % privileged
time % idle time      cpu
i-065716fba0e08a351  t2.large  1b     4 hours      0.2
0.1          99.7
```

instance-id	status	id	version	ago
i-065716fba0e08a351	Deployed	2	Sample Application	4 hours

Descripción de la salida

En la salida, en la parte superior de la pantalla, se muestra el nombre del entorno, el estado general del entorno y la fecha actual.

```
elasticBeanstalkExa-env                Ok
2015-07-08 23:13:20
```

En las tres líneas siguientes, se indica el tipo de entorno ("WebServer" en este caso), la configuración (Ruby 2.1 con Puma) y un desglose del número de instancias que tienen cada uno de los siete estados.

```
WebServer
Ruby 2.1 (Puma)
total      ok      warning  degraded  severe  info  pending  unknown
5          5          0         0         0       0     0         0
```

El resto de la salida está dividida en cuatro secciones. En la primera, el valor de status indica el estado, mientras que el valor de cause indica la causa del estado del entorno en su conjunto y de cada instancia. En el ejemplo siguiente, se muestran dos instancias del entorno con el estado Info y una causa que indica que se ha iniciado una implementación.

instance-id	status	cause	health
Overall	Ok		
i-d581497d	Info	Performing application deployment (running for 3 seconds)	
i-d481497c	Info	Performing application deployment (running for 3 seconds)	
i-136e00c0	Ok		
i-126e00c1	Ok		
i-8b2cf575	Ok		

Para obtener más información sobre los estados y los colores, consulte [Colores y estados](#).

En la sección requests, se muestra información de los logs de los servidores web de cada instancia. En este ejemplo, cada instancia está atendiendo normalmente las solicitudes y no hay errores.


```

instance-id    r/sec    %2xx    %3xx    %4xx    %5xx    p99    p90    p75    p50
p10
Overall        13.7    100.0    0.0    0.0    0.0    1.403    0.970    0.710    0.413
0.079
i-d581497d    2.4    100.0    0.0    0.0    0.0    1.102*    0.865    0.601    0.413
0.091
i-d481497c    2.7    100.0    0.0    0.0    0.0    0.842*    0.788    0.480    0.305
0.062
i-136e00c0    4.1    100.0    0.0    0.0    0.0    1.520*    1.088    0.883    0.524
0.104
i-126e00c1    2.2    100.0    0.0    0.0    0.0    1.334*    0.791    0.760    0.344
0.197
i-8b2cf575    2.3    100.0    0.0    0.0    0.0    1.162*    0.867    0.698    0.477
0.076

```

En la sección `cpu`, se muestran las métricas del sistema operativo de cada instancia. La salida difiere según el sistema operativo. A continuación, se muestra la salida para entornos Linux.

```

instance-id    type    az    running    load 1    load 5    user%    nice%    system%
idle%    iowait%
i-d581497d    t2.micro    1a    12 mins    0.0    0.03    0.2    0.0    0.0
99.7    0.1
i-d481497c    t2.micro    1a    12 mins    0.0    0.03    0.3    0.0    0.0
99.7    0.0
i-136e00c0    t2.micro    1b    12 mins    0.0    0.04    0.1    0.0    0.0
99.9    0.0
i-126e00c1    t2.micro    1b    12 mins    0.01    0.04    0.2    0.0    0.0
99.7    0.1
i-8b2cf575    t2.micro    1c    1 hour    0.0    0.01    0.2    0.0    0.1
99.6    0.1

```

Esta es la salida para entornos Windows.

```

instance-id    type    az    running    % user time    % privileged time    %
idle time
i-065716fba0e08a351    t2.large    1b    4 hours    0.2    0.0
99.8

```

Para obtener más información sobre las métricas del servidor y del sistema operativo mostradas, consulte [Métricas de la instancia](#).

En la última sección, deployments, se muestra el estado de implementación de cada instancia. Si se produce un error en una implementación continua, puede utilizar el ID de implementación, el estado y la etiqueta de versión para identificar las instancias del entorno que están ejecutando la versión incorrecta.

instance-id	status	id	version	ago
			deployments	
i-d581497d	Deployed	1	Sample Application	12 mins
i-d481497c	Deployed	1	Sample Application	12 mins
i-136e00c0	Deployed	1	Sample Application	12 mins
i-126e00c1	Deployed	1	Sample Application	12 mins
i-8b2cf575	Deployed	1	Sample Application	1 hour

Vista de estado interactiva

El comando `eb health --refresh` muestra una instantánea del estado del entorno. Para actualizar la información que aparece cada 10 segundos, utilice la opción `--refresh`.

```
$ eb health --refresh
elasticBeanstalkExa-env                               Ok
2015-07-09 22:10:04 (1 secs)
WebServer
  Ruby 2.1 (Puma)
total      ok      warning  degraded  severe  info  pending  unknown
  5         5         0         0         0         0     0         0

instance-id  status  cause
health
Overall      Ok
i-bb65c145   Ok      Application deployment completed 35 seconds ago and took 26
seconds
i-ba65c144   Ok      Application deployment completed 17 seconds ago and took 25
seconds
i-f6a2d525   Ok      Application deployment completed 53 seconds ago and took 26
seconds
i-e8a2d53b   Ok      Application deployment completed 32 seconds ago and took 31
seconds
i-e81cca40   Ok

instance-id  r/sec  %2xx  %3xx  %4xx  %5xx  p99  p90  p75  p50
p10          requests
```

```

Overall      671.8    100.0    0.0    0.0    0.0    0.003    0.002    0.001    0.001
0.000
i-bb65c145   143.0    1430     0      0      0      0.003    0.002    0.001    0.001
0.000
i-ba65c144   128.8    1288     0      0      0      0.003    0.002    0.001    0.001
0.000
i-f6a2d525   125.4    1254     0      0      0      0.004    0.002    0.001    0.001
0.000
i-e8a2d53b   133.4    1334     0      0      0      0.003    0.002    0.001    0.001
0.000
i-e81cca40   141.2    1412     0      0      0      0.003    0.002    0.001    0.001
0.000

```

```

instance-id  type      az  running  load 1  load 5  user%  nice%  system%
idle%  iowait%  cpu
i-bb65c145   t2.micro  1a  12 mins  0.0    0.03    0.2    0.0    0.0
99.7    0.1
i-ba65c144   t2.micro  1a  12 mins  0.0    0.03    0.3    0.0    0.0
99.7    0.0
i-f6a2d525   t2.micro  1b  12 mins  0.0    0.04    0.1    0.0    0.0
99.9    0.0
i-e8a2d53b   t2.micro  1b  12 mins  0.01   0.04    0.2    0.0    0.0
99.7    0.1
i-e81cca40   t2.micro  1c  1 hour   0.0    0.01    0.2    0.0    0.1
99.6    0.1

```

```

instance-id  status  id  version  ago
deployments
i-bb65c145   Deployed  1  Sample Application  12 mins
i-ba65c144   Deployed  1  Sample Application  12 mins
i-f6a2d525   Deployed  1  Sample Application  12 mins
i-e8a2d53b   Deployed  1  Sample Application  12 mins
i-e81cca40   Deployed  1  Sample Application  1 hour

```

(Commands: **Help**, **Quit**, # # # #)

En este ejemplo, se muestra un entorno que recientemente se ha escalado verticalmente de una a cinco instancias. La operación de escalado se hizo correctamente y todas las instancias están superando las comprobaciones de estado y están listas para atender solicitudes. En el modo interactivo, la información de estado se actualiza cada 10 segundos. En la esquina superior derecha, aparece un temporizador con la cuenta atrás hasta la siguiente actualización.

En la esquina inferior izquierda, aparece una lista de opciones. Para salir del modo interactivo, pulse Q. Para desplazarse, pulse las teclas de flecha. Para ver una lista de comandos adicionales, pulse H.

Opciones de la vista de estado interactiva

Si utiliza la vista interactiva con el estado del entorno, puede utilizar las teclas del teclado para ajustar la vista e indicarle a Elastic Beanstalk que reemplace o reinicie determinadas instancias. Para ver una lista de los comandos disponibles mientras consulta el informe de estado en modo interactivo, pulse H.

```
up,down,home,end  Scroll vertically
left,right         Scroll horizontally
F                 Freeze/unfreeze data
X                 Replace instance
B                 Reboot instance
<,>              Move sort column left/right
-,+              Sort order descending/ascending
P                 Save health snapshot data file
Z                 Toggle color/mono mode
Q                 Quit this program
```

Views

```
1                 All tables/split view
2                 Status Table
3                 Request Summary Table
4                 CPU%/Load Table
H                 This help menu
```

(press Q or ESC to return)

Administración de múltiples entornos Elastic Beanstalk como un grupo con la CLI de EB

Puede utilizar la CLI de EB para crear grupos de entornos de AWS Elastic Beanstalk en los que cada uno ejecute un componente distinto de una aplicación con una arquitectura orientada a servicios. La CLI de EB administra dichos grupos mediante la API [ComposeEnvironments](#).

Note

Los grupos de entorno son distintos de varios contenedores en un entorno de Multicontainer Docker. Con los grupos de entornos, cada componente de la aplicación se ejecuta en un entorno de Elastic Beanstalk; independiente, con su propio conjunto dedicado de instancias Amazon EC2. Cada componente se puede escalar por separado. Con Multicontainer Docker, puede combinar varios componentes de una aplicación en un solo entorno. Todos los componentes comparten el mismo conjunto de instancias Amazon EC2; y cada instancia ejecuta varios contenedores de Docker. Elija una de estas arquitecturas de acuerdo con las necesidades de su aplicación.

Para obtener información acerca de Multicontainer Docker, consulte [Uso de la ramificación de la plataforma de Amazon ECS](#).

Organice los componentes de su aplicación en la siguiente estructura de carpetas:

```
~/project-name
|-- component-a
|   `-- env.yaml
`-- component-b
    `-- env.yaml
```

Cada subcarpeta contiene el código fuente de un componente distinto de una aplicación que se ejecutará en su propio entorno y un archivo de definición del entorno denominado `env.yaml`.

Para obtener información detallada sobre el formato `env.yaml`, consulte [Manifiesto del entorno \(env.yaml\)](#).

Para utilizar la API Compose Environments, ejecute primero `eb init` desde la carpeta del proyecto, especificando cada componente por el nombre de la carpeta que lo contiene con la opción `--modules`:

```
~/workspace/project-name$ eb init --modules component-a component-b
```

La CLI de EB le pedirá que [configure cada componente](#) y después cree el directorio `.elasticbeanstalk` en la carpeta de cada componente. La CLI de EB no crea archivos de configuración en el directorio principal.

```
~/project-name
```

```
|-- component-a
|   |-- .elasticbeanstalk
|   `-- env.yaml
`-- component-b
    |-- .elasticbeanstalk
    `-- env.yaml
```

A continuación, ejecute el comando `eb create` con una lista de los entornos que desea crear, uno para cada componente:

```
~/workspace/project-name$ eb create --modules component-a component-b --env-group-suffix group-name
```

Este comando crea un entorno para cada componente. Los nombres de los entornos se crean concatenando el valor de `EnvironmentName` especificado en el archivo `env.yaml` con el nombre del grupo, separados por un guion. La longitud total de estas dos opciones y el guion no debe superar el máximo permitido para el nombre del entorno, que es de 23 caracteres.

Para actualizar el entorno, use el comando `eb deploy`:

```
~/workspace/project-name$ eb deploy --modules component-a component-b
```

Puede actualizar cada componente individualmente o como un grupo. Especifique los componentes que desea actualizar con la opción `--modules`.

La CLI de EB almacena el nombre del grupo que usó con `eb create` en la sección `branch-defaults` del archivo de configuración de la CLI de EB bajo `/.elasticbeanstalk/config.yml`. Para implementar la aplicación en otro grupo, utilice la opción `--env-group-suffix` cuando ejecute `eb deploy`. Si el grupo no existe, la CLI de EB creará un nuevo grupo de entornos:

```
~/workspace/project-name$ eb deploy --modules component-a component-b --env-group-suffix group-2-name
```

Para terminar los entornos, ejecute `eb terminate` en la carpeta de cada módulo. De forma predeterminada, la CLI de EB mostrará un error si intenta terminar un entorno del que depende otro entorno en ejecución. Termine primero el entorno dependiente o utilice la opción `--ignore-links` para invalidar el comportamiento predeterminado:

```
~/workspace/project-name/component-b$ eb terminate --ignore-links
```

Solución de problemas con la CLI de EB

En este tema se muestran mensajes de error comunes encontrados al utilizar la CLI de EB y sus posibles soluciones. Si detecta un mensaje de error que no se muestra aquí, utilice los enlaces de comentarios para informarnos de él.

ERROR: An error occurred while handling git command. Error code: 128 Error: fatal: Not a valid object name HEAD

Causa: este mensaje de error aparece cuando se ha inicializado un repositorio Git, pero aún no se ha validado. La CLI de EB busca la revisión HEAD cuando la carpeta del proyecto contiene un repositorio Git.

Solución: añada los archivos de su carpeta de proyecto al área de ensayo y realice la validación:

```
~/my-app$ git add .  
~/my-app$ git commit -m "First commit"
```

ERROR: This branch does not have a default environment. You must either specify an environment by typing "eb status my-env-name" or set a default environment by typing "eb use my-env-name".

Causa: cuando crea una nueva ramificación en git, esta no se asocia a un entorno de Elastic Beanstalk de forma predeterminada.

Solución: ejecute `eb list` para ver una lista de los entornos disponibles. Después ejecute `eb use env-name` para usar alguno de los entornos disponibles.

ERROR: 2.0+ Platforms require a service role. You can provide one with --service-role option

Causa: si especifica un nombre de entorno con `eb create` (por ejemplo, `eb create my-env`), la CLI de EB no intentará crear un rol de servicio por usted. Si no dispone del rol de servicio predeterminado, se muestra el error anterior.

Solución: ejecute `eb create` sin un nombre de entorno y siga las instrucciones para crear el rol de servicio predeterminado.

Solución de problemas de las implementaciones

Si la implementación de Elastic Beanstalk no funciona según lo previsto, es posible que reciba una respuesta 404 (si no se pudo lanzar la aplicación) o 500 (si la aplicación produce un error en

tiempo de ejecución), en lugar de ver su sitio web. A fin de solucionar muchos problemas comunes, puede utilizar la CLI de EB para comprobar el estado de la implementación, ver los registros, obtener acceso a la instancia de EC2 con SSH o abrir la página de la consola de administración de AWS del entorno de aplicaciones.

Para utilizar la CLI de EB para ayudar a resolver los problemas de su implementación

1. Ejecute `eb status` para ver el estado de la implementación actual y el estado de los hosts de EC2. Por ejemplo:

```
$ eb status --verbose
```

```
Environment details for: python_eb_app
Application name: python_eb_app
Region: us-west-2
Deployed Version: app-150206_035343
Environment ID: e-wa8u6rrmqy
Platform: 64bit Amazon Linux 2014.09 v1.1.0 running Python 2.7
Tier: WebServer-Standard-
CNAME: python_eb_app.elasticbeanstalk.com
Updated: 2015-02-06 12:00:08.557000+00:00
Status: Ready
Health: Green
Running instances: 1
    i-8000528c: InService
```

Note

El uso del modificador `--verbose` proporciona información sobre el estado de las instancias en ejecución. Sin él, `eb status` mostrará solo información general acerca de su entorno.

2. Ejecute `eb health` para ver información sobre el estado de su entorno:

```
$ eb health --refresh
elasticBeanstalkExa-env                               Degraded
2016-03-28 23:13:20
WebServer
  Ruby 2.1 (Puma)
total      ok      warning  degraded  severe  info  pending  unknown
5          2          0         2         1       0     0         0
```



```

instance-id  status  cause
Overall      Degraded Incorrect application version found on 3 out of 5
instances. Expected version "Sample Application" (deployment 1).
i-d581497d   Degraded Incorrect application version "v2" (deployment 2).
Expected version "Sample Application" (deployment 1).
i-d481497c   Degraded Incorrect application version "v2" (deployment 2).
Expected version "Sample Application" (deployment 1).
i-136e00c0   Severe   Instance ELB health has not been available for 5 minutes.
i-126e00c1   Ok
i-8b2cf575   Ok

```

```

instance-id  r/sec  %2xx  %3xx  %4xx  %5xx  p99  p90  p75
p50  p10
Overall      646.7  100.0  0.0  0.0  0.0  0.003  0.002  0.001
0.001  0.000
i-dac3f859   167.5  1675  0  0  0  0.003  0.002  0.001
0.001  0.000
i-05013a81   161.2  1612  0  0  0  0.003  0.002  0.001
0.001  0.000
i-04013a80   0.0    -    -    -    -    -    -    -
-    -
i-3ab524a1   155.9  1559  0  0  0  0.003  0.002  0.001
0.001  0.000
i-bf300d3c   162.1  1621  0  0  0  0.003  0.002  0.001
0.001  0.000

```

```

instance-id  type      az  running  load 1  load 5  user%  nice%
system% idle% iowait%
i-d581497d   t2.micro  1a  25 mins  0.16  0.1  7.0  0.0
1.7  91.0  0.1
i-d481497c   t2.micro  1a  25 mins  0.14  0.1  7.2  0.0
1.6  91.1  0.0
i-136e00c0   t2.micro  1b  25 mins  0.0  0.01  0.0  0.0
0.0  99.9  0.1
i-126e00c1   t2.micro  1b  25 mins  0.03  0.08  6.9  0.0
2.1  90.7  0.1
i-8b2cf575   t2.micro  1c  1 hour  0.05  0.41  6.9  0.0
2.0  90.9  0.0

```

```

instance-id  status  id  version  ago
deployments
i-d581497d   Deployed  2  v2  9 mins
i-d481497c   Deployed  2  v2  7 mins

```

i-136e00c0	Failed	2	v2	5 mins
i-126e00c1	Deployed	1	Sample Application	25 mins
i-8b2cf575	Deployed	1	Sample Application	1 hour

El ejemplo anterior muestra un entorno con cinco instancias en las que la implementación de la versión "v2" produjo un error en la tercera instancia. Después de un error en la implementación, la versión se restablece a la última versión correcta, que en este caso es "Sample Application" de la primera implementación. Para obtener más información, consulte [Uso de la CLI de EB para monitorear el estado del entorno](#).

- Ejecute `eb logs` para descargar y ver los registros asociados con la implementación de su aplicación.

```
$ eb logs
```

- Ejecute `eb ssh` para conectar con la instancia EC2 que ejecuta su aplicación y examinarla directamente. En la instancia, su aplicación implementada puede encontrarse en el directorio `/opt/python/current/app` y su entorno Python se encontrará en `/opt/python/run/venv/`.
- Ejecute `eb console` para ver el entorno de aplicaciones en la [consola de administración de AWS](#). Puede utilizar la interfaz web para examinar distintos aspectos de su implementación, incluida la configuración, el estado, los eventos y los registros de la aplicación. También puede descargar las versiones de la aplicación actuales o anteriores que ha implementado en el servidor.

Referencia de los comandos de la CLI de EB

Puede utilizar la interfaz de línea de comandos (CLI de EB) de Elastic Beanstalk para realizar una variedad de operaciones para implementar y administrar sus aplicaciones y entornos de Elastic Beanstalk. La CLI de EB se integra con Git si desea implementar código fuente de aplicaciones con control de código fuente de Git. Para obtener más información, consulte [Uso de la interfaz de línea de comandos de Elastic Beanstalk \(CLI de EB\)](#) y [Uso de la CLI de EB con Git](#).

Comandos

- [eb abort](#)
- [eb appversion](#)
- [eb clone](#)
- [eb codesource](#)

- [eb config](#)
- [eb console](#)
- [eb create](#)
- [eb deploy](#)
- [eb events](#)
- [eb health](#)
- [eb init](#)
- [eb labs](#)
- [eb list](#)
- [eb local](#)
- [eb logs](#)
- [eb open](#)
- [eb platform](#)
- [eb printenv](#)
- [eb restore](#)
- [eb scale](#)
- [eb setenv](#)
- [eb ssh](#)
- [eb status](#)
- [eb swap](#)
- [eb tags](#)
- [eb terminate](#)
- [eb upgrade](#)
- [eb use](#)
- [Opciones comunes](#)

eb abort

Descripción

Cancela una actualización cuando una configuración del entorno cambia a instancias que siguen en curso.

Note

Si tiene más de dos entornos a los que va a aplicar una actualización, se le pedirá que seleccione el nombre del entorno para el que desea restablecer los cambios.

Sintaxis

```
eb abort
```

```
eb abort environment-name
```

Opciones

Nombre	Descripción
Opciones comunes	

Salida

El comando muestra una lista de los entornos que están siendo actualizados en este momento y se le pide que seleccione la actualización que desea cancelar. Si solo se está actualizando un entorno en este momento, no tiene que especificar el nombre del entorno. Si todo sale bien, el comando revierte los cambios de configuración del entorno. El proceso de restauración continúa hasta que todas las instancias del entorno tengan la configuración del entorno anterior o hasta que el proceso de restauración produce un error.

Ejemplo

El siguiente ejemplo cancela la actualización de la plataforma.

```
$ eb abort
Aborting update to environment "tmp-dev".
<list of events>
```

eb appversion

Descripción

El comando `appversion` de la CLI de EB administra las [versiones de la aplicación](#) de Elastic Beanstalk. Puede crear una versión nueva de la aplicación sin necesidad de implementar, eliminar una versión de la aplicación o crear la política del ciclo de vida de la versión de la aplicación???. Si invoca el comando sin ninguna opción, ingresa en el [modo interactivo](#).

Utilice la opción `--create` para crear una versión nueva de la aplicación.

Utilice la opción `--delete` para eliminar una versión de la aplicación.

Utilice la opción `lifecycle` para mostrar o crear la política de ciclo de vida de versiones de la aplicación. Para obtener más información, consulte [the section called “Ciclo de vida de la versión”](#).

Sintaxis

```
eb appversion
```

```
eb appversion [-c | --create]
```

```
eb appversion [-d | --delete] version-label
```

```
eb appversion lifecycle [-p | --print]
```

Opciones

Nombre	Descripción
	Tipo: String
<code>-a <i>application-name</i></code> o bien <code>--application_name <i>application-name</i></code>	Nombre de la aplicación. Si no se encuentra una aplicación con el nombre especificado, la CLI de EB crea una versión de aplicación para una aplicación nueva. Solo se aplica con la opción <code>--create</code> . Tipo: String
<code>-c</code>	Cree una versión nueva de la aplicación.

Nombre	Descripción
<p>o</p> <p>--create</p>	<p>Tipo: String</p>
<p>-d <i>etiqueta-de-versión</i></p> <p>o bien</p> <p>--delete <i>etiqueta-de-versión</i></p>	<p>Elimine la versión de la aplicación etiquetada como <i>versión-etiqueta</i> .</p>
<p>-l <i>etiqueta_de_versión</i></p> <p>o bien</p> <p>--label <i>etiqueta_de_versión</i></p>	<p>Especifique una etiqueta para usarla con la versión que crea la CLI de EB. Si no utiliza esta opción, la CLI de EB genera una etiqueta única nueva. Si proporciona una etiqueta de versión, asegúrese de que sea única.</p> <p>Solo se aplica con la opción --create.</p> <p>Tipo: String</p>
<p>ciclo de vida</p>	<p>Invoque el editor predeterminado para crear una nueva política de ciclo de vida de versiones de aplicación. Utilice esta política para evitar alcanzar la cuota de versiones de la aplicación.</p>
<p>lifecycle -p</p> <p>o bien</p> <p>lifecycle --print</p>	<p>Muestra la política del ciclo de vida de la aplicación actual.</p>

Nombre	Descripción
<p><code>-m "<i>descripción_de_ver</i> <i>sión</i> "</code></p> <p>o bien</p> <p><code>--message "<i>descripci</i> <i>ón_de_versión</i> "</code></p>	<p>Descripción de la versión de la aplicación. Está entre comillas dobles.</p> <p>Solo se aplica con la opción <code>--create</code>.</p> <p>Tipo: String</p>
<p><code>-p</code></p> <p>o bien</p> <p><code>--process</code></p>	<p>Procesar previamente y validar el manifiesto del entorno y los archivos de configuración del paquete de código fuente. La validación de los archivos de configuración puede identificar problemas. Recomendamos que lo haga antes de implementar la versión de la aplicación en el entorno.</p> <p>Solo se aplica con la opción <code>--create</code>.</p>
<p><code>--source codecommi t/ <i>repository-</i> <i>name/branch-name</i></code></p>	<p>Repositorio y ramificación de CodeCommit. Para obtener más información, consulte Uso de la CLI de EB con AWS CodeCommit.</p> <p>Solo se aplica con la opción <code>--create</code>.</p>
<p><code>--staged</code></p>	<p>Para crear la versión de la aplicación, utilice los archivos organizados en el índice git, en lugar de la confirmación HEAD.</p> <p>Solo se aplica con la opción <code>--create</code>.</p>
<p><code>--timeout <i>minutos</i></code></p>	<p>Número de minutos que deben transcurrir antes de que se agote el tiempo de espera del comando.</p> <p>Solo se aplica con la opción <code>--create</code>.</p>
<p>Opciones comunes</p>	

Uso interactivo del comando

Si utiliza el comando sin argumentos, el resultado muestra las versiones de la aplicación. Se enumeran en orden cronológico inverso y primero aparece la última versión. Consulte la sección Examples (Ejemplos) para ver ejemplos del aspecto de la pantalla. Tenga en cuenta que la línea de estado se muestra en la parte inferior. La línea de estado muestra información confidencial del contexto.

Pulse `d` para eliminar una versión de la aplicación, pulse `l` para administrar la política de ciclo de vida de aplicación o pulse `q` para salir sin hacer cambios.

Note

Si la versión se implementa en cualquier entorno, no puede eliminarla.

Salida

El comando con la opción `--create` muestra un mensaje que confirma que se ha creado la versión de la aplicación.

El comando con la opción `--delete` *versión-de-etiqueta* muestra un mensaje en el que se confirma que se eliminó la versión de la aplicación.

Ejemplos

En el ejemplo siguiente, se muestra la ventana interactiva de una aplicación sin implementaciones.

```

No Environment Specified                               Application Name: versions
Environment Status: Unknown Health Unknown
Current version # deployed: None

#  Version Label  Date Created  Age  Description
3  v4             2016/12/22 13:28  56 secs  new features
2  v3             2016/12/22 13:27  1 min    important update
1  v1             2016/12/15 23:51  6 days   wow

(Commands: Quit, Delete, Lifecycle, ▼▲◀▶)
  
```

En el ejemplo siguiente, se muestra la ventana interactiva de una aplicación con la cuarta versión implementada y la etiqueta de versión Sample Application (Aplicación de muestra).


```

Sample-env Application Name: versions
Environment Status: Launching Health Green
Current version # deployed: 4

# Version Label Date Created Age Description
4 Sample Application 2016/12/22 13:30 2 mins -
3 v4 2016/12/22 13:28 4 mins new features
2 v3 2016/12/22 13:27 5 mins important update
1 v1 2016/12/15 23:51 6 days wow

(Commands: Quit, Delete, Lifecycle, ▼▲ ◀▶)

```

En el siguiente ejemplo, se muestra la salida de un comando `eb appversion lifecycle -p`, donde *ACCOUNT-ID* es el ID de la cuenta de usuario:

```

Application details for: lifecycle
Region: sa-east-1
Description: Application created from the EB CLI using "eb init"
Date Created: 2016/12/20 02:48 UTC
Date Updated: 2016/12/20 02:48 UTC
Application Versions: ['Sample Application']
Resource Lifecycle Config(s):
  VersionLifecycleConfig:
    MaxCountRule:
      DeleteSourceFromS3: False
      Enabled: False
      MaxCount: 200
    MaxAgeRule:
      DeleteSourceFromS3: False
      Enabled: False
      MaxAgeInDays: 180
  ServiceRole: arn:aws:iam::ACCOUNT-ID:role/aws-elasticbeanstalk-service-role

```

eb clone

Descripción

Clona un entorno en un nuevo entorno para que ambos tengan la misma configuración.

Note

De forma predeterminada, independientemente de la versión de la pila de soluciones del entorno desde el que crea el clon, el comando `eb clone` crea el entorno clonado con la pila

de soluciones más reciente. Puede invalidar este comportamiento incluyendo la opción `--exact` cuando ejecute el comando.

⚠ Important

Los entornos clonados de Elastic Beanstalk no transfieren los grupos de seguridad para su entrada, lo que deja el entorno abierto a todo el tráfico de Internet. Deberá restablecer los grupos de seguridad de entrada para el entorno clonado.

Para ver los recursos que no se pueden clonar, compruebe el estado de desviación de la configuración de su entorno. Para obtener más información, consulte [Detectar desviaciones en una CloudFormation pila completa](#) en la Guía del AWS CloudFormation usuario.

Sintaxis

`eb clone`

`eb clone environment-name`

Opciones

Nombre	Descripción
<code>-n <i>cadena</i></code> o bien <code>--clone_name <i>cadena</i></code>	Nombre deseado del entorno clonado.
<code>-c <i>cadena</i></code> o bien <code>--cname <i>cadena</i></code>	Prefijo CNAME deseado del entorno clonado.
<code>--envvars</code>	Propiedades del entorno en una lista separada por comas con el formato <code><i>nombre=valor</i></code> .

Nombre	Descripción
	<p>Tipo: String</p> <p>Restricciones:</p> <ul style="list-style-type: none"> • Los pares clave-valor deben ir separados por comas. • Las claves y los valores pueden contener caracteres alfabéticos en cualquier idioma, cualquier carácter numérico, espacios en blanco, separadores invisibles y los siguientes símbolos: <code>_ . : / + \ - @</code> • Las claves pueden contener un máximo de 128 caracteres. Los valores pueden contener un máximo de 256 caracteres. • Las claves y los valores distinguen entre mayúsculas y minúsculas. • Los valores no pueden coincidir con el nombre del entorno. • Los valores no pueden incluir <code>aws:</code> ni <code>elasticbeanstalk:</code> . • El tamaño combinado de todas las propiedades del entorno no puede superar los 4096 bytes.
<code>--exact</code>	Impide que Elastic Beanstalk actualice la versión de la pila de soluciones del nuevo entorno clonado a la versión más reciente disponible (para la plataforma del entorno original).
<code>--scale <i>número</i></code>	El número de instancias que se ejecutan en el entorno clonado cuando se lanza.
<code>--tags <i>nombre=valor</i></code>	Etiquetas para los recursos de su entorno en una lista separada por comas con el formato <code><i>nombre=valor</i></code> .
<code>--timeout</code>	Número de minutos que deben transcurrir antes de que se agote el tiempo de espera del comando.
Opciones comunes	

Salida

Si se ejecuta correctamente, el comando crea un entorno que tiene la misma configuración que el entorno original o con modificaciones, según se especifique en alguna de las opciones de `eb clone`.

Ejemplo

El siguiente ejemplo clona el entorno especificado.

```
$ eb clone
Enter name for Environment Clone
(default is tmp-dev-clone):
Enter DNS CNAME prefix
(default is tmp-dev-clone):
Environment details for: tmp-dev-clone
  Application name: tmp
  Region: us-west-2
  Deployed Version: app-141029_144740
  Environment ID: e-vjvraqnn5pv
  Platform: 64bit Amazon Linux 2014.09 v1.0.9 running PHP 5.5
  Tier: WebServer-Standard-1.0
  CNAME: tmp-dev-clone.elasticbeanstalk.com
  Updated: 2014-10-29 22:00:23.008000+00:00
Printing Status:
2018-07-11 21:04:20    INFO: createEnvironment is starting.
2018-07-11 21:04:21    INFO: Using elasticbeanstalk-us-west-2-888888888888 as Amazon S3
storage bucket for environment data.
...
2018-07-11 21:07:10    INFO: Successfully launched environment: tmp-dev-clone
```

eb codesource

Descripción

Configure la CLI de EB [para realizar la implementación desde un repositorio de CodeCommit](#) o deshabilite la integración de CodeCommit y cargue el paquete de código fuente de su equipo local.

Note

Algunas regiones de AWS no ofrecen CodeCommit. La integración entre Elastic Beanstalk y CodeCommit no funciona en estas regiones.

Para obtener información sobre los servicios de AWS que se ofrecen en cada región, consulte la [Tabla de regiones](#).

Sintaxis

```
eb codesource
```

```
eb codesource codecommit
```

```
eb codesource local
```

Opciones

Nombre	Descripción
Opciones comunes	

Salida

`eb codesource` le pide que elija entre implementaciones con integración de CodeCommit y estándar.

`eb codesource codecommit` inicia la configuración interactiva del repositorio para la integración de CodeCommit.

`eb codesource local` muestra la configuración original y desactiva la integración de CodeCommit.

Ejemplos

Utilice `eb codesource codecommit` para configurar la integración de CodeCommit para la ramificación actual.

```
~/my-app$ eb codesource codecommit
Select a repository
1) my-repo
2) my-app
3) [ Create new Repository ]
(default is 1): 1
```

```
Select a branch
1) mainline
2) test
3) [ Create new Branch with local HEAD ]
(default is 1): 1
```

Utilice `eb codesource local` para deshabilitar la integración de CodeCommit para la ramificación actual.

```
~/my-app$ eb codesource local
Current CodeCommit setup:
  Repository: my-app
  Branch: mainline
Default set to use local sources
```

eb config

Descripción

Administra la configuración activa [???](#) y las configuraciones guardadas [???](#) del entorno. Puede utilizar este comando para cargar, descargar o enumerar las configuraciones guardadas del entorno. También puede usarlo para descargar, mostrar o actualizar los ajustes de configuración activos.

Si el directorio raíz contiene un archivo `platform.yaml` que especifica una plataforma personalizada, este comando también cambia la configuración del creador. Esto se hace en función de los valores que se establece `platform.yaml`.


Note

`eb config` muestra las propiedades del entorno. Para configurar las propiedades del entorno desde las que puede leer dentro de la aplicación, utilice [eb setenv](#) en su lugar.

Sintaxis

A continuación se presentan partes de la sintaxis que se utiliza para que el comando `eb config` funcione con los valores de configuración [???](#) activos del entorno. Para obtener ejemplos específicos, consulte la sección [Ejemplos](#), más adelante en este tema.

- `eb config`: muestra los valores de configuración activos del entorno en un editor de texto que ha configurado como variable de entorno EDITOR. Cuando guarda los cambios en el archivo y cierra el editor, el entorno se actualiza con la configuración de opciones que ha guardado en el archivo.


 Note

Si no ha configurado una variable de entorno EDITOR, la CLI de EB muestra la configuración de opciones en el editor predeterminado para los archivos YAML.

- `eb config environment-name`: muestra y actualiza la configuración del entorno designado. La configuración se muestra en un editor de texto configurado o en los archivos YAML del editor predeterminado.
- `eb config save`: guarda los valores de configuración activos del entorno actual en el que se ejecuta `se.elasticbeanstalk/saved_configs/` encuentra el nombre del archivo `[configuration-name].cfg.yaml`. De forma predeterminada, la CLI de EB guarda las opciones de configuración con un **nombre-configuración** basado en el entorno. Puede especificar otro nombre de configuración incluyendo la opción `--cfg` con el nombre de configuración que desee cuando ejecute el comando.

Puede etiquetar su configuración guardada mediante la opción `--tags`.

- `eb config --display`: escribe la configuración activa de un entorno en stdout en lugar de un archivo. De forma predeterminada, muestra los ajustes de configuración en el terminal.
- `eb config --update configuration_string | file_path`: actualiza la configuración activa del entorno actual con la información especificada en **configuration_string** o dentro del archivo identificado por **file_path**.

 Note

Las opciones `--display` y `--update` proporcionan flexibilidad para leer y revisar los ajustes de configuración del entorno mediante programación.

A continuación se describe la sintaxis de utilizar el comando `eb config` para trabajar con [configuraciones guardadas](#). Para ver ejemplos, consulte la sección [Ejemplos](#) más adelante en este tema.

- `eb config get config-name`: descarga la configuración guardada con nombre de Amazon S3.

- `eb config delete config-name` : elimina la configuración guardada con nombre de Amazon S3. También lo elimina localmente, si ya lo descargó.
- `eb config list`: muestra las configuraciones guardadas que tiene en Amazon S3.
- `eb config put filename`: carga la configuración guardada designada en un bucket de Amazon S3. El *nombre de archivo* debe tener la extensión de archivo `.cfg.yaml`. Para especificar el nombre de archivo sin una ruta, puede guardar el archivo en la carpeta `.elasticbeanstalk` o en la carpeta `.elasticbeanstalk/saved_configs/` antes de ejecutar el comando. También puede especificar el *nombre de archivo* proporcionando la ruta completa.

Opciones

Nombre	Descripción
<code>--cfg <i>config-name</i></code>	Nombre que se utilizará para una configuración guardada. Esta opción funciona solo con <code>eb config save</code> .
<code>-d</code> o bien <code>--display</code>	Muestra los valores de configuración del entorno actual (escribe en stdout). Utilícelo con la opción <code>--format</code> para especificar la salida que debe estar en JSON o YAML. Si no lo especifica, la salida está en formato YAML. Esta opción solo funciona si utiliza el comando <code>eb config</code> sin ninguno de los demás subcomandos.
<code>-f <i>format_type</i></code> o bien <code>--format <i>format_type</i></code>	Especifica el formato de visualización. Los valores válidos son JSON o YAML. El valor predeterminado es YAML. Esta opción solo funciona con la opción <code>--display</code> .
<code>--tags <i>key1=value1[,ke</i></code>	Puede agregar etiquetas a su configuración guardada. Al especificar etiquetas en la lista, hágalo como pares clave=valor y separe cada una con una coma.

Nombre	Descripción
	<p>Para obtener más información, consulte Etiquetado de configuraciones guardadas.</p> <p>Esta opción funciona solo con eb config save.</p>
<code>--timeout <i>timeout</i></code>	Número de minutos que deben transcurrir antes de que se agote el tiempo de espera del comando.

Nombre	Descripción
<p><code>-u <i>configuration_string</i> <i>file_path</i></code></p> <p>o bien</p> <p><code>--update <i>configuration_string</i> <i>file_path</i></code></p>	<p>Actualiza la configuración activa del entorno actual.</p> <p>Esta opción solo funciona si utiliza el comando <code>eb config</code> sin ninguno de los demás subcomandos.</p> <p>El parámetro <code><i>configuration_string</i> <i>file_path</i></code> es de la cadena de tipo. La cadena proporciona la lista de espacios de nombres y las opciones correspondientes para agregar, actualizar o quitar de la configuración del entorno. Como alternativa, la cadena de entrada puede representar un archivo que contiene la misma información.</p> <p>Para especificar un nombre de archivo, la cadena de entrada debe seguir el formato <code>"file://< <i>path</i>><<i>filename</i>>"</code>. Para especificar el nombre de archivo sin un <code><i>path</i></code>, guárdelo en la carpeta en la que ejecuta el comando. Como alternativa, especifique el nombre del archivo al proporcionar la ruta completa.</p> <p>La información de configuración debe cumplir las siguientes condiciones. Se requiere al menos una de las secciones <code>OptionSettings</code> o <code>OptionsToRemove</code>. Se utiliza <code>OptionSettings</code> para agregar o cambiar opciones. Se utiliza <code>OptionsToRemove</code> para quitar opciones de un espacio de nombres. Para obtener ejemplos específicos, consulte la sección Ejemplos, más adelante en este tema.</p> <p>Example</p> <p>Formato YAML</p> <pre> OptionSettings: namespace1: option-name-1: <i>option-value-1</i> option-name-2: <i>option-value-2</i> ... OptionsToRemove: namespace1: </pre>

Nombre	Descripción
	<pre data-bbox="597 205 1507 346">option-name-1 option-name-2 ...</pre> <p data-bbox="597 384 716 422">Example</p> <p data-bbox="597 464 805 501">Formato JSON</p> <pre data-bbox="597 537 1507 1213">{ "OptionSettings": { "namespace1": { "option-name-1": " <i>option-value-1</i> ", "option-name-2": " <i>option-value-2</i> ", ... } }, "OptionsToRemove": { "namespace1": { "option-name-1", "option-name-2", ... } } }</pre> <p data-bbox="115 1251 388 1289">Opciones comunes</p>

Salida

Si el comando `eb config` o `eb config environment-name` se ejecuta correctamente sin agregar subcomandos ni opciones, el comando muestra la configuración de la opción actual en el editor de texto que configuró como variable de entorno EDITOR. Si no ha configurado una variable de entorno EDITOR, la CLI de EB muestra la configuración de opciones en el editor predeterminado para los archivos YAML.

Cuando guarda los cambios en el archivo y cierra el editor, el entorno se actualiza con la configuración de opciones que ha guardado en el archivo. Se muestra el siguiente resultado para confirmar la actualización de la configuración.

```
$ eb config myApp-dev
Printing Status:
2021-05-19 18:09:45 INFO Environment update is starting.
2021-05-19 18:09:55 INFO Updating environment myApp-dev's configuration
settings.
2021-05-19 18:11:20 INFO Successfully deployed new configuration to
environment.
```

Si el comando se ejecuta correctamente con la opción `--display`, muestra los valores de configuración del entorno actual (escribe en stdout).

Si el comando se ejecuta correctamente con el parámetro `get`, muestra la ubicación de la copia local que ha descargado.

Si el comando se ejecuta correctamente con el parámetro `save`, muestra la ubicación del archivo guardado.

Ejemplos

En esta sección se describe cómo cambiar el editor de texto que utiliza para ver y editar el archivo de configuración de opciones.

Para Linux y UNIX, el siguiente ejemplo cambia el editor a vim:

```
$ export EDITOR=vim
```

Para Linux y UNIX, el siguiente ejemplo cambia el editor a lo que esté instalado en `/usr/bin/kate`.

```
$ export EDITOR=/usr/bin/kate
```

Para Windows, el siguiente ejemplo cambia el editor a Notepad++.

```
> set EDITOR="C:\Program Files\Notepad++\Notepad++.exe
```

En esta sección se proporcionan ejemplos del comando `eb config` cuando se ejecuta con subcomandos.

En el ejemplo siguiente se elimina la configuración guardada denominada `app-tmp`.

```
$ eb config delete app-tmp
```

El siguiente ejemplo descarga la configuración guardada con el nombre `app-tmp` de su bucket de Amazon S3.

```
$ eb config get app-tmp
```

El siguiente ejemplo muestra los nombres de las configuraciones guardadas almacenadas en el bucket de Amazon S3.

```
$ eb config list
```

El siguiente ejemplo carga la copia local de la configuración guardada con el nombre `app-tmp` en su bucket de Amazon S3.

```
$ eb config put app-tmp
```

El siguiente ejemplo guarda las opciones de configuración del entorno en ejecución actual. Si no proporciona un nombre para utilizarlo en la configuración guardada, Elastic Beanstalk nombra el archivo de configuración según el nombre del entorno. Por ejemplo, un entorno llamado `tmp-dev` se denominaría `tmp-dev.cfg.yml`. Elastic Beanstalk guarda el archivo en la carpeta `/.elasticbeanstalk/saved_configs/`.

```
$ eb config save
```

En el ejemplo siguiente, se utiliza la opción `--cfg` para guardar los valores de configuración del entorno `tmp-dev` en un archivo denominado `v1-app-tmp.cfg.yml`. Elastic Beanstalk guarda el archivo en la carpeta `/.elasticbeanstalk/saved_configs/`. Si no especifica un nombre de entorno, Elastic Beanstalk guarda los valores de configuración del entorno en ejecución actual.

```
$ eb config save tmp-dev --cfg v1-app-tmp
```

En esta sección se proporcionan ejemplos del comando `eb config` cuando se ejecuta sin subcomandos.

El siguiente comando muestra la configuración de opciones del entorno actual en un editor de texto.

```
$ eb config
```

El siguiente comando muestra la configuración de opciones del entorno `my-env` en un editor de texto.

```
$ eb config my-env
```

En el ejemplo siguiente se muestra la configuración de opciones del entorno actual. Se produce en formato YAML porque no se especificó ningún formato puntual con la opción `--format`.

```
$ eb config --display
```

En el siguiente ejemplo se actualiza la configuración de opciones del entorno actual con las especificaciones del archivo denominad `example.txt`. El archivo está en formato YAML o JSON. La CLI de EB detecta automáticamente el formato de archivo.

- La opción `Minsize` se establece en 1 para el espacio de nombre `aws:autoscaling:asg`.
- El tamaño del lote del espacio de nombres `aws:elasticbeanstalk:command` se establece en 30 %.
- Elimina la configuración de opción de `IdleTimeout`: ninguno del espacio de nombres `AWSEBV2LoadBalancer.aws:elbv2:loadbalancer`.

```
$ eb config --update "file://example.txt"
```

Example - nombre de archivo: **example.txt**, formato YAML

```
OptionSettings:
  'aws:elasticbeanstalk:command':
    BatchSize: '30'
    BatchSizeType: Percentage
  'aws:autoscaling:asg':
    MinSize: '1'
OptionsToRemove:
  'AWSEBV2LoadBalancer.aws:elbv2:loadbalancer':
    IdleTimeout
```

Example - nombre de archivo: **example.txt**, formato JSON

```
{
  "OptionSettings": {
    "aws:elasticbeanstalk:command": {
      "BatchSize": "30",
      "BatchSizeType": "Percentage"
```

```

    },
    "aws:autoscaling:asg": {
        "MinSize": "1"
    }
},
"OptionsToRemove": {
    "AWSEBV2LoadBalancer.aws:elbv2:loadbalancer": {
        "IdleTimeout"
    }
}
}
}

```

En los ejemplos siguientes se actualizan la configuración de opciones del entorno actual. El comando establece la opción `MinSize` en 1 para el espacio de nombres `aws:autoscaling:asg`.

Note

Estos ejemplos son específicos de Windows PowerShell. Escapa de las apariciones literales del carácter de comilla doble (") al precederlo con un carácter de barra inclinada (\). Los distintos sistemas operativos y entornos de línea de comandos pueden tener secuencias de escape diferentes. Por este motivo, recomendamos utilizar la opción de archivo que se muestra en los ejemplos anteriores. La especificación de las opciones de configuración en un archivo no requiere caracteres de escape y es coherente en distintos sistemas operativos.

El siguiente ejemplo está en formato JSON. La CLI de EB detecta si el formato está en JSON o YAML.

```

PS C:\Users\myUser\EB_apps\myApp-env>eb config --update '{"OptionSettings\":
{"aws:autoscaling:asg\":{"MaxSize\":"1\"}}}'

```

El siguiente ejemplo está en formato YAML. Para ingresar la cadena YAML en el formato correcto, el comando incluye el espaciado y las devoluciones de final de línea que se requieren en un archivo YAML.

- Finaliza cada línea con la tecla “enter” (ingresar) o “return” (volver).
- Comience la segunda línea con dos espacios y la tercera línea con cuatro espacios.

```

PS C:\Users\myUser\EB_apps\myApp-env>eb config --update 'OptionSettings:

```

```
>> aws:autoscaling:asg:
>>   MinSize: \"1\"
```

eb console

Descripción

Abra un navegador para mostrar el panel de configuración del entorno en la consola de gestión de Elastic Beanstalk.

Si el directorio raíz contiene un archivo `platform.yaml` donde se especifica una plataforma personalizada, este comando mostrará también la configuración del entorno del constructor, tal y como se especifica en el archivo `platform.yaml`, en la consola de gestión de Elastic Beanstalk.

Sintaxis

`eb console`

`eb console environment-name`

Opciones

Nombre	Descripción
Opciones comunes	

eb create

Descripción

Crea un nuevo entorno e implementa una versión de la aplicación en él.

Note

- Para usar el comando `eb create` en una aplicación .NET, debe crear un paquete de implementación como se describe en [Creación de un paquete de código fuente para una aplicación .NET](#), después definir la configuración de la CLI para implementar el paquete

como un artefacto, como se describe en [Implementar un artefacto en lugar de la carpeta del proyecto](#).

- Para crear entornos con la CLI de EB, se requiere una [función de servicio](#). Puede crear una función de servicio creando un entorno en la consola de Elastic Beanstalk. Si no cuenta con ninguna función de servicio, el EB CLI intentará crear uno cuando ejecute `eb create`.

Puede implementar la versión de la aplicación desde varios orígenes:

- De forma predeterminada: desde el código fuente de la aplicación en el directorio de proyectos locales.
- Mediante la opción `--version`: desde una versión de la aplicación que ya existe en su aplicación.
- Cuando el directorio de proyectos no tiene código de aplicación o al utilizar la opción `--sample`: implementada desde una aplicación de ejemplo, específica de la plataforma de su entorno.

Sintaxis

```
eb create
```

```
eb create environment-name
```

Un nombre de entorno debe tener una longitud entre 8 y 64 caracteres. Solo puede contener letras, números y guiones (-). Un nombre de entorno no puede empezar ni terminar con un guion.

Si incluye un nombre del entorno en el comando, la CLI de EB no le pedirá que seleccione ningún valor ni que cree un rol de servicio.

Si ejecuta el comando sin argumento de nombre de entorno, se ejecuta en un flujo interactivo y le pide que escriba o seleccione valores para algunos ajustes. En este flujo interactivo, en caso de que esté implementando una aplicación de ejemplo, la CLI de EB también le pregunta si desea descargar esta aplicación de ejemplo en el directorio de proyectos locales. Al descargarlo, puede utilizar la CLI de EB con el nuevo entorno más tarde para ejecutar las operaciones que requieran el código de la aplicación, como por ejemplo [eb deploy](#).

Algunas solicitudes de flujo interactivo solo se muestran en determinadas condiciones. Por ejemplo, si elige utilizar un Application Load Balancer y su cuenta tiene al menos un Application Load Balancer compartible, Elastic Beanstalk mostrará un mensaje que le preguntará si desea utilizar

un balanceador de carga compartido. Si no existe un Application Load Balancer compatible en su cuenta, este mensaje no se muestra.

Opciones

Ninguna de estas opciones es obligatoria. Si ejecuta `eb create` sin ninguna opción, la CLI de EB le pedirá que escriba o seleccione un valor para cada ajuste.

Nombre	Descripción
<code>-d</code> o bien <code>--branch_default</code>	Establece el entorno como predeterminado en el repositorio actual.
<code>--cfg <i>nombre-de-configuración</i></code>	Utilice los ajustes de la plataforma de una configuración guardada en <code>.elasticbeanstalk/saved_configs/</code> o en el bucket de Amazon S3. Especifique el nombre del archivo sin la extensión <code>.cfg.yml</code> .
<code>-c <i>nombre-de-subdominio</i></code> o bien <code>--cname <i>nombre-de-subdominio</i></code>	Nombre del subdominio que se va a incluir como prefijo en la entrada DNS de CNAME que direcciona el tráfico al sitio web. Tipo: String Valor predeterminado: nombre del entorno.
<code>-db</code> o bien <code>--database</code>	Adjunta una base de datos al entorno. Si ejecuta <code>eb create</code> con la opción <code>--database</code> pero sin las opciones <code>--database.username</code> y <code>--database.password</code> , la CLI de EB le pide el nombre del usuario principal y la contraseña de la base de datos.
<code>-db.engine <i>motor</i></code> o bien <code>--database.engine <i>motor</i></code>	Tipo de motor de base de datos. Si ejecuta <code>eb create</code> con esta opción, la CLI de EB lanza el entorno con una base de datos adjuntada. Este es el caso incluso si no ejecutó el comando con la opción <code>--database</code> .

Nombre	Descripción
	<p>Tipo: String</p> <p>Valores válidos: <code>mysql</code>, <code>oracle-se1</code> , <code>postgres</code>, <code>sqlserver-ex</code> , <code>sqlserver-web</code> , <code>sqlserver-se</code></p>
<p><code>-db.i <i>tipo_de_instancia</i></code> o bien <code>--database.instance <i>tipo_de_instancia</i></code></p>	<p>Tipo de Amazon EC2 instance que se va a utilizar con la base de datos. Si ejecuta <code>eb create</code> con esta opción, la CLI de EB lanza el entorno con una base de datos adjuntada. Este es el caso incluso si no ejecutó el comando con la opción <code>--database</code> .</p> <p>Tipo: cadena</p> <p>Valores válidos:</p> <p>Amazon RDS es compatible con un conjunto estándar de instancias de base de datos. Para seleccionar una instancia de base de datos adecuada para su motor de base de datos, es necesario tener en cuenta algunos aspectos específicos. Para obtener más información, consulte Clases de instancia de base de datos en la Guía del usuario de Amazon RDS.</p>
<p><code>-db.pass <i>contraseña</i></code> o bien <code>--database.password <i>contraseña</i></code></p>	<p>Contraseña de la base de datos. Si ejecuta <code>eb create</code> con esta opción, la CLI de EB lanza el entorno con una base de datos adjuntada. Este es el caso incluso si no ejecutó el comando con la opción <code>--database</code> .</p>

Nombre	Descripción
<p><code>-db.size</code> <i>número_de_gigabytes</i></p> <p>o bien</p> <p><code>--database.size</code> <i>número_de_gigabytes</i></p>	<p>Número de gigabytes (GB) que se van a asignar al almacenamiento de la base de datos. Si ejecuta <code>eb create</code> con esta opción, la CLI de EB lanza el entorno con una base de datos adjuntada. Este es el caso incluso si no ejecutó el comando con la opción <code>--database</code>.</p> <p>Tipo: Number</p> <p>Valores válidos:</p> <ul style="list-style-type: none"> • MySQL: de 5 a 1024. El valor predeterminado es 5. • Postgres: de 5 a 1024. El valor predeterminado es 5. • Oracle: de 10 a 1024. El valor predeterminado es 10. • Microsoft SQL Server Express Edition – 30. • Microsoft SQL Server Web Edition – 30. • Microsoft SQL Server Standard Edition – 200.
<p><code>-db.user</code> <i>nombreDeUsuario</i></p> <p>o bien</p> <p><code>--database.username</code> <i>nombreDeUsuario</i></p>	<p>Nombre de usuario de la base de datos. Si ejecuta <code>eb create</code> con esta opción, la CLI de EB lanza el entorno con una base de datos adjuntada aunque el comando no se ejecute con la opción <code>--database</code>. Si ejecuta <code>eb create</code> con la opción <code>--database</code> pero sin las opciones <code>--database.username</code> y <code>--database.password</code>, la CLI de EB le pide el nombre de usuario maestro y la contraseña maestra de la base de datos.</p>
<p><code>-db.version</code> <i>version</i></p> <p>o bien</p> <p><code>--database.version</code> <i>version</i></p>	<p>Utilizado para especificar la versión del motor de base de datos. Si esta marca está presente, el entorno se lanzará con una base de datos que tiene el número de versión especificado, aunque no se incluya la marca <code>--database</code>.</p>

Nombre	Descripción
<code>--elb-type</code> <i>Valor type de</i>	El tipo de balanceador de carga . Tipo: String Valores válidos: classic, application , network Valor predeterminado: application
<code>-es</code> o bien <code>--enable-spot</code>	Habilite las solicitudes de instancias de spot para su entorno. Para obtener más información, consulte Grupo de Auto Scaling . Opciones relacionadas: <ul style="list-style-type: none"> • <code>--instance-types</code> • <code>--on-demand-base-capacity</code> • <code>--on-demand-above-base-capacity</code> • <code>--spot-max-price</code>
<code>--env-group-suffix</code> <i>groupname</i>	El nombre del grupo que se agrega nombre del entorno. Solo se utiliza con Compose Environments .
<code>--envvars</code>	Propiedades del entorno en una lista separada por comas con el formato <i>nombre=valor</i> . Consulte Configuración de las propiedades del entorno (variables de entorno) para ver los límites.
<code>-ip</code> <i>nombre_de_perfil</i> o bien <code>--instance_profile</code> <i>nombre_de_perfil</i>	El perfil de instancia con la función de IAM con las credenciales de seguridad temporales que la aplicación necesita para acceder a AWS los recursos.


Nombre	Descripción
<pre>-it o --instance- types <i>type1</i>[,<i>type2</i> ...]</pre>	<p>Una lista separada por comas de los tipos de Amazon EC2 instance que desea utilizar en el entorno. Si no especifica esta opción, Elastic Beanstalk proporciona tipos de instancia predeterminados.</p> <p>Para obtener más información, consulte Instancias de Amazon EC2 y Grupo de Auto Scaling.</p> <div data-bbox="688 575 1507 982" style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; background-color: #fff9e6;"> <p> Important</p> <p>La CLI de EB sólo aplica esta opción a instancias Spot. A menos que esta opción se use con la opción <code>--enable-spot</code>, la CLI de EB la ignora. Para especificar un tipo de instancia para una instancia en diferido, utilice la herramienta <code>--instance-type</code> (sin “s”) en su lugar.</p> </div>
<pre>-i o bien --instance_type</pre>	<p>El tipo de Amazon EC2 instance que desea que se utilice en el entorno. Si no especifica esta opción, Elastic Beanstalk proporciona un tipo de instancia predeterminado.</p> <p>Para obtener más información, consulte Instancias de Amazon EC2.</p> <div data-bbox="688 1373 1507 1822" style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; background-color: #fff9e6;"> <p> Important</p> <p>La CLI de EB sólo aplica esta opción a instancias en diferido. No utilice esta opción con la opción <code>--enable-spot</code>, porque la CLI de EB la ignora cuando lo hace. Para especificar tipos de instancia para una instancia Spot, utilice la herramienta <code>--instance-types</code> (sin “s”) en su lugar.</p> </div>

Nombre	Descripción
<p><code>-k <i>nombre_de_clave</i></code></p> <p>o bien</p> <p><code>--keyname <i>nombre_de_clave</i></code></p>	<p>Nombre del par de claves de Amazon EC2 que se va a utilizar con el cliente de Secure Shell (SSH) para iniciar sesión de forma segura en las instancias de Amazon EC2 que ejecutan su aplicación Elastic Beanstalk. Si incluye esta opción con el comando <code>eb create</code>, el valor que proporcione anulará cualquier nombre de clave que haya especificado con <code>eb init</code>.</p> <p>Valores válidos: nombre de clave existente que esté registrado con Amazon EC2.</p>
<p><code>-im <i>number-of-instances</i></code></p> <p>o</p> <p><code>--min-instances <i>number-of-instances</i></code></p>	<p>El número mínimo de instancias de Amazon EC2 necesarias para su entorno.</p> <p>Tipo: número (entero)</p> <p>Valor predeterminado: 1</p> <p>Valores válidos: de 1 a 10000</p>
<p><code>-ix <i>number-of-instances</i></code></p> <p>o</p> <p><code>--max-instances <i>number-of-instances</i></code></p>	<p>El número máximo de instancias de Amazon EC2 que permite para su entorno.</p> <p>Tipo: número (entero)</p> <p>Valor predeterminado: 4</p> <p>Valores válidos: de 1 a 10000</p>
<p><code>--modules <i>componente-a</i> <i>componente-b</i></code></p>	<p>Una lista de los entornos de componentes que se van a crear. Solo se utiliza con Entornos compuestos.</p>

Nombre	Descripción
<p>-sb</p> <p>o bien</p> <p>--on-demand-base-capacity</p>	<p>Número mínimo de instancias en diferido que el grupo de Auto Scaling aprovisiona antes de considerar la posibilidad de utilizar instancias de spot cuando se amplía el entorno.</p> <p>Esta opción solo se puede especificar con --enable-spot . Para obtener más información, consulte Grupo de Auto Scaling.</p> <p>Tipo: número (entero)</p> <p>Valor predeterminado: 0</p> <p>Valores válidos: de 0 a --max-instances (cuando no hay ningún valor: opción MaxSize del espacio de nombres aws:autoscaling:asg)</p>
<p>-sp</p> <p>o bien</p> <p>--on-demand-above-base-capacity</p>	<p>El porcentaje de instancias en diferido como parte de la capacidad adicional que el grupo de Auto Scaling aprovisiona además del número de instancias especificado mediante la opción --on-demand-base-capacity .</p> <p>Esta opción solo se puede especificar con --enable-spot . Para obtener más información, consulte Grupo de Auto Scaling.</p> <p>Tipo: número (entero)</p> <p>Valor predeterminado: 0 para un entorno de una sola instancia; 70 para un entorno con balanceo de carga</p> <p>Valores válidos: de 0 a 100</p>


Nombre	Descripción
<p><code>-p <i>platform-version</i></code></p> <p>o bien</p> <p><code>--platform <i>platform-version</i></code></p>	<p>La versión de la plataforma que se va a usar. Puede especificar una plataforma, una plataforma y la versión, una rama de la plataforma, el nombre de una pila de soluciones o el ARN de una pila de soluciones. Por ejemplo:</p> <ul style="list-style-type: none"> • <code>php, PHP, node.js</code>: la última versión de plataforma para la plataforma especificada • <code>php-7.2, "PHP 7.2"</code>: la versión recomendada (normalmente la más reciente) de la plataforma PHP 7.2 • <code>"PHP 7.2 running on 64bit Amazon Linux"</code>: la versión recomendada (normalmente la más reciente) de la plataforma PHP de esta rama de la plataforma • <code>"64bit Amazon Linux 2017.09 v2.6.3 running PHP 7.1"</code>: la versión de la plataforma PHP especificada por el nombre de la pila de esta solución • <code>"arn:aws:elasticbeanstalk:us-east-2:platform/PHP 7.1 running on 64bit Amazon Linux/2.6.3"</code> : la versión de la plataforma PHP especificada por el ARN de la pila de esta solución <p>Utilice eb platform list para obtener una lista con las configuraciones disponibles.</p> <p>Si especifica la opción <code>--platform</code> , anula el valor que se proporcionó durante <code>eb init</code>.</p>
<p><code>-pr</code></p> <p>o bien</p> <p><code>--process</code></p>	<p>Procesar previamente y validar el manifiesto del entorno y los archivos de configuración del paquete de código fuente. La validación de los archivos de configuración puede identificar los problemas antes de implementar la versión de la aplicación en un entorno.</p>

Nombre	Descripción
<code>-r <i>región</i></code> o bien <code>--region <i>región</i></code>	La AWS región en la que desea implementar la aplicación. Para obtener la lista de valores que puede especificar para esta opción, consulte Puntos de enlace y cuotas de AWS Elastic Beanstalk en la Referencia general de AWS.
<code>--sample</code>	Implemente la aplicación de ejemplo en el nuevo entorno en lugar del código del repositorio.
<code>--scale <i>number-of-instances</i></code>	Lance el número especificado de instancias.
<code>--service-role <i>rolDeServicio</i></code>	Asigne al entorno un rol de servicio que no sea el predeterminado.

 **Note**

No ingrese un ARN. Ingrese sólo el nombre de rol. Elastic Beanstalk agrega como prefijo los valores correctos al nombre de rol para crear internamente el ARN resultante.

Nombre	Descripción
<p><code>-l</code> <i>balanceador de carga</i></p> <p>o bien</p> <p><code>--shared-lb</code> <i>balanceador de carga</i></p>	<p>Configure el entorno para utilizar un balanceador de carga compartido. Proporcione el nombre o ARN de un balanceador de carga compatible en su cuenta, un Application Load Balancer que haya creado explícitamente, no uno creado por otro entorno de Elastic Beanstalk. Para obtener más información, consulte Balanceador de carga de aplicaciones compartido.</p> <p>Ejemplos de parámetros:</p> <ul style="list-style-type: none">• <code>FrontEndLB</code> : nombre de un balanceador de carga• <code>arn:aws:elasticloadbalancing:us-east-2:123456789012:loadbalancer/app/FrontEndLB/0dbf78d8ad96abbc</code> : un ARN de Application Load Balancer. <p>Puede especificar esta opción solo con <code>--elb-type application</code> . Si especifica esa opción y no especifica <code>--shared-lb</code> , Elastic Beanstalk crea un balanceador de carga dedicado para el entorno.</p>

Nombre	Descripción
<p><code>-lp <i>puerto</i></code></p> <p>o bien</p> <p><code>--shared-lb-port <i>puerto</i></code></p>	<p>Puerto de agente de escucha predeterminado del balanceador de carga compartido para este entorno. Elastic Beanstalk agrega una regla de escucha que enruta todo el tráfico de este agente de escucha al proceso de entorno predeterminado. Para obtener más información, consulte Balanceador de carga de aplicaciones compartido.</p> <p>Tipo: número (entero)</p> <p>Valor predeterminado: 80</p> <p>Valores válidos: cualquier entero que represente un puerto de agente de escucha del balanceador de carga compartido.</p>
<p><code>--single</code></p>	<p>Cree el entorno con una sola Amazon EC2 instance y sin un balanceador de carga.</p> <div data-bbox="688 1052 1507 1654" style="border: 1px solid #f08080; border-radius: 10px; padding: 10px;"><p> Warning</p><p>Un entorno de una sola instancia no está preparado para producción. Si la instancia se vuelve inestable durante la implementación o Elastic Beanstalk termina y reinicia la instancia durante una actualización de la configuración, es posible que la aplicación no esté disponible durante algún tiempo. Utilice entornos de una sola instancia para desarrollo, pruebas o ensayos. Utilice entornos con balanceo de carga para producción.</p></div>

Nombre	Descripción
<p>-sm</p> <p>o bien</p> <p>--spot-max-price</p>	<p>El precio máximo unidad por hora, en dólares estadounidenses, que está dispuesto a pagar por una instancia de Spot.</p> <p>Esta opción solo se puede especificar con <code>--enable-spot</code>. Para obtener más información, consulte Grupo de Auto Scaling.</p> <p>Tipo: número (float)</p> <p>Predeterminado: precio en diferido, para cada tipo de instancias. En este caso, el valor de la opción es <code>null</code>.</p> <p>Valores válidos: de <code>0.001</code> a <code>20.0</code></p> <p>Para obtener recomendaciones sobre las opciones de precio máximo para las instancias puntuales, consulte el historial de precios de las instancias puntuales en la Guía del usuario de Amazon EC2.</p>
<p>--tags <i>key1=value1[,key2=va</i></p>	<p>Etiquete los recursos del entorno. Las etiquetas se especifican como una lista de pares <code>key=value</code> separados por comas.</p> <p>Para obtener más información, consulte Etiquetado de entornos.</p>
<p>-t worker</p> <p>o bien</p> <p>--tier worker</p>	<p>Cree un entorno de trabajo. Omita esta opción para crear un entorno de servidor web.</p>
<p>--timeout <i>minutos</i></p>	<p>Establezca el número de minutos que deben transcurrir antes de que se agote el tiempo de espera del comando.</p>

Nombre	Descripción
<code>--version</code> <i>etiqueta_de_versión</i>	<p>Especifica la versión de la aplicación que se va a implementar en el entorno en lugar del código fuente de la aplicación del directorio de proyectos local.</p> <p>Tipo: String</p> <p>Valores válidos: etiqueta de la versión de la aplicación existente.</p>
<code>--vpc</code>	<p>Configure una VPC para el entorno. Cuando se incluye esta opción, la CLI de EB le pide que configure todos los ajustes obligatorios antes de lanzar el entorno.</p>
<code>--vpc.dbsubnets</code> <i>subnet1, subnet2</i>	<p>Especifica subredes para las instancias de base de datos de una VPC. Es obligatorio cuando se especifica <code>--vpc.id</code>.</p>
<code>--vpc.ec2subnets</code> <i>subnet1, subnet2</i>	<p>Especifica subredes para las instancias Amazon EC2 de una VPC. Es obligatorio cuando se especifica <code>--vpc.id</code>.</p>
<code>--vpc.elbpublic</code>	<p>Inicia el balanceador de carga Elastic Load Balancing en una subred pública de la VPC.</p> <p>No puede especificar esta opción con las opciones <code>--tier worker</code> o <code>--single</code>.</p>
<code>--vpc.elbsubnets</code> <i>subnet1, subnet2</i>	<p>Especifica subredes para el balanceador de carga Elastic Load Balancing en una VPC.</p> <p>No puede especificar esta opción con las opciones <code>--tier worker</code> o <code>--single</code>.</p>
<code>--vpc.id</code> <i>ID</i>	<p>Lanza el entorno en la VPC especificada.</p>

Nombre	Descripción
<code>--vpc.publicip</code>	Lanza instancias de Amazon EC2 en una subred pública de la VPC. No puede especificar esta opción con la opción <code>--tier worker</code> .
<code>--vpc.securitygroups <i>securitygroup1, securitygroup2</i></code>	Especifica los ID de grupo de seguridad. Es obligatorio cuando se especifica <code>--vpc.id</code> .
Opciones comunes	

Salida

Si se ejecuta correctamente, el comando le hará preguntas y después devolverá el estado de la operación de creación. Si hubiera algún problema durante el lanzamiento, utilice la operación [eb events](#) para obtener más detalles.

Si habilitó el CodeBuild soporte en su aplicación, `eb create` muestra la información a CodeBuild medida que se crea el código. Para obtener información sobre el CodeBuild soporte de Elastic Beanstalk, consulte. [Uso de la CLI de EB con AWS CodeBuild](#)

Ejemplos

En el siguiente ejemplo, se crea un entorno en modo interactivo.

```
$ eb create
Enter Environment Name
(default is tmp-dev): ENTER
Enter DNS CNAME prefix
(default is tmp-dev): ENTER
Select a load balancer type
1) classic
2) application
3) network
(default is 2): ENTER
Environment details for: tmp-dev
  Application name: tmp
```

```
Region: us-east-2
Deployed Version: app-141029_145448
Environment ID: e-um3yfrzq22
Platform: 64bit Amazon Linux 2014.09 v1.0.9 running PHP 5.5
Tier: WebServer-Standard-1.0
CNAME: tmp-dev.elasticbeanstalk.com
Updated: 2014-10-29 21:54:51.063000+00:00
Printing Status:
...
```

En el siguiente ejemplo también se crea un entorno en modo interactivo. En este ejemplo, su directorio de proyecto no tiene código de aplicación. El comando implementa una aplicación de ejemplo y la descarga en el directorio de proyectos local.

```
$ eb create
Enter Environment Name
(default is tmp-dev): ENTER
Enter DNS CNAME prefix
(default is tmp-dev): ENTER
Select a load balancer type
1) classic
2) application
3) network
(default is 2): ENTER
NOTE: The current directory does not contain any source code. Elastic Beanstalk is
launching the sample application instead.
Do you want to download the sample application into the current directory?
(Y/n): ENTER
INFO: Downloading sample application to the current directory.
INFO: Download complete.
Environment details for: tmp-dev
Application name: tmp
Region: us-east-2
Deployed Version: Sample Application
Environment ID: e-um3yfrzq22
Platform: 64bit Amazon Linux 2014.09 v1.0.9 running PHP 5.5
Tier: WebServer-Standard-1.0
CNAME: tmp-dev.elasticbeanstalk.com
Updated: 2017-11-08 21:54:51.063000+00:00
Printing Status:
...
```

El siguiente comando crea un entorno sin plantear ninguna pregunta.


```
$ eb create dev-env
Creating application version archive "app-160312_014028".
Uploading test/app-160312_014028.zip to S3. This may take a while.
Upload Complete.
Application test has been created.
Environment details for: dev-env
  Application name: test
  Region: us-east-2
  Deployed Version: app-160312_014028
  Environment ID: e-6fgpkjxyyi
  Platform: 64bit Amazon Linux 2015.09 v2.0.8 running PHP 5.6
  Tier: WebServer-Standard
  CNAME: UNKNOWN
  Updated: 2016-03-12 01:40:33.614000+00:00
Printing Status:
...
```

El siguiente comando crea un entorno en una VPC personalizada.

```
$ eb create dev-vpc --vpc.id vpc-0ce8dd99 --vpc.elbsubnets subnet-
b356d7c6,subnet-02f74b0c --vpc.ec2subnets subnet-0bb7f0cd,subnet-3b6697c1 --
vpc.securitygroup sg-70cff265
Creating application version archive "app-160312_014309".
Uploading test/app-160312_014309.zip to S3. This may take a while.
Upload Complete.
Environment details for: dev-vpc
  Application name: test
  Region: us-east-2
  Deployed Version: app-160312_014309
  Environment ID: e-pqkqip3mns
  Platform: 64bit Amazon Linux 2015.09 v2.0.8 running Java 8
  Tier: WebServer-Standard
  CNAME: UNKNOWN
  Updated: 2016-03-12 01:43:14.057000+00:00
Printing Status:
...
```

eb deploy

Descripción

Implementa el paquete de código fuente de la aplicación desde el directorio del proyecto inicializado en la aplicación en ejecución.

Si se instala Git, la CLI de EB utiliza el comando `git archive` para crear un archivo `.zip` a partir del contenido del comando `git commit` más reciente.

Sin embargo, cuando `.ebignore` se encuentra en su directorio de proyecto, la CLI de EB no utiliza comandos ni semántica de git para crear el paquete de código fuente. Esto significa que la CLI de EB omite los archivos especificados en `.ebignore` e incluye todos los demás archivos. En concreto, incluye archivos de código fuente sin validar.

Note

Puede configurar la CLI de EB para implementar un artefacto de su proceso de compilación en lugar de crear un archivo ZIP de la carpeta del proyecto. Para obtener más información, consulte [Implementar un artefacto en lugar de la carpeta del proyecto](#).

Sintaxis

```
eb deploy
```

```
eb deploy environment-name
```

Opciones

Nombre	Descripción
<code>-l <i>etiqueta_de_versión</i></code> <code>n</code>	Especifique una etiqueta para usarla con la versión que crea la CLI de EB. Si la etiqueta ya se ha utilizado, la CLI de EB vuelve a implementar la versión anterior a dicha etiqueta.
o bien	Tipo: String
<code>--label <i>etiqueta_de_versión</i></code>	

Nombre	Descripción
<code>--env-group-suffix</code> <i>groupname</i>	Nombre del grupo que se agrega al nombre del entorno. Solo se utiliza con Compose Environments .
<code>-m "descripción_de_ver sión "</code> o bien <code>--message "descripci ón_de_versión "</code>	La descripción de la versión de la aplicación, incluida entre comillas dobles. Tipo: String
<code>--modules</code> <i>componente-a</i> <i>componente-b</i>	Lista de componentes que se van a actualizar. Solo se utiliza con Compose Environments .
<code>-p</code> o bien <code>--process</code>	Procesar previamente y validar el manifiesto del entorno y los archivos de configuración del paquete de código fuente. La validación de los archivos de configuración puede identificar los problemas antes de implementar la versión de la aplicación en un entorno.
<code>--source</code> codecommit/ <i>repository-</i> <i>name/branch-name</i>	Repositorio y ramificación de CodeCommit. Consulte Uso de la CLI de EB con AWS CodeCommit .
<code>--staged</code>	Implementar archivos almacenados en el índice de git en lugar de la validación HEAD.
<code>--timeout</code> <i>minutos</i>	Número de minutos que deben transcurrir antes de que se agote el tiempo de espera del comando.
<code>--version</code> <i>etiqueta_ de_versión</i>	La versión de la aplicación existente que se va a implementar. Tipo: String
Opciones comunes	

Salida

Si se ejecuta correctamente, el comando devuelve el estado de la operación `deploy`.

Si habilitó la compatibilidad con CodeBuild en la aplicación, `eb deploy` mostrará información de CodeBuild cuando se compile el código. Para obtener información sobre la compatibilidad con CodeBuild en Elastic Beanstalk, consulte [Uso de la CLI de EB con AWS CodeBuild](#).

Ejemplo

En el ejemplo siguiente, se implementa la aplicación actual.

```
$ eb deploy
2018-07-11 21:05:22      INFO: Environment update is starting.
2018-07-11 21:05:27      INFO: Deploying new version to instance(s).
2018-07-11 21:05:53      INFO: New application version was deployed to running EC2
instances.
2018-07-11 21:05:53      INFO: Environment update completed successfully.
```

eb events

Descripción

Muestra los eventos más recientes del entorno.

Si el directorio raíz contiene un archivo `platform.yaml` donde se especifica una plataforma personalizada, este comando también devuelve los eventos más recientes del entorno del constructor.

Sintaxis

```
eb events
```

```
eb events environment-name
```

Opciones

Nombre	Descripción
<code>-f</code>	Muestra secuencias de eventos. Para cancelar, pulse CTRL+C.

Nombre	Descripción
o bien	
--follow	

Salida

Si se ejecuta correctamente, el comando devuelve los eventos recientes.

Ejemplo

En el siguiente ejemplo, se devuelven los eventos más recientes.

```
$ eb events
2014-10-29 21:55:39      INFO      createEnvironment is starting.
2014-10-29 21:55:40      INFO      Using elasticbeanstalk-us-west-2-111122223333 as Amazon
S3 storage bucket for environment data.
2014-10-29 21:55:57      INFO      Created load balancer named: awseb-e-r-AWSEBLoa-
NSKU0K5X6Z9J
2014-10-29 21:56:16      INFO      Created security group named: awseb-e-rxgrhjr9bx-stack-
AWSEBSecurityGroup-1UUHU5LZ20ZY7
2014-10-29 21:57:18      INFO      Waiting for EC2 instances to launch. This may take a
few minutes.
2014-10-29 21:57:18      INFO      Created Auto Scaling group named: awseb-e-rxgrhjr9bx-
stack-AWSEBAutoScalingGroup-1TE320ZCJ9RPD
2014-10-29 21:57:22      INFO      Created Auto Scaling group policy named:
arn:aws:autoscaling:us-east-2:11122223333:scalingPolicy:2cced9e6-859b-421a-
be63-8ab34771155a:autoScalingGroupName/awseb-e-rxgrhjr9bx-stack-
AWSEBAutoScalingGroup-1TE320ZCJ9RPD:policyName/awseb-e-rxgrhjr9bx-stack-
AWSEBAutoScalingScaleUpPolicy-1I2ZSNVU4APRY
2014-10-29 21:57:22      INFO      Created Auto Scaling group policy named:
arn:aws:autoscaling:us-east-2:11122223333:scalingPolicy:1f08b863-
bf65-415a-b584-b7fa3a69a0d5:autoScalingGroupName/awseb-e-rxgrhjr9bx-stack-
AWSEBAutoScalingGroup-1TE320ZCJ9RPD:policyName/awseb-e-rxgrhjr9bx-stack-
AWSEBAutoScalingScaleDownPolicy-1E3G7PZKZPS0G
2014-10-29 21:57:25      INFO      Created CloudWatch alarm named: awseb-e-rxgrhjr9bx-
stack-AWSEBCloudwatchAlarmLow-VF5EJ549FZBL
2014-10-29 21:57:25      INFO      Created CloudWatch alarm named: awseb-e-rxgrhjr9bx-
stack-AWSEBCloudwatchAlarmHigh-LA9YEW306WJ0
2014-10-29 21:58:50      INFO      Added EC2 instance 'i-c7ee492d' to Auto ScalingGroup
'awseb-e-rxgrhjr9bx-stack-AWSEBAutoScalingGroup-1TE320ZCJ9RPD'.
```

```
2014-10-29 21:58:53 INFO Successfully launched environment: tmp-dev
2014-10-29 21:59:14 INFO Environment health has been set to GREEN
2014-10-29 21:59:43 INFO Adding instance 'i-c7ee492d' to your environment.
```

eb health

Descripción

Devuelve el estado más reciente del entorno.

Si el directorio raíz contiene un archivo `platform.yaml` donde se especifica una plataforma personalizada, este comando también devuelve el estado más reciente del entorno del constructor.

Sintaxis

```
eb health
```

```
eb health environment-name
```

Opciones

Nombre	Descripción
<code>-r</code> o bien <code>--refresh</code>	Muestra información de estado de forma interactiva y la actualiza cada 10 segundos cuando se registra nueva información.
<code>--mono</code>	No muestra ningún color en la salida.

Salida

Si se ejecuta correctamente, el comando devuelve el estado reciente.

Ejemplo

En el siguiente ejemplo se devuelve la información de estado más reciente para un entorno Linux.

```
~/project $ eb health
```

```

elasticBeanstalkExa-env                               Ok
2015-07-08 23:13:20
WebServer
Ruby 2.1 (Puma)
total          ok      warning  degraded  severe  info  pending  unknown
    5          5         0         0         0         0         0         0

instance-id    status    cause
health
Overall        Ok
i-d581497d     Ok
i-d481497c     Ok
i-136e00c0    Ok
i-126e00c1    Ok
i-8b2cf575    Ok

instance-id    r/sec    %2xx    %3xx    %4xx    %5xx    p99    p90    p75    p50
p10
Overall        671.8    100.0    0.0     0.0     0.0     0.003  0.002  0.001  0.001
0.000
i-d581497d     143.0    1430    0        0        0        0.003  0.002  0.001  0.001
0.000
i-d481497c     128.8    1288    0        0        0        0.003  0.002  0.001  0.001
0.000
i-136e00c0    125.4    1254    0        0        0        0.004  0.002  0.001  0.001
0.000
i-126e00c1    133.4    1334    0        0        0        0.003  0.002  0.001  0.001
0.000
i-8b2cf575    141.2    1412    0        0        0        0.003  0.002  0.001  0.001
0.000

instance-id    type     az    running    load 1  load 5    user%  nice%  system%
idle%  iowait%
i-d581497d     t2.micro  1a    12 mins    0.0     0.04     6.2    0.0    1.0
92.5    0.1
i-d481497c     t2.micro  1a    12 mins    0.01    0.09     5.9    0.0    1.6
92.4    0.1
i-136e00c0    t2.micro  1b    12 mins    0.15    0.07     5.5    0.0    0.9
93.2    0.0
i-126e00c1    t2.micro  1b    12 mins    0.17    0.14     5.7    0.0    1.4
92.7    0.1
i-8b2cf575    t2.micro  1c    1 hour     0.19    0.08     6.5    0.0    1.2
92.1    0.1
    
```

instance-id	status	id	version deployments	ago
i-d581497d	Deployed	1	Sample Application	12 mins
i-d481497c	Deployed	1	Sample Application	12 mins
i-136e00c0	Deployed	1	Sample Application	12 mins
i-126e00c1	Deployed	1	Sample Application	12 mins
i-8b2cf575	Deployed	1	Sample Application	1 hour

eb init

Descripción

Establece valores predeterminados para las aplicaciones de Elastic Beanstalk que se han creado con la CLI de EB a través de una serie de preguntas.

Note

Los valores que se establecen con `eb init` se aplican únicamente al directorio y repositorio actuales del equipo actualmente en uso.

El comando no crea nada en su cuenta de Elastic Beanstalk. Para crear un entorno de Elastic Beanstalk, ejecute [eb create](#) después de ejecutarlo `eb init`.

Sintaxis

`eb init`


`eb init nombre-de-aplicación`

Opciones

Si ejecuta `eb init` sin especificar la opción `--platform`, la CLI de EB le pedirá que especifique un valor para cada ajuste.

Note

Si desea utilizar `eb init` para crear un nuevo par de claves, `ssh-keygen` debe estar instalado en la máquina local y disponible desde la línea de comando.

Nombre	Descripción	
<p><code>-i</code></p> <p><code>--interactive</code></p>	<p>Obliga a la CLI de EB a pedirle un valor para cada opción del comando <code>eb init</code>.</p> <div data-bbox="521 373 1304 930" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>El comando <code>init</code> le pide que proporcione valores para las opciones del comando <code>eb init</code> que no tienen ningún valor (predeterminado). Después de ejecutar por primera vez el comando <code>eb init</code> en un directorio, es posible que la CLI de EB no le pida valores para las opciones del comando. En este caso, utilice la opción <code>--interactive</code> cuando desee cambiar un ajuste que configuró previamente.</p> </div>	
<p><code>-k <i>nombre-de-clave</i></code></p> <p><code>--keyname <i>nombre-de-clave</i></code></p>	<p>Nombre del par de claves de Amazon EC2 que se va a utilizar con el cliente de Secure Shell (SSH) para iniciar sesión de forma segura en las instancias de Amazon EC2 que ejecutan la aplicación Elastic Beanstalk.</p>	
<p><code>--modules <i>folder-1</i></code> <code><i>folder-2</i></code></p>	<p>Muestra los directorios secundarios que se van a inicializar. Solo se utiliza con Compose Environments.</p>	
<p><code>-p <i>platform-version</i></code></p> <p><code>--platform <i>platform-version</i></code></p>	<p>La versión de la plataforma que se va a usar. Puede especificar una plataforma, una plataforma y la versión, una rama de la plataforma, el nombre de una pila de soluciones o el ARN de una pila de soluciones. Por ejemplo:</p> <ul style="list-style-type: none"> • <code>php</code>, <code>PHP</code>, <code>node.js</code> – la última versión de plataforma para la plataforma especificada • <code>php-7.2</code>, <code>"PHP 7.2"</code>: la versión recomendada (normalmente la más reciente) de la plataforma PHP 7.2 	

Nombre	Descripción	
	<ul style="list-style-type: none"> • "PHP 7.2 running on 64bit Amazon Linux": la versión recomendada (normalmente la más reciente) de la plataforma PHP de esta rama de la plataforma • "64bit Amazon Linux 2017.09 v2.6.3 running PHP 7.1": la versión de la plataforma PHP especificada por el nombre de la pila de esta solución • "arn:aws:elasticbeanstalk:us-east-2:platform/PHP 7.1 running on 64bit Amazon Linux/2.6.3" : la versión de la plataforma PHP especificada por el ARN de la pila de esta solución <p>Utilice eb platform list para obtener una lista con las configuraciones disponibles.</p> <p>Especifique la opción <code>--platform</code> para omitir la configuración interactiva.</p> <div data-bbox="521 1167 1304 1581" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p>Note</p> <p>Si especifica esta opción, la CLI de EB no le pedirá valores para ninguna otra opción. En su lugar, presupondrá los valores predeterminados para cada opción. Puede especificar todas las opciones para las que no desee utilizar valores predeterminados.</p> </div>	
<pre>--source codecommit/ <i>repository-name/branch-name</i></pre>	<p>Repositorio y ramificación de CodeCommit. Consulte Uso de la CLI de EB con AWS CodeCommit.</p>	

Nombre	Descripción
<code>--tags</code> <i>key1=value1</i>	Etiquete su aplicación. Las etiquetas se especifican como una lista de pares <code>key=value</code> separados por comas. Para obtener más información, consulte Etiquetado de aplicaciones .
Opciones comunes	

Compatibilidad con CodeBuild

Si ejecuta `eb init` en una carpeta que contiene un archivo [buildspec.yml](#) Elastic Beanstalk analiza el archivo en busca de una entrada `eb_codebuild_settings` con opciones específicas de Elastic Beanstalk. Para obtener información sobre la compatibilidad con CodeBuild en Elastic Beanstalk, consulte [Uso de la CLI de EB con AWS CodeBuild](#).

Salida

Si se ejecuta correctamente, el comando le guía por el proceso de configuración de una nueva aplicación de Elastic Beanstalk a través una serie de mensajes.

Ejemplo

La solicitud de ejemplo siguiente inicializa la CLI de EB y le pide que especifique información sobre la aplicación. Sustituya el texto del *marcador de posición* por sus propios valores.

```
$ eb init -i
Select a default region
1) us-east-1 : US East (N. Virginia)
2) us-west-1 : US West (N. California)
3) us-west-2 : US West (Oregon)
4) eu-west-1 : Europe (Ireland)
5) eu-central-1 : Europe (Frankfurt)
6) ap-south-1 : Asia Pacific (Mumbai)
7) ap-southeast-1 : Asia Pacific (Singapore)
...
(default is 3): 3
```

```
Select an application to use
1) HelloWorldApp
2) NewApp
3) [ Create new Application ]
(default is 3): 3

Enter Application Name
(default is "tmp"):
Application tmp has been created.

It appears you are using PHP. Is this correct?
(y/n): y

Select a platform branch.
1) PHP 7.2 running on 64bit Amazon Linux
2) PHP 7.1 running on 64bit Amazon Linux (Deprecated)
3) PHP 7.0 running on 64bit Amazon Linux (Deprecated)
4) PHP 5.6 running on 64bit Amazon Linux (Deprecated)
5) PHP 5.5 running on 64bit Amazon Linux (Deprecated)
6) PHP 5.4 running on 64bit Amazon Linux (Deprecated)
(default is 1): 1

Do you want to set up SSH for your instances?
(y/n): y

Select a keypair.
1) aws-eb
2) [ Create new KeyPair ]
(default is 2): 1
```

eb labs

Descripción

Los comandos secundarios de eb labs admiten una funcionalidad experimental o en proceso. Estos comandos pueden eliminarse o modificarse en futuras versiones de la CLI de EB y no se garantiza su compatibilidad con versiones posteriores.

Para ver una lista de los comandos secundarios disponibles y sus descripciones, ejecute `eb labs --help`.

eb list

Descripción

Muestra todos los entornos de la aplicación actual o todos los entornos de todas las aplicaciones, según se especifique en la opción `--all`.

Si el directorio raíz contiene un archivo `platform.yaml` que especifica una plataforma personalizada, este comando también muestra los entornos de constructor.

Sintaxis

```
eb list
```

Opciones

Nombre	Descripción
<code>-a</code> o bien <code>--all</code>	Muestra todos los entornos de todas las aplicaciones.
<code>-v</code> o bien <code>--verbose</code>	Proporciona información más detallada sobre todos los entornos, incluidas las instancias.
Opciones comunes	

Salida

Si se ejecuta correctamente, el comando devuelve una lista de nombres de entorno en la que su entorno actual está marcado con un asterisco (*).

Ejemplo 1

El siguiente ejemplo muestra sus entornos e indica que `tmp-dev` es su entorno predeterminado.

```
$ eb list
* tmp-dev
```

Ejemplo 2

El siguiente ejemplo muestra sus entornos con detalles adicionales.

```
$ eb list --verbose
Region: us-west-2
Application: tmp
  Environments: 1
    * tmp-dev : ['i-c7ee492d']
```

eb local

Descripción

Utilice `eb local run` para ejecutar localmente los contenedores de la aplicación en Docker. Compruebe el estado del contenedor de la aplicación con `eb local status`. Abra la aplicación en un navegador web con `eb local open`. Recupere la ubicación de los registros de la aplicación con `eb local logs`.

`eb local setenv` y `eb local printenv` le permiten configurar y ver las variables de entorno proporcionadas para los contenedores de Docker que se ejecutaron localmente con `eb local run`.

Debe ejecutar todos los comandos `eb local` del directorio del proyecto de una aplicación de Docker que se haya inicializado como un repositorio de la CLI de EB mediante `eb init`.

Note

Utilice `eb local` en un equipo local que ejecute Linux o macOS. El comando no admite Windows.

Antes de utilizar el comando en macOS, instale Docker para Mac, y asegúrese de que `boot2docker` no está instalado (o no está en la ruta de ejecución). El comando `eb local` intenta utilizar `boot2docker` si está presente, pero no funciona correctamente con él en macOS.

Sintaxis

`eb local run`

`eb local status`

`eb local open`

`eb local logs`

`eb local setenv`

`eb local printenv`

Opciones

`eb local run`

Nombre	Descripción
<code>--envvars <i>key1=value1,key2=value2</i></code>	Define variables de entorno que la CLI de EB pasará a los contenedores locales de Docker. En los entornos con varios contenedores, todas las variables se pasan a todos los contenedores.
<code>--port <i>hostport</i></code>	Asigna un puerto del host al puerto expuesto en el contenedor. Si no se especifica esta opción, la CLI de EB utiliza el mismo puerto en el host y el contenedor. Esta opción solo funciona con aplicaciones de plataforma Docker. No se aplica a la plataforma Multicontainer Docker.
Opciones comunes	

`eb local status`

`eb local open`

`eb local logs`

`eb local setenv`

`eb local printenv`

Nombre	Descripción
Opciones comunes	

Salida

`eb local run`

Mensajes de estado de Docker. Permanece activo mientras se ejecuta la aplicación. Presione Ctrl+C para detener la aplicación.

`eb local status`

Estado de cada uno de los contenedores que utiliza la aplicación, estén en ejecución o no.

`eb local open`

Abre la aplicación en un navegador web o sale.

`eb local logs`

Ubicación de los registros generados en el directorio del proyecto por las aplicaciones que se ejecutan localmente con `eb local run`.

`eb local setenv`

Ninguno

`eb local printenv`

Nombre y valores de las variables de entorno configuradas con `eb local setenv`.

Ejemplos

`eb local run`


```
~/project$ eb local run  
Creating elasticbeanstalk_phpapp_1...  
Creating elasticbeanstalk_nginxproxy_1...  
Attaching to elasticbeanstalk_phpapp_1, elasticbeanstalk_nginxproxy_1  
phpapp_1      | [23-Apr-2015 23:24:25] NOTICE: fpm is running, pid 1  
phpapp_1      | [23-Apr-2015 23:24:25] NOTICE: ready to handle connections
```

eb local status

Consulte el estado de los contenedores locales:

```
~/project$ eb local status  
Platform: 64bit Amazon Linux 2014.09 v1.2.1 running Multi-container Docker 1.3.3  
(Generic)  
Container name: elasticbeanstalk_nginxproxy_1  
Container ip: 127.0.0.1  
Container running: True  
Exposed host port(s): 80  
Full local URL(s): 127.0.0.1:80  
  
Container name: elasticbeanstalk_phpapp_1  
Container ip: 127.0.0.1  
Container running: True  
Exposed host port(s): None  
Full local URL(s): None
```

eb local logs

Consulte la ruta de los registros del proyecto actual:

```
~/project$ eb local logs  
Elastic Beanstalk will write logs locally to /home/user/project/.elasticbeanstalk/logs/  
local.  
Logs were most recently created 3 minutes ago and written to /home/user/  
project/.elasticbeanstalk/logs/local/150420_234011665784.
```

eb local setenv

Establezca las variables de entorno para utilizarlas con eb local run.

```
~/project$ eb local setenv PARAM1=value
```

Imprima las variables de entorno configuradas con `eb local setenv`.

```
~/project$ eb local printenv
Environment Variables:
PARAM1=value
```

eb logs

Descripción

El comando `eb logs` tiene dos propósitos distintos: habilitar o deshabilitar el streaming de registros a CloudWatch Logs y recuperar registros de instancias o los registros de CloudWatch Logs. Con la opción `--cloudwatch-logs (-cw)`, el comando habilita o deshabilita el streaming de registros. Sin esta opción, recupera los registros.

Al recuperar los registros, especifique la opción `--all`, `--zip` o `--stream` para recuperar los registros completos. Si no especifica ninguna de estas opciones, Elastic Beanstalk recupera los registros de finalización.

El comando procesa los registros del entorno especificado o predeterminado. Los registros pertinentes dependen del tipo de contenedor. Si el directorio raíz contiene un archivo `platform.yaml` donde se especifica una plataforma personalizada, este comando también procesa los registros del entorno del constructor.

Para obtener más información, consulte [the section called “Registros de CloudWatch”](#).

Sintaxis

Para habilitar o deshabilitar el streaming de registros a CloudWatch Logs:

```
eb logs --cloudwatch-logs [enable | disable] [--cloudwatch-log-source instance |
environment-health | all] [environment-name]
```

Para recuperar los registros de instancias:

```
eb logs [-all | --zip | --stream] [--cloudwatch-log-source instance] [--
instance instance-id] [--log-group log-group] [environment-name]
```


Para recuperar los registros de estado de entorno:

```
eb logs [-all | --zip | --stream] --cloudwatch-log-source environment-health
[environment-name]
```

Opciones

Nombre	Descripción
-cw [enable disable] o bien --cloudwatch-logs [enable disable]	Habilita o deshabilita el streaming de registros a CloudWatch Logs. Si no se proporciona ningún argumento, se habilita el streaming de registros. Si tampoco se especifica la opción <code>--cloudwatch-log-source</code> (<code>-cls</code>), el streaming de logs de instancias está habilitado o deshabilitado.
-cls instance environment-health all o bien --cloudwatch-log-source instance environment-health all	Especifica el origen de los registros al trabajar con CloudWatch Logs. Con el formato de habilitar o deshabilitar, son los registros para los que se habilita o deshabilita el streaming de CloudWatch Logs. Con el formato de recuperación del comando, estos son los registros para recuperar de CloudWatch Logs. Valores válidos: <ul style="list-style-type: none"> • Con <code>--cloudwatch-logs</code> (habilitar o deshabilitar) – <code>instance environment-health all</code> • Sin <code>--cloudwatch-logs</code> (recuperar) – <code>instance environment-health</code> Significado de los valores: <ul style="list-style-type: none"> • <code>instance</code> (predeterminado): logs de instancias • <code>environment-health</code> : logs de estado de entorno (solo se admite cuando el estado mejorado está habilitado en el entorno) • <code>all</code>: ambas orígenes de logs
-a o bien	Recupera todos los logs completos y los guarda en el directorio <code>.elasticbeanstalk/logs</code> .

Nombre	Descripción
<code>--all</code>	
<code>-z</code> o bien <code>--zip</code>	Recupera los logs completos, los comprime en un archivo <code>.zip</code> y guarda el archivo en el directorio <code>.elasticbeanstalk/logs</code> .
<code>--stream</code>	Realiza streaming (salidas de forma continua) de logs completos. Con esta opción, el comando se sigue ejecutando hasta que lo interrumpa (pulse Ctrl+C).
<code>-i <i>instance-id</i></code> o bien <code>--instance <i>instance-id</i></code>	Recupera exclusivamente los logs de la instancia especificada.

Nombre	Descripción
<p>-g <i>log-group</i></p> <p>o bien</p> <p>--log-group <i>log-group</i></p>	<p>Especifica el grupo de registros de CloudWatch Logs desde el que se van a recuperar los registros. La opción solo es válida cuando está habilitada el streaming de registros de instancias a CloudWatch Logs.</p> <p>Si el streaming de logs de instancias está habilitado, y no especifica la opción <code>--log-group</code>, el grupo de logs predeterminado es uno de los siguientes:</p> <ul style="list-style-type: none"> • Amazon Linux 2: <code>/aws/elasticbeanstalk/<i>environment-name</i> /var/log/eb-engine.log</code> • Plataformas Windows – <code>/aws/elasticbeanstalk/<i>environment-name</i> /EBDeploy-Log</code> • Amazon Linux AMI (AL1): <code>/aws/elasticbeanstalk/<i>environment-name</i> /var/log/eb-activity.log</code> <div data-bbox="625 1010 1507 1514" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>El 18 de julio de 2022, Elastic Beanstalk estableció el estado de todas las ramificaciones de la plataforma basadas en la AMI de Amazon Linux (AL1) como retirado. Para obtener más información sobre la migración a una ramificación de la plataforma Amazon Linux 2023 actual y totalmente compatible, consulte Migración de su aplicación de Linux de Elastic Beanstalk a Amazon Linux 2023 o Amazon Linux 2.</p> </div> <p>Para obtener información sobre el grupo de logs correspondiente a cada archivo de log, consulte Cómo Elastic Beanstalk configura CloudWatch Logs.</p>
<p>Opciones comunes</p>	

Output

De forma predeterminada, muestra los logs directamente en el terminal. Utiliza un programa de paginación para mostrar la salida. Pulse **Q** o **q** para salir.

Con `--stream` muestra los logs existentes en el terminal y se sigue ejecutando. Pulse **Ctrl+C** para salir.

Con `--all` y `--zip` se guardan los logs en archivos locales y se muestra la ubicación del archivo.

Ejemplos

En el ejemplo siguiente habilita el streaming de registros de instancias a CloudWatch Logs.

```
$ eb logs -cw enable
Enabling instance log streaming to CloudWatch for your environment
After the environment is updated you can view your logs by following the link:
https://console.aws.amazon.com/cloudwatch/home?region=us-east-1#logs:prefix=/aws/
elasticbeanstalk/environment-name/
Printing Status:
2018-07-11 21:05:20      INFO: Environment update is starting.
2018-07-11 21:05:27      INFO: Updating environment environment-name's configuration
settings.
2018-07-11 21:06:45      INFO: Successfully deployed new configuration to environment.
```

El siguiente ejemplo recupera los logs de instancias en un archivo `.zip`.

```
$ eb logs --zip
Retrieving logs...
Logs were saved to /home/workspace/environment/.elasticbeanstalk/logs/150622_173444.zip
```

eb open

Descripción

Abre la dirección URL pública de su sitio web en el navegador predeterminado.

Sintaxis

```
eb open
```

eb open *environment-name*

Opciones

Nombre	Descripción
Opciones comunes	

Salida

El comando eb open no genera ninguna salida. Simplemente abre la aplicación en una ventana del navegador.

eb platform

Descripción

Este comando admite dos WorkSpaces diferentes:

[Plataforma](#)

Utilice este Workspace para administrar las plataformas personalizadas.

[Entorno](#)

Utilice este Workspace para seleccionar una plataforma predeterminada o mostrar información sobre la plataforma actual.

Elastic Beanstalk proporciona el acceso directo ebp para eb platform.

Note

Windows PowerShell utiliza ebp como alias del comando. Si está ejecutando la CLI de EB en Windows PowerShell, utilice el formato largo de este comand — eb platform.

Uso de eb platform en plataformas personalizadas

Muestra las versiones de la plataforma actual y permite administrar las plataformas personalizadas.

Sintaxis

```
eb platform create [version] [options]
```

```
eb platform delete [version] [options]
```

```
eb platform events [version] [options]
```

```
eb platform init [platform] [options]
```

```
eb platform list [options]
```

```
eb platform logs [version] [options]
```

```
eb platform status [version] [options]
```

```
eb platform use [platform] [options]
```

Opciones

Nombre	Descripción
<code>create</code> [<i>version</i>] [<i>options</i>]	Crea una nueva versión de la plataforma. Más información.
<code>delete</code> <i>version</i> [<i>options</i>]	Elimina una versión de la plataforma. Más información.
<code>events</code> [<i>version</i>] [<i>options</i>]	Muestra los eventos de una versión de la plataforma. Más información.
<code>init</code> [<i>platform</i>] [<i>options</i>]	Inicializa un repositorio de la plataforma. Más información.
<code>list</code> [<i>options</i>]	Muestra las versiones de la plataforma actual. Más información.
<code>logs</code> [<i>version</i>] [<i>options</i>]	Muestra los registros creados en el entorno del constructor sobre una versión de la plataforma. Más información.
<code>status</code> [<i>version</i>] [<i>options</i>]	Muestra el estado de una versión de la plataforma. Más información.

Nombre	Descripción
use [<i>platform</i>] [<i>options</i>]	Permite seleccionar una plataforma diferente en la que se van a crear las nuevas versiones. Más información .
Opciones comunes	

Opciones comunes

Todos los comandos eb platform incluyen las siguientes opciones comunes.

Nombre	Descripción
-h O BIEN --help	Muestra un mensaje de ayuda y termina.
--debug	Muestra una salida de depuración adicional.
--quiet	Suprime todos los datos de salida.
-v O BIEN --verbose	Muestra una salida adicional.
--profile <i>PROFILE</i>	Utiliza el <i>PERFIL</i> especificado en las credenciales.
-r <i>REGION</i> O BIEN --region <i>REGION</i>	Utiliza la región especificada en <i>REGIÓN</i> .
--no-verify-ssl	No verifique los certificados SSL de AWS.

Eb platform create

Crea una nueva versión de la plataforma y devuelve el ARN de la nueva versión. Si ningún entorno del constructor se está ejecutando en la región actual, este comando lanza uno. La opción *versión* y la opción increment (-M, -m y -p) son mutuamente excluyentes.

Opciones

Nombre	Descripción
<i>versión</i>	Si no se especifica ninguna <i>versión</i> , se crea una nueva basada en la plataforma más reciente con una versión de lote (N en n.n.N) mayor.
-M O BIEN --major-increment	Aumenta el número de la versión principal (N en N.n.n).
-m O BIEN --minor-increment	Aumenta el número de la versión secundaria (N en n.N.n).
-p O BIEN --patch-increment	Aumenta el número de la versión de lote (N en n.n.N).
-i <i>INSTANCE_TYPE</i> O BIEN --instance-type <i>TIPO_DE_I NSTANCIA</i>	Utilice el valor de <i>TIPO_DE_INSTANCIA</i> como tipo de instancia; por ejemplo, t1.micro .
-ip <i>INSTANCE_PROFILE</i> O BIEN	Utilice el valor de <i>PERFIL_DE_INSTANCIA</i> como perfil de instancia al crear las AMI de una plataforma personalizada.

Nombre	Descripción
<code>--instance-profile</code> <i>INSTANCE_PROFILE</i>	Si no se especifica la opción <code>-ip</code> , crea el perfil de instancia <code>aws-elasticbeanstalk-custom-platfome-ec2-role</code> y lo utiliza en la plataforma personalizada.
<code>--tags</code> <i>key1=value1[,key2=value2]</i>	Etiquete su versión personalizada de la plataforma. Las etiquetas se especifican como una lista de pares <code>key=value</code> separados por comas. Para obtener más información, consulte Etiquetado de versiones de la plataforma personalizada .
<code>--timeout</code> <i>minutos</i>	Establezca el número de minutos que deben transcurrir antes de que se agote el tiempo de espera del comando.
<code>--vpc.id</code> <i>VPC_ID</i>	ID de la VPC en la que Packer crea el contenido.
<code>--vpc.subnets</code> <i>VPC_SUBNETS</i>	Subredes de la VPC en las que Packer crea el contenido.
<code>--vpc.publicip</code>	Asocia IP públicas con instancias EC2 lanzadas.

Eb platform delete

Elimina una versión de la plataforma. La versión no se elimina si algún entorno la está utilizando.

Opciones

Nombre	Descripción
<i>version</i>	Versión que se va a eliminar. Este valor es obligatorio.
<code>--cleanup</code>	Elimina todas las versiones de la plataforma con el estado <code>Failed</code> .
<code>--all-platforms</code>	Si se especifica <code>--cleanup</code> , se eliminan todas las versiones con el estado <code>Failed</code> de todas las plataformas.

Nombre	Descripción
<code>--force</code>	No pide confirmación cuando se elimina una versión.

eb platform events

Muestra los eventos de una versión de la plataforma. Si se especifica una *versión*, se muestran los eventos de dicha versión; de lo contrario, se muestran los eventos de la versión actual.

Opciones

Nombre	Descripción
<i>version</i>	Versión de los eventos que van a mostrar. Este valor es obligatorio.
<code>-f</code> O BIEN <code>--follow</code>	Continúa mostrando los eventos a medida que se producen.

Eb platform init

Inicializa un repositorio de la plataforma.

Opciones

Nombre	Descripción
<i>platform</i>	Nombre de la plataforma que se va a inicializar. Este valor es obligatorio a menos que <code>-i</code> (modo interactivo) esté habilitado.
<code>-i</code> O BIEN <code>--interactive</code>	Utiliza el modo interactivo.

Nombre	Descripción
-k <i>KEYNAME</i>	Nombre de la clave de EC2 predeterminada.
O BIEN	
--keyname <i>KEYNAME</i>	

Puede ejecutar este comando en un directorio que se haya inicializado anteriormente, aunque, en ese caso, no se puede cambiar el tipo de Workspace.

Para volver a inicializar con otra configuración, utilice la opción `-i`.

Eb platform list

Muestra las versiones de la plataforma vinculadas a un Workspace (directorio) o una región.

El comando devuelve resultados diferentes en función del tipo de Workspace en el que se ejecute, tal y como se indica a continuación:

- En un Workspace de plataforma (un directorio inicializado por `eb platform init`), el comando devuelve una lista de todas las versiones de la plataforma personalizada definida en el Workspace. Agrega la opción `--all-platforms` o `--verbose` para obtener una lista de todas las versiones de la plataforma de todas las plataformas personalizadas que tiene su cuenta en la región vinculada al Workspace.
- En un Workspace de aplicaciones (un directorio inicializado por `eb init`), el comando devuelve una lista de todas las versiones de la plataforma, tanto para plataformas gestionadas por Elastic Beanstalk como para las plataformas personalizadas de su cuenta. La lista utiliza nombres de versión de la plataforma cortos y algunas variantes de versión de la plataforma pueden combinarse. Agrega la opción `--verbose` para obtener una lista detallada con nombres completos y todas las variantes enumeradas por separado.
- En un directorio sin inicializar, el comando solo funciona con la opción `--region`. Devuelve una lista de todas las versiones de la plataforma gestionadas por Elastic Beanstalk admitidas en la región. La lista utiliza nombres de versión de la plataforma cortos y algunas variantes de versión de la plataforma pueden combinarse. Agrega la opción `--verbose` para obtener una lista detallada con nombres completos y todas las variantes enumeradas por separado.

Opciones

Nombre	Descripción
-a O BIEN --all-platforms	Válido únicamente en un WorkSpace inicializado (un directorio o inicializado por <code>eb platform init</code> o <code>eb init</code>). Muestra las versiones de las plataformas personalizadas asociadas a la cuenta.
-s <i>STATUS</i> O BIEN --status <i>STATUS</i>	Muestra únicamente las plataformas que coinciden con el <i>ESTADO</i> especificado: <ul style="list-style-type: none"> • Ready • Con error • Deleting (Eliminándose) • Creación

Eb platform logs

Muestra los registros creados en el entorno del constructor sobre una versión de la plataforma.

Opciones

Nombre	Descripción
<i>version</i>	Versión de la plataforma cuyos registros se van a mostrar. Si se omite, se muestran los registros de la versión actual.
--stream	Transmite los registros de implementación que se configuraron con CloudWatch.

Eb platform status

Muestra el estado de una versión de la plataforma.

Opciones

Nombre	Descripción
<i>version</i>	Versión de la plataforma cuyo estado se va a recuperar. Si se omite, se muestra el estado de la versión actual.

Eb platform use

Permite seleccionar una plataforma diferente en la que se van a crear las nuevas versiones.

Opciones

Nombre	Descripción
<i>platform</i>	Especifica <i>plataforma</i> como la versión activa de este Workspace. Este valor es obligatorio.

Uso de eb platform en entornos

Muestra las plataformas admitidas y permite definir la plataforma y la versión predeterminadas que se van a utilizar cuando se lance un entorno. Utilice `eb platform list` para ver una lista de todas las plataformas admitidas. Utilice `eb platform select` para cambiar la plataforma del proyecto. Utilice `eb platform show` para ver la plataforma del proyecto seleccionada.

Sintaxis

`eb platform list`

`eb platform select`

`eb platform show`

Opciones

Nombre	Descripción
<code>list</code>	Muestra la versión de la plataforma actual.

Nombre	Descripción
select	Seleccione la plataforma predeterminada.
show	Muestra información sobre la plataforma actual.

Ejemplo 1

En el siguiente ejemplo, se muestran los nombres de todas las configuraciones de todas las plataformas que admite Elastic Beanstalk.

```
$ eb platform list
docker-1.5.0
glassfish-4.0-java-7-(preconfigured-docker)
glassfish-4.1-java-8-(preconfigured-docker)
go-1.3-(preconfigured-docker)
go-1.4-(preconfigured-docker)
iis-7.5
iis-8
iis-8.5
multi-container-docker-1.3.3-(generic)
node.js
php-5.3
php-5.4
php-5.5
python
python-2.7
python-3.4
python-3.4-(preconfigured-docker)
ruby-1.9.3
ruby-2.0-(passenger-standalone)
ruby-2.0-(puma)
ruby-2.1-(passenger-standalone)
ruby-2.1-(puma)
ruby-2.2-(passenger-standalone)
ruby-2.2-(puma)
tomcat-6
tomcat-7
tomcat-7-java-6
tomcat-7-java-7
tomcat-8-java-8
```


Ejemplo 2

En el siguiente ejemplo, se le pedirá que elija una plataforma en una lista y la versión que desee implementar en la plataforma especificada.

```
$ eb platform select
Select a platform.
1) PHP
2) Node.js
3) IIS
4) Tomcat
5) Python
6) Ruby
7) Docker
8) Multi-container Docker
9) GlassFish
10) Go
(default is 1): 5

Select a platform version.
1) Python 2.7
2) Python
3) Python 3.4 (Preconfigured - Docker)
```

Ejemplo 3

En el ejemplo siguiente, se muestra información sobre la plataforma predeterminada actual.

```
$ eb platform show
Current default platform: Python 2.7
New environments will be running: 64bit Amazon Linux 2014.09 v1.2.0 running Python 2.7

Platform info for environment "tmp-dev":
Current: 64bit Amazon Linux 2014.09 v1.2.0 running Python
Latest: 64bit Amazon Linux 2014.09 v1.2.0 running Python
```

eb printenv

Descripción

Muestra todas las propiedades del entorno en la ventana de comandos.

Sintaxis

```
eb printenv
```

```
eb printenv environment-name
```

Opciones

Nombre	Descripción
Opciones comunes	

Salida

Si se ejecuta correctamente, el comando devuelve el estado de la operación `printenv`.

Ejemplo

El siguiente ejemplo muestra las propiedades de entorno del entorno especificado.

```
$ eb printenv
Environment Variables:
  PARAM1 = Value1
```

eb restore

Descripción

Vuelve a crear un entorno terminado, creando un nuevo entorno con el mismo nombre, ID y configuración. El nombre del entorno, el nombre del dominio y la versión de la aplicación debe estar disponibles para que la reconstrucción se realice correctamente.

Sintaxis

```
eb restore
```

```
eb restore environment_id
```

Opciones

Nombre	Descripción
Opciones comunes	

Salida

La CLI de EB muestra una lista de entornos terminados disponibles para su restauración.

Ejemplo

```
$ eb restore
Select a terminated environment to restore

#   Name           ID                Application Version    Date Terminated      Ago
3   gamma          e-s7mimej8e9     app-77e3-161213_211138  2016/12/14 20:32 PST  13
mins
2   beta           e-sj28uu2wia     app-77e3-161213_211125  2016/12/14 20:32 PST  13
mins
1   alpha          e-gia8mphu6q     app-77e3-161213_211109  2016/12/14 16:21 PST  4
hours

(Commands: Quit, Restore, # #)

Selected environment alpha
Application:    scorekeep
Description:    Environment created from the EB CLI using "eb create"
CNAME:         alpha.h23tbtbm92.us-east-2.elasticbeanstalk.com
Version:       app-77e3-161213_211109
Platform:      64bit Amazon Linux 2016.03 v2.1.6 running Java 8
Terminated:    2016/12/14 16:21 PST
Restore this environment? [y/n]: y

2018-07-11 21:04:20    INFO: restoreEnvironment is starting.
2018-07-11 21:04:39    INFO: Created security group named: sg-e2443f72
...
```

eb scale

Descripción

Escala el entorno para que siempre se ejecute en un número especificado de instancias. Para ello, establece el número mínimo y máximo de instancias en el número especificado.

Sintaxis

eb scale *number-of-instances*

eb scale *number-of-instances environment-name*

Opciones

Nombre	Descripción
--timeout	Número de minutos que deben transcurrir antes de que se agote el tiempo de espera del comando.
Opciones comunes	

Salida

Si se ejecuta correctamente, el comando actualiza el número mínimo y máximo de instancias que se van a ejecutar al número especificado.

Ejemplo

En el siguiente ejemplo, el número de instancias se establece en 2.

```
$ eb scale 2
2018-07-11 21:05:22 INFO: Environment update is starting.
2018-07-11 21:05:27 INFO: Updating environment tmp-dev's configuration settings.
2018-07-11 21:08:53 INFO: Added EC2 instance 'i-5fce3d53' to Auto Scaling Group
'awseb-e-2cpfjbra9a-stack-AWSEBAutoScalingGroup-7AXY7U13ZQ6E'.
2018-07-11 21:08:58 INFO: Successfully deployed new configuration to environment.
2018-07-11 21:08:59 INFO: Environment update completed successfully.
```

eb setenv

Descripción

Establece [propiedades de entorno](#) para el entorno predeterminado.

Sintaxis

```
eb setenv key=value
```

Puede incluir tantas propiedades como desee, pero el tamaño total de todas las propiedades no puede superar los 4096 bytes. Puede eliminar una variable dejando su valor en blanco. Consulte [Configuración de las propiedades del entorno \(variables de entorno\)](#) para ver los límites.

Note

Si `value` contiene un [carácter especial](#), ese carácter debe ir precedido por el carácter de escape `\`.

Opciones

Nombre	Descripción
<code>--timeout</code>	Número de minutos que deben transcurrir antes de que se agote el tiempo de espera del comando.
Opciones comunes	

Salida

Si se ejecuta sin errores, el comando indica que la actualización del entorno se ejecutó correctamente.

Ejemplo

En el ejemplo siguiente, se establece la variable de entorno `ExampleVar`.

```
$ eb setenv ExampleVar=ExampleValue
```

```
2018-07-11 21:05:25 INFO: Environment update is starting.
2018-07-11 21:05:29 INFO: Updating environment tmp-dev's configuration settings.
2018-07-11 21:06:50 INFO: Successfully deployed new configuration to environment.
2018-07-11 21:06:51 INFO: Environment update completed successfully.
```

El comando siguiente establece varias propiedades del entorno. Agrega la propiedad de entorno `foo` y establece su valor en `bar`, cambia el valor de la propiedad `JDBC_CONNECTION_STRING` y elimina las propiedades `PARAM4` y `PARAM5`.

```
$ eb setenv foo=bar JDBC_CONNECTION_STRING=hello PARAM4= PARAM5=
```

eb ssh

Descripción

Note

Este comando no funciona con entornos que ejecutan instancias Windows Server.

Conéctese a una Amazon EC2 instance Linux del entorno a través de Secure Shell (SSH). Si un entorno tiene varias instancias en ejecución, la CLI de EB le pedirá que especifique a qué instancia quiere conectarse. Para poder utilizar este comando, SSH debe estar instalado en la máquina local y disponible en la línea de comando. Los archivos de clave privada deben estar en una carpeta llamada `.ssh` del directorio del usuario, y las instancias EC2 del entorno deben tener direcciones IP públicas.

Si el directorio raíz contiene un archivo `platform.yaml` donde se especifica una plataforma personalizada, este comando también se conecta a las instancias del entorno personalizado.

Claves de SSH

Si no configuró SSH anteriormente, puede utilizar la CLI de EB para crear una clave cuando ejecute `eb init`. Si ya ejecutó `eb init`, hágalo de nuevo con la opción `--interactive` y seleccione `Yes` y `Create New Keypair` cuando se lo pidan al configurar SSH. La CLI de EB almacenará las claves creadas durante este proceso en la propia carpeta.

Este comando abre temporalmente el puerto 22 del grupo de seguridad del entorno al tráfico entrante procedente de 0.0.0.0/0 (todas las direcciones IP) si no hay ninguna regla aplicable al puerto 22. Si ha configurado el grupo de seguridad del entorno para que abra el puerto 22 a un intervalo de CIDR limitado para aumentar la seguridad, la CLI de EB respetará esta configuración y no hará ningún cambio en el grupo de seguridad. Para anular esta configuración y hacer que la CLI de EB abra el puerto 22 a todo el tráfico entrante, utilice la opción `--force`.

Consulte [Grupos de seguridad](#) para obtener información sobre la configuración del grupo de seguridad del entorno.

Sintaxis

```
eb ssh
```

```
eb ssh environment-name
```

Opciones

Nombre	Descripción
<code>-i</code> o bien <code>--instance</code>	Especifica el ID de la instancia a la que se conecta. Le recomendamos que utilice esta opción.
<code>-n</code> o bien <code>--number</code>	Especifica la instancia a la que se va a conectar mediante un número.
<code>-o</code> o bien <code>--keep_open</code>	Deja el puerto 22 abierto en el grupo de seguridad una vez que la sesión de SSH ha terminado.
<code>--command</code>	Ejecuta un comando shell en la instancia especificada en lugar de iniciar una sesión de SSH.

Nombre	Descripción
<code>--custom</code>	Especifique un comando SSH para utilizarlo en lugar de 'ssh -i keyfile'. No incluya el usuario remoto ni el nombre de host.
<code>--setup</code>	Cambia el par de claves asignado a las instancias del entorno (requiere que las instancias se sustituyan).
<code>--force</code>	<p>Abre el puerto 22 al tráfico entrante procedente de 0.0.0.0/0 en el grupo de seguridad del entorno, incluso si el grupo de seguridad ya está configurada para SSH.</p> <p>Utilice esta opción si el grupo de seguridad del entorno está configurado para abrir el puerto 22 a un rango de CIDR limitado que no incluye la dirección IP a la que está intentando conectarse.</p>
<code>--timeout <i>minutos</i></code>	<p>Establezca el número de minutos que deben transcurrir antes de que se agote el tiempo de espera del comando.</p> <p>Solo se puede utilizar con el argumento <code>--setup</code>.</p>
Opciones comunes	

Salida

Si se ejecuta correctamente, el comando abre una conexión SSH con la instancia.

Ejemplo

En el siguiente ejemplo, se establece conexión con el entorno especificado.

```
$ eb ssh
Select an instance to ssh into
1) i-96133799
2) i-5931e053
(default is 1): 1
INFO: Attempting to open port 22.
INFO: SSH port 22 open.
The authenticity of host '54.191.45.125 (54.191.45.125)' can't be established.
```



```
RSA key fingerprint is ee:69:62:df:90:f7:63:af:52:7c:80:60:1b:3b:51:a9.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '54.191.45.125' (RSA) to the list of known hosts.
```

```
  _|  _|_ )
  _| (    /  Amazon Linux AMI
  _|\___|___|
```

```
https://aws.amazon.com/amazon-linux-ami/2014.09-release-notes/
No packages needed for security; 1 packages available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-8-185 ~]$ ls
[ec2-user@ip-172-31-8-185 ~]$ exit
logout
Connection to 54.191.45.125 closed.
INFO: Closed port 22 on ec2 instance security group
```

eb status

Descripción

Proporciona información sobre el estado del entorno.

Si el directorio raíz contiene un archivo `platform.yaml` donde se especifica una plataforma personalizada, este comando también proporciona información sobre el entorno del constructor.

Sintaxis

```
eb status
```

```
eb status environment-name
```

Opciones

Nombre	Descripción
-v	Proporciona más información sobre las distintas instancias, como su estado con el balanceador de carga de Elastic Load Balancing.
o bien	
--verbose	

Nombre	Descripción
Opciones comunes	

Salida

Si se ejecuta correctamente, el comando devuelve la siguiente información sobre el entorno:

- Environment name
- Nombre de la aplicación
- Versión de la aplicación implementada
- ID del entorno
- Plataforma
- Capa del entorno
- CNAME
- Momento en que se actualizó por última vez el entorno
- Estado
- Estado

Si utiliza el modo detallado, la CLI de EB proporciona también el número de instancias Amazon EC2.

Ejemplo

El siguiente ejemplo muestra el estado del entorno tmp-dev.

```
$ eb status
Environment details for: tmp-dev
  Application name: tmp
  Region: us-west-2
  Deployed Version: None
  Environment ID: e-2cpfjbra9a
  Platform: 64bit Amazon Linux 2014.09 v1.0.9 running PHP 5.5
  Tier: WebServer-Standard-1.0
  CNAME: tmp-dev.elasticbeanstalk.com
  Updated: 2014-10-29 21:37:19.050000+00:00
  Status: Launching
```

Health: Grey

eb swap

Descripción

Intercambia el CNAME del entorno con el CNAME de otro entorno (por ejemplo, para evitar tiempos de inactividad cuando se actualiza la versión de la aplicación).

Note

Si tiene más de dos entornos, se le pedirá que seleccione el nombre del entorno que actualmente está utilizando el CNAME deseado en una lista de entornos. Para evitar esto, puede especificar el nombre del entorno que se va utilizar con la opción `-n` al ejecutar el comando.

Sintaxis

eb swap

eb swap ***environment-name***

Note

El ***nombre-de-entorno*** es el entorno en el que quiere utilizar un CNAME diferente. Si no se especifica el ***nombre-de-entorno*** como un parámetro de la línea de comando al ejecutar eb swap, la CLI de EB actualiza el CNAME del entorno predeterminado.

Opciones

Nombre	Descripción
<code>-n</code> o bien <code>--destination_name</code>	Especifica el nombre del entorno con el que se van a intercambiar los CNAME. Si ejecuta eb swap sin esta opción, la CLI de EB le pedirá que elija un entorno en una lista.

Nombre	Descripción
Opciones comunes	

Salida

Si se ejecuta correctamente, el comando devuelve el estado de la operación swap.

Ejemplos

La solicitud del ejemplo siguiente intercambia el entorno tmp-dev con live-env.

```
$ eb swap
Select an environment to swap with.
1) staging-dev
2) live-env
(default is 1): 2
2018-07-11 21:05:25    INFO: swapEnvironmentCNAMEs is starting.
2018-07-11 21:05:26    INFO: Swapping CNAMEs for environments 'tmp-dev' and 'live-env'.
2018-07-11 21:05:30    INFO: 'tmp-dev.elasticbeanstalk.com' now points to 'awseb-e-j-
AWSEBLoa-M7U21VXNLWHN-487871449.us-west-2.elb.amazonaws.com'.
2018-07-11 21:05:30    INFO: Completed swapping CNAMEs for environments 'tmp-dev' and
'live-env'.
```

En el ejemplo siguiente, el entorno tmp-dev se intercambia con el entorno live-env, pero no se le pide que especifique ni seleccione ningún valor para las opciones de configuración.

```
$ eb swap tmp-dev --destination_name live-env
2018-07-11 21:18:12    INFO: swapEnvironmentCNAMEs is starting.
2018-07-11 21:18:13    INFO: Swapping CNAMEs for environments 'tmp-dev' and 'live-env'.
2018-07-11 21:18:17    INFO: 'tmp-dev.elasticbeanstalk.com' now points to 'awseb-e-j-
AWSEBLoa-M7U21VXNLWHN-487871449.us-west-2.elb.amazonaws.com'.
2018-07-11 21:18:17    INFO: Completed swapping CNAMEs for environments 'tmp-dev' and
'live-env'.
```

eb tags

Descripción

Agregar, eliminar, actualizar y enumerar etiquetas de un recurso de Elastic Beanstalk.

Para obtener más información sobre el etiquetado de recursos en Elastic Beanstalk, consulte [Etiquetar recursos de la aplicación Elastic Beanstalk](#).

Sintaxis

```
eb tags [environment-name] [--resource ARN] -l | --list
```

```
eb tags [environment-name] [--resource ARN] -a | --add key1=value1 [,key2=value2 ...]
```

```
eb tags [environment-name] [--resource ARN] -u | --update key1=value1 [,key2=value2 ...]
```

```
eb tags [environment-name] [--resource ARN] -d | --delete key1 [,key2 ...]
```

Puede combinar las opciones de subcomando `--add`, `--update` y `--delete` en un único comando. Se necesita como mínimo uno. No se puede combinar ninguna de estas tres opciones de subcomandos con `--list`.

Si no hay argumentos adicionales, todos estos comandos enumeran o modifican las etiquetas del entorno predeterminado en la aplicación del directorio actual. Con un argumento *environment-name*, los comandos enumeran o modifican las etiquetas de ese entorno. Con la opción `--resource`, los comandos enumeran o modifican las etiquetas de cualquier recurso de Elastic Beanstalk: una aplicación, un entorno, una versión de la aplicación, una configuración guardada o una versión de la plataforma personalizada. Especifique el recurso por su nombre de recurso de Amazon (ARN).

Opciones

Ninguna de estas opciones es obligatoria. Si ejecuta `eb create` sin especificar ninguna opción, le pedirán que especifique o seleccione un valor para cada ajuste.

Nombre	Descripción
-l o bien <code>--list</code>	Enumere todas las etiquetas aplicadas actualmente al recurso.
-a <i>key1=value1</i> [, <i>key2=value2</i>] o bien	Aplice nuevas etiquetas al recurso. Especifique las etiquetas como una lista de pares <code>key=value</code>

Nombre	Descripción
<pre>--add <i>key1=value1[,key2=va</i></pre>	<p>separados por comas. No puede especificar las claves de etiquetas existentes.</p> <p>Valores válidos: consulte Etiquetado de recursos de .</p>
<pre>-u <i>key1=value1[,key2=value2</i> o bien --update <i>key1=value1[,key2=value2</i> .</pre>	<p>Actualice los valores de etiquetas de recursos existentes. Especifique las etiquetas como una lista de pares <code>key=value</code> separados por comas. Debe especificar las claves de etiquetas existentes.</p> <p>Valores válidos: consulte Etiquetado de recursos de .</p>
<pre>-d <i>key1[,key2 ...]</i> o bien --delete <i>key1[,key2 ...]</i></pre>	<p>Elimine las etiquetas de recursos existentes. Especifique las etiquetas como una lista de claves separadas por comas. Debe especificar las claves de etiquetas existentes.</p> <p>Valores válidos: consulte Etiquetado de recursos de .</p>
<pre>-r <i>region</i> o bien --region <i>region</i></pre>	<p>La Región de AWS en el que existe el recurso.</p> <p>Default: la región configurada predeterminada.</p> <p>Para obtener la lista de valores que puede especificar para esta opción, consulte Puntos de enlace y cuotas de AWS Elastic Beanstalk en la Referencia general de AWS.</p>
<pre>--resource <i>ARN</i></pre>	<p>El ARN del recurso para el que el comando modifica o enumera las etiquetas. Si no se especifica, el comando hace referencia al entorno (valor predeterminado o especificado) en la aplicación del directorio actual.</p> <p>Valores válidos: consulte uno de los subtemas de Etiquetado de recursos de específicos del recurso en el que está interesado. Estos temas muestran cómo se construye el ARN del recurso y se explica cómo obtener una lista de los ARN de este recurso para su aplicación o cuenta.</p>

Output

La opción del subcomando `--list` muestra una lista de las etiquetas del recurso. La salida muestra tanto las etiquetas que Elastic Beanstalk aplica de forma predeterminada como las etiquetas personalizadas.

```
$ eb tags --list
Showing tags for environment 'MyApp-env':

Key                                Value
Name                               MyApp-env
elasticbeanstalk:environment-id    e-63cmxwjaut
elasticbeanstalk:environment-name  MyApp-env
mytag                               tagvalue
tag2                                2nd value
```

Las opciones de subcomandos `--add`, `--update` y `--delete` no tienen ninguna salida cuando se ejecutan con éxito. Puede agregar la opción `--verbose` para ver la salida detallada de la actividad del comando.

```
$ eb tags --verbose --update "mytag=tag value"
Updated Tags:

Key                                Value
mytag                              tag value
```

Ejemplos

El siguiente comando agrega una etiqueta correctamente con la clave `tag1` y el valor `value1` al entorno predeterminado de la aplicación y, al mismo tiempo, elimina la etiqueta `tag2`.

```
$ eb tags --add tag1=value1 --delete tag2
```

El siguiente comando agrega correctamente una etiqueta a una configuración guardada dentro de una aplicación.

```
$ eb tags --add tag1=value1 \
```

```
--resource "arn:aws:elasticbeanstalk:us-east-2:my-account-  
id:configurationtemplate/my-app/my-template"
```

El siguiente comando produce un error porque intenta actualizar una etiqueta no existente.

```
$ eb tags --update tag3=newval  
ERROR: Tags with the following keys can't be updated because they don't exist:  
  
tag3
```

El siguiente comando produce un error porque intenta actualizar y eliminar la misma clave.

```
$ eb tags --update mytag=newval --delete mytag  
ERROR: A tag with the key 'mytag' is specified for both '--delete' and '--update'. Each  
tag can be either deleted or updated in a single operation.
```

eb terminate

Descripción

Termina el entorno en ejecución para que no incurra en cargos por los recursos de AWS no utilizados.

Con la opción `--all` se elimina la aplicación en que se inicializó el directorio actual mediante [eb init](#). El comando termina todos los entornos de la aplicación. También termina las [versiones de la aplicación](#) y sus [configuraciones guardadas](#) y, a continuación, elimina la aplicación.

Si el directorio raíz contiene un archivo `platform.yaml` donde se especifica una plataforma personalizada, este comando termina el entorno personalizado en ejecución.

Note

Siempre podrá lanzar más adelante un nuevo entorno con la misma versión.

Si tiene datos de un entorno que desea conservar, establezca la política de eliminación de base de datos en `Retain` antes de terminar el entorno. Esto mantiene la base de datos en funcionamiento fuera de Elastic Beanstalk. Después de esto, cualquier entorno de Elastic Beanstalk debe conectarse a él como una base de datos externa. Si desea hacer una copia de seguridad de los datos sin

mantener la base de datos en funcionamiento, configure la política de eliminación para que tome una instantánea de la base de datos antes de terminar el entorno. Para obtener más información, consulte [Ciclo de vida de la base de datos](#) en el capítulo Configuración de entornos de esta guía.

Important

Si termina un entorno, también debe eliminar todas las asignaciones de CNAME que ha creado, ya que otros clientes pueden reutilizar un nombre de host disponible. Asegúrese de eliminar los registros de DNS que apuntan a su entorno terminado para evitar que se cree una entrada de DNS colgante. Una entrada de DNS colgante puede exponer el tráfico de Internet destinado a su dominio a vulnerabilidades de seguridad. También puede presentar otros riesgos.

Para más información, consulte [Protección contra registros de delegación colgantes en Route 53](#) en la Guía para desarrolladores de Amazon Route 53. También puede obtener más información sobre las entradas de DNS colgantes en [Protecciones de dominio mejoradas para solicitudes de Amazon CloudFront](#) en el Blog de seguridad de AWS.

Sintaxis

```
eb terminate
```

```
eb terminate environment-name
```

Opciones

Nombre	Descripción
<code>--all</code>	Termina todos los entornos de la aplicación, las versiones de la aplicación y sus configuraciones guardadas y, a continuación, elimina la aplicación.
<code>--force</code>	Termina el entorno sin pedir confirmación.
<code>--ignore-links</code>	Termina el entorno aunque haya otros entornos asociados que dependen de él. Consulte Compose Environments .
<code>--timeout</code>	Número de minutos que deben transcurrir antes de que se agote el tiempo de espera del comando.

Output

Si se ejecuta correctamente, el comando devuelve el estado de la operación terminate.

Ejemplo

La solicitud del ejemplo siguiente termina el entorno tmp-dev.

```
$ eb terminate
The environment "tmp-dev" and all associated instances will be terminated.
To confirm, type the environment name: tmp-dev
2018-07-11 21:05:25     INFO: terminateEnvironment is starting.
2018-07-11 21:05:40     INFO: Deleted CloudWatch alarm named: awseb-e-2cpfjbra9a-stack-
AWSEBCloudwatchAlarmHigh-16V08Y0F2KQ7U
2018-07-11 21:05:41     INFO: Deleted CloudWatch alarm named: awseb-e-2cpfjbra9a-stack-
AWSEBCloudwatchAlarmLow-6ZAWH9F20P7C
2018-07-11 21:06:42     INFO: Deleted Auto Scaling group policy named:
arn:aws:autoscaling:us-east-2:11122223333:scalingPolicy:5d7d3e6b-
d59b-47c5-b102-3e11fe3047be:autoScalingGroupName/awseb-e-2cpfjbra9a-stack-
AWSEBAutoScalingGroup-7AXY7U13ZQ6E:policyName/awseb-e-2cpfjbra9a-stack-AWSEBAutoSca
lingScaleUpPolicy-1876U27JEC34J
2018-07-11 21:06:43     INFO: Deleted Auto Scaling group policy named:
arn:aws:autoscaling:us-east-2:11122223333:scalingPolicy:29c6e7c7-7ac8-46fc-91f5-
cfabb65b985b:autoScalingGroupName/awseb-e-2cpfjbra9a-stack-
AWSEBAutoScalingGroup-7AXY7U13ZQ6E:policyName/awseb-e-2cpfjbra9a-stack-AWSEBAutoSca
lingScaleDownPolicy-SL4LH0DM0MU
2018-07-11 21:06:48     INFO: Waiting for EC2 instances to terminate. This may take a
few minutes.
2018-07-11 21:08:55     INFO: Deleted Auto Scaling group named: awseb-e-2cpfjbra9a-
stack-AWSEBAutoScalingGroup-7AXY7U13ZQ6E
2018-07-11 21:09:10     INFO: Deleted security group named: awseb-e-2cpfjbra9a-stack-
AWSEBSecurityGroup-XT4YYGFL7I99
2018-07-11 21:09:40     INFO: Deleted load balancer named: awseb-e-2-AWSEBLoa-
AK6RRYFQVV3S
2018-07-11 21:09:42     INFO: Deleting SNS topic for environment tmp-dev.
2018-07-11 21:09:52     INFO: terminateEnvironment completed successfully.
```

eb upgrade

Descripción

Actualiza la plataforma de su entorno a la versión más reciente de la plataforma en la que se está ejecutando en este momento.

Si el directorio raíz contiene un archivo `platform.yaml` que especifica una plataforma personalizada, este comando actualiza el entorno a la versión más reciente de la plataforma personalizada en la que se está ejecutando en este momento.

Sintaxis

```
eb upgrade
```

```
eb upgrade environment-name
```

Opciones

Nombre	Descripción
<code>--force</code>	Actualiza sin pedirle que confirme el nombre del entorno antes de empezar el proceso de actualización.
<code>--noroll</code>	Actualiza todas las instancias sin usar las actualizaciones continuas para mantener algunas instancias en servicio durante la actualización.

[Opciones comunes](#)

Salida

El comando muestra un resumen de los cambios y le pedirá que confirme la actualización escribiendo el nombre del entorno. En caso de éxito, el entorno se actualiza y se lanza con la versión más reciente de la plataforma.

Ejemplo

En el siguiente ejemplo se actualiza la versión de la plataforma actual del entorno especificado a la última versión de la plataforma disponible.

```
$ eb upgrade
```

```
Current platform: 64bit Amazon Linux 2014.09 v1.0.9 running Python 2.7
```

```
Latest platform: 64bit Amazon Linux 2014.09 v1.2.0 running Python 2.7
```

```
WARNING: This operation replaces your instances with minimal or zero downtime. You may  
cancel the upgrade after it has started by typing "eb abort".
```

You can also change your platform version by typing "eb clone" and then "eb swap".

To continue, type the environment name:

eb use

Descripción

Establece el entorno especificado en el entorno predeterminado.

Cuando se utiliza Git, eb use establece el entorno predeterminado de la ramificación actual. Ejecute este comando una vez en cada ramificación que desee implementar en Elastic Beanstalk.

Sintaxis

eb use ***environment-name***

Opciones

Nombre	Descripción
<code>--source codecommit/ <i>repository-name/branch-name</i></code>	Repositorio y ramificación de CodeCommit. Consulte Uso de la CLI de EB con AWS CodeCommit .
<code>-r <i>region</i></code> <code>--region <i>region</i></code>	Cambie la región en la que se van a crear los entornos.
Opciones comunes	

Opciones comunes

Puede utilizar las siguientes opciones con todos los comandos de la CLI de EB.

Nombre	Descripción
<code>--debug</code>	Imprimir información de depuración.
<code>-h, --help</code>	Mostrar el mensaje de ayuda.

Nombre	Descripción
	Tipo: String
	Valor predeterminado: None
<code>--no-verify-ssl</code>	Omitir la verificación de certificados SSL. Utilice esta opción si tiene problemas al usar la CLI con un proxy.
<code>--profile</code>	Utilice un perfil específico del archivo de credenciales de AWS.
<code>--quiet</code>	Suprimir todo el resultado del comando.
<code>--region</code>	Usar la región especificada.
<code>-v, --verbose</code>	Mostrar información detallada.

CLI de EB 2.6 (retirado)

Esta versión de la CLI de EB y su documentación se han sustituido por la versión 3 (en esta sección, EB CLI 3 representa la versión 3 y posteriores de la CLI de EB). Para obtener más información sobre la nueva versión, consulte [Uso de la interfaz de línea de comandos de Elastic Beanstalk \(CLI de EB\)](#).

Debe migrar a la última versión de EB CLI 3. Se pueden administrar los entornos lanzados con EB CLI 2.6 o versiones anteriores de la CLI de EB.

Diferencias de la versión 3 de la CLI de EB

EB es una interfaz de línea de comandos (CLI) para Elastic Beanstalk que se puede utilizar para implementar aplicaciones de forma rápida y más sencilla. La última versión de EB introducida en Elastic Beanstalk fue EB CLI 3. La CLI de EB recupera automáticamente la configuración de un entorno creado con EB si el entorno se está ejecutando. Tenga en cuenta que EB CLI 3 no almacena localmente los valores de las opciones, como en versiones anteriores.

La CLI de EB introduce los comandos `eb create`, `eb deploy`, `eb open`, `eb console`, `eb scale`, `eb setenv`, `eb config`, `eb terminate`, `eb clone`, `eb list`, `eb use`, `eb printenv` y `eb ssh`. En EB CLI 3.1 o posterior, también puede utilizar el comando `eb swap`. En EB CLI 3.2 solo, puede utilizar los comandos `eb abort`, `eb platform` y `eb upgrade`. Además de estos nuevos comandos, los comandos de EB CLI 3 se diferencian de los comandos de EB CLI 2.6 en varios casos:

- `eb init`: utilice `eb init` para crear un directorio `.elasticbeanstalk` en el directorio del proyecto existente y crear una nueva aplicación de Elastic Beanstalk para el proyecto. A diferencia de lo que sucede con las versiones anteriores, EB CLI 3 y las versiones posteriores no le piden que cree un entorno.
- `eb start`: EB CLI 3 no incluye el comando `eb start`. Use `eb create` para crear un entorno.
- `eb stop`: EB CLI 3 no incluye el comando `eb stop`. Utilice `eb terminate` para terminar completamente un entorno y limpiarlo.
- `eb push` y `git aws.push`: EB CLI 3 no incluye los comandos `eb push` y `git aws.push`. Use `eb deploy` para actualizar el código de la aplicación.
- `eb update`: EB CLI 3 no incluye el comando `eb update`. Use `eb config` para actualizar un entorno.
- `eb branch`: EB CLI 3 no incluye el comando `eb branch`.

Para obtener más información acerca del uso de los comandos de EB CLI 3 para crear y administrar una aplicación, consulte [Referencia de los comandos de la CLI de EB](#). Para obtener una tutorial de implementación de una aplicación de ejemplo con EB CLI 3, consulte [Administración de entornos de Elastic Beanstalk con la CLI de EB](#).

Migración a EB CLI 3 y CodeCommit

Elastic Beanstalk no solo ha retirado la CLI de EB 2.6, sino que también ha quitado algunas funcionalidades de 2.6. El cambio más importante de la versión 2.6 es que la CLI de EB ya no admite de forma nativa las actualizaciones de código incrementales (`eb push`, `git aws.push`) ni las ramificaciones (`eb branch`). En esta sección se describe cómo migrar de EB CLI 2.6 a la última versión de la CLI de EB y utilizar CodeCommit como repositorio de código.

Si aún no lo ha hecho, cree un repositorio de código en CodeCommit, tal y como se describe en [Migración a CodeCommit](#).

Una vez que haya [instalado](#) y [configurado](#) la CLI de EB, dispone de dos ocasiones para asociar la aplicación a su repositorio de CodeCommit, incluida una ramificación específica.

- Cuando ejecuta `eb init`, como en el siguiente ejemplo, donde *myRepo* es el nombre de su repositorio de CodeCommit y *myBranch* es la ramificación de CodeCommit.

```
eb init --source codecommit/myRepo/myBranch
```

- Cuando ejecuta `eb deploy`, como en el siguiente ejemplo, donde *myRepo* es el nombre de su repositorio de CodeCommit y *myBranch* es la ramificación de CodeCommit.

```
eb deploy --source codecommit/myRepo/myBranch
```

Para obtener más información, incluido cómo implementar actualizaciones de código incrementales en un entorno de Elastic Beanstalk sin tener que volver a cargar todo el proyecto, consulte [Uso de la CLI de EB con AWS CodeCommit](#).

Interfaz de línea de comandos de Elastic Beanstalk API (retirada)

Esta herramienta, la interfaz de línea de comandos de la API de Elastic Beanstalk (CLI de API), ha sido reemplazada por la AWS CLI, que proporciona comandos equivalentes a la API para todos los servicios de AWS. Consulte la Guía del usuario de AWS Command Line Interface para empezar a utilizar la AWS CLI. Pruebe también la [CLI de EB](#) para experimentar con una línea de comandos de nivel superior simplificada.

Conversión de scripts de la CLI de la API de Elastic Beanstalk

Convierta los antiguos scripts de la CLI de la API de EB para utilizar la AWS CLI o Tools for Windows PowerShell a fin de obtener acceso a las últimas API de Elastic Beanstalk. En la siguiente tabla se enumeran los comandos de CLI basados en la API de Elastic Beanstalk y los comandos equivalentes en la AWS CLI y Tools for Windows PowerShell.

CLI de la API de Elastic Beanstalk	AWS CLI	AWS Tools for Windows PowerShell
<code>elastic-beanstalk-check-dns-availability</code>	check-dns-availability	<code>Get-EBDNSAvailability</code>
<code>elastic-beanstalk-create-application</code>	create-application	<code>New-EBApplication</code>

CLI de la API de Elastic Beanstalk	AWS CLI	AWS Tools for Windows PowerShell
<code>elastic-beanstalk-create-application-version</code>	<u>create-application-version</u>	<code>New-EBApplicationVersion</code>
<code>elastic-beanstalk-create-configuration-template</code>	<u>create-configuration-template</u>	<code>New-EBConfigurationTemplate</code>
<code>elastic-beanstalk-create-environment</code>	<u>create-environment</u>	<code>New-EBEnvironment</code>
<code>elastic-beanstalk-create-storage-location</code>	<u>create-storage-location</u>	<code>New-EBStorageLocation</code>
<code>elastic-beanstalk-delete-application</code>	<u>delete-application</u>	<code>Remove-EBApplication</code>
<code>elastic-beanstalk-delete-application-version</code>	<u>delete-application-version</u>	<code>Remove-EBApplicationVersion</code>
<code>elastic-beanstalk-delete-configuration-template</code>	<u>delete-configuration-template</u>	<code>Remove-EBConfigurationTemplate</code>
<code>elastic-beanstalk-delete-environment-configuration</code>	<u>delete-environment-configuration</u>	<code>Remove-EBEnvironmentConfiguration</code>

CLI de la API de Elastic Beanstalk	AWS CLI	AWS Tools for Windows PowerShell
<code>elastic-beanstalk-describe-application-versions</code>	<u>describe-application-versions</u>	<code>Get-EBApplicationVersion</code>
<code>elastic-beanstalk-describe-applications</code>	<u>describe-applications</u>	<code>Get-EBApplication</code>
<code>elastic-beanstalk-describe-configuration-options</code>	<u>describe-configuration-options</u>	<code>Get-EBConfigurationOption</code>
<code>elastic-beanstalk-describe-configuration-settings</code>	<u>describe-configuration-settings</u>	<code>Get-EBConfigurationSetting</code>
<code>elastic-beanstalk-describe-environment-resources</code>	<u>describe-environment-resources</u>	<code>Get-EBEnvironmentResource</code>
<code>elastic-beanstalk-describe-environments</code>	<u>describe-environments</u>	<code>Get-EBEnvironment</code>
<code>elastic-beanstalk-describe-events</code>	<u>describe-events</u>	<code>Get-EBEvent</code>
<code>elastic-beanstalk-list-available-solution-stacks</code>	<u>list-available-solution-stacks</u>	<code>Get-EBAvailableSolutionStack</code>

CLI de la API de Elastic Beanstalk	AWS CLI	AWS Tools for Windows PowerShell
<code>elastic-beanstalk-rebuild-environment</code>	<u>rebuild-environment</u>	<code>Start-EBEnvironmentRebuild</code>
<code>elastic-beanstalk-request-environment-info</code>	<u>request-environment-info</u>	<code>Request-EBEnvironmentInfo</code>
<code>elastic-beanstalk-restart-app-server</code>	<u>restart-app-server</u>	<code>Restart-EBAppServer</code>
<code>elastic-beanstalk-retrieve-environment-info</code>	<u>retrieve-environment-info</u>	<code>Get-EBEnvironmentInfo</code>
<code>elastic-beanstalk-swap-environment-cnames</code>	<u>swap-environment-cnames</u>	<code>Set-EBEnvironmentCNAME</code>
<code>elastic-beanstalk-terminate-environment</code>	<u>terminate-environment</u>	<code>Stop-EBEnvironment</code>
<code>elastic-beanstalk-update-application</code>	<u>update-application</u>	<code>Update-EBApplication</code>
<code>elastic-beanstalk-update-application-version</code>	<u>update-application-version</u>	<code>Update-EBApplicationVersion</code>

CLI de la API de Elastic Beanstalk	AWS CLI	AWS Tools for Windows PowerShell
<code>elastic-beanstalk-update-configuration-template</code>	<u>update-configuration-template</u>	<code>Update-EBConfigurationTemplate</code>
<code>elastic-beanstalk-update-environment</code>	<u>update-environment</u>	<code>Update-EBEnvironment</code>
<code>elastic-beanstalk-validate-configuration-settings</code>	<u>validate-configuration-settings</u>	<code>Test-EBConfigurationSetting</code>

AWS Elastic Beanstalk Seguridad

La seguridad en la nube de AWS es nuestra máxima prioridad. Como cliente de AWS, se beneficia de una arquitectura de red y un centro de datos que se han diseñado para satisfacer los requisitos de seguridad de las organizaciones más exigentes.

La seguridad es una responsabilidad compartida entre AWS y usted. El [modelo de responsabilidad compartida](#) describe esto como la seguridad de la nube y la seguridad en la nube:

Seguridad de la nube: AWS es responsable de proteger la infraestructura que ejecuta todos los servicios ofrecidos en la nube de AWS y proporcionar servicios que usted puede utilizar de forma segura. Nuestra responsabilidad en torno a la seguridad es la mayor prioridad en AWS y auditores externos prueban y verifican la eficacia de nuestra seguridad con frecuencia como parte de los <https://aws.amazon.com/compliance/programs/> programas de conformidad de AWS. Revise los [AWS servicios de AWS en el ámbito de los programas de control](#) para obtener información relacionada con Elastic Beanstalk.

Seguridad en la nube: su responsabilidad viene determinada por el servicio de AWS que usa y otros factores, como la confidencialidad de los datos, los requisitos de la organización y las normas y reglamentos aplicables. Esta documentación tiene por objetivo ayudarle a comprender cómo aplicar el modelo de responsabilidad compartida cuando se utiliza Elastic Beanstalk.

Use los siguientes temas de seguridad para obtener más información sobre las tareas de seguridad de las que Elastic Beanstalk es responsable, así como las configuraciones de seguridad que debe tener en cuenta al usar Elastic Beanstalk para cumplir sus objetivos de seguridad y conformidad.

Temas

- [Protección de datos en Elastic Beanstalk](#)
- [Identity and Access Management para Elastic Beanstalk](#)
- [Registro y supervisión en Elastic Beanstalk](#)
- [Validación de conformidad para Elastic Beanstalk](#)
- [Resiliencia en Elastic Beanstalk](#)
- [Seguridad de la infraestructura en Elastic Beanstalk](#)
- [Análisis de configuración y vulnerabilidad en Elastic Beanstalk](#)
- [Prácticas recomendadas de seguridad para Elastic Beanstalk](#)

Protección de datos en Elastic Beanstalk

El [modelo de responsabilidad compartida](#) de AWS se aplica a la protección de datos de AWS Elastic Beanstalk. Como se describe en este modelo, AWS es responsable de proteger la infraestructura global que ejecuta toda la Nube de AWS. Usted es responsable de mantener el control sobre el contenido alojado en esta infraestructura. También es responsable de la configuración de seguridad y de las tareas de administración para los Servicios de AWS que utiliza. Para obtener más información sobre la privacidad de los datos, consulte las [Preguntas frecuentes sobre la privacidad de datos](#). Para obtener información sobre la protección de datos en Europa, consulte la publicación de blog sobre el [Modelo de responsabilidad compartida de AWS y GDPR](#) en el Blog de seguridad de AWS.

Para proteger los datos, recomendamos proteger las credenciales de Cuenta de AWS y configurar cuentas de usuario individuales con AWS IAM Identity Center o AWS Identity and Access Management (IAM). De esta manera, cada usuario recibe solamente los permisos necesarios para cumplir con sus obligaciones laborales. También recomendamos proteger sus datos de la siguiente manera:

- Utilice autenticación multifactor (MFA) en cada cuenta.
- Utilice SSL/TLS para comunicarse con los recursos de AWS. Exigimos TLS 1.2 y recomendamos TLS 1.3.
- Configure la API y el registro de actividad del usuario con AWS CloudTrail.
- Utilizar las soluciones de cifrado de AWS, junto con todos los controles de seguridad predeterminados dentro de los servicios de Servicios de AWS.
- Utilice servicios de seguridad administrados avanzados, como Amazon Macie, que lo ayuden a detectar y proteger los datos confidenciales almacenados en Amazon S3.
- Si necesita módulos criptográficos validados FIPS 140-2 al acceder a AWS a través de una interfaz de línea de comandos o una API, utilice un punto de conexión de FIPS. Para obtener más información sobre los puntos de conexión de FIPS disponibles, consulte [Estándar de procesamiento de la información federal \(FIPS\) 140-2](#).

Se recomienda encarecidamente no ingresar información confidencial o sensible, como por ejemplo direcciones de correo electrónico de clientes, en etiquetas o campos de formato libre, tales como el campo Nombre. Esto incluye cuando se trabaja con Elastic Beanstalk u otros Servicios de AWS con la consola, la API, la AWS CLI o los SDK de AWS. Cualquier dato que ingrese en etiquetas o campos de formato libre utilizados para nombres puede ser empleado para los registros de facturación o

diagnóstico. Si proporciona una URL a un servidor externo, recomendamos encarecidamente que no incluya información de credenciales en la URL a fin de validar la solicitud para ese servidor.

Para obtener otros temas de seguridad de Elastic Beanstalk, consulte [AWS Elastic BeanstalkSeguridad](#).

Temas

- [Protección de datos mediante cifrado](#)
- [Privacidad del tráfico entre redes](#)

Protección de datos mediante cifrado

Elastic Beanstalk almacena varios objetos en un bucket de Amazon Simple Storage Service (Amazon S3) que crea para cada región de AWS en la que usted crea entornos. Para obtener más información, consulte [the section called “Simple Storage Service \(Amazon S3\)”](#).

Proporciona algunos de los objetos almacenados y los envía a Elastic Beanstalk, por ejemplo, versiones de aplicaciones y paquetes de origen. Elastic Beanstalk genera otros objetos, por ejemplo, archivos de registro. Además de los datos que almacena Elastic Beanstalk, su aplicación puede transferir o almacenar datos como parte de su operación.

La protección de datos consiste en proteger los datos mientras están en tránsito (cuando viajan hacia y desde Elastic Beanstalk) y en reposo (mientras están almacenados en centros de datos de AWS).

Cifrado en tránsito

Puede conseguir la protección de datos en tránsito de dos formas: cifre la conexión mediante Capa de sockets seguros (SSL) o use cifrado del lado del cliente (donde el objeto se cifra antes de enviarse). Ambos métodos son válidos para proteger los datos de la aplicación. Para proteger la conexión, cífrala mediante SSL siempre que su aplicación, sus desarrolladores y administradores, y sus usuarios finales envíen o reciban cualquier objeto. Para obtener detalles acerca del cifrado de tráfico web a y desde su aplicación, consulte [the section called “HTTPS”](#).

El cifrado del lado del cliente no es un método válido para proteger el código fuente en versiones de aplicaciones y paquetes de origen que carga. Elastic Beanstalk necesita acceso a estos objetos, por lo que no se pueden cifrar. Por tanto, asegúrese de proteger la conexión entre su entorno de implementación o desarrollo y Elastic Beanstalk.

Cifrado en reposo

Para proteger los datos de la aplicación en reposo, obtenga información sobre la protección de datos en el servicio de almacenamiento que su aplicación usa. Por ejemplo, consulte [Protección de datos en Amazon RDS](#) en la Guía del usuario de Amazon RDS, [Protección de datos en Amazon S3](#) en la Guía del usuario de Amazon Simple Storage Service o [Cifrado de datos y metadatos en EFS](#) en la Guía del usuario de Amazon Elastic File System.

Elastic Beanstalk no activa el cifrado predeterminado para el bucket de Amazon S3 que crea. Esto significa que, de forma predeterminada, los objetos se almacenan en el bucket sin cifrar (y solo pueden acceder los usuarios autorizados para leer el bucket). Si su aplicación requiere cifrado en reposo, puede configurar los buckets de su cuenta para el cifrado predeterminado. Para obtener más información, consulte [Cifrado predeterminado de Amazon S3 para buckets de S3](#) en la Guía del usuario de Amazon Simple Storage Service.

Para obtener más información sobre la protección de datos, consulte la entrada de blog relativa al [modelo de responsabilidad compartida de AWS y GDPR](#) en el blog de seguridad de AWS.

Para obtener otros temas de seguridad de Elastic Beanstalk, consulte [AWS Elastic BeanstalkSeguridad](#).

Privacidad del tráfico entre redes

Puede utilizar Amazon Virtual Private Cloud (Amazon VPC) para crear límites entre los recursos de la aplicación de Elastic Beanstalk y controlar el tráfico entre ellos, la red local e Internet. Para obtener más información, consulte [the section called "Amazon VPC"](#).

Para obtener más información sobre la seguridad de Amazon VPC, consulte [Seguridad](#) en la Guía del usuario de Amazon VPC.

Para obtener más información sobre la protección de datos, consulte la entrada de blog relativa al [modelo de responsabilidad compartida de AWS y GDPR](#) en el blog de seguridad de AWS.

Para obtener otros temas de seguridad de Elastic Beanstalk, consulte [AWS Elastic BeanstalkSeguridad](#).

Identity and Access Management para Elastic Beanstalk

AWS Identity and Access Management (IAM) es un servicio de AWS que ayuda al administrador a controlar de forma segura el acceso a los recursos de AWS. Los administradores de IAM controlan

quién está autenticado (ha iniciado sesión) y autorizado (tiene permisos) para utilizar recursos de AWS Elastic Beanstalk. IAM es un servicio de AWS que puede utilizar sin cargo adicional.

Para más detalles sobre cómo trabajar con IAM, consulte [Uso de Elastic Beanstalk con AWS Identity and Access Management](#).

Para obtener otros temas de seguridad de Elastic Beanstalk, consulte [AWS Elastic Beanstalk Seguridad](#).

AWS políticas gestionadas para AWS Elastic Beanstalk

Una política AWS administrada es una política independiente creada y administrada por AWS. Las políticas administradas están diseñadas para proporcionar permisos para muchos casos de uso comunes, de modo que pueda empezar a asignar permisos a usuarios, grupos y funciones.

Ten en cuenta que es posible que las políticas AWS administradas no otorguen permisos con privilegios mínimos para tus casos de uso específicos, ya que están disponibles para que los usen todos los AWS clientes. Se recomienda definir [políticas administradas por el cliente](#) específicas para sus casos de uso a fin de reducir aún más los permisos.


No puedes cambiar los permisos definidos en AWS las políticas administradas. Si AWS actualiza los permisos definidos en una política AWS administrada, la actualización afecta a todas las identidades principales (usuarios, grupos y roles) a las que está asociada la política. AWS es más probable que actualice una política AWS administrada cuando Servicio de AWS se lance una nueva o cuando estén disponibles nuevas operaciones de API para los servicios existentes.

Para obtener más información, consulte [Políticas administradas por AWS](#) en la Guía del usuario de IAM.

Actualizaciones de Elastic AWS Beanstalk a las políticas administradas

Consulte los detalles sobre las actualizaciones de las políticas AWS administradas de Elastic Beanstalk desde el 1 de marzo de 2021.

Para ver la fuente de JSON de una política administrada específica, consulte la Guía de referencia de [políticas AWS administradas](#).

Cambio	Descripción	Fecha
<p>Se actualizaron las siguientes políticas:</p> <ul style="list-style-type: none"> • AWSElasticBeanstalkInternalMaintenanceRolePolicy • AWSElasticBeanstalkMaintenance • AWSElasticBeanstalkManagedUpdatesInternalServiceRolePolicy • AWSElasticBeanstalkManagedUpdatesServiceRolePolicy • AWSElasticBeanstalkRoleCore 	<p>Estas políticas se actualizaron para permitir a Elastic Beanstalk añadir o eliminar etiquetas al crear o actualizar una pila o un conjunto de AWS CloudFormation cambios.</p> <p>Para obtener más información acerca de AWSElasticBeanstalkManagedUpdatesServiceRolePolicy, consulte Permisos de rol vinculado a servicios para Elastic Beanstalk.</p> <p>Para obtener más información acerca de AWSElasticBeanstalkRoleCore, consulte Políticas para la integración con otros servicios.</p>	30 de abril de 2024
<p>AWSElasticBeanstalkService—Se actualizó la política existente</p>	<p>Se actualizó esta política para permitir que Elastic Beanstalk etiquete recursos al crearlos para Elastic Load Balancing, grupos de escalado automático (ASG) y Amazon ECS.</p> <div data-bbox="591 1556 1029 1879" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>Esta política se ha reemplazado anteriormente con AWSElasticBeanstalk</p> </div>	10 de mayo de 2023

Cambio	Descripción	Fecha
	<p data-bbox="667 212 976 814"> kManagedU pdatesCus tomerRole Policy . Aunque esta política ya no está disponibl e para asociarla a nuevos usuarios, grupos o roles de IAM, se puede seguir asociando a los existentes previamen te. </p> <p data-bbox="591 932 1027 1058"> Para obtener más informaci ón, consulte Políticas de rol de servicio administradas. </p>	
<p data-bbox="112 1104 545 1234"> AWSElasticBeanstalkMulticon tainerDocker— Se actualizó la política existente </p>	<p data-bbox="591 1104 1016 1283"> Se actualizó esta política para permitir que Elastic Beanstalk etiquete recursos al crearlos para Amazon ECS. </p> <p data-bbox="591 1329 1027 1507"> Para obtener más informaci ón, consulte Administración de perfiles de instancia de Elastic Beanstalk. </p>	<p data-bbox="1068 1104 1370 1136">23 de marzo de 2023</p>

Cambio	Descripción	Fecha
<p>AWSElasticBeanstalkRoleECS— Se actualizó la política existente</p>	<p>Se actualizó esta política para permitir que Elastic Beanstalk etiquete recursos al crearlos para Amazon ECS.</p> <p>Para obtener más información, consulte Políticas para la integración con otros servicios.</p>	<p>23 de marzo de 2023</p>
<p>AdministratorAccess-AWSElasticBeanstalk —Se actualizó la política existente</p>	<p>Se actualizó esta política para permitir que Elastic Beanstalk etiquete recursos al crearlos para Amazon ECS.</p> <p>Para obtener más información, consulte Administración de políticas de usuario de Elastic Beanstalk.</p>	<p>23 de marzo de 2023</p>
<p>AWSElasticBeanstalkManagedUpdatesServiceRolePolicy — Se actualizó la política existente</p>	<p>Esta política se actualizó para permitir que Elastic Beanstalk agregue etiquetas a los recursos de Amazon ECS al crearlos.</p> <p>Para obtener más información, consulte Permisos de rol vinculado a servicios para Elastic Beanstalk.</p>	<p>23 de marzo de 2023</p>

Cambio	Descripción	Fecha
<p>AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy— Se actualizó la política existente</p>	<p>Esta política se actualizó para permitir que Elastic Beanstalk agregue etiquetas a los recursos de Amazon ECS al crearlos.</p> <p>Para obtener más información, consulte Políticas de rol de servicio administradas.</p>	<p>23 de marzo de 2023</p>
<p>AWSElasticBeanstalkManagedUpdatesServiceRolePolicy— Se actualizó la política existente</p>	<p>Esta política se actualizó para permitir que Elastic Beanstalk agregue etiquetas a grupos de escalado automático al crearlos.</p> <p>Para obtener más información, consulte Rol vinculado al servicio de actualizaciones administradas.</p>	<p>27 de enero de 2023</p>
<p>AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy— Se actualizó la política existente</p>	<p>Esta política se actualizó para permitir que Elastic Beanstalk agregue etiquetas al crear un grupo de escalado automático (ASG).</p> <p>Para obtener más información, consulte Políticas de rol de servicio administradas.</p>	<p>23 de enero de 2023</p>

Cambio	Descripción	Fecha
<p>AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy— Se actualizó la política existente</p>	<p>Esta política se actualizó para permitir que Elastic Beanstalk agregue etiquetas al crear un equilibrador de carga elástico (ELB).</p> <p>Para obtener más información, consulte Políticas de rol de servicio administradas.</p>	<p>21 de diciembre de 2022</p>
<p>AWSElasticBeanstalkManagedUpdatesServiceRolePolicy— Se actualizó la política existente</p>	<p>Se agregaron permisos a esta política administrada para permitir que Elastic Beanstalk haga lo siguiente durante las actualizaciones administradas:</p> <ul style="list-style-type: none"> • Crear y eliminar plantillas de lanzamiento y versiones de plantillas. • Lanzar instancias de Amazon EC2 con plantillas de lanzamiento. • Si hay un Amazon RDS presente, recuperar una lista de los motores de base de datos disponibles e información sobre las instancias de RDS aprovisionadas. <p>Para obtener más información, consulte Rol vinculado al servicio de actualizaciones administradas.</p>	<p>23 de agosto de 2022</p>

Cambio	Descripción	Fecha
<p>AWSElasticBeanstalkReadOnlyAccess— En desuso</p> <p>GovCloud (EE. UU.) Región de AWS</p>	<p>Esta política ha sido reemplazada por AWSElasticBeanstalkReadOnlyAccess.</p> <p>Esta política se eliminará gradualmente en los GovCloud EE. UU. Región de AWS</p> <p>Cuando esta política se elimine gradualmente, ya no estará disponible para asociarla a usuarios, grupos o roles de IAM nuevos después del 17 de junio de 2021.</p> <p>Para obtener más información, consulte Políticas de usuario.</p>	<p>17 de junio de 2021</p>
<p>AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy— Se actualizó la política existente</p>	<p>Esta política se ha actualizado para permitir a Elastic Beanstalk que lea los atributos de las zonas de disponibilidad de EC2. Habilita a Elastic Beanstalk para proporcionar una validación más efectiva de la selección del tipo de instancia en las zonas de disponibilidad.</p> <p>Para obtener más información, consulte Políticas de rol de servicio administradas.</p>	<p>16 de junio de 2021</p>

Cambio	Descripción	Fecha
AWSElasticBeanstalkFullAccess— En desuso GovCloud (EE. UU.) Región de AWS	<p>Esta política ha sido reemplazada por AdministratorAccess-AWSElasticBeanstalk .</p> <p>Esta política se eliminará gradualmente en los GovCloud EE. UU. Región de AWS</p> <p>Cuando esta política se elimine gradualmente, ya no estará disponible para asociarla a usuarios, grupos o roles de IAM nuevos después del 10 de junio de 2021.</p> <p>Para obtener más información, consulte Políticas de usuario.</p>	10 de junio de 2021

Cambio	Descripción	Fecha
<p>Las siguientes políticas gestionadas quedaron obsoletas en todos los países de China Región de AWS:</p> <ul style="list-style-type: none">• AWSElasticBeanstalkFullAccess• AWSElasticBeanstalkReadOnlyAccess	<p>La política AWSElasticBeanstalkFullAccess ha sido reemplazada por AdministratorAccess-AWSElasticBeanstalk .</p> <p>La política AWSElasticBeanstalkReadOnlyAccess ha sido reemplazada por AWSElasticBeanstalkReadOnly .</p> <p>Estas políticas se eliminaron gradualmente en toda China Región de AWS.</p> <p>Estas políticas ya no estarán disponibles para asociarlas a usuarios, grupos o roles de IAM nuevos después del 3 de junio de 2021.</p> <p>Para obtener más información, consulte Políticas de usuario.</p>	<p>3 de junio de 2021</p>

Cambio	Descripción	Fecha
AWSElasticBeanstalkService— En desuso	<p>Esta política se reemplazó por <code>AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy</code>.</p> <p>Esta política se eliminó gradualmente y ya no está disponible para asociarla a usuarios, grupos o roles de IAM nuevos.</p> <p>Para obtener más información, consulte Políticas de rol de servicio administradas.</p>	De junio de 2021 a enero de 2022

Cambio	Descripción	Fecha
<p>Las siguientes políticas gestionadas quedaron obsoletas en todos los Región de AWS años, excepto en China y GovCloud (EE. UU.):</p> <ul style="list-style-type: none"> • AWSElasticBeanstalkFullAccess • AWSElasticBeanstalkReadOnlyAccess 	<p>La política AWSElasticBeanstalkFullAccess ha sido reemplazada por AdministratorAccess-AWSElasticBeanstalk .</p> <p>La política AWSElasticBeanstalkReadOnlyAccess ha sido reemplazada por AWSElasticBeanstalkReadOnly .</p> <p>Estas políticas se eliminarán gradualmente en todos los Región de AWS años, excepto en China y GovCloud (EE. UU.).</p> <p>Estas políticas dejarán de estar disponibles para asociarlas a usuarios, grupos o roles de IAM nuevos después del 16 de abril de 2021.</p> <p>Para obtener más información, consulte Políticas de usuario.</p>	<p>16 de abril de 2021</p>

Cambio	Descripción	Fecha
<p>Se actualizaron las siguientes políticas administradas:</p> <ul style="list-style-type: none"> • AdministratorAccess-AWSElasticBeanstalk • AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy 	<p>Ambas políticas ahora admiten PassRole permisos en China Regiones de AWS.</p> <p>Para obtener más información sobre AdministratorAccess-AWSElasticBeanstalk , consulte Políticas de usuario.</p> <p>Para obtener más información sobre AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy , consulte Políticas de rol de servicio administradas.</p>	<p>9 de marzo de 2021</p>
<p>AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy: política nueva</p>	<p>Elastic Beanstalk agregó una política nueva para reemplazar la política administrada AWSElasticBeanstalkService .</p> <p>Esta política administrada nueva mejora la seguridad de los recursos mediante la aplicación de un conjunto de permisos más restrictivo.</p> <p>Para obtener más información, consulte Políticas de rol de servicio administradas.</p>	<p>3 de marzo de 2021</p>

Cambio	Descripción	Fecha
Elastic Beanstalk comenzó a rastrear los cambios	Elastic Beanstalk comenzó a realizar un seguimiento de los cambios en las políticas administradas. AWS	1 de marzo de 2021

Registro y supervisión en Elastic Beanstalk

El monitoreo es importante para el mantenimiento de la confiabilidad, la disponibilidad y el rendimiento de AWS Elastic Beanstalk y sus soluciones de AWS. Debe recopilar datos de monitoreo de todas las partes de su solución de AWS para que pueda depurar un error multipunto de una forma más fácil si se produce. AWS proporciona varias herramientas para monitorear sus recursos de Elastic Beanstalk y responder a posibles incidentes, tales como los siguientes.

Para obtener más información sobre la supervisión, consulte [Monitorización de un entorno](#).

Para obtener otros temas de seguridad de Elastic Beanstalk, consulte [AWS Elastic Beanstalk Seguridad](#).

Informes de estado mejorados

Los informes de estado mejorados son una característica que puede habilitar en su entorno para permitir que Elastic Beanstalk recopile información adicional acerca de los recursos de su entorno. Elastic Beanstalk analiza la información recopilada para proporcionar una imagen más completa del estado general del entorno y ayudar a identificar los problemas que pueden causar que su aplicación deje de estar disponible. Para obtener más información, consulte [Informes y monitorización de estado mejorados](#).

Registros de instancias de Amazon EC2

Las instancias EC2 de Amazon en el entorno de Elastic Beanstalk generan registros que se pueden consultar para solucionar problemas relacionados con la aplicación o los archivos de configuración. Los registros creados por el servidor web, el servidor de aplicaciones, los scripts de la plataforma Elastic Beanstalk y AWS CloudFormation se almacenan localmente en instancias individuales. Puede recuperarlos de forma sencilla a través de la [consola de administración del entorno](#) o la CLI de EB. También puede configurar su entorno para transmitir registros a Amazon CloudWatch Logs

en tiempo real. Para obtener más información, consulte [Visualización de registros de instancias de Amazon EC2 en su entorno de Elastic Beanstalk](#).

Notificaciones del entorno

Puede configurar su entorno de Elastic Beanstalk para utilizar Amazon Simple Notification Service (Amazon SNS) para notificarle eventos importantes que afecten a su aplicación. Especifique una dirección de correo electrónico durante o después de la creación del entorno para recibir un correo electrónico de AWS cuando se produzca un error o cuando cambie el estado del entorno. Para obtener más información, consulte [Notificaciones de entorno de Elastic Beanstalk con Amazon SNS](#).

Alarmas de Amazon CloudWatch

Las alarmas de Amazon CloudWatch le permiten ver una sola métrica durante el período de tiempo que especifique. Si la métrica supera un determinado umbral, se envía una notificación a un tema de Amazon SNS o política de AWS Auto Scaling. Las alarmas de CloudWatch no invocan acciones simplemente porque estén en un estado particular. En su lugar, las alarmas invocan acciones si el estado ha cambiado y se ha mantenido para un número de periodos especificado. Para obtener más información, consulte [Uso de Elastic Beanstalk con Amazon CloudWatch](#).

Registros de AWS CloudTrail

CloudTrail proporciona un registro de las medidas que adopta un usuario, un rol o un servicio de AWS en Elastic Beanstalk. Mediante la información recopilada por CloudTrail, puede determinar la solicitud que se realizó a Elastic Beanstalk, la dirección IP desde la que se realizó, quién la realizó y cuándo, etc. Para obtener más información, consulte [Registro de llamadas a la API de Elastic Beanstalk con AWS CloudTrail](#).

Debugging deAWS X-Ray

X-Ray es un servicio de AWS que recopila datos sobre las solicitudes que la aplicación atiende y los utiliza para crear un mapa de servicios que puede utilizar para identificar problemas con la aplicación y oportunidades de optimización. Puede utilizar la consola AWS Elastic Beanstalk o un archivo de configuración para ejecutar el daemon de X-Ray en las instancias del entorno. Para obtener más información, consulte [Configuración de la depuración de AWS X-Ray](#).

Validación de conformidad para Elastic Beanstalk

Audidores externos evalúan la seguridad y la conformidad de AWS Elastic Beanstalk como parte de varios programas de conformidad de AWS. Entre estos se incluyen SOC, PCI, FedRAMP, HIPAA y otros. AWS proporciona una lista actualizada con frecuencia de los servicios de AWS adscritos al ámbito de programas de conformidad específicos en [Servicios de AWS en el ámbito del programa de conformidad](#).

Los informes de auditoría de terceros están disponibles para su descarga mediante AWS Artifact. Para obtener más información, consulte [Descarga de informes en AWS Artifact](#).

Para obtener más información sobre los programas de conformidad de AWS, consulte [Programas de conformidad de AWS](#).

Su responsabilidad de conformidad al utilizar Elastic Beanstalk se determina en función de la sensibilidad de los datos, los objetivos de conformidad de su organización, así como de la legislación y los reglamentos aplicables. Si el uso que hace de Elastic Beanstalk está sujeto a estándares de conformidad como HIPAA, PCI o FedRAMP, AWS proporciona recursos útiles:

- [Guías de inicio rápido de seguridad y conformidad](#): guías de implementación que ofrecen consideraciones sobre la arquitectura y los pasos para implementar entornos de referencia centrados en la seguridad y la conformidad de AWS.
- [Documento técnico sobre el diseño de arquitecturas para cumplir los requisitos de seguridad y conformidad de HIPAA](#): documento técnico en el que se describe cómo las empresas pueden utilizar AWS para crear aplicaciones conformes con HIPAA.
- [Recursos de conformidad de AWS](#): un conjunto de manuales y guías de conformidad que podrían aplicarse a su sector y ubicación.
- [AWS Config](#): un servicio que evalúa en qué medida las configuraciones de los recursos cumplen las prácticas internas, las directrices del sector y la normativa.
- [AWS Security Hub](#): un servicio que ofrece una vista integral de su estado de seguridad en AWS que lo ayuda a comprobar la conformidad con las normas del sector de seguridad y las prácticas recomendadas.

Para consultar otros temas de seguridad de Elastic Beanstalk, consulte [AWS Elastic BeanstalkSeguridad](#).

Resiliencia en Elastic Beanstalk

La infraestructura global de AWS se compone de regiones y zonas de disponibilidad de AWS.

Las regiones de AWS proporcionan varias zonas de disponibilidad físicamente independientes y aisladas que se encuentran conectadas mediante redes con un alto nivel de rendimiento y redundancia, además de baja latencia.

Con las zonas de disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que realizan una conmutación por error automática entre zonas de disponibilidad sin interrupciones. Las zonas de disponibilidad tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de centros de datos únicos o múltiples.

Para obtener más información sobre las zonas de disponibilidad y las regiones de AWS, consulte [Infraestructura global de AWS](#).

AWS Elastic Beanstalk administra y automatiza el uso de la infraestructura global de AWS en su nombre. Al utilizar Elastic Beanstalk, se beneficia de la disponibilidad y los mecanismos de tolerancia a errores que ofrece AWS.

Para consultar otros temas de seguridad de Elastic Beanstalk, consulte [AWS Elastic Beanstalk Seguridad](#).

Seguridad de la infraestructura en Elastic Beanstalk

Al tratarse de un servicio administrado, AWS Elastic Beanstalk está protegido por los procedimientos de seguridad de red globales de AWS que se describen en el documento técnico [Amazon Web Services: Información general sobre los procesos de seguridad](#).

Puede utilizar llamadas a la API publicadas de AWS para obtener acceso a Elastic Beanstalk a través de la red. Los clientes deben ser compatibles con Transport Layer Security (TLS) 1.0 o una versión posterior. Le recomendamos TLS 1.2 o una versión posterior. Los clientes también deben ser compatibles con conjuntos de cifrado con confidencialidad directa total (PFS), como Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La mayoría de las plataformas modernas como Java 7 y posteriores son compatibles con estos modos.

Además, las solicitudes deben estar firmadas mediante un ID de clave de acceso y una clave de acceso secreta que esté asociada a una entidad principal de IAM. También puede utilizar [AWS](#)

[Security Token Service](#) (AWS STS) para generar credenciales de seguridad temporales para firmar solicitudes.

Para consultar otros temas de seguridad de Elastic Beanstalk, consulte [AWS Elastic BeanstalkSeguridad](#).

Análisis de configuración y vulnerabilidad en Elastic Beanstalk

AWS y nuestros clientes comparten la responsabilidad de conseguir un alto nivel de seguridad y conformidad en los componentes del software. AWS Elastic Beanstalk lo ayuda a cumplir su parte del modelo de responsabilidad compartida al ofrecerle la característica de actualizaciones administradas. Esta función aplica automáticamente actualizaciones secundarias y de parche para una versión de plataforma compatible de Elastic Beanstalk.

Para obtener más información, consulte [Modelo de responsabilidad compartida para el mantenimiento de plataformas Elastic Beanstalk](#).

Para obtener otros temas de seguridad de Elastic Beanstalk, consulte [AWS Elastic BeanstalkSeguridad](#).

Prácticas recomendadas de seguridad para Elastic Beanstalk

AWS Elastic Beanstalk proporciona varias características de seguridad para tener en cuenta a medida que desarrolla e implementa las propias políticas de seguridad. Las siguientes prácticas recomendadas son directrices generales y no suponen una solución de seguridad completa. Puesto que es posible que estas prácticas recomendadas no sean adecuadas o suficientes para el entorno, considérelas como consideraciones útiles, no como normas.

Para obtener otros temas de seguridad de Elastic Beanstalk, consulte [AWS Elastic BeanstalkSeguridad](#).

Prácticas recomendadas de seguridad preventiva

Los controles de seguridad preventivos intentan evitar incidentes antes de que ocurran.

Implementación del acceso a los privilegios mínimos

Elastic Beanstalk proporciona políticas administradas AWS Identity and Access Management (IAM) para [perfiles de instancia](#), [funciones del servicio](#) y [usuarios de IAM](#). Estas políticas administradas

especifican todos los permisos que pueden ser necesarios para el correcto funcionamiento de su entorno y aplicación.

Es posible que su aplicación no requiera todos los permisos de nuestras políticas administradas. Puede personalizarlos y conceder solo los permisos necesarios para las instancias de su entorno, el servicio de Elastic Beanstalk y sus usuarios para realizar sus tareas. Esto es especialmente importante para las políticas de usuario, donde diferentes roles de usuario pueden tener necesidades de permiso distintas. La implementación del acceso con privilegios mínimos es esencial a la hora de reducir los riesgos de seguridad y el impacto que podrían causar los errores o los intentos malintencionados.

Actualizar sus plataformas con frecuencia

Elastic Beanstalk publica con regularidad nuevas versiones de la plataforma para actualizar todas sus plataformas. Las nuevas versiones de la plataforma proporcionan actualizaciones del sistema operativo, el tiempo de ejecución, el servidor de aplicaciones y el servidor web, y actualizaciones a componentes de Elastic Beanstalk. Muchas de estas actualizaciones de la plataforma incluyen importantes correcciones de seguridad. Asegúrese de que sus entornos de Elastic Beanstalk se ejecutan en una versión de la plataforma compatible (normalmente la última versión de su plataforma). Para obtener más información, consulte [Actualización de la versión de la plataforma del entorno de Elastic Beanstalk](#).

La forma más sencilla de mantener la plataforma de su entorno actualizada consiste en configurarlo para usar [actualizaciones de la plataforma actualizadas](#).

Aplicar IMDSv2 en instancias de entorno

Las instancias de Amazon Elastic Compute Cloud (Amazon EC2) en sus entornos Elastic Beanstalk utilizan el servicio de metadatos de instancia (IMDS), un componente en instancia, para acceder de forma segura a los metadatos de instancia. IMDS admite dos métodos para acceder a los datos: IMDSv1 e IMDSv2. IMDSv2 utiliza solicitudes orientadas a la sesión y mitiga varios tipos de vulnerabilidades que podrían utilizarse para intentar acceder al IMDS. Para obtener más información sobre las ventajas de IMDSv2, consulte [las mejoras para agregar defensa en profundidad al servicio de metadatos de instancias EC2](#).

IMDSv2 es más seguro, por lo que es una buena idea imponer el uso de IMDSv2 en sus instancias. Para aplicar IMDSv2, asegúrese de que todos los componentes de la aplicación sean compatibles con IMDSv2 y, a continuación, desactive IMDSv1. Para obtener más información, consulte [the section called “IMDS”](#).

Prácticas recomendadas de detección de seguridad

Los controles de detección de seguridad identifican las infracciones de seguridad tras producirse. Pueden ayudarte a detectar una posible amenaza o incidente de seguridad.

Implementar la monitorización

El monitoreo es una parte importante del mantenimiento de la confiabilidad, la seguridad, la disponibilidad y el rendimiento de las soluciones de Elastic Beanstalk. AWS proporciona varias herramientas y servicios a fin de ayudarlo a monitorear los servicios de AWS .

A continuación se muestran algunos ejemplos de elementos que monitorizar:

- Métricas de Amazon CloudWatch para Elastic Beanstalk: establezca alarmas para métricas clave de Elastic Beanstalk y para las métricas personalizadas de su aplicación. Para obtener más información, consulte [Uso de Elastic Beanstalk con Amazon CloudWatch](#).
- Entradas de AWS CloudTrail: realice un seguimiento de las acciones que podrían afectar la disponibilidad, como `UpdateEnvironment` o `TerminateEnvironment`. Para obtener más información, consulte [Registro de llamadas a la API de Elastic Beanstalk con AWS CloudTrail](#).

Habilitar AWS Config

AWS Config proporciona una vista detallada de la configuración de los recursos de AWS de su cuenta. Puede observar las relaciones entre los recursos, obtener un historial de los cambios de configuración y comprobar cómo cambian las relaciones y configuraciones con el paso del tiempo.

Puede utilizar AWS Config para definir reglas que evalúan las configuraciones de recursos para la conformidad de los datos. Las reglas de AWS Config representan la configuración ideal para los recursos de Elastic Beanstalk. Si un recurso infringe una regla y está marcado como no conforme, AWS Config puede avisarle mediante un tema de Amazon Simple Notification Service (Amazon SNS). Para obtener más información, consulte [Búsqueda y seguimiento de recursos de Elastic Beanstalk con AWS Config](#).

Solución de problemas

Este capítulo brinda orientación para solucionar problemas con el entorno de Elastic Beanstalk. Proporciona la siguiente información.

- Una introducción a la herramienta AWS Systems Manager, además de un procedimiento para ejecutar un manual de procedimientos predefinido de Elastic Beanstalk que muestra pasos y recomendaciones para la solución de problemas.
- Guía general sobre las acciones que puede tomar y los recursos que puede consultar si el estado del entorno empeora.
- Consejos de solución de problemas más específicos por categoría temática.

Si el estado del entorno cambia a rojo, le recomendamos que utilice primero la herramienta AWS Systems Manager que incluye manuales de procedimientos predefinidos para solucionar problemas de Elastic Beanstalk. Para obtener más información, consulte [Uso de la herramienta Systems Manager](#) en la siguiente sección de este capítulo.

Temas

- [Uso de manual de procedimientos de AWS Systems Manager Elastic Beanstalk](#)
- [General guidance](#)
- [Categorías](#)

Uso de manual de procedimientos de AWS Systems Manager Elastic Beanstalk

Puede usar Systems Manager para solucionar problemas en sus entornos de Elastic Beanstalk. Para ayudarlo a empezar rápidamente, Systems Manager brinda manuales de procedimientos de automatización predefinidos para Elastic Beanstalk. Un manual de procedimientos de automatización es un tipo de documento de Systems Manager que define las acciones que se deben realizar en las instancias del entorno y otros recursos de AWS.


El documento `AWSSupport-TroubleshootElasticBeanstalk` es un manual de procedimientos de automatización diseñado para ayudar a identificar una serie de problemas comunes que pueden deteriorar el entorno de Elastic Beanstalk. Para ello, comprueba los componentes de su

entorno, incluidos los siguientes: las instancias de EC2, la VPC, la pila de AWS CloudFormation, los equilibradores de carga, los grupos de escalado automático y la configuración de red asociada a las reglas de los grupos de seguridad, las tablas de enrutamiento y las ACL.

También ofrece la opción de cargar archivos de registro agrupados desde su entorno a AWS Support.

Para obtener más información, consulte [AWSSupport-TroubleshootElasticBeanstalk](#) en la Referencia del manual de procedimientos de automatización de AWS Systems Manager.

Utilice Systems Manager para ejecutar el manual de procedimientos **AWSSupport-TroubleshootElasticBeanstalk**

 Note

Ejecute este procedimiento en la misma Región de AWS donde se encuentra su entorno de Elastic Beanstalk.

1. Abra la consola de [AWS Systems Manager](#).
2. En el panel de navegación, en Administración de cambios, elija Automatización.
3. Elija Execute automation (Ejecutar automatización).
4. En la pestaña Propiedad de Amazon, en el cuadro de búsqueda de Documentos de automatización, introduzca `AWSSupport-TroubleshootElasticBeanstalk`.
5. Seleccione la tarjeta `AWSSupport-TroubleshootElasticBeanstalk` y, a continuación, elija Siguiente.
6. Seleccione Ejecutar.
7. En la sección Parámetros de entrada:
 - a. En el menú desplegable `AutomationAssumeRole`, seleccione el ARN del rol que permite a Systems Manager realizar acciones en su nombre.
 - b. En `ApplicationName`, introduzca el nombre de la aplicación Elastic Beanstalk.
 - c. En Nombre del entorno, introduzca el entorno de Elastic Beanstalk.
 - d. (Opcional) En `S3UploaderLink`, introduzca un enlace si un ingeniero de AWS Support le ha proporcionado un enlace de S3 para la recopilación de registros.
8. Elija Execute (Ejecutar).

Si se produce un error en alguno de los pasos, seleccione el enlace situado en la columna ID del paso en el que se produjo el error. Esto muestra una página de Detalles de ejecución para el paso. La sección `VerificationErrorMessage` mostrará un resumen de los pasos que requieren atención. Por ejemplo, `IAMPermissionCheck` podría mostrar un mensaje de advertencia. En este caso, puede comprobar que el rol seleccionado en el menú desplegable `AutomationAssumeRole` tiene los permisos necesarios.

Una vez completados correctamente todos los pasos, el resultado muestra los pasos de solución de problemas y las recomendaciones para restaurar el entorno a un estado correcto.

General guidance

Los mensajes de error pueden aparecer en la página Eventos en la consola, en los registros o en la página Estado. También puede tomar medidas para recuperarse de un entorno degradado provocado por un cambio reciente. Si el estado de su entorno cambia a Rojo, pruebe lo siguiente:

- Revise los [eventos](#) recientes del entorno. Los mensajes de Elastic Beanstalk sobre los problemas de implementación, carga y configuración suelen aparecer aquí.
- Revise el [historial de cambios](#) reciente del entorno. El historial de cambios enumera todos los cambios de configuración realizados en los entornos e incluye otra información, como qué usuario de IAM realizó cambios y qué parámetros de configuración se establecieron.
- [Obtenga los logs](#) para las ver las entradas recientes de los archivos log. Los logs del servidor web contienen información sobre solicitudes entrantes y errores.
- [Conéctese a una instancia](#) y compruebe los recursos del sistema.
- [Restaurar](#) una versión operativa anterior de la aplicación.
- Deshaga los últimos cambios de configuración o restaure una [configuración guardada](#).
- Implemente un nuevo entorno. Si el entorno parece estar en buen estado, realice un [intercambio de CNAME](#) para dirigir el tráfico al nuevo entorno y siga depurando el anterior.

Categorías

En este tema, se proporcionan consejos de solución de problemas más específicos por categoría.

Temas

- [Conectividad](#)
- [Creación del entorno y lanzamiento de instancias](#)
- [Implementaciones](#)
- [Estado](#)
- [Configuración](#)
- [Solución de problemas con contenedores de Docker](#)
- [Preguntas frecuentes](#)

Conectividad

Problema: Los servidores creados en la consola de Elastic Beanstalk no aparecen en Toolkit for Eclipse

Puede importar manualmente los servidores siguiendo las instrucciones de [Importación de entornos existentes en Eclipse](#).

Problema: No se puede conectar a Amazon RDS desde Elastic Beanstalk.

Para conectar Amazon RDS desacoplado a su aplicación Elastic Beanstalk, haga lo siguiente:

- Asegúrese de que RDS se encuentra en la misma región que su aplicación de Elastic Beanstalk.
- Asegúrese de que el grupo de seguridad de RDS de la instancia tiene autorización para el grupo de seguridad de Amazon EC2 que está utilizando en su entorno de Elastic Beanstalk. Para obtener instrucciones sobre cómo encontrar el nombre del grupo de seguridad de EC2 a través de la consola de administración de AWS, consulte [Grupos de seguridad](#). Para obtener más información sobre la configuración del grupo de seguridad de EC2, consulte la sección "Autorizar acceso a la red a un grupo de seguridad de Amazon EC2" en [Trabajando con grupos de seguridad de base de datos](#) en la Guía del usuario de Amazon Relational Database Service.
- Para Java, asegúrese de que el archivo JAR de MySQL está en su WEB-INF/lib. Consulte [Añadir una instancia de base de datos de Amazon RDS a un entorno de aplicaciones Java](#) para obtener más detalles.

Creación del entorno y lanzamiento de instancias

Evento: error al lanzar el entorno

Este evento se produce cuando Elastic Beanstalk intenta lanzar un entorno y encuentra errores en el proceso. Los eventos anteriores de la página Events (Eventos) le indicarán la causa del problema.

Evento: operación de crear entorno completa, pero con tiempos de espera de comandos. Intente aumentar el periodo de tiempo de espera.

Su aplicación puede tardar mucho tiempo en implementarse si utiliza archivos de configuración que ejecutan comandos en la instancia, descargan archivos grandes o instalan paquetes. Aumente el [tiempo de espera de los comandos](#) para dar más tiempo a la aplicación para que empiece a ejecutarse durante las implementaciones.

Evento: error al crear los siguientes recursos: [AWSEBInstanceLaunchWaitCondition]

Este mensaje indica que las instancias de Amazon EC2; del entorno no comunicaron a Elastic Beanstalk que se implementaron correctamente. Esto puede ocurrir si las instancias no disponen de conectividad a Internet. Si configuró el entorno para lanzar las instancias en una subred de VPC privada, [asegúrese de que la subred tiene un NAT](#) que permita que las instancias se conecten a Elastic Beanstalk.

Evento: se requiere un rol de servicio en esta región. Agregue una opción de Rol de servicio al entorno.

Elastic Beanstalk utiliza un rol de servicio para monitorear los recursos del entorno y permitir las [actualizaciones administradas de la plataforma](#). Para obtener más información, consulte [Administración de roles de servicio de Elastic Beanstalk](#).

Implementaciones

Problema: la aplicación deja de estar disponible durante las implementaciones

Como Elastic Beanstalk utiliza un proceso de actualización "drop-in", podría haber algunos segundos de inactividad. Utilice las [implementaciones continuas](#) para minimizar el efecto de las implementaciones en los entornos de producción.

Evento: error al crear una versión de la aplicación AWS Elastic Beanstalk.

Puede que el paquete de código fuente de la aplicación sea demasiado grande o que haya alcanzado la [cuota de versiones de la aplicación](#).

Evento: operación de actualización de entorno completa, pero con tiempo de espera de los comandos. Intente aumentar el periodo de tiempo de espera.

Su aplicación puede tardar mucho tiempo en implementarse si utiliza archivos de configuración que ejecutan comandos en la instancia, descargan archivos grandes o instalan paquetes. Aumente el [tiempo de espera de los comandos](#) para dar más tiempo a la aplicación para que empiece a ejecutarse durante las implementaciones.

Estado

Evento: el uso de la CPU supera el 95,00 %

Intente [ejecutar más instancias](#) o [elija un tipo de instancia distinto](#).

Evento: Elastic Load Balancer *awseb-myapp* no tiene ninguna instancia con un estado correcto

Si la aplicación parece funcionar, asegúrese de que la URL de comprobación de estado de la aplicación está configurada correctamente. De lo contrario, consulte la pantalla de estado y los registros del entorno para obtener más información.

Evento: Elastic Load Balancer *awseb-myapp* no se encuentra

Es posible que el balanceador de carga del entorno se haya quedado sin conexión. Realice únicamente cambios en los recursos del entorno con las opciones de configuración y [extensibilidad](#) proporcionadas por Elastic Beanstalk. Vuelva a crear el entorno o lance uno nuevo.

Evento: error al lanzar una instancia de EC2. A la espera de que se lance una nueva instancia de EC2...

Es posible que el tipo de instancia del entorno tenga poca disponibilidad o que se haya alcanzado la cuota de instancias de la cuenta. Consulte el [Panel de estado del servicio](#) para asegurarse de que el servicio Elastic Compute Cloud (Amazon EC2) aparece de color verde o para [solicitar un aumento de la cuota](#).

Configuración

Evento: no puede configurar un entorno de Elastic Beanstalk con valores para la opción Elastic Load Balancing Target y la opción Application Healthcheck URL

La opción Target del espacio de nombres `aws:elb:healthcheck` ya no está disponible. Elimine la opción Target de su entorno e intente actualizar de nuevo.

Evento: ELB no se puede asociar a varias subredes de la misma zona de disponibilidad.

Este mensaje puede aparecer si intenta mover un balanceador de carga entre subredes de la misma zona de disponibilidad. Para cambiar las subredes del balanceador, debe sacar el balanceador de carga de la zona o zonas de disponibilidad originales y volver a incluirlo con las subredes deseadas. Durante el proceso, todas las instancias se migran de una zona de disponibilidad a otra, lo que causa tiempos de inactividad importantes. En lugar de ello, considere la posibilidad de crear un nuevo entorno y [realizar un intercambio de CNAME](#).

Solución de problemas con contenedores de Docker

Evento: error al extraer la imagen de Docker :latest: nombre de repositorio no válido (), solo se admite [a-z0-9-_.]. Siga los registros para obtener más detalles.

Marque la sintaxis del archivo `dockerrun.aws.json` con un validador JSON. Compruebe también si el contenido de `dockerfile` se ajusta a los requisitos descritos en [Configuración de Docker](#).

Evento: no se encuentra ninguna directiva EXPOSE en Dockerfile, anular implementación

El archivo `Dockerfile` o `dockerrun.aws.json` no declara el puerto del contenedor. Utilice la instrucción EXPOSE (`Dockerfile`) o el bloque Ports (`archivo dockerrun.aws.json`) para exponer un puerto para el tráfico entrante.

Evento: error al descargar el *repositorio* de credenciales de autenticación desde *nombre de bucket*

`dockerrun.aws.json` proporciona un par de claves de EC2 y/o un bucket de S3 que no son válidos para el archivo `.dockercfg`. También es posible que el perfil de instancia no tenga autorización de GetObject para el bucket de S3. Compruebe que el archivo `.dockercfg` contiene un bucket de S3 y un par de claves EC2 que son válidos. Conceda permisos para la acción `s3:GetObject` al rol de IAM en el perfil de instancia. Para obtener más información, consulte [Administración de perfiles de instancia de Elastic Beanstalk](#).

Evento: error en ejecución de actividad, porque: ADVERTENCIA: archivo de configuración de autenticación no válido

El archivo de autenticación (`config.json`) no tiene un formato correcto. Consulte [Uso de imágenes de un repositorio privado](#).

Preguntas frecuentes

Pregunta: ¿Cómo puedo cambiar la URL de mi aplicación de `myapp.us-west-2.elasticbeanstalk.com` a `www.myapp.com`?

En un servidor DNS, registre un registro CNAME como **www.mydomain.com** CNAME **mydomain.elasticbeanstalk.com**.

Pregunta: ¿Cómo puedo especificar una zona de disponibilidad específica para mi aplicación de Elastic Beanstalk?

Puede elegir una zona de disponibilidad específica mediante las API, la CLI, el complemento de Eclipse o el complemento de Visual Studio. Para obtener instrucciones sobre cómo usar la consola de Elastic Beanstalk para especificar una zona de disponibilidad, consulte [Grupo de Auto Scaling para su entorno de Elastic Beanstalk](#).

Pregunta: ¿Cómo puedo cambiar el tipo de instancia del entorno?

Para cambiar el tipo de instancia del entorno, vaya a la página de configuración del entorno y elija Edit (Editar) en la categoría de configuración Instances (Instancias). Luego, seleccione un nuevo tipo de instancia y haga clic en Apply (Aplicar) para actualizar el entorno. Después de esto, Elastic Beanstalk termina todas las instancias en ejecución y las reemplaza por instancias nuevas.

Pregunta: ¿Cómo puedo determinar si alguien ha realizado cambios de configuración en un entorno?

Para ver esta información, en el panel de navegación de la consola de Elastic Beanstalk seleccione Change history (Historial de cambios) para visualizar una lista de cambios de configuración de todos los entornos. Esta lista incluye la fecha y la hora del cambio, el parámetro de configuración y el valor al que se cambió, y el usuario de IAM que realizó el cambio. Para obtener más información, consulte [Historial de cambios](#).

Pregunta: ¿Puedo evitar que los volúmenes de Amazon EBS se eliminen al terminar las instancias?

Las instancias de su entorno utilizan Amazon EBS para el almacenamiento; sin embargo, el volumen raíz se elimina cuando Auto Scaling termina una instancia. No se recomienda almacenar el estado u otros datos de las instancias. Si es necesario, puede evitar que se eliminen los volúmenes con la AWS CLI: `$ aws ec2 modify-instance-attribute -b '/dev/sdc=<vol-id>:false` tal y como se describe en la [Referencia de la AWS CLI](#).

Pregunta: ¿Cómo elimino la información personal de mi aplicación de Elastic Beanstalk?

Los recursos de AWS que usa la aplicación Elastic Beanstalk podrían almacenar información personal. Cuando termina un entorno, Elastic Beanstalk termina los recursos que creó. También se terminan los recursos que agregó a través de los [archivos de configuración](#). Sin embargo, si creó los recursos de AWS fuera del entorno de Elastic Beanstalk y los asoció a la aplicación, puede que deba

asegurarse manualmente de que no se retenga información personal que pueda haber almacenado la aplicación. En esta guía para desarrolladores, siempre que se explica la creación de recursos adicionales, también mencionamos cuándo debe considerar eliminarlos.

Recursos de Elastic Beanstalk

Los recursos relacionados siguientes pueden serle de ayuda cuando trabaje con este servicio.

- [Elastic Beanstalk API Reference \(Referencia de la API de Elastic Beanstalk\)](#): exhaustiva descripción de todas las API de SOAP y de consulta. Además, contiene una lista de todos los tipos de datos de SOAP.
- [elastic-beanstalk-samples on GitHub](#): un GitHub repositorio con archivos de configuración de muestra de Elastic Beanstalk (.ebextensions). El README .md archivo del repositorio contiene enlaces a repositorios adicionales GitHub con aplicaciones de muestra.
- [Elastic Beanstalk Technical FAQ \(Preguntas técnicas frecuentes sobre Elastic Beanstalk\)](#): principales preguntas realizadas por los desarrolladores acerca de este producto.
- [AWS Elastic Beanstalk Notas de la versión](#): detalles sobre las nuevas funciones, actualizaciones y correcciones de las versiones del servicio, la plataforma, la consola y la CLI de EB de Elastic Beanstalk.
- [Clases y talleres](#): enlaces a cursos especializados y basados en funciones, además de laboratorios personalizados que le ayudarán a perfeccionar sus habilidades y adquirir experiencia práctica. AWS
- [AWS Centro para desarrolladores](#): explore los tutoriales, descargue herramientas y obtenga información sobre los eventos para desarrolladores. AWS
- [AWS Herramientas para desarrolladores](#): enlaces a herramientas para desarrolladores, SDK, kits de herramientas IDE y herramientas de línea de comandos para desarrollar y administrar AWS aplicaciones.
- [Centro de recursos de introducción](#): aprenda a configurar su primera aplicación Cuenta de AWS, a unirse a la AWS comunidad y a lanzar su primera aplicación.
- [Tutoriales prácticos](#): sigue step-by-step los tutoriales para lanzar tu primera aplicación. AWS
- [AWS Documentos técnicos](#): enlaces a una lista completa de AWS documentos técnicos, que abarcan temas como la arquitectura, la seguridad y la economía, redactados por arquitectos de AWS soluciones u otros expertos técnicos.
- [AWS Support Center](#): el centro para crear y gestionar sus casos. AWS Support También incluye enlaces a otros recursos útiles, como foros, preguntas frecuentes técnicas, estado del servicio y AWS Trusted Advisor.

- [AWS Support](#)— La página web principal con información sobre AWS Support un one-on-one canal de soporte de respuesta rápida que le ayudará a crear y ejecutar aplicaciones en la nube.
- [Contacte con nosotros](#) – Un punto central de contacto para las consultas relacionadas con la facturación AWS , cuentas, eventos, abuso y demás problemas.
- [AWS Condiciones del sitio](#): información detallada sobre nuestros derechos de autor y marca comercial; su cuenta, licencia y acceso al sitio; y otros temas.

Aplicaciones de muestra

A continuación, se incluyen enlaces para descargar aplicaciones de ejemplo que se implementan como parte de [Introducción a Elastic Beanstalk](#).

Note

Es posible que, en algunos ejemplos, se utilicen características que podrían haberse publicado después del lanzamiento de la plataforma que está utilizando. Si el ejemplo no puede ejecutarse correctamente, intente actualizar la plataforma a una versión reciente, tal y como se describe en [the section called “Plataformas admitidas”](#).

- Docker: [docker.zip](#)
- Docker multicontenedor: [2.zip docker-multicontainer-v](#)
- Docker preconfigurado (Glassfish): [1.zip docker-glassfish-v](#)
- Go: [go.zip](#)
- Corretto: [corretto.zip](#)
- Tomcat: [tomcat.zip](#)
- .NET Core en Linux: [.zip dotnet-core-linux](#)
- .NET Core: [dotnet-asp-windows .zip](#)
- Node.js: [nodejs.zip](#)
- PHP: [php.zip](#)
- Python: [python.zip](#)
- Ruby: [ruby.zip](#)

Historial de la plataforma

El historial de la plataforma de AWS Elastic Beanstalk se ha movido. Consulte [Historial de la plataforma](#) en el documento AWS Elastic BeanstalkPlatforms.

Temas

- [Plataformas personalizadas de Elastic Beanstalk](#)

Plataformas personalizadas de Elastic Beanstalk

Note

El [18 de julio de 2022](#), Elastic Beanstalk estableció el estado de todas las ramificaciones de la plataforma basadas en la AMI de Amazon Linux (AL1) como retirado. Esto incluye plataformas personalizadas. Elastic Beanstalk no admite plataformas personalizadas. Para obtener más información acerca del retiro de la AMI de Amazon Linux por parte de Elastic Beanstalk, consulte [Preguntas frecuentes sobre la retirada de plataformas](#).

En este documento se incluye este tema como referencia para cualquier cliente que utilizó la función de plataforma personalizada de Elastic Beanstalk antes de su retiro. Anteriormente, las plataformas personalizadas de Elastic Beanstalk solo admitían la creación de una AMI desde las AMI de base Amazon Linux, RHEL 7, RHEL 6 o Ubuntu 16.04. Elastic Beanstalk ya no admite estos sistemas operativos. Para obtener más información sobre la función de plataformas personalizadas, que ya no se admiten, amplíe el siguiente tema.

Plataformas personalizadas

Una plataforma personalizada es una personalización más avanzada que una [imagen personalizada](#) en varios sentidos. Una plataforma personalizada le permite desarrollar una nueva plataforma completa desde cero, personalizando el sistema operativo, el software adicional y los scripts que Elastic Beanstalk ejecuta en instancias de plataforma. Esta flexibilidad le permite crear una plataforma para una aplicación que utilice un lenguaje u otro software de infraestructura, para el que Elastic Beanstalk no proporciona una plataforma administrada. Si comparamos este tipo de plataforma con las imágenes personalizadas, en las que usted modifica una Amazon Machine Image

(AMI) para su uso con una plataforma Elastic Beanstalk existente, vemos que Elastic Beanstalk proporciona igualmente los scripts de la plataforma y controla la pila de software de la plataforma. Además, con las plataformas personalizadas puede usar un script automatizado para crear y mantener su personalización, mientras que en las imágenes personalizadas realiza los cambios manualmente a través de una instancia en ejecución.

Para crear una plataforma personalizada, crea una AMI de uno de los sistemas operativos admitidos: Ubuntu, RHEL o Amazon Linux (consulte la entrada `flavor` de [Formato del archivo platform.yaml](#) para conocer los números de versión exactos) y añada personalizaciones. Puede crear su propia plataforma Elastic Beanstalk con [Packer](#), que es una herramienta de código abierto para crear imágenes de máquinas para muchas plataformas, incluidas AMI para su uso con Amazon Elastic Compute Cloud (Amazon EC2). Las plataformas Elastic Beanstalk se componen de una AMI que está configurada para ejecutar el conjunto de software que respalda a una aplicación y metadatos que pueden contener opciones de configuración personalizadas y valores predeterminados.

Elastic Beanstalk administra Packer como una plataforma integrada independiente y, por tanto, no tendrá que preocuparse por la configuración y las versiones de Packer.

Para crear una plataforma, proporcione a Elastic Beanstalk una plantilla de Packer y los scripts y archivos que la plantilla invoca para crear las AMI. Estos componentes se empaquetan con un [archivo de definición de plataforma](#), que especifica la plantilla y los metadatos, en un archivo ZIP, conocido como [archivo de definición de plataforma](#).

Cuando crea una plataforma personalizada, lanza un entorno de una sola instancia sin una dirección IP elástica que ejecuta Packer. A continuación, Packer lanza otra instancia para crear una imagen. Puede reutilizar este entorno con diferentes plataformas y diferentes versiones de cada plataforma.

Note

Las plataformas personalizadas son específicas de cada AWS región. Si utiliza Elastic Beanstalk en varias regiones, debe crear las plataformas por separado en cada región. En determinadas circunstancias, las instancias lanzadas por Packer no se han limpiado y se deben terminar manualmente. Para obtener información sobre cómo limpiar manualmente estas instancias, consulte [Limpieza de instancias de Packer](#).

Los usuarios de la cuenta pueden usar sus plataformas personalizadas. Para ello, deben especificar un [ARN de plataforma](#) durante la creación del entorno. Estos ARN se devuelven mediante el comando `eb platform create` que utilizó para crear la plataforma personalizada.

Cada vez que crea su plataforma personalizada, Elastic Beanstalk crea una nueva versión de plataforma. Los usuarios pueden especificar una plataforma por su nombre para obtener únicamente la última versión de la plataforma o incluir un número de versión para obtener una versión específica.

Por ejemplo, para implementar la última versión de la plataforma personalizada con el ARN **MyCustomPlatformARN**, que podría ser la 3.0, la línea de comando de la CLI de EB sería esta:

```
eb create -p MyCustomPlatformARN
```

Para implementar la versión 2.1, la línea de comando de la CLI de EB sería esta:

```
eb create -p MyCustomPlatformARN --version 2.1
```

Puede aplicar etiquetas a una versión de la plataforma personalizada cuando la cree y editar etiquetas de las versiones de la plataforma personalizada existentes. Para obtener más detalles, consulte [Etiquetado de versiones de la plataforma personalizada](#).

Creación de una plataforma personalizada

Para crear una plataforma personalizada, la raíz de su aplicación debe contener un archivo de definiciones de plataforma, `platform.yaml`, que define el tipo de constructor utilizado para crear la plataforma personalizada. El formato de este archivo se describe en [Formato del archivo `platform.yaml`](#). Puede crear la plataforma personalizada desde cero o usar una de las [plataformas personalizadas de ejemplo](#) como punto de partida.

Uso de las plataformas personalizadas de ejemplo

Si no quiere crear su propia plataforma personalizada, puede utilizar uno de los ejemplos de archivo de definiciones de plataforma para arrancar la plataforma personalizada. Los únicos elementos que tiene que configurar en las muestras para poder usarlas son una AMI de origen y una región.

Note

No utilice una plataforma personalizada de ejemplo sin modificar en producción. El objetivo de los ejemplos es mostrar algunas de las funcionalidades disponibles en una plataforma personalizada, pero no están preparados para producción.

[NodePlatform_Ubuntu.zip](#)

Esta plataforma se basa en Ubuntu 16.04 y es compatible con Node.js 4.4.4. Utilizaremos esta plataforma personalizada para los ejemplos de esta sección.

[NodePlatform_RHEL.zip](#)

Esta plataforma se basa en RHEL 7.2 y es compatible con Node.js 4.4.4.

[NodePlatform_AmazonLinux.zip](#)

Esta plataforma se basa en Amazon Linux 2016.09.1 y es compatible con Node.js 4.4.4.

[TomcatPlatform_Ubuntu.zip](#)

Esta plataforma se basa en Ubuntu 16.04 y es compatible con Tomcat 7/Java 8.

[CustomPlatform_NodeSampleApp.zip](#)

Ejemplo de Node.js que utiliza express y ejs para mostrar una página web estática.

[CustomPlatform_.zip TomcatSampleApp](#)

Ejemplo de Tomcat que muestra una página web estática cuando se implementa.

Descargue el archivo de definición de plataforma de ejemplo: `NodePlatform_Ubuntu.zip`. Este archivo contiene un archivo de definición de la plataforma, una plantilla de Packer, scripts que Packer ejecuta durante la creación de imágenes y scripts y archivos de configuración que Packer copia en la instancia del constructor durante la creación de la plataforma.

Example NodePlatform_Ubuntu.zip

```
|-- builder           Contains files used by Packer to create the custom platform
|-- custom_platform.json  Packer template
|-- platform.yaml      Platform definition file
|-- ReadMe.txt         Briefly describes the sample
```

El archivo de definición de plataforma, `platform.yaml`, le dice a Elastic Beanstalk el nombre de la plantilla Packer, `custom_platform.json`.

```
version: "1.0"

provisioner:
  type: packer
  template: custom_platform.json
```

```
flavor: ubuntu1604
```

La plantilla de Packer le indica a Packer cómo debe generar las AMI de la plataforma mediante una [AMI de Ubuntu](#) que sirve de base para la imagen de plataforma en los tipos de instancia HVM. La sección `provisioners` le indica a Packer que copie todo el contenido de la carpeta `builder` del archivo en la instancia y que ejecute el script `builder.sh` en la instancia. Cuando los scripts se completan, Packer crea una imagen de la instancia modificada.

Elastic Beanstalk crea tres variables de entorno que se pueden utilizar para etiquetar las AMI de Packer:

AWS_EB_PLATFORM_ARN

ARN de la plataforma personalizada.

AWS_EB_PLATFORM_NAME

Nombre de la plataforma personalizada.

AWS_EB_PLATFORM_VERSION

Versión de la plataforma personalizada.

El archivo `custom_platform.json` de ejemplo utiliza estas variables para definir los que se utilizan en los scripts:

- `platform_name`, que se establece mediante `platform.yaml`
- `platform_version`, que se establece mediante `platform.yaml`
- `platform_arn`, que se establece mediante el script de compilación principal, `builder.sh`, que aparece al final del archivo de ejemplo `custom_platform.json`.

El archivo `custom_platform.json` contiene dos propiedades para las que tiene que proporcionar valores: `source_ami` y `region`. Para obtener más información sobre cómo elegir los valores de AMI y región correctos, consulte [Actualización de la plantilla de Packer](#) en el `eb-custom-platforms-samples` GitHub repositorio.

Example `custom_platform.json`

```
{
  "variables": {
    "platform_name": "{{env `AWS_EB_PLATFORM_NAME`}}",
```

```

    "platform_version": "{{env `AWS_EB_PLATFORM_VERSION`}}",
    "platform_arn": "{{env `AWS_EB_PLATFORM_ARN`}}"
  },
  "builders": [
    {
      ...
      "region": "",
      "source_ami": "",
      ...
    }
  ],
  "provisioners": [
    {...},
    {
      "type": "shell",
      "execute_command": "chmod +x {{ .Path }}; {{ .Vars }} sudo {{ .Path }}",
      "scripts": [
        "builder/builder.sh"
      ]
    }
  ]
}

```

Los scripts y otros archivos que incluya en su archivo de deficiones de plataforma variarán considerablemente en función de las modificaciones que quiera realizar en la instancia. La plataforma de ejemplo contiene los siguientes scripts:

- `00-sync-apt.sh`: comentado: `apt -y update`. Hemos convertido en comentario el comando porque solicita una entrada al usuario, lo que interrumpe la actualización automatizada del paquete. Esto puede ser un problema de Ubuntu. Sin embargo, la ejecución de `apt -y update` sigue siendo una práctica recomendada. Por este motivo, hemos dejado el comando en el script de muestra como referencia.
- `01-install-nginx.sh`: instala `nginx`.
- `02-setup-platform.sh`: instala `wget`, `tree` y `git`. Copia enlaces y [configuraciones de registro](#) en la instancia y crea los siguientes directorios:
 - `/etc/SampleNodePlatform`: ubicación en la que se carga el archivo de configuración del contenedor durante la implementación.
 - `/opt/elasticbeanstalk/deploy/appsource/`: ubicación en la que el script `00-unzip.sh` carga el código fuente de la aplicación durante la implementación (consulte la sección [Herramientas de script de plataforma](#) para obtener más información sobre este script).

- `/var/app/staging/`: ubicación en la que el código fuente de la aplicación se procesa durante la implementación.
- `/var/app/current/`: ubicación en la que el código fuente de la aplicación se ejecuta después de procesarse.
- `/var/log/nginx/healthd/`: ubicación en la que el [agente de estado avanzado](#) escribe los registros.
- `/var/nodejs`: ubicación en la que se cargan los archivos Node.js durante la implementación.

Utilice la CLI de EB para crear su primera plataforma personalizada con el archivo de definición de plataforma de ejemplo.

Para crear una plataforma personalizada

1. [Instale la CLI de EB](#).
2. Cree el directorio en el que va a extraer la plataforma personalizada de ejemplo.

```
~$ mkdir ~/custom-platform
```

3. Extraiga `NodePlatform_Ubuntu.zip` al directorio y, a continuación, cambie al directorio de extracción.

```
~$ cd ~/custom-platform
~/custom-platform$ unzip ~/NodePlatform_Ubuntu.zip
~/custom-platform$ cd NodePlatform_Ubuntu
```

4. Edite el archivo `custom_platform.json` y proporcione valores para las propiedades `region` y `source_ami`. Para obtener más información, consulte [Actualización de plantilla de Packer](#).
5. Ejecute [eb platform init](#) y siga los mensajes para inicializar el repositorio de la plataforma

Puede utilizar `eb platform` como forma abreviada de `ebp`.

Note

Windows `ebp` lo PowerShell utiliza como alias de comando. Si ejecuta la CLI de EB en Windows PowerShell, utilice la forma larga de este comando: `eb platform`.

```
~/custom-platform$ eb platform init
```

Este comando también crea el directorio `.elasticbeanstalk` en la carpeta actual y agrega el archivo de configuración `config.yml` al directorio. No cambie ni elimine este archivo, ya que Elastic Beanstalk lo utiliza como base cuando crea la plataforma personalizada.

De forma predeterminada, `eb platform init` utiliza el nombre de la carpeta actual como el nombre de la plataforma personalizada, que, en este ejemplo, sería `custom-platform`.

6. Ejecute [eb platform create](#) para lanzar un entorno de Packer y obtener el ARN de la plataforma personalizada. Necesitará este valor más tarde, cuando cree un entorno desde la plataforma personalizada.

```
~/custom-platform$ eb platform create
...
```

De forma predeterminada, Elastic Beanstalk crea el perfil de instancia `aws-elasticbeanstalk-custom-platform-ec2-role` para plataformas personalizadas. Si, en su lugar, desea utilizar un perfil de instancia existente, agregue la opción `-ip INSTANCE_PROFILE` en el comando [eb platform create](#).

Note

Packer no podrá crear la plataforma personalizada correctamente si utiliza el perfil de instancia predeterminado Elastic Beanstalk de `aws-elasticbeanstalk-ec2-role`.

En la CLI de EB se muestra la salida de eventos del entorno de Packer hasta que se completa el proceso de creación. Si desea salir de la vista de eventos, presione `Ctrl+C`.

7. Puede comprobar los errores de los registros con el comando [eb platform logs](#).

```
~/custom-platform$ eb platform logs
...
```

8. Puede comprobar el proceso en otro momento con [eb platform events](#).

```
~/custom-platform$ eb platform events
```

```
...
```

9. Compruebe el estado de la plataforma con [eb platform status](#).

```
~/custom-platform$ eb platform status  
...
```

Una vez que se complete la operación, tendrá a su disposición una plataforma que puede utilizar para lanzar un entorno de Elastic Beanstalk.

Puede utilizar la plataforma personalizada cuando cree un entorno desde la consola. Consulte [El asistente de creación de nuevo entorno](#).

Para lanzar un entorno en la plataforma personalizada

1. Cree un directorio para la aplicación.

```
~$ mkdir custom-platform-app  
~$ cd ~/custom-platform-app
```

2. Inicialice un repositorio de la aplicación.

```
~/custom-platform-app$ eb init  
...
```

3. Descargue la aplicación de ejemplo [NodeSampleApp.zip](#).

4. Extraiga la aplicación de muestra.

```
~/custom-platform-app$ unzip ~/NodeSampleApp.zip
```

5. Ejecute `eb create -p CUSTOM-PLATFORM-ARN`, donde **CUSTOM-PLATFORM-ARN** es el ARN devuelto por un comando `eb platform create`, para lanzar un entorno que ejecute la plataforma personalizada.

```
~/custom-platform-app$ eb create -p CUSTOM-PLATFORM-ARN  
...
```

Contenido del archivo de definición de la plataforma

Un archivo de definición de plataforma es el equivalente de plataforma de un [paquete de origen de aplicaciones](#). El archivo de definiciones de plataforma es un archivo ZIP que contiene un archivo de definición de plataforma, una plantilla Packer y los scripts y archivos utilizados por la plantilla Packer para crear la plataforma.

Note

Cuando se utiliza la CLI de EB para crear una plataforma personalizada, esta crea un archivo de definiciones de plataforma a partir de los archivos y carpetas del repositorio de la plataforma, por lo que no es necesario crear el archivo manualmente.

El archivo de definición de plataforma es un archivo con formato YAML que debe ser nombrado `platform.yaml` y estar en la raíz del archivo de definiciones de plataforma. Consulte [Creación de una plataforma personalizada](#) para obtener una lista de claves obligatorias y opcionales admitidas en un archivo de definición de plataforma.

No es necesario que asigne el nombre a la plantilla de Packer de una forma determinada, pero el nombre del archivo debe coincidir con la plantilla del proveedor especificada en el archivo de definiciones de la plataforma. Consulte en la [documentación oficial de Packer](#) las instrucciones para crear plantillas de Packer.

Los demás archivos del archivo de definición de plataforma son scripts y archivos que utiliza la plantilla para personalizar una instancia antes de crear una AMI.

Enlaces de plataforma personalizados

Elastic Beanstalk utiliza una estructura de directorios estandarizada para los enlaces de las plataformas personalizadas. Son scripts que se ejecutan durante los eventos del ciclo de vida en respuesta a las operaciones de administración (cuando se lanzan instancias en el entorno o cuando un usuario inicia una implementación o utiliza la característica de reinicio del servidor de aplicaciones).

Coloque los scripts que desea que disparen los enlaces en una de las subcarpetas de la carpeta `/opt/elasticbeanstalk/hooks/`.

⚠ Warning

No se permite utilizar enlaces de plataforma personalizados en las plataformas administradas. Los enlaces de plataforma están diseñados para las plataformas personalizadas. En las plataformas administradas de Elastic Beanstalk pueden funcionar de forma diferente o tener algunos problemas, y el comportamiento puede diferir de una plataforma a otra. En las plataformas AMI de Amazon Linux (antes de Amazon Linux 2), pueden seguir funcionando de manera útil en algunos casos; úselas con precaución. Los ganchos de plataforma personalizados son una característica heredada que existe en las plataformas de la AMI de Amazon Linux. En las plataformas Amazon Linux 2, los ganchos de plataforma personalizados en la `/opt/elasticbeanstalk/hooks/` carpeta se suspenden por completo. Elastic Beanstalk no los lee ni ejecuta. Las plataformas de Amazon Linux 2 admiten un nuevo tipo de ganchos de plataforma, diseñados específicamente para ampliar las plataformas gestionadas de Elastic Beanstalk. Puede agregar scripts y programas personalizados directamente a un directorio de ganchos en el paquete de origen de la aplicación. Elastic Beanstalk los ejecuta durante varias etapas de aprovisionamiento de instancias. Para obtener más información, expanda la sección Enlaces de plataforma de [the section called “Ampliación de plataformas Linux”](#).

Los enlaces están organizados en las siguientes carpetas:

- `appdeploy`: scripts que se ejecutan durante la implementación de una aplicación. Elastic Beanstalk ejecuta la implementación de una aplicación cuando se lanzan nuevas instancias y cuando un cliente inicia la implementación de una nueva versión.
- `configdeploy`: scripts que se ejecutan cuando un cliente realiza una actualización de la configuración que afecta a la configuración del software de la instancia; por ejemplo, cuando se configuran propiedades de entorno o se habilita la rotación de logs en Amazon S3.
- `restartappserver`: scripts que se ejecutan cuando un cliente realiza una operación de reinicio del servidor de aplicaciones.
- `preinit`: scripts que se ejecutan durante el proceso de arranque de las instancias.
- `postinit`: scripts que se ejecutan después del proceso de arranque de las instancias.

Las carpetas `appdeploy`, `configdeploy` y `restartappserver` contienen las subcarpetas `pre`, `enact` y `post`. En cada fase de las operaciones, se ejecutan todos los scripts de la carpeta `pre` en orden alfabético, después los de la carpeta `enact` y, a continuación, los de la carpeta `post`.

Cuando se lanza una instancia, Elastic Beanstalk ejecuta `preinit`, `appdeploy` y `postinit`, en este orden. En las siguientes implementaciones de las instancias en ejecución, Elastic Beanstalk ejecuta los enlaces `appdeploy`. Los enlaces de `configdeploy` se ejecutan cuando un usuario actualiza las opciones de configuración del software de la instancia. Los enlaces de `restartappserver` solo se ejecutan cuando el usuario inicia un reinicio del servidor de aplicaciones.

Cuando los scripts experimentan errores, pueden salir con un estado cero y escribir en `stderr` el error de la operación. El mensaje que escriba en `stderr` aparecerá en el evento que se genera cuando se produce un error en la operación. Elastic Beanstalk también captura esta información en el archivo log `/var/log/eb-activity.log`. Si no desea que la operación muestre errores, devuelva 0 (cero). Los mensajes que se escriben en `stderr` o `stdout` aparecen en los [registros de implementación](#), pero no en la secuencia de eventos, a no ser que se produzca un error en la operación.

Limpieza de instancias de Packer

En determinadas circunstancias, por ejemplo, si se detiene el proceso del constructor de Packer antes de que haya terminado, las instancias lanzadas por Packer no se limpian. Estas instancias no forman parte del entorno Elastic Beanstalk y solo se pueden ver y finalizar mediante el servicio Amazon EC2.

Para limpiar manualmente estas instancias

1. Abra la [consola de Amazon EC2](#).
2. Asegúrese de estar en la misma AWS región en la que creó la instancia con Packer.
3. En Resources (Recursos), elija **N** Running Instances (Instancias de ejecución), donde **N** indica el número de instancias de ejecución.
4. Haga clic en el cuadro de texto de consulta.
5. Seleccione la etiqueta Name (Nombre).
6. Introduzca `packer`.

La consulta debería ser similar a `tag:Name: packer`.

7. Seleccione las instancias que coinciden con la consulta.
8. Si el valor de Instance State (Estado de instancia) es `running` (en ejecución), elija Actions (Acciones), Instance State (Estado de instancia), Stop (Detener), Actions (Acciones), Instance State (Estado de instancia) y Terminate (Terminar).

Formato del archivo platform.yaml.

El archivo `platform.yaml` tiene el formato siguiente.

```
version: "version-number"

provisioner:
  type: provisioner-type
  template: provisioner-template
  flavor: provisioner-flavor

metadata:
  maintainer: metadata-maintainer
  description: metadata-description
  operating_system_name: metadata-operating_system_name
  operating_system_version: metadata-operating_system_version
  programming_language_name: metadata-programming_language_name
  programming_language_version: metadata-programming_language_version
  framework_name: metadata-framework_name
  framework_version: metadata-framework_version

option_definitions:
  - namespace: option-def-namespace
    option_name: option-def-option_name
    description: option-def-description
    default_value: option-def-default_value

option_settings:
  - namespace: "option-setting-namespace"
    option_name: "option-setting-option_name"
    value: "option-setting-value"
```

Sustituya los marcadores por estos valores:

version-number

Obligatorio. Versión de la definición de YAML. Debe ser **1.0**.

provisioner-type

Obligatorio. Tipo de constructor utilizado para crear la plataforma personalizada. Debe ser **packer**.

provisioner-template

Obligatorio. Archivo JSON que contiene la configuración de *provisioner-type*.

provisioner-flavor

Opcional. Sistema operativo base que se utiliza con la AMI. Uno de los siguientes:

amazon (predeterminado)

Amazon Linux Si no se especifica, se utiliza la versión más reciente de Amazon Linux en el momento en que se creó la plataforma.

Amazon Linux 2 no es un tipo de sistema operativo compatible.

ubuntu1604

Ubuntu 16.04 LTS

rhel7

RHEL 7

rhel6

RHEL 6

metadata-maintainer

Opcional. Información de contacto de la persona propietaria de la plataforma (100 caracteres).

metadata-description

Opcional. Descripción de la plataforma (2000 caracteres).

metadata-operating_system_name

Opcional. Nombre del sistema operativo de la plataforma (50 caracteres). Este valor está disponible al filtrar la salida de la [ListPlatformVersionsAPI](#).

metadata-operating_system_version

Opcional. Versión del sistema operativo de la plataforma (20 caracteres).

metadata-programming_language_name

Opcional. Lenguaje de programación admitido por la plataforma (50 caracteres).

metadata-programming_language_version

Opcional. Versión del lenguaje de la plataforma (20 caracteres).

metadata-framework_name

Opcional. Nombre del marco web utilizado por la plataforma (50 caracteres).

metadata-framework_version

Opcional. Versión del marco web de la plataforma (20 caracteres).

option-def-namespace

Opcional. Espacio de nombres de `aws:elasticbeanstalk:container:custom` (100 caracteres).

option-def-option_nombre

Opcional. Nombre de la opción (100 caracteres). Puede definir hasta 50 opciones de configuración personalizadas para que la plataforma las ofrezca a los usuarios.

option-def-description

Opcional. Descripción de la opción (1024 caracteres).

option-def-default_valor

Opcional. Valor predeterminado que se utiliza cuando el usuario no especifica ningún valor.

En el ejemplo siguiente, se crea la opción **NPM_START**.

```
options_definitions:
- namespace: "aws:elasticbeanstalk:container:custom:application"
  option_name: "NPM_START"
  description: "Default application startup command"
  default_value: "node application.js"
```

option-setting-namespace

Opcional. Espacio de nombres de la opción.

option-setting-option_nombre

Opcional. Nombre de la opción. Puede especificar hasta 50 [opciones proporcionadas por Elastic Beanstalk](#).

option-setting-value

Opcional. Valor que se utiliza cuando el usuario no especifica ningún valor.

En el ejemplo siguiente, se crea la opción **TEST**.

```
option_settings:
- namespace: "aws:elasticbeanstalk:application:environment"
  option_name: "TEST"
  value: "This is a test"
```

Etiquetado de versiones de la plataforma personalizada

Puede aplicar etiquetas a sus versiones de plataforma AWS Elastic Beanstalk personalizadas. Las etiquetas son pares clave-valor asociados AWS a los recursos. Para obtener información sobre el etiquetado de recursos de Elastic Beanstalk, los casos de uso, las restricciones de las claves y los valores de las etiquetas y los tipos de recursos admitidos, consulte [Etiquetar recursos de la aplicación Elastic Beanstalk](#).

Puede especificar etiquetas cuando cree una versión de la plataforma personalizada. En una versión de la plataforma personalizada, puede añadir o eliminar etiquetas, y actualizar los valores de etiquetas existentes. Puede añadir hasta 50 etiquetas a cada versión de la versión de la plataforma personalizada.

Adición de etiquetas durante la creación de la versión de la plataforma personalizada

Si utiliza la CLI de EB para crear una versión de la plataforma personalizada, use la opción `--tags` con [eb platform create](#) para añadir etiquetas.

```
~/workspace/my-app$ eb platform create --tags mytag1=value1,mytag2=value2
```

Con este AWS CLI u otros clientes basados en API, añada etiquetas mediante el `--tags` parámetro del comando. [create-platform-version](#)

```
$ aws elasticbeanstalk create-platform-version \
  --tags Key=mytag1,Value=value1 Key=mytag2,Value=value2 \
  --platform-name my-platform --platform-version 1.0.0 --platform-definition-bundle
S3Bucket=DOC-EXAMPLE-BUCKET,S3Key=sample.zip
```

Administración de etiquetas de una versión de la plataforma personalizada existente

Puede añadir, actualizar y eliminar etiquetas en una versión de la plataforma personalizada Elastic Beanstalk existente.

Si utiliza la CLI de EB para actualizar su versión de la plataforma personalizada, utilice [eb tags](#) para añadir, actualizar, eliminar o enumerar etiquetas.

Por ejemplo, el siguiente comando muestra las etiquetas en una versión de la plataforma personalizada.

```
~/workspace/my-app$ eb tags --list --resource "arn:aws:elasticbeanstalk:us-east-2:my-account-id:platform/my-platform/1.0.0"
```

El siguiente comando actualiza la etiqueta mytag1 y elimina la etiqueta mytag2.

```
~/workspace/my-app$ eb tags --update mytag1=newvalue --delete mytag2 \
--resource "arn:aws:elasticbeanstalk:us-east-2:my-account-id:platform/my-platform/1.0.0"
```

Para obtener una lista de las opciones y más ejemplos, consulte [eb tags](#).

Con este AWS CLI u otros clientes basados en API, utilice el [list-tags-for-resource](#) comando para enumerar las etiquetas de una versión de plataforma personalizada.

```
$ aws elasticbeanstalk list-tags-for-resource --resource-arn
"arn:aws:elasticbeanstalk:us-east-2:my-account-id:platform/my-platform/1.0.0"
```

Utilice el comando [update-tags-for-resource](#) para añadir, actualizar o eliminar etiquetas en una versión de la plataforma personalizada.

```
$ aws elasticbeanstalk update-tags-for-resource \
--tags-to-add Key=mytag1,Value=newvalue --tags-to-remove mytag2 \
--resource-arn "arn:aws:elasticbeanstalk:us-east-2:my-account-id:platform/my-platform/1.0.0"
```

Especifique las etiquetas que desea añadir y las que desea actualizar en el parámetro `--tags-to-add` de `update-tags-for-resource`. Se añade una etiqueta no existente y se actualiza el valor de una etiqueta existente.

Note

Para usar algunos de los AWS CLI comandos y la CLI de EB con una versión de plataforma personalizada de Elastic Beanstalk, necesita el ARN de la versión de plataforma personalizada. Puede recuperar el ARN mediante el siguiente comando.

```
$ aws elasticbeanstalk list-platform-versions
```

Utilice la opción `--filters` para filtrar la salida por el nombre de la plataforma personalizada.

Historial de documentos

En la siguiente tabla se describen los cambios importantes introducidos en la Guía para AWS Elastic Beanstalk desarrolladores desde abril de 2024.

Cambio	Descripción	Fecha
QuickStart para .NET Core en Windows	Nuevo QuickStart para .NET Core en Windows.	28 de junio de 2024
QuickStart para Docker	Nuevo QuickStart para Docker.	19 de junio de 2024
Impedir el acceso al bucket de Amazon S3 entre entornos	Nuevo: Impedir el acceso a los buckets de Amazon S3 entre entornos.	12 de junio de 2024
QuickStart para .NET Core en Linux	Nuevo QuickStart para .NET Core en Windows.	28 de mayo de 2024
QuickStart para PHP	Nuevo QuickStart para PHP.	10 de mayo de 2024
QuickStart para Node.js	Nuevo QuickStart para Node.js.	5 de mayo de 2024
QuickStart para Go	Nuevo QuickStart para Go.	5 de mayo de 2024
Calendario de lanzamientos de la plataforma Elastic Beanstalk	Se agregó un tema nuevo que incluye un cronograma de Próximos lanzamientos de las sucursales de plataformas Se trasladó Programación de sucursales de plataforma de retirada y Historial de la ramificación de la plataforma retirada a este tema.	1 de mayo de 2024

<u>AWSElasticBeanstalkRoleCore AWS política gestionada</u>	Permisos actualizados en la política AWS gestionada.	30 de abril de 2024
<u>AWSElasticBeanstalkManagedUpdatesServiceRolePolicy AWS política gestionada</u>	Permisos actualizados en la política AWS gestionada.	30 de abril de 2024
<u>AWSElasticBeanstalkManagedUpdatesInternalServiceRolePolicy AWS política gestionada</u>	Permisos actualizados en la política AWS gestionada.	30 de abril de 2024
<u>AWSElasticBeanstalkMaintenance AWS política gestionada</u>	Permisos actualizados en la política AWS gestionada.	30 de abril de 2024
<u>AWSElasticBeanstalkInternalMaintenanceRolePolicy AWS política gestionada</u>	Permisos actualizados en la política AWS gestionada.	30 de abril de 2024

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la versión original de inglés, prevalecerá la versión en inglés.